```
? EXECUTE ESPOL/DISK

          PACKET  1510
          INPUT    314 CARDS FROM ZIP
          TIME    1314
          DATE   77138 WEDNESDAY, 05/18/77



       *** MESSAGE OF THE DAY ***

            *** SYSTEM WAS COLD STARTED 5/4/77 USING 4/23/77 BACKUP TAPES
                SOME FILES MAY BE AVAILABLE FROM THE 5/4/77 TAPES MADE PRIOR
                TO THE COLD START, BUT SOME FILES WERE LOST
              PLEASE USE USER CARDS IN FRONT OF ALL BATCH JOBS OR PACKETS

            WHEN YOU GET TIRED OF THIS MESSAGE -
            SUGGEST A BETTER ONE.

       *** BURROUGHS B5700 TSMCP MARK XVI.0.69  AND INTRINSICS MARK XVI.0.00 ***

        ? EXECUTE ESPOL/DISK
            ? STACK= 1024
            ? FILE TAPE= SYMBOL/INTRINS DISK SERIAL
            ? FILE STUFF= TSSINT/STUFF DISK
            ? FILE LINE= LINE BACK UP TAPE
            ? FILE DISK= TSS/INT
            ? CORE= 15000
       ? DATA CARD
             4:ESPOL/DISK/SITE= 3 BOJ 1314 12/31/76
             DKA OUT SER CODE SITE:ESPOL/DISK= 3
             CDB IN CARD:ESPOL/DISK= 3
             DKA IN SER SYMBOL INTRINS:ESPOL/DISK= 3
             NEW PBT ON MTD
             MTD OUT 211 LINE:ESPOL/DISK= 3
             DKA OUT SER TSSINT STUFF:ESPOL/DISK= 3
             DKA OUT SER CODISK SITE:ESPOL/DISK= 3
            3ST
            #OPRTR ST=ED ESPOL/DISK= 3
            3OK
             DKA LOK TSSINT STUFF:ESPOL/DISK= 3
             CDB REL CARD:ESPOL/DISK= 3
       ? END.
             DKA REL SYMBOL INTRINS:ESPOL/DISK= 3
             MTD LOK LINE:ESPOL/DISK= 3
             DKA OUT SER TSS INT:ESPOL/DISK= 3
             DKA LOK TSS INT:ESPOL/DISK= 3
             CODISK/SITE/SITE= 1200 SEGS--CREATED 05/18/77 AT 13:15:14:23
             DKA REL CODISK SITE:ESPOL/DISK= 3
             CODE/SITE/SITE= 0 SEGS--CREATED 05/18/77 AT 13:14:35:57
             DKA REL CODE SITE:ESPOL/DISK= 3
             ESPOL/DISK/SITE= 3,PST= 9:50 EOJ
             FOR ESPOL/DISK= 3:PST= 589,IOT= 146,CORE=15360
```

PKT#1510 REMOVED

TSS /INT

```
%        I N T R I N S I C S    M A R K  XVI,0,00          10/01/74          00000000 T  0000:0
  COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE                    * 00000010 T  0000:0
           * FILE ID: SYMBOL/INTRINS  TAPE ID: SYMBOL1/FILE000             * 00000011 T  0000:0
           * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION         * 00000012 T  0000:0
           * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED               * 00000013 T  0000:0
           * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON             * 00000014 T  0000:0
           * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF               * 00000015 T  0000:0
           * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232                * 00000016 T  0000:0
           *                                                               * 00000017 T  0000:0
           * COPYRIGHT (C)  1971, 1972, 1974                               * 00000018 T  0000:0
           *       BURROUGHS CORPORATION                                   * 00000019 T  0000:0
           *  AA320206   AA393180   AA332366                               *; 00000021 T  0000:0
BEGIN                                                                      00000100 T  0000:0
   DEFINE    ETRLNG    = 5#,                                               00000200 T  0000:0
             INTDESC(INTDESC1) = FLAG(INTDESC1 & 85[1:41:7]) #,            00000210 T  0000:0
             INTCALL(INTCALL1,INTCALL2) = P(INTCALL2 & 85[1:41:7],         00000215 T  0000:0
                                            INTCALL1,COC) #,               00000216 T  0000:0
             CALLINT(CALLINT1) = P(CALLINT1 & 85[1:41:7],XCH,COC) #,       00000218 T  0000:0
             COBOLDCI= @167 #,                                             00000219 T  0000:0
             FORTERRI= @134 #,                                             00000220 T  0000:0
             EXPI    = @20 #,                                              00000221 T  0000:0
             LNI     = @17 #,                                              00000222 T  0000:0
             DEXPI   = @77 #,                                              00000223 T  0000:0
             DLOGI   = @101 #,                                             00000224 T  0000:0
             CABSI   = @53 #,                                              00000225 T  0000:0
             SINI    = @14 #,                                              00000226 T  0000:0
             SQRTI   = @13 #,                                              00000227 T  0000:0
             ATAN2I  = @114 #,                                             00000228 T  0000:0
             DMODI   = @65 #,                                              00000229 T  0000:0
             DSINI   = @105 #,                                             00000230 T  0000:0
             DSQRTI  = @123 #,                                             00000231 T  0000:0
             XTOII   = @6 #,                                               00000232 T  0000:0
             CXTOII  = @56 #,                                              00000233 T  0000:0
             COSI    = @15 #,                                              00000234 T  0000:0
             TANI    = @111 #,                                             00000235 T  0000:0
             ARCTANI = @16 #,                                              00000236 T  0000:0
             DATANI  = @113 #,                                             00000237 T  0000:0
             ARSINI  = @116 #,                                             00000238 T  0000:0
             GAMMAI  = @126 #,                                             00000239 T  0000:0
             EDITIT(EDITIT1,EDITIT2,EDITIT3,EDITIT4,EDITIT5) = P(MKS,      00000240 T  0000:0
                    EDITIT1,EDITIT2,EDITIT3,(-1),(EDITIT4),(EDITIT5),      00000241 T  0000:0
                      @153&85[1:41:7],XCH,COC) #,                          00000242 T  0000:0
             % EDITIT(BUFFADDRESS,FIELDWIDTH(W),TYPE,LOWPART,HIGHPART)     00000243 T  0000:0
             % WILL EDIT THE VALUE (LOWPART,HIGHPART) INTO A FIELD         00000244 T  0000:0
             % STARTING AT BUFFADDRESS.  EDITIT RETURNS THE ENDING         00000245 T  0000:0
             % ADDRESS.  THE WIDTH OF THE EDITED FIELD IS CONSTRAINED      00000246 T  0000:0
             % TO W CHARACTERS (EDITED VALUE IS RIGHT JUSTIFIED WITH       00000247 T  0000:0
```

```
                % LEADING BLANKS IF W IS LARGER THAN NEEDED) -- BUT IF                00000248 T  0000:0
                % W=0, THEN EDITIT WILL ADJUST THE FIELD WIDTH TO                      00000249 T  0000:0
                % ACCOMODATE FULL NUMERICAL SIGNIFICANCE. TYPE=2 => EDITIT             00000250 T  0000:0
                % WILL CHOOSE BETWEEN REAL, INTEGER, AND DOUBLEPRECISION               00000251 T  0000:0
                % EDITING (DOUBLEPRECISION IS USED IF LOWPART≠0).                     00000252 T  0000:0
                % TYPE=1 => USE ONLY INTEGER, TYPE=3 => USE ONLY REAL,                 00000253 T  0000:0
                % TYPE=4 => USE ONLY LOGICAL, TYPE=5 => USE ONLY DOUBLE-               00000254 T  0000:0
                % PRECISION.                                                           00000255 T  0000:0
                CTC       = 33:33:15#,                                               00000300 T  0000:0
                CTF       = 18:33:15#,                                               00000400 T  0000:0
                FTC       = 33:18:15#,                                               00000410 T  0000:0
                FTF       = 18:18:15#,                                               00000420 T  0000:0
                CF        = 33:15#,                                                  00000500 T  0000:0
                FF        = 18:15#;                                                  00000600 T  0000:0
   REAL         JUNK      = 5;                                                       00000700 T  0000:0
  NAME MEM=2, M=2, MEMORY=2 ;                                                       00000710 T  0000:0
   REAL         BLKCNTRL  = 5;                                                     00000750 T  0000:0
    DEFINE DUMPNOW(DUMPNOW1)=P(DUMPNOW1,0,48,COM,DEL,DEL)#,                         00000775 T  0000:0
           TRACENOW(TRACENOW1,TRACENOW2)=                                           00000780 T  0000:0
             P(TRACENOW1,1,TRACENOW2 ,+ ,48,COM,DEL,DEL)#;                          00000785 T  0000:0
PROCEDURE OUTPUTINT(TEN, FILX, CHSKP, LNSKP, FI, FRMT, LISX);%      %WF           00000800 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00005
   VALUE      CHSKP, LNSKP, FI, LISX;%                                %WF        00000900 T  0000:0
   NAME       FILX;%                                                  %WF        00001000 T  0000:0
   ARRAY      TEN[*], FRMT[*];%                                       %WF        00001100 T  0000:0
   REAL       LISX;%                                                  %WF        00001200 T  0000:0
   INTEGER    CHSKP, LNSKP, FI;%  00200000                            %WF        00001300 T  0000:0
        FORWARD;%       CODE=28000000, INTRINSIC NUMBER=@ 1          %WF        00001400 T  0000:0
PROCEDURE INTRINSIC(DUPE, D, NUMDIM, SIZE, TYPE);%                    %WF        00001500 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00005
   VALUE      DUPE, D, NUMDIM, SIZE, TYPE;%                           %WF        00001600 T  0000:0
   NAME       D;%                                                     %WF        00001700 T  0000:0
   ARRAY      DUPE[*];%                                               %WF        00001800 T  0000:0
   INTEGER    NUMDIM, SIZE, TYPE;%  00400000                          %WF        00001900 T  0000:0
        FORWARD;%       CODE=32000000, INTRINSIC NUMBER=@ 2          %WF        00002000 T  0000:0
PROCEDURE INPUTINT(TEN, FILX, DKADR, ACT,%                            %WF        00002100 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00005
                   FI, FRMT, LISX, EOFL, PARL);%                      %WF        00002200 T  0000:0
   VALUE      ACT, FI;%                                               %WF        00002300 T  0000:0
   NAME       FILX, LISX;%                                            %WF        00002400 T  0000:0
   ARRAY      TEN[*], FRMT[*];%                                       %WF        00002500 T  0000:0
   REAL       EOFL, PARL;%                                            %WF        00002600 T  0000:0
   INTEGER    DKADR, ACT, FI;%  00600000                              %WF        00002700 T  0000:0
        FORWARD;%       CODE=36000000, INTRINSIC NUMBER=@ 3          %WF        00002800 T  0000:0
PROCEDURE DISKSORT(T1, T2, RELA, ENDQ, BINGO, IPFIDX,%                %WF        00002900 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00005
             OUTPRO, INPRO, OUTF, INF, OPTOG, IPTOG, DKO, DKI,%       %WF        00003000 T  0000:0
             TP1, TP2, TP3, TP4, TP5, NT, HIVALU, EQUALS,%            %WF        00003100 T  0000:0
             R, ALFA, CORESIZE, DISKSIZE);%                           %WF        00003200 T  0000:0
   VALUE      OPTOG, IPTOG, NT, HIVALU, EQUALS, R, ALFA,%             %WF        00003300 T  0000:0
              CORESIZE, DISKSIZE;%                                    %WF        00003400 T  0000:0
   NAME       TP1, TP2, TP3, TP4, TP5;%                               %WF        00003500 T  0000:0
   REAL       T1, T2, RELA, ENDQ, BINGO, IPFIDX, OUTPRO, INPRO,%      %WF        00003600 T  0000:0
              OUTF, INF, DKO, DKI, NT, HIVALU, EQUALS, CORESIZE;%     %WF        00003700 T  0000:0
   BOOLEAN    OPTOG, IPTOG, ALFA;%                                    %WF        00003800 T  0000:0
   INTEGER    R, DISKSIZE;%  00700000                                 %WF        00003900 T  0000:0
        FORWARD;%       CODE=39400000, INTRINSIC NUMBER=@ 4          %WF        00004000 T  0000:0
```

```
REAL PROCEDURE DUMPINT(SN, CV, BV, TIPE;%
                                TENS, ALFA, CHAR, FIEL, FORMT);%
   VALUE     SN, CV, BV, TIPE, TENS, ALFA, CHAR, FORMT;%
   NAME      FIEL;%
   REAL      SN, CV, BV, TIPE, TENS, ALFA, CHAR, FORMT;%
        FORWARD;%      CODE=42000000, INTRINSIC NUMBER=@ 5
PROCEDURE XTOTHEIINT(BASE, EXPON, M, LOG, EXP);%

   VALUE     BASE, EXPON, M, LOG, EXP;%
   REAL      BASE, EXPON, M, LOG, EXP;%
        FORWARD;%      CODE=42254000, INTRINSIC NUMBER=@ 6
REAL PROCEDURE ABSINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@ 7
REAL PROCEDURE SIGNINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@10
INTEGER PROCEDURE ENTIERINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@11
REAL PROCEDURE TIMEINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@12
PROCEDURE SQRTINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@13
PROCEDURE SININT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@14
PROCEDURE COSINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@15
REAL PROCEDURE ARCTANINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@16
PROCEDURE LNINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=          INTRINSIC NUMBER=@17
```

```
        %WF          00004100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00004200 T  0000:0
        %WF          00004300 T  0000:0
        %WF          00004400 T  0000:0
        %WF          00004500 T  0000:0
        %WF          00004600 T  0000:0
        %WF          00004700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00004800 T  0000:0
        %WF          00004900 T  0000:0
        %WF          00005000 T  0000:0
        %WF          00005100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00005200 T  0000:0
        %WF          00005300 T  0000:0
        %WF          00005400 T  0000:0
        %WF          00005500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00005600 T  0000:0
        %WF          00005700 T  0000:0
        %WF          00005800 T  0000:0
        %WF          00005900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00006000 T  0000:0
        %WF          00006100 T  0000:0
        %WF          00006200 T  0000:0
        %WF          00006300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00006400 T  0000:0
        %WF          00006500 T  0000:0
        %WF          00006600 T  0000:0
        %WF          00006700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00006800 T  0000:0
        %WF          00006900 T  0000:0
        %WF          00007000 T  0000:0
        %WF          00007100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00007200 T  0000:0
        %WF          00007300 T  0000:0
        %WF          00007400 T  0000:0
        %WF          00007500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00007600 T  0000:0
        %WF          00007700 T  0000:0
        %WF          00007800 T  0000:0
        %WF          00007900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00008000 T  0000:0
        %WF          00008100 T  0000:0
        %WF          00008200 T  0000:0
        %WF          00008300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00008400 T  0000:0
        %WF          00008500 T  0000:0
        %WF          00008600 T  0000:0
```

```
REAL PROCEDURE EXPINT(X);%
   VALUE     X;%
   REAL      X;%
        FORWARD;%      CODE=           INTRINSIC NUMBER=@20
REAL PROCEDURE GOTOSOLVERINT(L, X, F, B);%
   VALUE     L, X, F, B;%
   ARRAY F[*];%
   REAL      L, X, B;%
        FORWARD;%      CODE=           INTRINSIC NUMBER=@21
PROCEDURE ALGOLWRITE(TEN, FILX, CHSKP, LNSKP, FI, AEXP,%
             ARRY, LINESKIP, CHANSKIP, SUPRS, NUMWDS, TANK);%
   VALUE     CHSKP, LNSKP, FI, AEXP, LINESKIP,%
             CHANSKIP, SUPRS, NUMWDS, TANK;%
   NAME      FILX, TANK;%
   ARRAY     TEN[*], ARRY[*];%
   INTEGER   CHSKP, LNSKP, FI, AEXP, LINESKIP,%
             CHANSKIP, SUPRS, NUMWDS;% 00100100
        FORWARD;%      CODE=~~26000000~~00, INTRINSIC NUMBER=@22
PROCEDURE ALGOLREAD(TEN, FILX, DKADD, ACT, FI, AEXP,%
             ARRY, EOFL, PARL, DKADR, CODE, TANK);%
   VALUE     ACT, FI, AEXP, DKADR, CODE, TANK;%
   NAME      FILX, TANK;%
   ARRAY     TEN[*], ARRY[*];%
   REAL      DKADD, EOFL, PARL, DKADR, CODE;%
   INTEGER   ACT, FI, AEXP;% 00500000
        FORWARD;%      CODE=~~34000000~~, INTRINSIC NUMBER=@23
PROCEDURE ALGOLSELECT(ACT1, ACT2, TANK, I);%
   VALUE     ACT1, ACT2, TANK, I;%
   NAME      TANK;%
   INTEGER   ACT1, ACT2, I;%
        FORWARD;%      CODE=           INTRINSIC NUMBER=@24
PROCEDURE COBOLFCR;%
        FORWARD;%      CODE=43000000, INTRINSIC NUMBER=@25
PROCEDURE COBOLIO;          % GO TO 02700000
        FORWARD;%      CODE=43230000, INTRINSIC NUMBER=@26
PROCEDURE POLYMERGE(T1, T2, T3, ENDQ, BINGO, IPFIDX,%
             OUTPRO, INPRO, OUTF, INF, OPTOG, IPTOG, DKO, DKI,%
             TP1, TP2, TP3, TP4, TP5, NT, HIVALU, EQUALS,%
             R, ALFA, CORESIZE, DISKSIZE);%
   VALUE     OPTOG, IPTOG, NT, HIVALU, EQUALS, R, ALFA,%
             CORESIZE, DISKSIZE;%
   NAME      TP1, TP2, TP3, TP4, TP5;%
   REAL      T1, T2, T3, ENDQ, BINGO, IPFIDX, OUTPRO, INPRO,%
             OUTF, INF, DKO, DKI, NT, HIVALU, EQUALS, R, CORESIZE;
   BOOLEAN   OPTOG, IPTOG, ALFA;%
   INTEGER   DISKSIZE;%
        FORWARD;%      CODE=40140000, INTRINSIC NUMBER=@27
PROCEDURE STATUSINT(T, C);%
```

```
        %WF          00008700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00008800 T  0000:0
        %WF          00008900 T  0000:0
        %WF          00009000 T  0000:0
        %WF          00009100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00009200 T  0000:0
        %WF          00009300 T  0000:0
        %WF          00009400 T  0000:0
        %WF          00009500 T  0000:0
        %WF          00009600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00009700 T  0000:0
        %WF          00009800 T  0000:0
        %WF          00009900 T  0000:0
        %WF          00010000 T  0000:0
        %WF          00010100 T  0000:0
        %WF          00010200 T  0000:0
        %WF          00010300 T  0000:0
        %WF          00010400 T  0000:0
        %WF          00010500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00010600 T  0000:0
        %WF          00010700 T  0000:0
        %WF          00010800 T  0000:0
        %WF          00010900 T  0000:0
        %WF          00011000 T  0000:0
        %WF          00011100 T  0000:0
        %WF          00011200 T  0000:0
        %WF          00011300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00011400 T  0000:0
        %WF          00011500 T  0000:0
        %WF          00011600 T  0000:0
        %WF          00011700 T  0000:0
        %WF          00011800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00011900 T  0000:0
             00012000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00012100 T  0000:0
        %WF          00012200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
        %WF          00012300 T  0000:0
        %WF          00012400 T  0000:0
        %WF          00012500 T  0000:0
        %WF          00012600 T  0000:0
        %WF          00012700 T  0000:0
        %WF          00012800 T  0000:0
        %WF          00012900 T  0000:0
        %WF          00013000 T  0000:0
        %WF          00013100 T  0000:0
        %WF          00013200 T  0000:0
        %WF          00013300 T  0000:0
        %WF          00013400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
```

```
   VALUE     T, C;%
   REAL      T;%
   INTEGER   C;%
        FORWARD;%       CODE=           INTRINSIC NUMBER=@30
REAL PROCEDURE MAXINT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%       CODE=           INTRINSIC NUMBER=@31
REAL PROCEDURE MININT(X);%

   VALUE     X;%
   REAL      X;%
        FORWARD;%       CODE=           INTRINSIC NUMBER=@32
PROCEDURE DELAYINT(ARRY, MASK, TIME);%

   VALUE ARRY, MASK, TIME;%
   ARRAY     ARRY[*];%
   REAL      MASK;%
   INTEGER TIME;%
        FORWARD;%       CODE=           INTRINSIC NUMBER=@33
PROCEDURE SUPERMOVERINT(SORCE, DEST, AEXP);%

   VALUE     AEXP;%
   ARRAY     SORCE[*], DEST[*];%
   INTEGER   AEXP;%
        FORWARD;%       CODE=           INTRINSIC NUMBER=@34
PROCEDURE SISO;   FORWARD;    %INT#35,SEQ#08400000

 INTEGER PROCEDURE DELTA(P1,P2);%INT#36,SEQ#00022300

    VALUE P1,P2;   INTEGER P1,P2;     FORWARD;
 PROCEDURE ICVD;   FORWARD;     %INT#37,SEQ#00022500

PROCEDURE DYNAMICDIALER(B, A, X, F);

VALUE B, A, X, F;
INTEGER B, A, X; BOOLEAN F;
        FORWARD;%       CODE=00022700, INTRINSIC NUMBER=@40
 PROCEDURE SCAN(UPDPDD,PTR,UPDCDD,HISCOUNT,CASECODE,CHAR);

    VALUE PTR, HISCOUNT, CASECODE, CHAR;
    NAME UPDPDD, UPDCDD;
    INTEGER PTR, HISCOUNT, CASECODE, CHAR;
    FORWARD;
 PROCEDURE REPL;   FORWARD;     %INT#42,SEQ#08420000

 PROCEDURE COMPARE;FORWARD;     %INT#43,SEQ#08430000

 PROCEDURE BASICPRINT(TYPE);

    VALUE     TYPE;
    REAL      TYPE;
        FORWARD;%       CODE=08500000, INTRINSIC NUMBER=@44
 PROCEDURE SWAP;  FORWARD;    %INT#45,SEQ#00023700

 PROCEDURE BASICINPUT(TYPES);
```

```
      %WF          00013500 T  0000:0
      %WF          00013600 T  0000:0
      %WF          00013700 T  0000:0
      %WF          00013800 T  0000:0
      %WF          00013900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
      %WF          00014000 T  0000:0
      %WF          00014100 T  0000:0
      %WF          00014200 T  0000:0
      %WF          00014300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
      %WF          00014400 T  0000:0
      %WF          00014500 T  0000:0
      %WF          00014600 T  0000:0
      %WF          00014700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
      %WF          00014800 T  0000:0
      %WF          00014900 T  0000:0
      %WF          00015000 T  0000:0
      %WF          00015100 T  0000:0
      %WF          00015200 T  0000:0
      %WF          00015300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
      %WF          00015400 T  0000:0
      %WF          00015500 T  0000:0
      %WF          00015600 T  0000:0
      %WF          00015700 T  0000:0
                   00015800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00015900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00015950 T  0000:0
                   00016000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016110 T  0000:0
                   00016120 T  0000:0
                   00016130 T  0000:0
                   00016200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016210 T  0000:0
                   00016220 T  0000:0
                   00016230 T  0000:0
                   00016240 T  0000:0
                   00016300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016510 T  0000:0
                   00016520 T  0000:0
                   00016530 T  0000:0
                   00016600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                   00016700 T  0000:0
```

```
      VALUE     TYPES;
      REAL      TYPES;
          FORWARD;%       CODE=08700000, INTRINSIC NUMBER=@46
    PROCEDURE READATA(TYPE);

      VALUE     TYPE;
      REAL      TYPE;
          FORWARD;%       CODE=08600000, INTRINSIC NUMBER=@47
    PROCEDURE FTINT ; FORWARD; % 050
    PROCEDURE FTOUT ; FORWARD; % 051
    PROCEDURE DABS  ; FORWARD; % 052
    PROCEDURE CABS  ; FORWARD; % 053
    PROCEDURE AINT  ; FORWARD; % 054
    PROCEDURE MATH  ; FORWARD; % 055
    PROCEDURE XTOI  ; FORWARD; % 056
    PROCEDURE IDINT ; FORWARD; % 057
    PROCEDURE FLOAT ; FORWARD; % 060
    PROCEDURE SNGL  ; FORWARD; % 061
    PROCEDURE DBLE  ; FORWARD; % 062
    PROCEDURE AMOD  ; FORWARD; % 063
    PROCEDURE TIME  ; FORWARD; % 064
    PROCEDURE DMOD  ; FORWARD; % 065
    PROCEDURE DMAX1 ; FORWARD; % 066
    PROCEDURE DMIN1 ; FORWARD; % 067
    PROCEDURE SIGNV ; FORWARD; % 070
    PROCEDURE DSIGN ; FORWARD; % 071
    PROCEDURE DIIM  ; FORWARD; % 072
    PROCEDURE REALP ; FORWARD; % 073
    PROCEDURE AIMAG ; FORWARD; % 074
    PROCEDURE CMPLX ; FORWARD; % 075
    PROCEDURE CONJG ; FORWARD; % 076
    PROCEDURE DEXP  ; FORWARD; % 077
```

FINNAME 02982500

see also FORTRANFREEREAD line 02920000

see also FTOUTFIX line 02886000

FORTRANFREEWRITE 02976020

FOUTNAME 02991260

line 02800000

line 02900000

```
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00016710 T  0000:0
                      00016720 T  0000:0
                      00016730 T  0000:0
                      00016800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00016810 T  0000:0
                      00016820 T  0000:0
                      00016830 T  0000:0
                      00016900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00017900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00018900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00019000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00019100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00019200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
```

```
PROCEDURE CEXP  ; FORWARD; % 100
PROCEDURE DLOG  ; FORWARD; % 101
PROCEDURE CLOG  ; FORWARD; % 102
PROCEDURE ALOG10; FORWARD; % 103
PROCEDURE DLOG10; FORWARD; % 104
PROCEDURE DSIN  ; FORWARD; % 105
PROCEDURE CSIN  ; FORWARD; % 106
PROCEDURE DCOS  ; FORWARD; % 107
PROCEDURE CCOS  ; FORWARD; % 110
PROCEDURE TANF  ; FORWARD; % 111
PROCEDURE COTAN ; FORWARD; % 112
PROCEDURE DATAN ; FORWARD; % 113
PROCEDURE ATAN2 ; FORWARD; % 114
PROCEDURE DATAN2; FORWARD; % 115
PROCEDURE ARSIN ; FORWARD; % 116
PROCEDURE ARCOS ; FORWARD; % 117
PROCEDURE SINH  ; FORWARD; % 120
PROCEDURE COSH  ; FORWARD; % 121
PROCEDURE TANH  ; FORWARD; % 122
PROCEDURE DSQRT ; FORWARD; % 123
PROCEDURE CSQRT ; FORWARD; % 124
PROCEDURE ERF   ; FORWARD; % 125
PROCEDURE GAMMA ; FORWARD; % 126
PROCEDURE ALGAMA; FORWARD; % 127
PROCEDURE ANDI  ; FORWARD; % 130
PROCEDURE ORI   ; FORWARD; % 131
PROCEDURE CMPL  ; FORWARD; % 132
PROCEDURE EQUIVP; FORWARD; % 133
PROCEDURE FORTERR;FORWARD; % 134
```

```
                    00019300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00019900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00020900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021200 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021300 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021400 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021500 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021600 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021700 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021800 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00021900 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00022000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                    00022010 T  0000:0
```

```
PROCEDURE MAX; FORWARD; % 135

PROCEDURE MIN; FORWARD; % 136

PROCEDURE IMOD; FORWARD; % 137

PROCEDURE CONCAT; FORWARD; % 140

 PROCEDURE C@NCAT;

        FORWARD;%       CODE=08400000, INTRINSIC NUMBER=@140
 PROCEDURE MATRIXDIDDLER(A, B, C, TYPE);

    VALUE     A, B, C, TYPE;
    ARRAY     A[*], B[*], C[*];
    INTEGER   TYPE;
        FORWARD;%       CODE=08800000, INTRINSIC NUMBER=@+4+
 PROCEDURE INVERT(A, B);

    VALUE     A, B;
    ARRAY     A[*], B[*];
        FORWARD;%       CODE=09100000, INTRINSIC NUMBER=@142
 PROCEDURE TRANSPOSE(A, B);

    VALUE     A, B;
    ARRAY     A[*], B[*];
        FORWARD;%       CODE=08900000, INTRINSIC NUMBER=@143
 PROCEDURE MATRIXMULTIPLY(A, B, C);

    VALUE     A, B, C;
    ARRAY     A[*], B[*], C[*];
        FORWARD;%       CODE=09000000, INTRINSIC NUMBER=@144
 PROCEDURE RANDOM(NUMBER, BASE);

    VALUE     NUMBER;
    REAL      NUMBER;
    INTEGER   BASE;
        FORWARD;%       CODE=00022900, INTRINSIC NUMBER=@145
 PROCEDURE FORTRANFREEREAD;

        FORWARD;%       CODE=09200000, INTRINSIC NUMBER=@146
 PROCEDURE BASICLOSE(FILX);

    VALUE FILX;  NAME FILX;
    BEGIN REAL SELECT=14, ALGOLWRITE=12;  ARRAY AIT=6[*];
          REAL T,I;  ARRAY FIB[*];  NAME M=2;
          SUBROUTINE MAYBEPRINT;
          BEGIN FIB:=FILX[NOT 2];
                IF FIB[5].[41:3]=0 THEN  %NOT CLOSED-NOT INPUT
                IF FIB[4].[8:4] NEQ 10 THEN  %NOT DATA COM
                IF FIB[20].[3:15]#0 THEN   % DATA LEFT
                   P(MKS,1,0,0,(FIB[20].[18:10]+1),FILX,ALGOLWRITE);
          END;
          IF P(.FILX,LOD)=0 THEN  %EOJ FILE CLOSE
          BEGIN I:=AIT[0]+1; WHILE (T:=AIT[I:=I-1]).[8:10] NEQ 0
               DO IF T.[1:1] THEN
```

```
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022011 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022012 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022013 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022014 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022020 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022025 T  0000:0
                      00022030 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022032 T  0000:0
                      00022034 T  0000:0
                      00022036 T  0000:0
                      00022038 T  0000:0
                      00022040 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022050 T  0000:0
                      00022060 T  0000:0
                      00022070 T  0000:0
                      00022080 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022090 T  0000:0
                      00022100 T  0000:0
                      00022110 T  0000:0
                      00022120 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022130 T  0000:0
                      00022140 T  0000:0
                      00022150 T  0000:0
                      00022160 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022162 T  0000:0
                      00022164 T  0000:0
                      00022166 T  0000:0
                      00022168 T  0000:0
                      00022170 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022175 T  0000:0
                      00022180 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00005
                      00022185 T  0000:0
                      00022190 T  0000:0
                      00022195 T  0000:0
                      00022200 T  0000:0
                      00022205 T  0001:0
                      00022210 T  0002:3
                      00022212 T  0004:1
                      00022215 T  0006:1
                      00022220 T  0008:1
                      00022225 T  0011:3
                      00022230 T  0012:0
                      00022235 T  0013:3
                      00022240 T  0018:1
```

```
          BEGIN FILX:=M[M[T,[18:15]] INX 4]; MAYBEPRINT END;                    00022245 T  0020:0
        END ELSE  %FILE RESTORE                                                 00022250 T  0025:2
        BEGIN MAYBEPRINT;                                                       00022255 T  0025:2
              P(MKS,2,0,[FILX[NOT 2]],4,SELECT);                                00022260 T  0027:0
              FIB[0]:=FIB[8]:=FIB[20]:=FIB[21]:=0;                              00022265 T  0029:1
        END;                                                                    00022270 T  0033:2
    END BASIC FILE RESTORE;                                                     00022275 T  0033:2
                                                                                      SIZE= 0034 WORDS



PROCEDURE FILEATTRIBUTES(T,E,D,V,G,I,TN); VALUE T,I,V,D,G; REAL D,G,I,E;        00022280 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
INTEGER V; ARRAY TN[*]; NAME T; FORWARD; % CODE @ 0043000, INT # @150            00022281 T  0000:0
PROCEDURE COBOLDECIMALTOOCTALCONVERT(A); % INT #=@151, CODE=09300000            00022282 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
VALUE A;  NAME A;  FORWARD ;                                                    00022283 T  0000:0
PROCEDURE COBOLOCTOLTODECIMALCONVERT(A,L,H,R,N,S,T); % INT #=@152                00022284 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
VALUE L,H,R,N,S,T;  REAL L,H,R,N,S,T; NAME A; FORWARD; % CODE=09400000           00022285 T  0000:0
PROCEDURE FORTRANFREEWRITE(F,D,R,W,L,I,N,S); VALUE I,D,R,W,L; INTEGER R,        00022286 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
W; REAL I,D,L; NAME F; ARRAY S[*],N[*]; FORWARD ;%COD @02976019,INT@153          00022287 T  0000:0
PROCEDURE FINNAME; FORWARD;                                                     00022288 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE FOUTNAME; FORWARD;                                                    00022289 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE FTINTFIX(F1,D1,F2,F3,L1,E1,E2,P1); VALUE D1,F2,L1,E1,E2,P1 ;          00022292 T  0000:0
                                                 Ø2780000                       START OF REL SEGMENT; DISK ADDRESS = 00007
REAL D1,F2,L1,E1,E2,P1; ARRAY F3[*]; NAME F1; FORWARD; % INTRINSIC @156          00022293 T  0000:0
PROCEDURE FTOUTFIX(F,D,R,Q,L,E,EL,PL); VALUE D,R,L,E,EL,PL; REAL D,R,L,E         00022294 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
,EL,PL; NAME F;  ARRAY Q[*]; FORWARD ; % CODE AT SEQ # 02886040, INT@157        00022295 T  0000:0
PROCEDURE FBINBACKBLOCK(F1,D,F2,F3,L,E1,E2,P1);  VALUE D,F2,L,E1,E2,P1 ;        00022296 T  0000:0
                                                 Ø2767050                       START OF REL SEGMENT; DISK ADDRESS = 00007
REAL D,F2,L,E1,E2,P1;  ARRAY F3[*];  NAME F1;  FORWARD;  % INT # @160.          00022297 T  0000:0
PROCEDURE COBOLVARSZ;  FORWARD;%  CODE=09500000  INT #=@161                     00022298 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE COBOLIONONDSK;  FORWARD;%  CODE=09600000  INT #=@162                  00022299 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE COBOLIODSK;  FORWARD;%  CODE=09700000  INT #=@163                     00022300 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE FORTRANMEMHANDLER(A,H);VALUE H;REAL H;ARRAY A[*];FORWARD;%164         00022301 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE COBOLATT; FORWARD; % CODE = 02650000  INT # = @165   %CJC 103I        00022302 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE INTERRUPTER; FORWARD; % CODE=09800000; INT #=@166                     00022303 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
PROCEDURE COBOLDC; FORWARD; % CODE = 02690000  INT #=@167                       00022304 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
INTEGER PROCEDURE DELTA(P1,P2);  VALUE P1,P2;  REAL P1,P2;        %@036         00022310 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00007
BEGIN                                                                           00022320 T  0000:0
      DEFINE                                                                    00022330 T  0000:0
      DOT=[18:13]#, AMPER=[18:35:13]#;                                          00022340 T  0000:0
        COMMENT @4000000=2*20, WHICH IS 1 LARGER THAN ANY 6500 COUNT.;          00022350 T  0000:0
```

```
        COMMENT DELTA=2*20 IF DESC(P1)≠DESC(P2) OR CSIZE=S ARE ≠;                00022360 T  0000:0
        IF (P2-P1).[31:17]≠0 THEN DELTA←@4000000 ELSE                            00022370 T  0000:0
        DELTA←P2.DOT-P1.DOT;                                                     00022380 T  0003:1
END DELTA;                                                                       00022390 T  0007:1
                                                                                        SIZE= 0008 WORDS



                                                                                 00022400 T  0000:0
PROCEDURE ICVD;              %37                                                 00022500 T  0000:0
                                                                                        START OF REL SEGMENT; DISK ADDRESS = 00008
BEGIN                                                                            00022510 T  0000:0
        DEFINE DOT=[18:13]#, AMPER=[18:35:13]#, CSIZE=[31:02]#,SIX=0#;           00022520 T  0000:0
        ARRAY STRING[*];                                                         00022530 T  0000:0
        NAME M = 2;                                                              00022540 T  0000:0
        REAL PTR=-3;   INTEGER N=-1;                                             00022550 T  0000:0
        IF PTR.CSIZE≠SIX THEN POLISH(M&1[17:47:01],9999,CDC,DEL);                00022560 T  0000:0
        STRING ← M[PTR];                                                         00022570 T  0004:0
        N←N; COMMENT MAKE SURE N IS INTEGERIZED;                                 00022575 T  0005:2
        IF N>8 THEN POLISH(M&1[14:47:01],N,CDC,DEL);                             00022580 T  0006:1
        POLISH([STRING[(PTR.DOT+N-1).[35:10]]], DEL);                            00022590 T  0009:2
        STREAM(RESULT←0:S←[STRING[PTR.[18:10]]], N,                              00022600 T  0012:1
             SKS←PTR.[28:03]);                                                   00022610 T  0014:1
        BEGIN                                                                    00022620 T  0015:1
        DI ← LOC RESULT;                                                         00022630 T  0015:1
        SI ← S; SI ← SI+SKS;                                                     00022640 T  0015:2
        DS ← N OCT;                                                              00022650 T  0016:1
        END;                                                                     00022660 T  0016:3
        PTR ← P;                                                                 00022670 T  0017:0
END ICVD;                                                                        00022680 T  0017:2
                                                                                        SIZE= 0019 WORDS



PROCEDURE DYNAMICDIALER(A,B,X,F) ;                                               00022700 T  0000:0
                                                                                        START OF REL SEGMENT; DISK ADDRESS = 00009
VALUE B, A, X, F;                                                                00022705 T  0000:0
INTEGER B, A, X; BOOLEAN F;                                                      00022710 T  0000:0
    BEGIN  % A,B,X,Y,Z ARE AS IN Y&Z[A:B:X].                                     00022715 T  0000:0
           % F=TRUE => X WAS LITERAL, AND TRB WILL BE DONE AFTER XITING.         00022720 T  0000:0
    REAL Y=-7, Z=-6, C=+1 ;                                                      00022725 T  0000:0
    DEFINE Q= @3403007777777777 #, % MASK FOR ZERO-ING OUT THE G,H,K&V-          00022730 T  0000:0
                                    % REGISTER PARTS OF THE RCW.                 00022735 T  0000:0
           R= @0055005500610065 #, % NOP,DIA,DIB,TRB.                            00022740 T  0000:0
           S= @0055703404210435 #; % NOP,LITC Y,STD,XIT.                          00022745 T  0000:0
    IF (A←A)<1 OR (B←B)<1 OR (X←X)<1 OR X+A>48 OR X+B>48                         00022750 T  0000:0
       THEN P((-63),26,COM) ;                                                    00022755 T  0006:2
    IF F THEN P(Q,AND,0&(B MOD 6)[4:9:3],A MOD 6,DIB 7,TRB 3,                    00022760 T  0008:3
              P&(B DIV 6)[12:45:3],A DIV 6,DIB 15,TRB 3,OR,0,0,XIT) ;            00022765 T  0013:0
    GO P(P(R)&(B DIV 6)[12:45:3],A DIV 6,DIB 24,TRB 3,P&(B MOD 6)                00022770 T  0016:3
       [15:9:3],A MOD 6,DIB 27,TRB 3,P&X[36:42:6],,A,←,S,,B,←,Y,Z,[A]);          00022775 T  0020:1
END DYNAMICDIALER;                                                               00022830 T  0025:2
                                                                                        SIZE= 0029 WORDS
```

```
PROCEDURE RANDOM(NUMBER, BASE);

   VALUE     NUMBER;
   REAL      NUMBER;
   INTEGER   BASE;
   BEGIN INTEGER  N;
         REAL     T;
         IF (T := NUMBER MOD 1.0)>0 THEN
         BEGIN BASE := T.[9:38];  P(RTN);  END;
         IF NUMBER≠0 THEN
         BEGIN T := POLISH(1, 1, COM);
               N := 0 & T[10:36:6] & T[16:42:6] & T[22:30:6]
                    & ((T.[30:18])×P(DUP))[28:22:20];
         END ELSE IF (N := BASE)=0 THEN N := @2631353020000;
         T := 3 & (N.[10:26]×6137 + 2197513)[10:12:36];
         POLISH((((BASE := T) OR 0.5) - 0.5) + P(DUP), RTN);
   END RANDOM;
```

```
                                                    00022840 T  0000:0
                                                    00022850 T  0000:0
                                                    00022900 T  0000:0
                                    START OF REL SEGMENT; DISK ADDRESS = 00010
                                                    00022925 T  0000:0
                                                    00022950 T  0000:0
                                                    00022975 T  0000:0
                                                    00023000 T  0000:0
                                                    00023025 T  0000:0
                                                    00023100 T  0000:0
                                                    00023150 T  0002:1
                                                    00023200 T  0004:2
                                                    00023250 T  0005:1
                                                    00023300 T  0007:0
                                                    00023350 T  0009:2
                                                    00023400 T  0012:3
                                                    00023450 T  0017:2
                                                    00023500 T  0020:3
                                                    00023550 T  0023:2
                                                          SIZE= 0028 WORDS
```

```
PROCEDURE SWAP;                 % 045
BEGIN
      ARRAY A = -2 [*,*], B = -1 [*,*];
      STREAM(A, B, CA←0, CB←0, FA←A.[18:15], FB←B.[18:15]);
   BEGIN
      SI ← A; CA ← SI;
      SI ← B; CB ← SI;
      DI ← LOC B; DI ← DI+5; SKIP 3 DB;
      SI ← LOC CA; SI ← SI+5; SKIP 3 SB;
      3(IF SB THEN DS ← SET ELSE DS ← RESET; SKIP SB); DS ← 2 CHR;
      DI ← FB; SI ← LOC B; DS ← WDS;
      DI ← LOC A; DI ← DI+5; SKIP 3 DB;
      SI ← LOC CB; SI ← SI+5; SKIP 3 SB;
      3(IF SB THEN DS ← SET ELSE DS ← RESET; SKIP SB); DS ← 2 CHR;
      DI ← FA; SI ← LOC A; DS ← WDS;
   END;
END SWAP;
```

```
                                                    00023600 T  0000:0
                                                    00023700 T  0000:0
                                    START OF REL SEGMENT; DISK ADDRESS = 00011
                                                    00023710 T  0000:0
                                                    00023720 T  0000:0
                                                    00023730 T  0000:0
                                                    00023740 T  0004:0
                                                    00023750 T  0004:0
                                                    00023760 T  0004:2
                                                    00023770 T  0005:0
                                                    00023780 T  0005:3
                                                    00023790 T  0006:2
                                                    00023800 T  0008:3
                                                    00023810 T  0009:2
                                                    00023820 T  0010:1
                                                    00023830 T  0011:0
                                                    00023840 T  0013:1
                                                    00023850 T  0014:0
                                                    00023860 T  0014:1
                                                          SIZE= 0015 WORDS
```

```
COMMENT ALGOL WRITE INTRINSIC;%
PROCEDURE ALGOLWRITE(TEN, FILX, CHSKP, LNSKP, FI, AEXP,                %WF
                     ARRY, LINESKIP, CHANSKIP, SUPRS, NUMWDS, TANK);   %WF
   VALUE     LINESKIP, CHANSKIP, SUPRS, NUMWDS, TANK,                  %WF
```

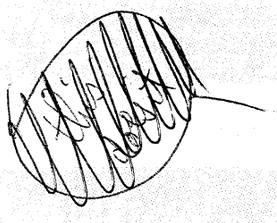
```
                                                    00023900 T  0000:0
                                                    00024000 T  0000:0
                                                    00024100 T  0000:0
                                                    00024200 T  0000:0
                                                    00100000 T  0000:0
                                                    00100100 T  0000:0
                                    START OF REL SEGMENT; DISK ADDRESS = 00012
                                                    00100200 T  0000:0
                                                    00100300 T  0000:0
```

```
                CHSKP, LNSKP, FI, ARRY;                                  %WF
    INTEGER     CHSKP, LNSKP, FI, AEXP,                                  %WF
                LINESKIP, CHANSKIP, NUMWDS, SUPRS;                       %WF
    NAME        FILX, TANK;                                              %WF
    ARRAY       ARRY[*], TEN[*];                                         %WF
BEGIN REAL SELECT=14,REED=13,ADDRESS;%
      NAME MEM=2;%
      LABEL AB,ACTION;
      LABEL DS,WINDUP1;
      ARRAY FPB=3[*],FIB[*],HEADER[*];%
      INTEGER I,RSIZE;%
      INTEGER SPOUT;
    ARRAY       TINK=TANK[*];                                            %WF
    REAL CHNSKP=CHANSKIP;
    REAL        ALGOLWRITE=12;                                           %WF
    DEFINE FNUM = FIB[4].[13:11] #;
      DEFINE IOD=(*TANK)#;%
$ SET OMIT = NOT(TIMESHARING)
SUBROUTINE WAIT; POLISH(TANK, @2000000000, 36, COM, DEL, DEL);
$ POP OMIT
$ SET OMIT = TIMESHARING
      LABEL ERR,LP1,MT1,CLOSED,DK1,SP1,CP1,DC1,PP1;%
    LABEL DCN1,DCN2,SPIN;
$ SET OMIT = NOT SHAREDISK
    SWITCH SW1← ERR,LP1,MT1,CLOSED,DK1,SP1,CP1,LP1,PP1,ERR,DC1,
                ERR,LP1,DCN1;
      LABEL LP2,MT2,DK2,SP2,CP2,DC2,PP2;%
    SWITCH SW2← ERR,LP2,MT2,ERR,DK2,SP2,CP2,LP2,PP2,ERR,DC2,ERR,
                LP2,DCN2;
    LABEL DS1,DR1,DU1;%
    SWITCH DSW1←DS1,DR1,DU1,CLOSED;
    LABEL UT,PBIT,OWT,D19,RELEASE,STA,EXIT,L1,WINDUP,DBIT;%
    LABEL TYPEU,TYPEA,TYPEC;%
    SWITCH TYPE←TYPEU,TYPEA,ERR,TYPEC;%
    LABEL DS2,DR2,DU2;%
    SWITCH DSW2←DS2,DR2,DU2;%
    SUBROUTINE BLOCK;%
    BEGIN GO TO TYPE[I←FIB[5].[46:2]];%
TYPEC:      STREAM(D1←IOD,S←(NUMWDS←NUMWDS+1)×8,%
                D2←(TANK[0]←NUMWDS INX IOD));%
                BEGIN SI←LOC S; DI←DI-8; DS←4 DEC; DI←D1;%
                      SI←D2; SI←SI-8; DI←DI-4; DS←4 CHR;%
                END;%
            IF (FIB[17]←FIB[17]-NUMWDS)>RSIZE+1 THEN BEGIN%
OWT:                FIB[7]←FIB[7]+1; P(XIT);%
TYPEA:      IF (FIB[17]←FIB[17]-RSIZE)≥RSIZE THEN%
              BEGIN TANK[0]←RSIZE INX IOD; GO OWT END END;%
            NUMWDS←FIB[18].[18:15]-FIB[17]+(I=3);%
TYPEU:  END BLOCK;%
     REAL SUBROUTINE DISKADDRESS;%
    BEGIN%
            ADDRESS←(CHANSKIP DIV HEADER[0].[30:12])×HEADER[0].[42:6];%
            IF (SUPRS←ADDRESS DIV HEADER[1]+10)≥30 THEN
                BEGIN P(0); GO TO EXIT END;
            IF HEADER[SUPRS]=0 THEN
            IF HEADER[9]>(SUPRS-10) THEN%
              P(FPB[FNUM+3],FPB[FNUM],FPB[FNUM+1],SUPRS,HEADER,
```

```
00100400 T  0000:0
00100500 T  0000:0
00100600 T  0000:0
00100700 T  0000:0
00100800 T  0000:0
00100900 T  0000:0
00101000 T  0000:0
00101100 T  0000:0
00101200 T  0000:0
00101300 T  0000:0
00101400 T  0000:0
00101450 T  0000:0
00101500 T  0000:0
00101550 T  0000:0
00101600 T  0000:0
00101650 T  0000:0
00101700 T  0000:0
00101750 T  0000:0
00101752 T  0000:0
00101753 T  0004:0
00101799 T  0004:0
00101900 T  0004:0
00101910 T  0004:0
00101919 T  0004:0
00102000 T  0004:0
00102010 T  0004:0
00102100 T  0004:0
00102200 T  0004:0
00102210 T  0004:0
00102300 T  0004:0
00102400 T  0004:0
00102500 T  0004:0
00102600 T  0004:0
00102700 T  0004:0
00102800 T  0004:0
00102900 T  0004:0
00103000 T  0004:0
00103100 T  0004:0
00103200 T  0008:1
00103300 T  0010:3
00103400 T  0012:2
00103500 T  0013:2
00103600 T  0014:2
00103700 T  0014:3
00103800 T  0018:1
00103900 T  0020:2
00104000 T  0023:0
00104100 T  0025:2
00104200 T  0029:0
00104300 T  0029:1
00104400 T  0030:0
00104500 T  0030:0
00104600 T  0033:1
00104700 T  0035:3
00104800 T  0037:0
00104900 T  0038:0
00105000 T  0040:0
```

```
                    4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL) ELSE                          00105050 T  0046:0
                   BEGIN P(0); GO TO EXIT END;%                                     00105100 T  0048:1
              ADDRESS←HEADER[SUPRS]+SUPRS←ADDRESS MOD HEADER[1];%                   00105200 T  0049:2
              STREAM(D←[ADDRESS]); BEGIN SI←D; DS←8 DEC END; P(1);%                 00105300 T  0052:1
         EXIT: DISKADDRESS←P;%                                                      00105400 T  0054:0
         END DISKADDRESS;%                                                          00105500 T  0054:1
                 IF TINK=0 THEN                                          %WF        00105600 T  0054:2
                 BEGIN FIB ← FILX[NOT 2];                                %WF        00105700 T  0057:2
                 IF FIB[5].[11:2]<2 THEN P(MKS,"WRITNG",FILX,7,SELECT) ;            00105703 T  0059:3
                      IF FIB[5].[43:1] THEN                                         00105710 T  0063:0
                        P(MKS, CHSKP, 0, FILX, 1, SELECT);                          00105720 T  0064:0
                      IF LNSKP>1 AND ARRY≤0 AND (I←FIB[4].[8:4])≠1                  00105750 T  0066:0
$ SET OMIT = NOT(TIMESHARING)                                                       00105752 T  0069:2
                      AND I≠7 AND I≠12 AND I≠10 THEN                                00105753 T  0069:2
$ SET OMIT = TIMESHARING                                                            00105754 T  0073:1
                      P(XIT);%CARRIAGE CONTROL ON NON-PRINTER FILE                  00105760 T  0073:1
                                                                                    00105800 T  0074:0
                                                                                    00105900 T  0074:0
                      RSIZE ← P(MKS, LNSKP, CHSKP, SUPRS,                %WF      00106000 T  0074:0
                                (-1), FILX, ALGOLWRITE);                 %WF      00106100 T  0075:0
                      IF ARRY≤0 THEN SUPRS ← 1 ELSE                      %WF      00106200 T  0076:2
                      BEGIN %   11/24/72 - CORRECTED 10/3/73                        00106300 P  0078:3
                           IF ARRY.[8:10]=P(DUP,0) THEN % INDEXED WRITE             00106320 C  0081:0
                             P(DEL,AEXP) % WRITE MIN(AEXP,RSIZE) WORDS              00106340 C  0082:3
                            ELSE % WRITE MIN(ARRY SIZE,AEXP,RSIZE) WORDS            00106360 C  0083:3
                           IF P GTR P(DUP,AEXP) %                                   00106380 C  0083:3
                             THEN P(DEL,AEXP);                             %WF      00106400 T  0084:3
                           IF P(DUP)≥RSIZE THEN P(DEL) ELSE RSIZE ← P;     %WF      00106500 T  0086:0
                                STREAM(P4 ← [ARRY[0]], P3 ← RSIZE,         %WF      00106600 T  0088:2
                                       P2 ← P(DUP).[36:6], P1 ← *FILX);    %WF      00106700 T  0089:2
                                BEGIN SI ← P4;  DS ← P3 WDS;                %WF      00106800 T  0091:0
                                    P2(DS ← 32 WDS;  DS ← 32 WDS);          %WF      00106900 T  0091:3
                                END;                                        %WF      00107000 T  0093:0
                           END;                                             %WF      00107100 T  0093:1
                           IF RSIZE>0 THEN P(MKS, LNSKP,                    %WF      00107200 T  0093:1
                             CHSKP, SUPRS, RSIZE, FILX, ALGOLWRITE);        %WF      00107300 T  0095:0
                           FILX[NOT 4] ← FILX[NOT 3] ← 0;                   %WF      00107400 T  0096:1
                           P(XIT);                                          %WF      00107500 T  0099:2
                      END;                                                  %WF      00107600 T  0099:3
                 FIB←TANK[NOT 2];%                                                  00107700 T  0099:3
         UT:     I←FIB[4].[8:4]; RSIZE←FIB[18].[33:15];%                            00107800 T  0101:2
                 SPOUT:=(I=5);                                                      00107820 T  0104:2
          $ SET OMIT = TIMESHARING                                                  00107840 T  0105:3
                 IF CHNSKP.[4:1] THEN                                               00107860 T  0105:3
                 BEGIN CHNSKP.[4:1]←0;                                              00107870 T  0106:2
          $ SET OMIT = NOT SHAREDISK                                                00107879 T  0108:3
                 END;                                                               00107890 T  0108:3
                 IF NUMWDS<0 THEN GO TO SW1[I]; GO TO SW2[I];%                      00107900 T  0108:3
          LP1: MT1: SP1: CP1: PP1:                                                  00108000 T  0126:0
         %                                                                          00108100 T  0126:0
         D19:    IF IOD.[19:1] THEN%                                                00108200 T  0126:0
         PBIT:   IF IOD.[2:1] THEN P(RSIZE,RTN) ELSE%                               00108300 T  0127:0
                 IF IOD.[25:1] THEN%                                                00108400 T  0129:2
         CLOSED:     BEGIN                                                          00108410 T  0131:0
                          FIB[13].[27:1]←0;                                         00108420 T  0131:2
                          IF (I←(FPB[FNUM+3] AND 31))≠10 AND I≠12                   00108430 T  0134:0
                          AND I≠13 AND I≠26 THEN FIB[5].[45:1]←0 ELSE               00108440 T  0137:2
```

```
                    FIB[5].[45:1]←P(TANK[NOT 3],DUP)≠0 AND P(XCH)≠15;                    00108450 T  0143:1
                    P(TANK,0,11,COM,DEL,DEL) ;                                          00108510 T  0149:0
                    IF NOT FIB[5].[45:1] THEN GO UT ;                                   00108515 T  0150:2
                    P(TANK[NOT 3]);  TANK[NOT 3]←TANK[NOT 4]←0 ;                        00108520 T  0152:0
                    P(MKS,9,BLKCNTRL,DEL) ;% TAKE PARITY ACTION LBL BRNCH.              00108525 T  0156:2
                    P(1); GO TO DS;                                                     00108530 T  0157:2
               END ELSE                                                                 00108535 T  0158:1
          IF IOD.[27:1] AND (I=2 OR I=7 OR I=8) THEN%                                   00108600 T  0158:1
               BEGIN IF NOT FIB[4].[2:1] THEN%                                         00108700 T  0162:3
                         BEGIN HEADER←TANK[NOT 1];HEADER[4].[42:6]←1 END;              00108800 T  0164:2
                     IF I=7 THEN FIB[9].[1:1]←1; % MULTI-REEL PBT FILE                 00108850 T  0169:1
                     I←FIB[13].[28:10]+1;%                                             00108900 T  0173:0
                     P(MKS,6,0,(NOT 2) INX TANK,4,SELECT);%                            00109000 T  0175:0
                     FIB[13].[28:10]←I; GO TO CLOSED;%                                 00109100 T  0177:1
               END ELSE%                                                               00109200 T  0180:1
               BEGIN                                                                   00109300 T  0180:1
ERR:                P(3);                                                            00109310 T  0180:3
DS:                 P(TANK,XCH,11,COM);                                              00109320 T  0181:0
               END;                                                                    00109330 T  0182:0
          WAIT; GO TO PBIT;%                                                           00109400 T  0182:0
DK1:      HEADER←*[FIB[14]]; GO TO DSW1[FIB[4].[27:3]];%                               00109500 T  0183:2
DK2:      HEADER←*[FIB[14]]; GO TO DSW2[FIB[4].[27:3]];%                               00109600 T  0188:2
CP2:      BLOCK; TANK[0]←FLAG(FIB[16])&CHANSKIP[32:47:1]; GO TO RELEASE;%              00109700 T  0193:0
LP2:      IF SUPRS THEN STREAM(RSIZE,D←IOD); BEGIN RSIZE(DS←8 LIT " ") END;          00109800 T  0196:3
          CHANSKIP←CHANSKIP+LINESKIP.[45:1];                                           00109850 T  0201:0
          IF CHANSKIP≠0 THEN%                                                          00109900 T  0202:3
          BEGIN IF (I←FIB[17]-RSIZE)>0 THEN%                                           00110000 T  0203:2
               STREAM(I,D←RSIZE INX IOD); BEGIN I(DS←8 LIT " ") END;%                  00110100 T  0206:0
          END ELSE BLOCK;%                                                             00110200 T  0210:2
          TANK[0]←FLAG(FIB[16])&LINESKIP[27:47:1]&LINESKIP[28:46:1]%                   00110300 T  0212:0
                  &CHANSKIP[29:44:4]&NUMWDS[8:38:10];%                                 00110400 T  0214:0
          GO TO RELEASE;%                                                              00110500 T  0217:1
SP2: PP2:%                                                                             00110600 T  0217:3
MT2:      BLOCK;%                                                                      00110700 T  0217:3
          P(TANK[0]←FLAG(FIB[16])&NUMWDS[8:38:10],NUMWDS,XCH,INX,%                     00110800 T  0219:0
            @3700000000000000,XCH,←);%                                                 00110900 T  0222:0
          IF SPOUT THEN % SPO OUTPUT                                                   00110910 T  0222:3
               IF FPB[FNUM+3].[42:6]=43 THEN P(XIT) ELSE %DUMMY                        00110920 C  0223:0
               P(0,0,NOT,IOD,INX,15,COM,XIT)                                           00110940 T  0227:0
               ELSE                                                                    00110990 T  0231:1
RELEASE:        P(FLAG(FIB[19])&IOD[3:3:5],TANK,PRL,DEL);%                             00111000 T  0231:1
WINDUP:         I←FIB[19].[33:15]-FIB[16].[33:15];%                                    00111100 T  0234:3
                FIB[16].[33:15]←SUPRS←MEM[P(DUP) INX NOT 1].[18:15];%                  00111200 T  0237:2
                FIB[19].[33:15]←SUPRS+I;%                                              00111300 T  0242:1
           WINDUP1:                                                                    00111400 T  0244:3
                FIB[6]←FIB[6]+1; FIB[7]←FIB[7]+1; FIB[17]←FIB[18].[18:15];%           00111500 T  0244:3
                P(XIT);%                                                               00111600 T  0251:0
DU1:%                                                                                  00111700 T  0251:1
DS1:      IF LINESKIP≠0 THEN%                                                          00111800 T  0251:1
          BEGIN IF IOD.[27:1] AND IOD.[19:1] THEN GO AB;                               00111900 T  0252:0
                IF FIB[17]=FIB[18].[18:15] THEN                                        00111950 T  0255:2
                BEGIN CHANSKIP←FIB[7];%                                                00112000 T  0257:2
L1:                   IF DISKADDRESS THEN%                                             00112100 T  0259:0
                      IF IOD.[19:1] THEN DBIT: IF IOD.[2:1] THEN%                      00112200 T  0260:0
                      BEGIN                                                            00112300 T  0263:0
 $ SET OMIT = NOT SHAREDISK                                                            00112309 T  0263:2
                           MEM[FIB[16]]←ADDRESS;                                       00112340 T  0263:2
```



```
                        P(RSIZE,RTN);                                                  00112350 T  0265:1
                    END ELSE                                                           00112360 T  0265:3
                    IF IOD.[25:1] THEN GO TO CLOSED ELSE                               00112400 T  0265:3
$ SET OMIT = NOT SHAREDISK                                                             00112409 T  0267:1
                    BEGIN                                                              00112420 T  0267:1
$ SET OMIT = NOT SHAREDISK                                                             00112429 T  0268:0
                        GO TO AB;                                                      00112440 T  0268:0
                    END ELSE                                                           00112450 T  0268:2
                    BEGIN WAIT; GO TO DBIT; END ELSE                                   00112460 T  0268:2
                    BEGIN                                                              00112470 T  0270:2
$ SET OMIT = NOT SHAREDISK                                                             00112479 T  0271:0
                        GO TO AB;                                                      00112490 T  0271:0
                    END;                                                               00112500 T  0271:2
            END;  P(RSIZE,RTN);%                                                       00112600 T  0271:2
        END;%                                                                          00112700 T  0272:0
        P(MKS,CHANSKIP,4,TANK,1,SELECT); GO TO L1;                                     00112800 T  0272:0
DS2:    IF FIB[7]>HEADER[7] THEN HEADER[7]←FIB[7];%                                    00112900 T  0274:0
        BLOCK; TANK[0]←FLAG(FIB[16]); GO RELEASE;%                                     00113000 T  0277:1
DR1:    IF LINESKIP≠0 THEN CHANSKIP←FIB[7] ELSE FIB[7]←CHANSKIP;%                      00113100 T  0279:3
        IF HEADER[7]<CHANSKIP THEN HEADER[7]←CHANSKIP;%                                00113200 T  0283:3
 $ SET OMIT = NOT SHAREDISK                                                            00113249 T  0286:2
        IF FIB[5].[46:2]=0 THEN GO TO L1;%                                             00113300 T  0286:2
        IF DISKADDRESS THEN%                                                           00113400 T  0288:2
        BEGIN FIB[16].[24:1]←1;%                                                       00113500 T  0290:0
 $ SET OMIT = SHAREDISK                                                                00113599 T  0293:0
            P(MKS,CHANSKIP+1,1,TANK,REED,RTN);%                                        00113600 T  0293:0
 $ POP OMIT                                                                            00113601 T  0295:0
 $ SET OMIT = NOT SHAREDISK                                                            00113649 T  0295:0
        END;%                                                                          00113700 T  0295:0
 $ SET OMIT = NOT SHAREDISK                                                            00113749 T  0295:0
        GO TO AB;                                                                      00113800 T  0295:0
DR2:                                                                                   00113900 T  0295:2
 $ SET OMIT = NOT SHAREDISK                                                            00113909 T  0295:2
        TANK[0]←FLAG(FIB[16])&0[24:24:1];                                              00113980 T  0295:2
        P(FLAG(FIB[19])&IOD[3:3:5]&1[27:47:1],TANK,PRL,DEL);%                          00114000 T  0297:3
 $ SET OMIT = NOT SHAREDISK                                                            00114049 T  0301:3
        GO TO WINDUP;%                                                                 00114100 T  0301:3
DU2:: FIB[5].[43:2]←2;%                                                                00114300 T  0302:1
        IF FIB[7]>HEADER[7] THEN HEADER[7]←FIB[7];%                                    00114400 T  0305:2
        BLOCK;%                                                                        00114500 T  0308:3
        CHANSKIP←FIB[7]+FIB[13].[10:9]×HEADER[0].[30:12];%                             00114600 T  0310:0
        IF DISKADDRESS THEN%                                                           00114700 T  0313:2
        BEGIN P(TANK[0]←FLAG(FIB[16])&0[24:24:1],(NOT 0),XCH,INX,%                     00114800 T  0315:0
              ADDRESS,XCH,←);%                                                         00114900 T  0318:3
            P(FLAG(FIB[19])&1[24:47:1],TANK,PRL,DEL);%                                 00115000 T  0319:2
        END ELSE%                                                                      00115100 T  0322:1
        BEGIN TANK[0]←FLAG(FIB[16])&0[24:24:1];%                                       00115200 T  0322:1
            P(FLAG(FIB[19])&1[24:44:4],TANK,PRL,DEL);%                                 00115300 T  0325:0
        END;%                                                                          00115400 T  0327:3
        GO TO WINDUP;%                                                                 00115500 T  0327:3
 $ SET OMIT = NOT(TIMESHARING)                                                         00115501 T  0328:1
 DC1::   P(RSIZE, RTN);                                                                00115510 T  0328:1
 $ SET OMIT = TIMESHARING                                                              00115590 T  0329:2
 AB::    BEGIN IF(ADDRESS←TANK[NOT 4])=0 THEN GO ERR;                                  00115800 T  0329:2
 ACTION::    TANK[NOT 3]←TANK[NOT 4] ←0;                                               00115900 T  0332:3
             TANK[0] := IOD OR MEM;                                                    00116000 T  0336:1
             P(ADDRESS,MKS,9,JUNK); GO TO ERR;                                         00116100 T  0337:3
```

```
            END;
            IF TANK[NOT 4]=0 THEN BEGIN WAIT; GO TO DC1 END;
            IF @(CHANSKIP,[CF]×60,CHANSKIP,TANK,18,11,COM,DEL,DEL,DEL)
                 THEN P(RSIZE,RTN);
            GO TO AB;
 $ SET OMIT = NOT(TIMESHARING)
DC2:  IF CHANSKIP.[CF] NEQ 0 THEN CHANSKIP:=ABS(CHANSKIP&1[CTF])
        ELSE CHANSKIP ← (8×(LINESKIP=4)+LINESKIP.[45:3])
                                &LINESKIP[32:43:1];
            NUMWDS ← IF SUPRS THEN 0 ELSE 8×NUMWDS;
            POLISH(IOD, NUMWDS, CHANSKIP, 0, (-11), COM, DEL);
            I:=POLISH+1;
            ADDRESS:=TANK[NOT(4-(I=2))];
            TANK[NOT 4]:=TANK[NOT 3]:=0;
            IF I THEN P(XIT);
            IF ADDRESS NEQ 0 THEN
               P(ADDRESS,MKS,9,BLKCNTRL);
            ADDRESS:=1+((I=0)×2);
            P(TANK,ADDRESS,11,COM);
DCN1:DCN2:SPIN:    P(XIT);
 $ SET OMIT = TIMESHARING
END ALGOLWRITE;
```

```
00116200 T  0339:1
00116300 T  0339:1
00116400 T  0343:2
00116500 T  0346:3
00116600 T  0347:3
00116601 T  0348:1
00116610 T  0348:1
00116620 T  0350:2
00116630 T  0353:1
00116635 T  0355:3
00116640 T  0358:2
00116644 C  0360:3
00116646 C  0361:3
00116648 C  0364:2
00116650 P  0367:3
00116652 C  0368:3
00116654 C  0369:2
00116656 C  0371:0
00116658 C  0373:1
00116660 T  0374:1
00116690 T  0375:1
00118800 T  0375:1
                SIZE= 0376 WORDS
```

```
PROCEDURE OUTPUTINT(TEN,FILX,CHSKP,LNSKP,FI,FRMT,LISX);%
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00025
      COMMENT ESPOL VERSION OF ALGOL WRITE INTRINSIC%
              BY L.R. GUCK  12/1/64;%
  VALUE     CHSKP,LNSKP,FI,LISX;%
  NAME      FILX;%
  ARRAY     TEN[*],%
            FRMT[*];%
  REAL LISX;%
  INTEGER   CHSKP,LNSKP,FI;%
     BEGIN%
  REAL      ALGOLWRITE=12;%
  ARRAY REALROW=TEN-1[*];%
  REAL      SELECT=14;%
  REAL      JUNK2=9;%
  INTEGER   V2=1 ;
  INTEGER   TEMPD=7 ;
  INTEGER   JUNK1=17;%
  INTEGER   LSTRN=19;%
  INTEGER   AEXP =FRMT;%
  ARRAY     ARRY =LISX[*];%
  INTEGER   TLSTRN=+1 ;%
  REAL      UTYP = TLSTRN+1;
  DEFINE    UTIP =UTYP.[47:1] #, %%%% USED FOR NON-BOOLEAN USE OF UTYP
            FFTYP=UTYP.[46:1] #, %%%% FLAG TO SHOW USING FREE FIELD.
            STORW=UTYP.[40:6] #, %%%% USED TO STORE ORIG VALUE OF W.
            UES  =UTYP.[39:1] #, %%%% FLAG TO INCLUDE EXPONENT SIGN.
            UDC  =UTYP.[38:1] #, %%%% FLAG TO INCLUDE DECIMAL POINT.
            UED  =UTYP.[36:2] #, %%%% NUMBER OF EXPONENT DIGITS.
            UMD  =UTYP.[35:1] #, %%%% FLAG TO INCLUDE MANTISSA.
```

```
00200000 T  0000:0
00200100 T  0000:0
00200200 T  0000:0
00200300 T  0000:0
00200400 T  0000:0
00200500 T  0000:0
00200600 T  0000:0
00200700 T  0000:0
00200800 T  0000:0
00200900 T  0000:0
00201000 T  0000:0
00201100 T  0000:0
00201200 T  0000:0
00201300 T  0000:0
00201310 T  0000:0
00201320 T  0000:0
00201400 T  0000:0
00201500 T  0000:0
00201600 T  0000:0
00201700 T  0000:0
00201800 T  0000:0
00201900 T  0000:0
00201905 T  0000:0
00201906 T  0000:0
00201907 T  0000:0
00201908 T  0000:0
00201909 T  0000:0
00201910 T  0000:0
00201911 T  0000:0
```

```
                STORD=UTYP.[29:6] #, %%%% USED TO STORE ORIG VALUE OF D.          00201912 T  0000:0
                UBUFF=UTYP.[16:13]#, %%%% ADJUSTED BUFFER SIZE.                   00201913 T  0000:0
                UTOP =UTYP.[15:1] #, %%%% FLAG TO INCLUDE TRAILING BLANK.        00201914 T  0000:0
                USKIP=UTYP.[09:6] #, %%%% # XTRA LEADING BLANKS FOR I OR F       00201915 T  0000:0
                FFCHR=UTYP.[03:6] #; %%%% FREE FIELD DELIMITER (, OR BLNK)       00201917 T  0000:0
INTEGER         SUPRS = UTYP+1;                                                  00201920 T  0000:0
REAL            BUFF=SUPRS+1;%                                                   00202000 T  0000:0
INTEGER         BSIZE=BUFF+1;%                                                   00202100 T  0000:0
ARRAY           FIB=BSIZE+1[*];%                                                 00202200 T  0000:0
REAL            WH2=FIB+1;%                                                      00202300 T  0000:0
REAL            WH1=WH2+1;%                                                      00202400 T  0000:0
REAL            DH1=WH1+1;%                                                      00202500 T  0000:0
INTEGER         DH2=DH1+1;%                                                      00202600 T  0000:0
REAL            W=DH2+1;%                                                        00202700 T  0000:0
REAL            W1=W+1;%                                                         00202800 T  0000:0
REAL            W2=W1+1;%                                                        00202900 T  0000:0
REAL            WT=W2+1;%                                                        00203000 T  0000:0
REAL            D=WT+1;%                                                         00203100 T  0000:0
REAL            D1=D+1;%                                                         00203200 T  0000:0
REAL            D2=D1+1;%                                                        00203300 T  0000:0
REAL            DA=D2+1;%                                                        00203400 T  0000:0
REAL            SKIP=DA+1;%                                                      00203500 T  0000:0
REAL            CHR=SKIP+1;%                                                     00203600 T  0000:0
INTEGER         E=CHR+1;%                                                        00203700 T  0000:0
REAL            ZEROS=E+1;%                                                      00203800 T  0000:0
REAL            CODE=ZEROS+1;%                                                   00203900 T  0000:0
REAL            FAW=CODE+1;%                                                     00204000 T  0000:0
REAL            SGN=FAW+1;%                                                      00204100 T  0000:0
INTEGER         SCFTR=SGN+1;%                                                    00204200 T  0000:0
REAL TPHRASE=SCFTR+1;                                                            00204210 T  0000:0
INTEGER LZ = TPHRASE + 1;                                                        00204220 C  0000:0
LABEL           RTNPRNT,EFA,EFC,EERTN,RNA,RNB,%                                  00204300 T  0000:0
                START,ISFRM,AEXL,ISA,ISB,ASLST,BS,BR,BB,ERROR,%                  00204400 T  0000:0
                FMOUT,S1,S,LFPAR,RTPAR,SLASH,SCALE,STRNG,%                       00204500 T  0000:0
                PHRAS,INLOOP,ASTB,ASTA,AST,FLDW,JMP,%                            00204600 T  0000:0
                LOGI,ALFA,DOTYPE,XTYPE,ITYPE,%                                   00204700 T  0000:0
                FTYPE,RFIN,FA,FB,FC,FD,UTYPE,UI,UF,ESUBTYPE,ETYPE1,COMMM,        00204800 T  0000:0
                FORMATERR,TTYPE,BACK,                                            00204810 T  0000:0
                ETYPE,REIN,EA,ERTN,REOT,EB,MAXN,TEN8,%                           00204900 T  0000:0
                RTYPE,RC,TRYE,RRTN,MAXM,COMM;%                                   00205000 T  0000:0
      COMMENT LABELS ARE LISTED IN SAME ORDER THEY APPEAR;%                      00205100 T  0000:0
DEFINE L0G8 = @1157163034761674#,%                                               00205200 T  0000:0
       MAX  =@0007777777777777#,%                                                00205300 T  0000:0
       SAVEBUFF=TPHRASE.[30:18]#,                                                00205305 T  0000:0
       MAXCHR  =TPHRASE.[18:12]#,                                                00205310 T  0000:0
       P    = POLISH#;%                                                          00205400 T  0000:0
SUBROUTINE CKPB;%                                                                00205500 T  0000:0
     BEGIN%                                                                      00205600 T  0001:0
          IF FILX.[18:15] ≤ 1 THEN%                                              00205700 T  0001:0
               BEGIN IF NOT FILX.[18:15] THEN%                                   00205800 T  0002:1
                    BEGIN;STREAM(A←[REALROW[0]]:B←0);%                           00205900 T  0003:3
                         BEGIN SI←A; DI←A; SI←SI-16;                             00206000 T  0005:3
                              SKIP 2 SB;%                                        00206100 T  0006:2
                              IF SB THEN TALLY ← 1;%                             00206200 T  0006:3
                              A ← TALLY;%                                        00206300 T  0007:2
                         END;%                                                   00206400 T  0007:3
                    IF NOT P THEN%                                               00206500 T  0008:0
```

```
                                        BEGIN P(FILX,14,COM,DEL);%                    00206600 T  0008:1
                                             FILX.[18:15] ← 1;%                       00206700 T  0009:3
                                        END;%                                         00206800 T  0011:0
                                   END;%                                              00206900 T  0011:0
                              BSIZE ← REALROW.[8:10];%                                00207000 T  0011:0
                         END ELSE%                                                    00207100 T  0012:2
                    BSIZE←POLISH(MKS,LNSKP,CHSKP,SUPRS,(-1),FILX,ALGOLWRITE);        00207200 T  0012:2
                    BUFF←(*FILX)&BSIZE[8:38:10] ;                                    00207300 T  0015:2
               END;%                                                                 00207400 T  0017:2
          SUBROUTINE PRNT;%                                                          00207500 T  0017:3
               BEGIN COMMENT RELEASE BUFFER;%                                        00207600 T  0018:0
                    COMMENT S= RETURN LITERAL.%                                      00207700 T  0018:0
                              S=1 = IF TRUE THEN RETURN AFTER RELEASE.%              00207800 T  0018:0
                                 IF FALSE THEN EXIT;%                                00207900 T  0018:0
                    P(XCH);%                                                         00208000 T  0018:0
                    IF FILX.[18:15] > 1 THEN%                                        00208100 T  0018:1
                         IF BSIZE>0 THEN                                             00208200 T  0019:2
                         POLISH(MKS,LNSKP,CHSKP,SUPRS,BSIZE,FILX,ALGOLWRITE);        00208300 T  0020:3
                         COMMENT WRITE RELEASE;%                                     00208400 T  0023:0
                         IF P THEN CKPB%                                             00208500 T  0023:0
                         ELSE BEGIN LSTRN ← TLSTRN;%                                 00208600 T  0024:1
                                   IF FILX.[18:15]>1 THEN                            00208700 T  0026:1
                                      FILX[NOT 4]←FILX[NOT 3]←0 ELSE                 00208800 T  0027:2
                                   IF FILX.[18:15] = 1 THEN%                         00208900 T  0031:1
                                        P(FILX,14,COM);%                             00209000 T  0033:0
                                   P(XIT);%                                          00209100 T  0034:1
                              END;%                                                  00209200 T  0034:2
          RTNPRNT:END;%                                                              00209300 T  0034:2
          SUBROUTINE DEBLANK; IF CHR<132 THEN                                        00209310 T  0034:3
               STREAM(P3←P(BSIZE-CHR,DUP),P2←P DIV 64,P1←BUFF) ;                     00209320 T  0035:3
                    BEGIN P2(2(DS←32LIT" ")); P3(DS←LIT" ") END ;                    00209330 T  0038:2
          SUBROUTINE PRNTA;%                                                         00209400 T  0045:3
               BEGIN COMMENT BLANK TO END OF BUFFER OR TO 132 TH CHARACTER,%          00209500 T  0046:0
                              WHICH EVER IS LESS;%                                   00209600 T  0046:0
                    P(XCH);    COMMENT S= XIT KEY IF TRUE THEN RETURN.%              00209700 T  0046:0
                                        IF FALSE THEN EXIT ;%                        00209800 T  0046:1
                    IF TPHRASE>0 THEN DEBLANK ELSE CHR←MAXCHR ;                      00209900 T  0046:1
     %VOID                                                                           00210000 T  0050:3
     %VOID                                                                           00210100 T  0050:3
     %VOID                                                                           00210200 T  0050:3
     %VOID                                                                           00210300 T  0050:3
     %VOID                                                                           00210400 T  0050:3
     %VOID                                                                           00210500 T  0050:3
     %VOID                                                                           00210600 T  0050:3
               BSIZE←(IF CHR=0 THEN BSIZE ELSE CHR+7) DIV 8;                         00210700 T  0050:3
                    PRNT; COMMENT RELEASE BUFFER;%                                   00210800 T  0054:2
                    CHR ← 0;%                                                        00210900 T  0056:0
                    BSIZE ← BSIZE × 8;%                                              00211000 T  0056:3
                    TPHRASE←BUFF←P(,BUFF,LOD,0,INX) ;                                00211100 T  0058:0
               END;%                                                                 00211200 T  0060:0
          SUBROUTINE FINDE;%                                                         00211300 T  0060:1
               BEGIN COMMENT DETERMINE THE EXPONENT OF A REAL NUMBER;%                00211400 T  0061:0
                  IF WH1 = (LZ←ZEROS←0) THEN GO TO EFC;                              00211500 P  0061:0
                    E ← (( 0&WH1[42:3:6]&WH1[1:2:1] + 12) × LOG8) + .5 ;%            00211600 T  0063:1
          EFA:      IF ABS(WH1) ≥ ( IF  E≥ 0 THEN TEN[E]%                            00211700 T  0067:2
                                             ELSE 1/TEN[-E])%                        00211800 T  0069:2
                         THEN GO TO EERTN;%                                          00211900 T  0071:2
```

```
          E ← E - 1;%                                                                    00212000 T  0072:1
          GO TO EFA;%                                                                    00212100 T  0073:2
EFC:      E ← 0;%                                                                        00212200 T  0076:0
EERTN:END;%                                                                              00212300 T  0076:3
SUBROUTINE RNDOFF;%                                                                      00212400 T  0077:0
     COMMENT ADJUST NUMBER TO 12 SIGNIFICAT DIGITS PLUS%                                 00212500 T  0077:0
             TRAILING ZEROS. NOTE DA = ADJUSTED <DECIMAL PLACES>;%                       00212600 T  0077:0
RNA: BEGIN IF ABS(P((JUNK2 ← TEN[DA]) × WH1,DUP)) ≤ MAX%                                 00212700 T  0077:0
           THEN GO TO RNB;   COMMENT DA = ADJUSTED DECIMAL PLACES;                       00212800 T  0079:1
          P(DEL);%                                                                       00212900 T  0080:0
          ZEROS ← ZEROS +1; COMMENT TRAILING ZEROS +1;%                                  00213000 T  0080:1
          DA  ← DA-1;    COMMENT SUBTRACT 1 FROM DECIMAL PLACES;%                        00213100 T  0081:2
          GO TO RNA;%                                                                    00213200 T  0082:3
RNB:      DH2 ← P;       COMMENT ROUND OFF NUMBER;%                                     00213300 T  0085:0
     END;%                                                                               00213400 T  0085:2
REAL SUBROUTINE LISTELEMENT;                                                             00213500 T  0085:3
          BEGIN IF LSTRN<0 THEN GO TO ERROR;                                             00213600 T  0086:0
                P(WH1,,WH1,ISN); WH1←LISX;                                               00213700 T  0087:1
                LISTELEMENT←P;                                                           00213800 T  0088:3
          END LISTELEMENT;                                                               00213900 T  0089:0
SUBROUTINE SETMAXCHR; IF MAXCHR<CHR THEN MAXCHR←CHR ;                                    00213910 T  0089:1
LABEL L,X,A,Z,I,G,R,C,O,ZW2,ZD,SWT;                                                      00214000 T  0093:3
     COMMENT START OF CODE;%                                                             00214100 T  0093:3
START::   P(LSTRN,0,0,0);%                                                               00214200 T  0093:3
          P(0);                                                                          00214210 T  0095:0
          IF FILX.[18:15] > 1 THEN%                                                      00214300 T  0095:1
     BEGIN P(FILX[NOT 2]);%                                                              00214400 T  0096:2
          IF FIB[5].[11:2]<2 THEN P(MKS,"WRITNG",FILX,7,SELECT) ;                        00214420 T  0098:1
          IF FIB[5].[43:1] THEN POLISH(MKS,CHSKP,0,FILX,1,SELECT);%                      00214500 T  0101:2
               COMMENT CALL SELECT IF FILE NOT IN WRITE STATUS;%                         00214600 T  0104:2
          END ELSE P(0);%                                                                00214700 T  0104:2
          CKPB; COMMENT CHECK FOR PRESENCE BIT;%                                         00214800 T  0106:1
     COMMENT CHECK FOR TYPE OF WRITE;%                                                   00214900 T  0107:0
          IF FRMT ≠ 0 THEN GO TO ISFRM;%                                                 00215000 T  0107:0
          IF ARRY ≠ 0 THEN%                                                              00215100 T  0108:2
               IF ARRY < 0 THEN GO TO ASLST%                                             00215200 T  0109:2
                           ELSE GO TO AEXL;%                                             00215300 T  0111:0
     COMMENT CASE = FORMAT = LIST = EMPTY;%                                              00215400 T  0112:0
               SUPRS←1;%                                                                 00215500 T  0112:0
          GO TO BB;%                                                                     00215600 T  0112:3
ISFRM:    IF NOT P(FRMT,TOP,XCH,DEL) THEN GO TO FMOUT;%                                  00215700 T  0113:1
          IF FI≠0 THEN %%% FREE FIELD: [FI]/[TEMPD=AEXP-1],                              00215710 T  0115:0
               BEGIN TEMPD←AEXP-1;  FRMT←0;  GO TO FMOUT END ;                           00215720 T  0115:3
     COMMENT CASE = AEXP,ARRAY ROW;%                                                     00215800 T  0118:3
AEXL:     IF P(ARRY,[8:10],DUP) ≤ AEXP THEN GO TO ISA;%                                  00215900 T  0118:3
          IF AEXP<0 THEN P(ARRY[AEXP]) ;                                                 00216000 T  0121:0
          P(DEL,AEXP);%                                                                  00216100 T  0122:3
               COMMENT STACK ← SMALLER OF ARRAY SIZE OR AEXP;%                           00216200 T  0123:1
ISA:      IF P(DUP) ≤ BSIZE THEN GO TO ISB;%                                             00216300 T  0123:1
          P(DEL,BSIZE);%                                                                 00216400 T  0124:2
               COMMENT STACK ← SMALLEST OF BUFFER SIZE, ARRAY SIZE%                      00216500 T  0125:0
                    OR AEXP;%                                                            00216600 T  0125:0
ISB:      BSIZE ← P;%                                                                    00216700 T  0125:0
               COMMENT BSIZE←# OF WORDS TO TRANSFER;%                                    00216800 T  0125:2
     COMMENT TRANSFER ARRAY TO BUFFER;%                                                  00216900 T  0125:2
          STREAM(P4 ← [ARRY[0]], P3 ← BSIZE,%                                            00217000 T  0125:2
               P2 ← BSIZE DIV 64, P1 ← * FILX);%                                         00217100 T  0126:2
```

```
          BEGIN%                                                                   00217200 T  0128:0
               SI ← P4;%                                                           00217300 T  0128:0
               P2 (DS ← 32 WDS;%                                                   00217400 T  0128:1
                  DS ← 32 WDS));%                                                  00217500 T  0129:0
               DS ← P3 WDS;%                                                       00217600 T  0129:2
                    END;%                                                          00217700 T  0130:0
          GO TO BB;%                                                               00217800 T  0130:1
          COMMENT CASE =  *,LIST;%                                                 00217900 T  0130:3
ASLST:         P(0,.LSTRN,SND);              COMMENT S=I=0;%                       00218000 T  0130:3
BS:            P(DUP,LISX);                  COMMENT S=VALUE,S-1=I,S-2=I;%         00218100 T  0131:2
               IF LSTRN ≥ 0 THEN GO TO BR;%                                        00218200 T  0132:0
               BSIZE ← P(DEL,DEL);  COMMENT BSIZE ← I =%                           00218300 T  0133:1
                                                 # OF WORDS IN BUFFER;%            00218400 T  0134:1
               GO TO BB;%                                                          00218500 T  0134:1
BR:            P(XCH);                       COMMENT S=I,S-1=VALUE,S-2=I;%         00218600 T  0134:3
               P([BUFF],STD);                COMMENT VALUE TO BUFFER[I];%          00218700 T  0135:0
               IF P(1,+,DUP) < BSIZE%                                              00218800 T  0135:2
                    THEN GO TO BS;%                                                00218900 T  0136:1
               BSIZE ← P;%                                                         00219000 T  0137:1
BB:            P(0);     COMMENT FLAG TO EXIT ON CALLING PRNT;%                    00219100 T  0137:3
               PRNT;%                                                              00219200 T  0138:0
ERROR:         P(0);%                                                              00219300 T  0139:0
               PRNTA;    COMMENT CALL PRNTA AND EXIT;%                             00219400 T  0139:1
          COMMENT CASE OF  FORMAT,LIST OR FORMAT EMPTY;%                           00219500 T  0140:0
FMOUT:         LSTRN ← -(    ARRY = 0);%                                           00219600 T  0140:0
                    COMMENT LSTRN ← -1 IF NO LIST;%                                00219700 T  0141:3
               P(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);%                      00219800 T  0141:3
               P(0) ;                                                              00219810 T  0147:0
               DH1 ← TEMPD ;                                                       00219900 T  0147:1
               WH1 ← LISX; COMMENT GET FIRST LIST ITEM;%                           00220000 T  0148:0
               TPHRASE←BUFF←P(0,[BUFF]),[33:15] ;                                  00220100 T  0148:3
                    COMMENT BUFF ← ABSOLUTE CORE ADDRESS;%                         00220200 T  0150:3
               BSIZE ← BSIZE × 8;%                                                 00220300 T  0150:3
                    COMMENT BSIZE NOW # OF CHARACTERS IN BUFFER;%                  00220400 T  0152:0
               IF FRMT=0 THEN %%% FREE FIELD WRITE, WHICH IS EQUIVALENT            00220405 T  0152:0
                    BEGIN      %%% TO <INFINITY>U OR <INFINITY>UX.X.               00220410 T  0153:0
                    CODE←11; JUNK1←BSIZE; IF(TEMPD←DH1)>63 THEN TEMPD←63;          00220415 T  0153:2
                    FFCHR←IF FI<0 THEN " " ELSE "," ;                              00220416 T  0157:0
                    IF (FI←ABS(FI)-1)≠0 THEN %%% WE HAVE AT LEAST [FI]/.           00220418 T  0161:1
                         BEGIN IF TEMPD=0 THEN %%% WE HAVE [FI]/[0].               00220420 T  0163:1
                              BEGIN IF (TEMPD←BSIZE/FI-2.4999999999)≤0             00220422 T  0164:2
                              THEN TEMPD←1 ELSE IF TEMPD>21 THEN                   00220424 T  0166:3
                              JUNK1←((TEMPD←21)+2)×FI END                          00220426 T  0170:3
                         ELSE IF (V2←FI×(TEMPD+2))<BSIZE THEN JUNK1←V2 ;           00220428 T  0173:2
                         V2←TEMPD END %%% ABOVE ELSE WAS [FI]/[TEMPD].             00220430 T  0177:2
                    ELSE IF (V2←TEMPD)=0 THEN TEMPD←63;%%%HAVE [0]/[0] OR          00220432 T  0178:1
                    UTYP←(UTYP&JUNK1[16:35:13]) OR 2 ;   %%%[0]/[TEMPD].           00220455 T  0181:1
                    FAW←1&TEMPD[6:42:6]&V2[32:42:6] ;                              00220460 T  0183:2
                    GO TO PHRAS ;                                                  00220465 T  0186:1
                    END ELSE                                                       00220470 T  0186:3
               GO TO S;%                                                           00220500 T  0186:3
S1:            FI ← FI + 1;%                                                       00220600 T  0186:3
                    COMMENT SET INDEX TO NEXT EDITING PHRASE;%                     00220700 T  0188:0
S:             CODE ← (FAW ← FRMT[FI]).[2:4];%                                     00220800 T  0188:0
               UTYP ← 0&BSIZE[16:35:13] ;                                          00220810 T  0190:0
               IF FAW > 0 THEN GO TO PHRAS;%                                       00220900 T  0191:3
                    COMMENT  IF S=0 THEN GO TO PHRASE;%                            00221000 T  0193:0
```

```
          GO TO P(CODE); COMMENT SWITCH ON CODE;%                                  00221100 T  0193:0
          GO TO RTPAR;%                                                            00221200 T  0193:2
          GO TO STRNG;%                                                            00221300 T  0194:0
          GO TO LFPAR;%                                                            00221400 T  0194:2
          GO TO SLASH;%                                                            00221500 T  0195:0
          GO TO SCALE;%                                                            00221600 T  0195:2
     COMMENT LEFT PARENTHESIS;%                                                    00221700 T  0196:0
LFPAR:    IF FAW.[12:1] THEN                                                       00221800 T  0196:0
          BEGIN IF P(LISTELEMENT,DUP)<0 THEN                                       00221900 T  0196:3
                BEGIN P(DEL); FI←FAW.[28:10]+FI; END;                              00222000 T  0198:3
          END ELSE P(FAW.[38:10]);                                                 00222100 T  0201:1
               COMMENT MASK OUT REPEAT AND LEAVE IN STACK;%                        00222200 T  0202:2
          GO TO S1;%                                                               00222300 T  0202:2
     COMMENT RIGHT PARENTHESIS;%                                                   00222400 T  0203:0
RTPAR:     P(1,SUB); COMMENT SUBTRACT ONE FROM LFPAR REPEAT;%                      00222500 T  0203:0
          IF P(DUP) = 0 THEN BEGIN%                                                00222600 T  0203:2
                             P(DEL); COMMENT DELETE 0 REPEAT;%                     00222700 T  0204:3
                             GO TO S1; COMMENT PICK UP NEXT PHRASE;                00222800 T  0205:0
                             END;%                                                 00222900 T  0205:2
          FI ← FI -(FAW AND 1023);     COMMENT SET FI BACK TO LFPAR;               00223000 T  0205:2
          GO TO S1;%                                                               00223100 T  0207:1
     COMMENT SLASH;%                                                               00223200 T  0207:3
SLASH:    POLISH((LSTRN≥0) OR NOT FAW);%                                           00223300 T  0207:3
          PRNTA;COMMENT RELEASE BUFFER;%                                           00223400 T  0209:1
               COMMENT EXIT IF FORMAT & LIST EXAUSTED;%                            00223500 T  0210:0
          GO TO S1;  COMMENT S1 IF LIST OR FORMAT NOT EXAUSTED;%                   00223600 T  0210:0
     COMMENT SCALE FACTOR;%                                                        00223700 T  0210:2
SCALE:    SCFTR←IF FAW.[12:1] THEN LISTELEMENT                                     00223800 T  0210:2
                             ELSE 0&FAW[38:38:10]&FAW[1:11:1];                     00223900 T  0212:2
          GO TO S1;%                                                               00224000 T  0216:1
     COMMENT STRINGS;%                                                             00224100 T  0216:3
STRNG:    IF P(CHR + (W←FAW.[6:6]),DUP) > BSIZE%                                   00224200 T  0216:3
               THEN GO TO ERROR; COMMENT BUFFER OVERFLOW;%                         00224300 T  0218:3
          CHR ← P  ;  COMMENT CHR ← W+CHR;%                                        00224400 T  0219:3
          SETMAXCHR ;                                                              00224410 T  0220:1
                STREAM(P4 ← 0; P3 ← FAW,P2 ← W,P1 ← BUFF);%                        00224500 T  0221:0
               BEGIN%                                                              00224600 T  0222:3
                    SI ← LOC P2;%                                                  00224700 T  0222:3
                    SI ← SI-P2;%                                                   00224800 T  0223:0
                    DS ← P2 CHR;%                                                  00224900 T  0223:2
                    P4 ← DI;%                                                      00225000 T  0224:0
               END;%                                                               00225100 T  0224:1
          BUFF ← P;%                                                               00225200 T  0224:2
          GO TO S1;%                                                               00225300 T  0225:0
     COMMENT BREAK APART FORMAT WORD;%                                             00225400 T  0225:2
PHRAS:    IF FAW.[12:1] THEN P(LISTELEMENT) ELSE P(FAW.[38:10]);                   00225500 T  0225:2
          IF CODE=13 THEN CODE←IF (CODE←LISTELEMENT)="D" THEN 0 ELSE               00225600 T  0229:1
                               IF CODE="T" THEN 1 ELSE                             00225650 T  0233:3
                               IF CODE="X" THEN 2 ELSE                             00225700 T  0236:1
                               IF CODE="A" THEN 4 ELSE                             00225800 T  0238:1
                               IF CODE="I" THEN 6 ELSE                             00225900 T  0240:1
                               IF CODE="F" THEN 8 ELSE                             00226000 T  0242:1
                               IF CODE="E" THEN 10 ELSE                            00226100 T  0244:1
                               IF CODE="U" THEN 11 ELSE                            00226110 T  0246:1
                               IF CODE="B" THEN 110 ELSE                           00226120 T  0248:1
                               IF CODE="O" THEN 12 ELSE                            00226200 T  0250:1
                               IF CODE="L" THEN 14 ELSE                            00226300 T  0252:1
```

```
                        IF CODE="R" THEN 15 ELSE 16;                                00226400 T  0254:1
IF CODE=110 THEN BEGIN CODE←11;  FAW.[31:1]←1 END ;                                 00226410 T  0257:0
W←IF FAW.[13:1] THEN LISTELEMENT-(CODE=1) ELSE FAW.[6:6] ;                          00226500 T  0260:3
D←IF FAW.[14:1] THEN LISTELEMENT                                                    00226600 T  0264:2
                ELSE IF CODE=11 THEN FAW.[32:6]                                     00226610 T  0267:3
                ELSE (D1←FAW.[20:4])+(D2←FAW.[16:4]);                               00226700 T  0269:3
IF P(DUP)≤0  THEN GO BACK;                                                          00226800 T  0274:1
IF W<0 THEN                                                                         00226810 T  0275:2
     IF CODE=1 AND W=(-1)  THEN GO BACK                                             00226815 T  0276:1
     ELSE IF NOT(CODE=0 OR CODE=12) THEN GO FORMATERR;                              00226820 T  0278:3
IF D<0 THEN IF NOT(CODE≠15 AND CODE≠8 AND CODE≠10)                                  00226830 T  0281:3
THEN GO TO FORMATERR ;                                                              00226840 T  0285:1
IF W=0 THEN IF CODE≠2 AND CODE≠1 THEN                                               00226850 T  0286:1
BACK:                 BEGIN P(DEL); GO S1 END ;                                     00226852 T  0289:1
IF CODE=11 THEN BEGIN UTOP←DH1←FAW.[31:1]=0 ;                                       00226860 T  0290:2
   IF (WH2←UBUFF-DH1-FFTYP)<W THEN W←WH2 ;  IF D>WH2 THEN                           00226865 T  0295:0
   GO TO ERROR;  UTYP←UTYP&W[40:42:6]&D[29:42:6] OR 1 ;                             00226870 T  0300:1
   GO TO INLOOP END ;                                                               00226875 T  0304:0
IF FAW.[13:2]≠0 OR FAW.[2:4]=13 THEN                                                00226900 T  0304:2
BEGIN                                                                               00227000 T  0307:1
      GO P(IF CODE=15 THEN 8 ELSE IF CODE=1 THEN 2 ELSE                             00227060 T  0307:3
           CODE) ;                                                                  00227065 T  0311:3
      GO C; GO X; GO A; GO I; GO R; GO G; GO O; GO L;                               00227100 T  0312:1
      GO TO FORMATERR ;                                                             00227200 T  0316:1
L:    W1←IF W≤5 THEN W ELSE 5; GO TO Z;                                             00227300 T  0316:3
X:    W1←W DIV 64; W←SKIP←W.[42:6];                                                 00227400 T  0320:0
      GO TO ZW2;                                                                    00227500 T  0323:0
A:    W1←IF W≤6 THEN W ELSE 6;                                                      00227600 T  0323:2
Z:    SKIP←W-W1; GO TO ZW2;                                                         00227700 T  0326:1
I:    W1←IF W≤8 THEN W ELSE 8;                                                      00227800 T  0328:0
      SKIP←IF W≤16 THEN 0 ELSE W-16;  CODE←6 ;                                      00227900 T  0330:3
      W2←W-SKIP-W1; GO TO ZD;                                                       00228000 T  0334:3
G:    D←D+(UTIP OR FAW.[2:4]=13 OR FAW.[14:1]);                                     00228100 T  0337:0
      D2←D-D1←IF D≤8 THEN D ELSE 8;                                                 00228200 T  0341:1
      SKIP←IF (W-D)≤5 THEN 0 ELSE W-D-5;                                            00228300 T  0345:0
      W1←W2←0;  CODE←10;  GO TO SWT ;                                               00228400 T  0349:1
R:    D2←D-D1←IF D≤8 THEN D ELSE 8;                                                 00228500 T  0351:3
      SKIP←IF (W-D)≤17 THEN 0 ELSE W-D-17;                                          00228600 T  0355:2
      W1←IF (W-D)≤8 THEN W-D-1 ELSE 8;                                              00228700 T  0359:3
      W2←IF (W-D-SKIP)≤9 THEN 0 ELSE W-D-SKIP-9;                                    00228800 T  0364:0
      CODE←8;  GO TO SWT ;                                                          00228900 T  0369:1
C: O: W←8; W1←SKIP←0;                                                               00229000 T  0370:2
ZW2:  W2←0;                                                                         00229100 T  0372:2
ZD:   D←D1←D2←0;                                                                    00229200 T  0373:1
SWT:  WT←W1+W2;                                                                     00229300 T  0375:0
IF UTYP THEN BEGIN IF NOT((DH1←TEMPD-W)≤0 OR CODE≠10) THEN                          00229310 T  0376:1
                   W←TEMPD ELSE DH1←0; IF (WH2←W+UTOP+FFTYP+                         00229320 T  0379:3
                   USKIP)+CHR>UBUFF THEN BEGIN P(1);  PRNTA END;                    00229330 T  0384:3
                   CHR←CHR+WH2; SETMAXCHR; IF CODE=10 THEN BEGIN                    00229340 T  0390:0
                   SKIP←SGN+DH1; GO ETYPE1 END ELSE GO JMP END ;                    00229350 T  0392:3
END ELSE                                                                            00229400 T  0395:0
BEGIN WT←(W1←FAW.[28:4])+(W2←FAW.[24:4]);                                           00229500 T  0395:0
      SKIP←FAW.[32:6];                                                              00229600 T  0398:3
END;                                                                                00229700 T  0400:0
INLOOP:  IF CODE ≤ 2 THEN GO TO FLDW;%                                              00229800 T  0400:0
UTYP.[35:5]←27 ; %%% SETS UMD=UDC=UES=TRUE, SETS UED=2.                             00229810 T  0401:1
USKIP←0 ;                                                                           00229820 T  0403:0
```

```
          IF LSTRN≥0 THEN IF UTYP THEN GO TO UTYPE                          00229900 T  0404:3
                                  ELSE GO TO FLDW ;                         00229910 T  0406:1
          P(0);      COMMENT SET KEY = EXIT;%                               00230000 T  0407:1
          PRNTA;     COMMENT LIST EXAUSTED, RELEASE BUFFER AND EXIT;        00230100 T  0407:2
     COMMENT FILL FIELD WITH *;%                                            00230200 T  0409:0
ASTB: P(DEL); ASTA: P(DEL);%                                                00230300 T  0409:0
AST:             STREAM(P3←0; P2←W,PU←UTIP,PUU←UTOP,PS←USKIP,               00230400 T  0409:2
                    PFF←FFTYP,                                              00230410 T  0412:3
                    PCH←FFCHR,                                              00230415 T  0413:2
                    P1←BUFF) ;                                              00230420 T  0414:1
                    BEGIN                                                   00230500 T  0414:3
                         PS(DS←LIT" ") ;                                    00230510 T  0414:3
                         P2(DS←LIT"*";  PU(DI←DI+1;  DS←LIT"x")) ;          00230600 T  0416:0
                         PFF(SI←LOC P1; SI←SI+1; DS←CHR) ;                  00230605 T  0418:3
                         PUU(DS←LIT" ") ;                                   00230610 T  0420:1
                         P3 ← DI;%                                          00230700 T  0421:2
                    END;%                                                   00230800 T  0421:3
                                                                            00230900 T  0422:0
          GO TO COMMM ;                                                     00231000 T  0422:0
FORMATERR:  IF FILX.[18:15]>1 THEN                                          00231020 T  0422:2
               BEGIN %%% NOT ARRAYROWBUFF, SO TRY PAR LBL BRANCH,           00231025 T  0423:3
               P(FILX[NOT 3]) ;                                             00231027 T  0424:1
               FILX[NOT 3] ← FILX[NOT 4] ← 0 ;                              00231030 T  0425:2
               P(MKS,9,JUNK) ;                                              00231040 T  0428:3
               END ;                                                        00231045 T  0429:2
            TEN←0;  TEN←P([TEN[1]],CFX,SFB) & 10[8:38:10] ;                 00231050 T  0429:2
            STREAM(TEN);  DS←17LIT"-FMT ERR NO LBL:←" ;                     00231060 T  0432:3
            P([TEN[0]],[33:15],34,COM) ;                                    00231070 T  0436:2
FLDW:     IF CODE=1 THEN GO TTYPE;  IF P(W+CHR,DUP)>BSIZE                   00231100 T  0438:0
               THEN GO TO ERROR;   COMMENT BUFFER OVERFLOW;%                00231200 T  0440:1
          CHR ← P;                 COMMENT CHR ← CHR + W;%                 00231300 T  0441:1
          SETMAXCHR ;                                                       00231350 T  0441:3
     COMMENT SELECT EDITING PHRASE;%                                        00231400 T  0443:0
JMP:      IF CODE = 15 THEN GO TO RTYPE;%                                   00231500 T  0443:0
          IF CODE THEN GO ERROR ;                                           00231510 T  0444:1
          GO TO P(CODE);%                                                   00231600 T  0445:1
          GO TO DOTYPE;   COMMENT CODE = 0;%                                00231700 T  0445:3
          GO TO XTYPE;    COMMENT CODE = 2;%                                00231800 T  0446:1
          GO TO ALFA ;    COMMENT CODE = 4;%                                00231900 T  0446:3
          GO TO ITYPE;    COMMENT CODE = 6;%                                00232000 T  0447:1
          GO TO FTYPE;    COMMENT CODE = 8;%                                00232100 T  0447:3
          GO TO ETYPE;    COMMENT CODE = 10;%                               00232200 T  0448:1
          GO TO DOTYPE;   COMMENT CODE = 12;%                               00232300 T  0448:3
          GO TO LOGI ;    COMMENT CODE = 14;%                               00232400 T  0449:1
     COMMENT  L PHRASE;%                                                    00232500 T  0449:3
LOGI:          STREAM(P5 ← 0:P4 ← IF WH1 THEN "TRUE "%                      00232600 T  0449:3
                                         ELSE "FALSE" ,%                    00232700 T  0451:2
                    P3 ← W1, P2 ← SKIP,P1 ← BUFF);%                         00232800 T  0452:1
               BEGIN%                                                       00232900 T  0453:1
                    P2(DS ← LIT " ");%                                      00233000 T  0453:1
                    SI ← LOC P4;%                                           00233100 T  0454:2
                    SI ← SI+3;%                                             00233200 T  0454:3
                    DS ← P3 CHR;%                                           00233300 T  0455:0
                    P5 ← DI;%                                               00233400 T  0455:2
               END;%                                                        00233500 T  0455:3
                                                                            00233600 T  0456:0
          GO TO COMMM ;                                                     00233700 T  0456:0
```

```
UTYPE:   COMMENT THE U <EDITING PHRASE TYPE> (U, UW OR UW.M) SELECTS THE          00233703 T  0459:0
           <EDITING PHRASE TYPE> ::= I / F / (SPECIAL) E,                       00233706 T  0459:0
         AND FOR THE CHOSEN PHRASE TYPE IT SELECTS A SUITABLE                   00233709 T  0459:0
           <FIELD PART> ::= FP,                                                 00233712 T  0459:0
         AND A SUITABLE                                                         00233715 T  0459:0
           <DECIMAL PLACES> ::= D,                                              00233718 T  0459:0
         SUCH THAT THE FOLLOWING CONDITIONS ARE SATISFIED:                      00233721 T  0459:0
           1. FP MUST SATISFY THE INEQUALITIES, M ≤ FP ≤ W.                     00233727 T  0459:0
              IF M > W THEN INITIALLY FP ≤ W AND LATER FP IS                    00233730 T  0459:0
              ADJUSTED SUCH THAT M-W LEADING BLANKS ARE SUPPLIED                00233733 T  0459:0
           2. SUBJECT TO CONDITION #1, THE SELECTED PHRASE                      00233739 T  0459:0
              SHALL OUTPUT, IN A HIGHLY READABLE FORMAT, THE                    00233742 T  0459:0
              MAXIMUM AMOUNT OF MEANINGFUL NUMERIC SIGNIFICANCE                 00233745 T  0459:0
              IN THE LEAST POSSIBLE FIELD WIDTH. A BLANK SPACE                  00233746 T  0459:0
              IS POSTFIXED TO THE EDITED LIST ELEMENT.                          00233747 T  0459:0
         NOTE: WH1 = VALUE OF LIST ELEMENT TO BE EDITED, AND                    00233748 T  0459:0
         IN THE FOLLOWING COMMENTS, "FULL WORD" IS USED IN THE                  00233751 T  0459:0
         CONTEXT OF CONDITION #2 UNRESTRICTED BY CONDITION #1.                  00233754 T  0459:0
       END OF COMMENT ;                                                         00233757 T  0459:0
       WH1←TEN[0]×ABS(WH2←WH1); %%% RETAIN ORIG WH1,NORM WH1 FOR FINDE          00233760 T  0459:0
       IF (W←STORW)≤1 THEN GO UI; %%% RESTORE W, U1 IS SENT TO I-TYPE.          00233765 T  0461:1
       FINDE ; %%% FINDE SETS E = ENTIER[LOG10(ABS(WH1))],@E≤WH1<@E+1.          00233775 T  0463:2
       TEMPD←STORD ; %%% DECREASES USE OF PARTIAL WORDS.                        00233778 T  0465:0
       IF ((WH1←ABS(WH2))=0 %%% ZERO IS INTEGRAL, REGARDLESS OF EXPONT          00233781 T  0466:1
          OR (WH1.[3:6]=0 AND E<10))    %%% WH1 IS INTEGRAL AND NOT BIG        00233784 T  0467:2
          AND (V2←(SGN←WH2<0)+1+E)≤W    %%% V2 = MINIMUM WIDTH REQUIRED        00233787 T  0470:1
          THEN                          %%% FOR FULL WORD I-TYPE.              00233788 T  0473:0
            BEGIN W←V2 ; %%% WE NOW USE FULL WORD I-TYPE.                       00233790 T  0473:3
            UI:  IF W<TEMPD THEN USKIP←TEMPD-W ; %%% PHRASE GETS BLNKS          00233791 T  0475:0
            WH1←WH2;  GO TO I ; %%% RESTORE WH1 AND EXIT TO I-TYPE.             00233793 T  0478:2
            END ;                                                               00233796 T  0479:3
       SKIP←(W1←IF DH2←E<0 THEN WH1×TEN[-E] ELSE WH1/TEN[E])≥5 ;                00233800 T  0479:3
       JUNK1←IF DH1←(D←11-(W1>5.49755813885))<E THEN WH1/TEN[E-D]               00233802 T  0485:3
       ELSE WH1×TEN[D-E] ;                                                      00233804 T  0490:0
            %%% JUNK1 = MANTISSA OF WH1 AS AN 11 OR 12 DIGIT INTEGER,          00233806 T  0493:0
            %%% D = # DIGITS-1 IN JUNK1.                                       00233809 T  0493:0
            %%% W1 = MANTISSA OF WH1 AS N.NN...N.                               00233812 T  0493:0
            %%% DH1 = TRUE IF WH1 > MAX INTEGER, ELSE DH1 = FALSE.              00233813 T  0493:0
            %%% DH2 = TRUE IF WH1 < 1, ELSE DH2 = FALSE.                        00233815 T  0493:0
            %%% SKIP = TRUE IF WH1 WOULD ROUND UP, ELSE SKIP = FALSE.           00233816 T  0493:0
       IF DH1 OR DH2 THEN IF (D1←JUNK1 MOD 10)<3%%% HERE WE HANDLE ANY          00233818 T  0493:0
          THEN JUNK1←JUNK1-D1 ELSE IF D1>7 THEN %%% CONVERSION TRUNCA-          00233821 T  0495:3
          JUNK1←JUNK1-D1+10 ;                       %%% TION PROBLEMS.          00233824 T  0500:3
       IF JUNK1=TEN[11] THEN IF D≠11 THEN                                       00233825 T  0503:0
            BEGIN WH1←(IF E≥(-1) THEN TEN[E+1] ELSE 1/TEN[-(E+1)])              00233826 T  0505:1
            &SGN[1:47:1];  GO TO UTYPE ;                                        00233827 T  0510:0
            END ;                                                               00233828 T  0512:2
       D1←1 ;                                                                   00233829 T  0512:2
       WHILE JUNK1 MOD TEN[D1]=0 DO D1←D1+1;%%%D1=1+#TRAILN 0 IN JUNK1          00233830 T  0513:1
       UES←DH2;  IF NOT JUNK1←ABS(E)>9 THEN UED←1;  UDC←DA←D1-D≠1 ;             00233831 T  0517:0
       WT←(W2←2+SGN←DH2)+1+DA←JUNK1 ; %%% WT IS MAIN FIELD WIDTH FOR E          00233832 T  0526:0
       IF DH1                       %%% WH1 BEYOND MAXIMUM F-TYPE RANGE       00233833 T  0529:3
          OR ((2+DA<(-E)            %%% OR WH1 HAS LESS WIDTH IN THE          00233836 T  0529:3
          OR (D←D+1-D1)+DA<E)       %%% E-TYPE THAN IN THE F-TYPE,            00233839 T  0531:1
          AND (ABS(E)≥4 OR W<2+SGN  %%% AND IT WOULDNT LOOK BETTER IN F       00233840 T  0534:2
          +(D1←IF DH2 THEN 0 ELSE E)+D2←IF D≤E THEN 1 ELSE D-E)) THEN           00233841 T  0536:0
            BEGIN                   %%% THEN WE SHALL TRY E-TYPE.             00233842 T  0543:2
```

```
          %%% IN THE ABOVE, D = # DECIMAL PLACES FOR FULL WORD ETYPE
          IF D+WT≤W THEN     %%% D+WT = MINIMUM FIELD WIDTH REQUIRED
               BEGIN  W←D+WT;%%% FOR FULL WORD E-TYPE.
               GO TO G ; %%% EXIT TO FIRST PHASE OF E-TYPE PHRASE.
               END ;
          IF NOT (DH1 OR  %%% E-TYPE WIDTH WAS TOO SMALL TO HANDLE
             V2≠W) THEN GO UI ; %%% NN...NOO...0.0, SO DROP ,0, GO I
          ESUBTYPE:
          IF (D←W-WT)≥0 THEN GO TO G ; %%% WH1 FITS ROUNDED E-TYPE.
          UDC←0 ; %%% NO ROOM FOR DECIMAL POINT, SO WE RESET FLAG.
          IF D+DA=D←0 THEN GO G ; %%% FORM IS <SIGN>N@<EXP>,GO TO E.
          UMD←0 ; %%% NO ROOM FOR MANTISSA, SO WE RESET FLAG.
          W1←NOT(W2+JUNK1≠W OR (JUNK1←W1)≠1) ;
          IF DH2 THEN BEGIN IF (DH2←(-E>10)+W2)≤W AND SKIP THEN
             BEGIN %%% WH1<1, ROUND UP TO <SIGN>@<EXP+1>,GO TO ETYPE
             W←W-(DH2<W);  IF (E←E+1)=(-9) THEN UED←1;  GO TO G END
             ELSE IF W1 THEN GO TO G %%% DEL 1 IN <SIGN>1@<EXP>,GO E
          END ELSE IF (E>8)+W2=W AND SKIP THEN BEGIN E←E+1;  GO TO G
          END %%% WH1>1, ROUND UP TO <SIGN>@<EXP+1>, GO TO E-TYPE.
          ELSE IF W1 THEN GO G ; %%% DEL 1 IN <SIGN>1@<EXP>, GO TO E
          D←W-1-(W+E+SKIP=1)×SGN;  GO TO UF ; %%% GO TO F-TYPE FOR
          END                                 %%% **...*/(0).00..,0
     ELSE IF W≥D1←(D←D2)+D2←2+SGN+D1 THEN %%% HANDLE VARIOUS F-TYPES
     %%% D = # DECIMAL PLACES FOR FULL WORD F-TYPE.
     %%% D1 = MINIMUM WIDTH REQUIRED FOR FULL WORD F-TYPE.
     %%% D2 = 1 + #DIGITS TO THE LEFT OF THE DECIMAL PLACE.
          BEGIN  W ← D1 ; %%% FULL WORD F-TYPE.
          UF:  IF W<TEMPD THEN USKIP←TEMPD-W ; %%% PHRASE GETS BLNKS
          WH1←WH2;  GO TO R ; %%% RESTORE WH1 AND EXIT TO F-TYPE.
          END;
        IF DH2 THEN IF SGN THEN %%% DH2 SAYS WH1≤0.NN...N, SO SINCE WE
          D2←D2-(WH1<.5 OR W≠2)%%% CANNOT FIT FULL WORD IN F-TYPE, WE
            -(W+E < 1-SKIP)   %%% THEN DELETE LEADING ZERO (IF DO NOT
          ELSE D2←D2-1 ;      %%% HAVE TO ROUND INTO IT),AND IF SHALL
                              %%% HAVE TO ROUND TO 0,DELETE -SIGN TOO
        IF (D←W-D2)≥0 THEN GO TO UF ; %%% AFTER ABOVE SURGERY, IF CAN
                                      %%% ROUND THEN SEND WH1 TO F-TYPE
        IF D2-1=W THEN GO TO UI ; %%% TRY WH1 ROUNDED TO AN INTEGER.
        GO TO ESUBTYPE ; %%% AS A LAST DITCH EFFORT, TRY ROUNDED E-TYPE
        COMMENT A PHRASE ;
ALFA:             STREAM(P5 ← 0: P4 ← WH1, P3 ← W1, P2 ← SKIP,
                         P1 ← BUFF);%
                    BEGIN%
                         P2(DS ← LIT " ");%
                         SI ← LOC P3;%
                         SI ← SI - P3;%
                         DS ← P3 CHR;%
                         P5 ← DI;%
                    END;%

          GO TO COMMM ;
        COMMENT D & O PHRASES;%
DOTYPE:           STREAM(P4 ← 0: P3 ← IF CODE  = 0 THEN 0%
                                                ELSE WH1,%
                         P2 ← SKIP, P1 ← BUFF);%
                    BEGIN%
                         P2(DS ← LIT " ");%
```

```
00233845 T  0544:0
00233848 T  0544:0
00233851 T  0545:1
00233854 T  0547:0
00233857 T  0547:2
00233860 T  0547:2
00233863 T  0547:3
00233864 T  0549:1
00233865 T  0549:1
00233866 T  0551:2
00233867 T  0553:1
00233868 T  0555:2
00233869 T  0557:1
00233870 T  0560:3
00233871 T  0564:2
00233872 T  0565:0
00233873 T  0571:2
00233874 T  0572:1
00233875 T  0577:2
00233876 T  0578:0
00233877 T  0579:2
00233878 T  0583:3
00233879 T  0583:3
00233883 T  0588:0
00233886 T  0588:0
00233889 T  0588:0
00233892 T  0588:0
00233893 T  0589:1
00233895 T  0592:3
00233896 T  0594:0
00233899 T  0594:0
00233900 T  0595:0
00233903 T  0597:2
00233906 T  0599:2
00233909 T  0603:1
00233912 T  0603:1
00233915 T  0605:2
00233918 T  0605:2
00233921 T  0607:1
00233994 T  0607:3
00233997 T  0607:3
00234000 T  0609:1
00234100 T  0609:3
00234200 T  0609:3
00234300 T  0611:0
00234400 T  0611:1
00234500 T  0611:3
00234600 T  0612:1
00234700 T  0612:2
00234800 T  0612:3
00234900 T  0612:3
00235000 T  0613:1
00235100 T  0613:1
00235200 T  0615:2
00235300 T  0616:1
00235400 T  0617:0
00235500 T  0617:0
```

```
                        SI ← LOC P3;%                                                                    00235600 T  0618:1
                        DS ← 8 CHR;%                                                                     00235700 T  0618:2
                        P4 ← DI;%                                                                        00235800 T  0618:3
                    END;%                                                                                00235900 T  0619:0
                                                                                                         00236000 T  0619:1
        GO TO COMMM ;                                                                                    00236100 T  0619:1
    COMMENT X PHRASE;%                                                                                   00236200 T  0619:3
XTYPE:  IF P(CHR+(W1×64),DUP) > BSIZE%                                                                   00236300 T  0619:3
            THEN GO TO ERROR;    COMMENT BUFFER OVERFLOW;%                                               00236400 T  0621:1
        CHR←P;  SETMAXCHR ;                                                                              00236500 T  0622:1
              STREAM(P4 ← 0: P3 ← W1, P2 ← SKIP, P1 ← BUFF);%                                            00236600 T  0624:0
                BEGIN%                                                                                   00236700 T  0625:3
                    P2(DS ← LIT " ");%                                                                   00236800 T  0625:3
                    P3(32(DS ← 2 LIT " "));%                                                             00236900 T  0627:0
                    P4 ← DI;%                                                                            00237000 T  0628:3
                END;%                                                                                    00237100 T  0629:0
                                                                                                         00237200 T  0629:1
        GO TO COMMM ;                                                                                    00237300 T  0629:1
    COMMENT T PHRASE ;                                                                                   00237305 T  0629:3
TTYPE:  IF TPHRASE>0 THEN BEGIN DEBLANK; TPHRASE←-TPHRASE END ;                                          00237308 T  0629:3
        IF (CHR←W+W1×64)≥BSIZE THEN GO ERROR ;                                                           00237310 T  0633:0
        STREAM(P3←SAVEBUFF:P2←W,P1←W1);                                                                  00237315 T  0635:3
            BEGIN SI←P3; SI←SI+P2; P1(2(SI←SI+32)); P3←SI END ;                                           00237320 T  0637:3
        GO COMMM ;                                                                                       00237325 T  0640:2
    COMMENT I PHRASE;%                                                                                   00237400 T  0641:0
ITYPE:  IF ABS(P(WH1,DUP)) > P(MAXN)%                                                                    00237500 T  0641:0
        THEN GO TO ASTA;   COMMENT FILL FIELD WITH *;%                                                   00237600 T  0642:0
        P(.WH1,ISN,DUP);    COMMENT ROUND NUMBER;%                                                       00237700 T  0642:3
        SGN   ←    P < 0;%                                                                               00237800 T  0643:2
        WH2 ←(WH1 ← ABS(P)) DIV P(TEN8);%                                                                00237900 T  0644:2
        IF WH1 ≥ TEN[WT-SGN]%                                                                            00238000 T  0646:1
        THEN GO TO AST;     COMMENT NUMBER > FIELD WIDTH;%                                               00238100 T  0647:1
        STREAM(P8 ← 0: P7 ← WT-1, P6 ← [WH2],P5 ← SGN,%                                                  00238200 T  0648:1
               P4 ← W2,P3 ← W1,P2 ← SKIP+USKIP,PU←UTOP,                                                  00238300 T  0650:1
               PFF←FFTYP,                                                                                00238305 T  0652:3
               PCH←FFCHR,                                                                                00238307 T  0653:2
               P1 ← BUFF) ;                                                                              00238310 T  0654:1
            BEGIN%                                                                                       00238400 T  0654:3
                P2(DS ← LIT " ");%                                                                       00238500 T  0654:3
                P1 ← DI;  COMMENT SAVE STARTING ADDRESS;%                                                00238600 T  0656:0
                SI ← P6;%                                                                                00238700 T  0656:1
                DS ← P4 DEC; COMMENT CONVERT HIGH HALF;%                                                 00238800 T  0656:2
                SI ← P6;%                                                                                00238900 T  0657:0
                SI ← SI+8;%                                                                              00239000 T  0657:1
                DS ← P3 DEC; COMMENT CONVERT LOW HALF;%                                                  00239100 T  0657:2
                PFF(SI←LOC P1; SI←SI-1; DS←CHR) ;                                                        00239105 T  0658:0
                PU(DS ← LIT" ") ;                                                                        00239110 T  0659:2
                P8 ← DI;%                                                                                00239200 T  0660:3
                DI ← P1;%                                                                                00239300 T  0661:0
                DS ← P7 FILL;  COMMENT LEADING ZEROS←BLANKS;%                                            00239400 T  0661:1
                P5(DI ← DI-1;  DS ← LIT"-") ;                                                            00239500 T  0661:3
                                                                                                         00239600 T  0663:1
                                                                                                         00239700 T  0663:1
                                                                                                         00239800 T  0663:1
                                                                                                         00239900 T  0663:1
                                                                                                         00240000 T  0663:1
                                                                                                         00240100 T  0663:1
```

```
                    END;%

                GO TO COMMM ;
            COMMENT F PHRASE;%
FTYPE:      IF ABS(WH1 ← WH1 × 1.0) > P(MAXN)%
              THEN GO TO AST; COMMENT INSURE NUMBER IS REAL AND NOT%
                                  TO BIG;%
                IF NOT UTYP THEN FINDE ;%FINDE SETS E=[LOG10(ABS(WH1))],
RFIN:           IF (E + (DA ← D)) > 10 THEN GO TO FD;%
                    COMMENT  DA ← DECIMAL PLACES. IF D+E>10, MORE THEN%
                        11  DECIMAL PLACES SO MUST DO SPECIAL ROUND;%
                DH2 ← WH1 × (JUNK2 ← TEN[D]);%
                    COMMENT SHIFT NUMBER LEFT D PLACES THEN ROUND IT%
                        OFF BY DOING INTEGER STORE IN DH2;%
FA:             SGN  ←    DH2 < 0;%
                DH1 ← (DH2 ← ABS(DH2)) DIV P(TEN8);%
                IF DH2 ≥ JUNK2 THEN GO TO FB;%
                    COMMENT NUMBER IS LESS THEN ONE, WILL SIGN FIT;%
                IF P(WT-SGN, DUP) < 0 THEN GO TO ASTA;%
                    COMMENT ASTA IF SIGN DONT FIT;%
              LZ ← P ≠ (JUNK1←0);
                    COMMENT JUNK1 = # OF INTEGER DIGITS TO PRINT.%
                        IF WT-SIGN = 0 THEN JUNK1=0= DONT PRINT LEADING
                          ZERO.%
                        IF WT-SIGN > 0 THEN JUNK1=1= DO PRINT LEADING%
                          ZERO;%
                GO FC ;
 MAXN:::    @0007777777777777 ;
                    COMMENT NUMBER ≥ 1. CHECK IF ON ROUND WE OVERFLOWED%
                        INTO NEXT POWER OF TEN;%
FB:         IF ((JUNK1 ← E + (IF DH2 ≥ TEN[E+1+DA] THEN 2%
                                                ELSE 1))%
                 + SGN ) > WT THEN GO TO AST;%
                COMMENT   FOR NUMBERS ≥ 1, E = ONE LESS THAN THE%
                          NUMBER OF DIGITS LEFT OF THE DECIMAL%
                          POINT. IN JUNK1 WE SAVE EITHER E+1 OR%
                          E+2,DEPENDING ON IF ROUND OVERFLOW%
                          OCCURED. ALSO WE COMPARE JUNK1 + SIGN%
                          WITH WT. THIS TELLS US IF THE NUMBER%
                          WILL FIT THE FIELD. WT = TOTAL NUMBER%
                          OF POSITIONS AVAILABLE FOR INTEGER DIGITS%
                          + SIGN;%
FC:                       COMMENT   NOW WE CONVERT. NOTE THAT NUMBER%
                          IS NOW AN INTEGER IN THE FORM N--N*N--N.%
                          * DENOTES TRUE DECIMAL POINT%
                          . DENOTES MACHINE POINT%
                          ZEROS CONTAINS # OF TRAILING ZEROS%
                          JUNK1 CONTAINS # OF DIGITS LEFT OF *%
                          DA   CONTAINS # OF DIGITS BETWEEN * &.%
                          NOTE THAT WH2 + ZEROS ALWAYS = D.%
                          THE STREAM PROCEDURE WILL CONVERT THE%
                          NUMBER IN TWO PARTS (ALREADY SET UP IN%
                          DH1 AND DH2). IT WILL THEN MOVE JUNK1 #%
                          OF DIGITS LEFT AND INSERT THE DECIMAL%
                          POINT. ALSO THE SIGN AND TRAILING ZEROS%
                          ARE INSERTED;%
```

```
00240200 T  0663:1
00240300 T  0663:1
00240400 T  0663:2
00240500 T  0663:2
00240600 T  0664:0
00240700 T  0664:0
00240800 T  0665:3
00240900 T  0666:2
00241000 T  0666:2
00241100 T  0669:0
00241200 T  0671:1
00241300 T  0671:1
00241400 T  0671:1
00241500 T  0673:1
00241600 T  0673:1
00241700 T  0673:1
00241800 T  0674:2
00241900 T  0676:2
00242000 T  0677:3
00242100 T  0677:3
00242200 T  0679:3
00242300 P  0679:3
00242400 T  0681:1
00242500 T  0681:1
00242600 T  0681:1
00242700 T  0681:1
00242800 T  0681:1
00242895 T  0681:1
00242900 T  0685:0
00243000 T  0686:0
00243100 T  0686:0
00243200 T  0686:0
00243300 T  0689:0
00243400 T  0690:2
00243500 T  0692:0
00243600 T  0692:0
00243700 T  0692:0
00243800 T  0692:0
00243900 T  0692:0
00244000 T  0692:0
00244100 T  0692:0
00244200 T  0692:0
00244300 T  0692:0
00244400 T  0692:0
00244500 T  0692:0
00244600 T  0692:0
00244700 T  0692:0
00244800 T  0692:0
00244900 T  0692:0
00245000 T  0692:0
00245100 T  0692:0
00245200 T  0692:0
00245300 T  0692:0
00245400 T  0692:0
00245500 T  0692:0
00245600 T  0692:0
00245700 T  0692:0
```

```
                D1←DA+JUNK1-(D2←IF P(JUNK1 +DA,DUP) > 8 THEN P(8,SUB)%                    00245800 T  0692:0
                                                        ELSE P(DEL,0));%                  00245900 T  0695:1
STREAM(P9←0:P8←JUNK1+LZ,P7←ZEROS,P6←[DH1],P5←SGN,                                         00246000 P  0697:2
       P4←D1,P3←D2,P2←SKIP+WT-JUNK1-LZ+USKIP,PU←UTOP,                                     00246100 P  0699:3
            PFF←FFTYP,                                                                    00246105 T  0703:3
            PCH←FFCHR,                                                                    00246107 T  0704:2
                              LZ,                                                         00246108 C  0705:1
            P1 ← BUFF) ;                                                                  00246110 T  0705:2
          BEGIN%                                                                          00246200 T  0706:0
              P2(DS←LIT " "); COMMENT INSERT LEADING BLANKS;%                             00246300 T  0706:0
              P1←DI;          COMMENT SAVE ADDRESS OF MSD;%                               00246400 T  0707:1
              DI←DI+1;        COMMENT LEAVE ROOM FOR INTEGER%                             00246500 T  0707:2
                                    PART;%                                                00246600 T  0707:3
                                    LZ(DS←LIT"0");                                        00246650 C  0707:3
              SI ←P6;%                                                                    00246700 T  0709:0
              DS←P3 DEC;      COMMENT CONVERT HIGH PART;%                                 00246800 T  0709:1
              SI←P6;%                                                                     00246900 T  0709:3
              SI←SI+8;%                                                                   00247000 T  0710:0
              DS←P4 DEC;      COMMENT CONVERT LOW HALF;%                                  00247100 T  0710:1
              P7(DS←LIT"0"); COMMENT INSERT TRAILING ZEROS;%                              00247200 T  0710:3
              PFF(SI←LOC P1; SI←SI-1; DS←CHR) ;                                           00247205 T  0712:0
              PU(DS ← LIT" ") ;                                                           00247210 T  0713:2
              P9←DI;          COMMENT ADDRESS OF NEXT FIELD;%                             00247300 T  0714:3
              SI←P1;%                                                                     00247400 T  0715:0
              SI←SI+1;%                                                                   00247500 T  0715:1
              DI←P1;          COMMENT MOVE INTEGER PART LEFT;%                            00247600 T  0715:2
              DS←P8 CHR;%                                                                 00247700 T  0715:3
              DS←LIT".";%                                                                 00247800 T  0716:1
              P5(DI ← P1;  DI ← DI-1;  DS ← LIT"-") ;                                     00247900 T  0716:3
                                                                                          00248000 T  0718:2
                                                                                          00248100 T  0718:2
                                                                                          00248200 T  0718:2
                                                                                          00248300 T  0718:2
                                                                                          00248400 T  0718:2
                                                                                          00248500 T  0718:2
                  END;%                                                                   00248600 T  0718:2
                                                                                          00248700 T  0718:3
              GO TO COMMM ;                                                               00248800 T  0718:3
      FD:           COMMENT MORE THEN 11 SIGNIFICANT DIGITS SO WE HAVE%                   00248900 T  0721:0
                          TO DO SPECIAL ROUND;%                                           00249000 T  0721:0
              DA ← D -(ZEROS ← E+D-11);%                                                  00249100 T  0721:0
                  COMMENT FIRST GUESS AT TRAILING ZEROS;%                                 00249200 T  0723:3
              RNDOFF;%                                                                    00249300 T  0723:3
              GO TO FA;%                                                                  00249400 T  0725:0
            COMMENT E PHRASES;%                                                           00249500 T  0725:2
ETYPE: IF D + 6 > W THEN GO TO AST;                                                       00249600 P  0725:2
           SGN ← (WH1 ← WH1 × 1.0) < 0;                                                   00249650 C  0727:3
              FINDE;    COMMENT E ← EXPONENT;%                                          00249700 T  0730:0
      ETYPE1: P(1) ; %%% RETURN LITERAL USED AT REDT.                                     00249800 T  0731:0
      REIN:   IF (DA+D-1) > 10 THEN GO TO EB; COMMENT SPECIAL ROUND OFF                   00249900 T  0731:1
                          IF MORE THEN 11 SIGNIFICANT DIGITS;%                            00250000 T  0733:2
              P(0);       COMMENT SET LITERAL TO NOT ADJUST D2 AT ERTN;%                  00250100 T  0733:2
                          DH2 ← (IF (E-D) ≥ 0%                                            00250200 T  0733:3
                          THEN WH1 / TEN[E-D+1]%                                          00250300 T  0734:3
                          ELSE WH1 × TEN[D-1-E]);%                                        00250400 T  0737:0
      EA:     COMMENT NUMBER NOW IN FORM OF N*N----N.%                                    00250500 T  0740:2
                          WHERE  * = TRUE DECIMAL POINT%                                  00250600 T  0740:2
```

```
                                E = EXPONENT%                                    00250700 T  0740:2
                                . = MACHINE POINT%                               00250800 T  0740:2
                                DA = # OF DIGITS BETWEEN * S .%                  00250900 T  0740:2
                                DA + ZEROS = <DECIMAL PLACES>%                   00251000 T  0740:2
                                STORING IN DH2 ROUNDS NUMBER;%                   00251100 T  0740:2
                        IF ( DH2 ← ABS(DH2)) ≥ TEN[DA+1]%                        00251200 T  0740:2
                             THEN BEGIN%                                         00251300 T  0742:1
                                       DH2 ← TEN[DA];%                           00251400 T  0743:1
                                       E ← E + 1;%                               00251500 T  0744:1
                                  END;%                                          00251600 T  0745:2
                        COMMENT IF ROUND OVERFLOWED THE LEADING DIGIT FROM 9 TO% 00251700 T  0745:2
                                  10 WE SET OUR NUMBER TO 1.0 AND INCREASE%      00251800 T  0745:2
                                  EXPONENT BY ONE;%                              00251900 T  0745:2
                        DH1 ← DH2 DIV P(TEN8);COMMENT SINCE HARDWARE CAN CONVERT% 00252000 T  0745:2
                                  ONLY 8 DIGITS WE SPLIT NUMBER IN TWO%          00252100 T  0746:3
                                  PARTS AT 8 TH DIGIT;%                          00252200 T  0746:3
                        IF FALSE THEN                                            00252210 T  0746:3
TEN8:::  @1045753604000000 ;                                                     00252220 T  0747:0
                        STREAM(P10←0:P9 ← ABS(E),P8 ← (   E<0),P7 ← SGN,%        00252300 T  0749:0
                             P6 ← ZEROS,P5 ← [DH1],P4 ← D2,P3 ← D1,%             00252400 T  0751:1
                             P2 ← SKIP, PU ← UTOP              ,PES ← UES,       00252500 T  0752:1
                             PFF ← FFTYP,                                        00252505 T  0754:0
                             PCH←FFCHR,                                          00252507 T  0754:3
                             PED ← UED,PDC ← UDC,PMD ← UMD, P1 ← BUFF) ;         00252510 T  0755:2
                             BEGIN%                                              00252600 T  0758:1
                                  P2(DS←LIT" "); COMMENT INSERT LEADING BLANKS;% 00252700 T  0758:1
                                  P1←DI;         COMMENT SAVE ADDRESS OF INTEGER% 00252800 T  0759:2
                                                    DIGIT;%                      00252900 T  0759:3
                        P7(DI ← DI-1;  DS←LIT"-");                               00253000 T  0759:3
                                                                                 00253100 T  0761:1
                                                                                 00253200 T  0761:1
                                                                                 00253300 T  0761:1
                                                                                 00253400 T  0761:1
                                                                                 00253500 T  0761:1
                                                                                 00253600 T  0761:1
                        PDC(DI←DI+1 ;    COMMENT SAVE ROOM FOR INTEGER           00253700 T  0761:1
                                            DIGIT;%                              00253800 T  0762:0
                        SI←P5;%                                                  00253900 T  0762:0
                        DS←P4 DEC) ;     COMMENT CONVERT HIGH HALF ;             00254000 T  0762:1
                        PMD(SI←P5 ;                                              00254100 T  0763:0
                        SI←SI+8;%                                                00254200 T  0763:3
                        DS←P3 DEC;       COMMENT CONVERT LOW HALF;%              00254300 T  0764:0
                        P6(DS←LIT"0")) ;COMMENT INSERT TRAILING ZEROS ;          00254400 T  0764:2
                        DS←LIT "@";%                                             00254500 T  0766:0
                        PES(DS←LIT"+";  P8(DI←DI-1;  DS←LIT"-")) ;               00254600 T  0766:2
                                                                                 00254700 T  0769:1
                                                                                 00254800 T  0769:1
                                                                                 00254900 T  0769:1
                                                                                 00255000 T  0769:1
                        SI←LOC P9;       COMMENT CONVERT EXPONENT;%              00255100 T  0769:1
                        DS ← PED DEC;                                            00255200 T  0769:2
                        PFF(SI←LOC PED; SI←SI-1; DS←CHR) ;                       00255205 T  0770:0
                        PU(DS ← LIT" ") ;                                        00255210 T  0771:2
                        P10←DI;          COMMENT ADDRESS OF NEXT FIELD;%         00255300 T  0772:3
                        PDC(SI←P1 ;                                              00255400 T  0773:0
                        SI←SI+1;%                                                00255500 T  0773:3
                        DI←P1;           COMMENT MOVE INTEGER DIGIT LEFT;        00255600 T  0774:0
```

```
                    DS←CHR;%                                                    00255700 T  0774:1
                    DS←LIT".") ;                                                00255800 T  0774:2
               END;%                                                            00255900 T  0775:1
          BUFF ← P;%                                                            00256000 T  0775:2
ERTN:     IF P THEN D2 ← D-8;%                                                  00256100 T  0776:0
REOT:     IF NOT P THEN GO TO RRTN; COMMENT OUT IF FROM RTYPE;%                 00256200 T  0777:3
          GO TO COMM;%                                                          00256300 T  0778:1
EB:                 COMMENT MORE THEN 11 SIGNIFICANT DIGITS%                  00256600 T  0781:0
                      REQUESTED SO DO SPECIAL ROUND;%                         00256700 T  0781:0
          DA ← D-1-(ZEROS← D-12);%                                              00256800 T  0781:0
          WH1 ← (IF E>0 THEN WH1 / TEN[E]%                                      00256900 T  0783:3
                        ELSE WH1 × TEN[-E]);%                                   00257000 T  0785:2
               COMMENT NUMBER NOW IN FORM N.NNN WITH EXPONENT IN E;              00257100 T  0788:1
          RNDOFF;                    COMMENT ROUND OFF NUMBER;%                00257200 T  0788:1
          D2 ← DA - 7;%                                                         00257300 T  0789:0
          P(1);     COMMENT SET KEY TO ADJUST D2 AT ERTN;%                      00257400 T  0790:1
          GO TO EA;%                                                            00257500 T  0790:2
          COMMENT R PHRASE;%                                                    00257600 T  0791:0
RTYPE:    WH1 ← IF SCFTR ≥ 0 THEN WH1 × TEN[SCFTR]%                             00257700 T  0791:0
                             ELSE WH1 / TEN[-SCFTR];%                           00257800 T  0792:3
          FINDE;%                                                               00257900 T  0795:2
          SGN ← WH1 < 0;%                                                       00258000 T  0797:0
          IF ABS(WH1) > P(MAXM) THEN GO TO TRYE;%                               00258100 T  0798:1
          IF E ≥ 0 THEN%                                                        00258200 T  0799:3
               BEGIN COMMENT CHECK IF IT WILL GO AS F FIELD;%                   00258300 T  0800:2
                    IF (E+2+D+SGN) ≤ W THEN%                                    00258400 T  0801:0
                    BEGIN COMMENT YES- IT WILL;%                                00258500 T  0803:1
RC:                    SKIP ← W-(D+1+WT);%                                      00258600 T  0803:3
                       GO TO REIN%                                              00258700 T  0806:0
                    END ELSE GO TO TRYE; COMMENT TO BIG FOR F;%                 00258800 T  0806:2
               END%                                                             00258900 T  0806:2
                    ELSE%                                                       00259000 T  0806:2
               BEGIN COMMENT NUMBER IS LSS THEN 1, SEE IF IT WILL%              00259100 T  0806:2
                       GO AS F FIELD WITHOUT LOSS IN REQUESTED%                 00259200 T  0807:0
                       ACCURACY;%                                               00259300 T  0807:0
                    IF ABS(E) ≤ D THEN GO TO RC;%                               00259400 T  0807:0
                      COMMENT TO RC IF IT WILL GO AS F FIELD;%                  00259500 T  0808:2
               END;%                                                            00259600 T  0808:2
          COMMENT SEE IF NUMBER WILL FIT IN E FIELD;%                           00259700 T  0808:2
TRYE:     IF W < (D+6+SGN) THEN%                                                00259800 T  0808:2
               BEGIN COMMENT FIELD TO SMALL FOR E, IF%                          00259900 T  0810:1
                    NUMBER < 1 PRINT AS F FIELD EVEN THOUGH%                    00260000 T  0810:3
                      ACCURACY IS LOST, FILL FIELD WITH * IF%                   00260100 T  0810:3
                     NUMBER ≥ 1;%                                               00260200 T  0810:3
                  IF E < 0 THEN GO TO RC%                                       00260300 T  0810:3
                           ELSE GO TO AST;%                                     00260400 T  0811:2
               END;%                                                            00260500 T  0812:2
          COMMENT NUMBER WILL FIT AS E FIELD,ADJUST PARAMETERS SO%               00260600 T  0812:2
                  ETYPE CAN HANDLE;%                                            00260700 T  0812:2
               SKIP ← W-(D+6);%                                                 00260800 T  0812:2
          IF (D ← D+1) > 8 THEN BEGIN D1←8; D2 ← D-8 END%                       00260900 T  0814:1
                           ELSE BEGIN D1←D; D2 ← 0 END;%                        00261000 T  0818:2
          P(0);     COMMENT FLAG USED AT REOT TO RETURN CONTROL TO%              00261100 T  0821:2
                           RRTN;%                                               00261200 T  0821:3
          GO TO REIN;%                                                          00261300 T  0821:3
RRTN:     IF (D ← D-1) > 8 THEN BEGIN D1←8; D2 ← D-8 END%                       00261400 T  0822:1
                           ELSE BEGIN D1←D; D2 ← 0 END;%                        00261500 T  0826:2
```

```
                GO TO COMM;%                                                                    00261600 T  0828:2
MAXM:::         @00077777777777777;%                                                            00261700 T  0829:0
          COMMENT AFTER FORMATING A PHRASE WE COME HERE;%                                       00261800 T  0830:0
COMMM:    BUFF←P ;                                                                              00261810 T  0830:0
COMM:     IF CODE > 2 THEN WH1 ← LISX;%                                                         00261900 T  0830:2
          IF P((FFTYP=0),SUB,DUP) > 0 THEN GO TO INLOOP ;                                       00262000 T  0832:2
          P(DEL);%                                                                              00262100 T  0835:1
          GO TO S1;%                                                                            00262200 T  0835:2
          COMMENT THE <REPEAT PART> OF PHRASE IS IN TOP OF STACK,%                              00262300 T  0837:0
               IF REPEAT=1 > 0 THEN GO TO INLOOP TO USE SAME PHRASE                             00262400 T  0837:0
               AGAIN ELSE DELETE THE "0" REPEAT AND GO TO S1 TO%                                00262500 T  0837:0
               PICK UP NEXT PHRASE;%                                                            00262600 T  0837:0
     END OUTPUTINT;%                                                                            00262700 T  0837:0
                                                                                                          SIZE= 0838 WORDS


COMMENT ALGOL SELECT INTRINSIC;%                                                                00300000 T  0000:0
PROCEDURE ALGOLSELECT(ACT1,ACT2,TANK,I); VALUE ACT1,ACT2,TANK,I;%                               00300100 T  0000:0
                                                                          START OF REL SEGMENT; DISK ADDRESS = 00053
     INTEGER ACT1,ACT2,I; NAME TANK;%                                                           00300200 T  0000:0
BEGIN ARRAY FIB[*]; NAME MEM=2; ARRAY FPB=3[*];%                                                00300300 T  0000:0
     ARRAY HEADER[*];%                                                                          00300400 T  0000:0
     LABEL REW,L6,MYUSERR ;                                                                     00300500 T  0000:0
     REAL  RITE=12,REED=13,SELECT=14;%                                                          00300600 T  0000:0
     INTEGER STATUS,NBUFFS,BSIZE,T1,INOUT,DIREC,UTYPE;%                                         00300700 T  0000:0
     LABEL OWT,EASY,EXIT,FILL;%                                                                 00300800 T  0000:0
     DEFINE IOD=(*TANK)#;%                                                                      00300900 T  0000:0
     LABEL WR,ERR,RF,RR;%                                                                       00301000 T  0000:0
    LABEL DC19;                                                                                 00301010 T  0000:0
     SWITCH CURRENT←WR,ERR,RF,RR;%                                                              00301100 T  0000:0
     LABEL CR,LP,MT,DK,SP,CP,PP,PR,DC;%                                                         00301200 T  0000:0
    SWITCH USW← CR,LP,MT,EASY,DK,SP,CP,LP,PP,PR,DC,CR,LP,DC19;                                  00301300 T  0000:0
REAL SUBROUTINE COUNT;%                                                                         00301400 T  0000:0
     BEGIN FOR I←0 STEP 1 UNTIL NBUFFS-1 DO%                                                    00301500 T  0001:0
          BEGIN IF NOT TANK[I],[19:1] THEN%                                                     00301600 T  0005:1
                    P([TANK[I]],@2000000000,2,COM,DEL,DEL);%                                    00301700 T  0007:0
               IF TANK[I],[27:1] THEN                                                           00301800 T  0009:2
                    BEGIN                                                                       00301805 T  0011:0
                         I ← I+1=(FIB[4],[2:1] AND FIB[5],[44:1]);                              00301810 T  0011:2
                         P(1); GO OWT;                                                          00301820 T  0015:1
                    END;                                                                        00301830 T  0017:0
          END;  P(0);%                                                                          00301900 T  0017:0
     OWT:  COUNT←P;%                                                                            00302000 T  0017:3
     END COUNT;%                                                                                00302100 T  0018:0
     SUBROUTINE SPACE; P(XCH,TANK,9,11,COM,DEL,DEL,DEL);%                                       00302200 T  0018:1
     SUBROUTINE MOVEUP;%                                                                        00302300 T  0021:1
          IF (I←MEM[FIB[16] INX 1])≠BSIZE THEN%                                                 00302400 T  0022:0
          BEGIN TANK[0]←IOD&(P(DUP),[33:15]=BSIZE+I)[33:33:15];%                                00302500 T  0024:3
               T1←FIB[16],[33:15];%                                                             00302600 T  0028:1
               STREAM(N←I+1,L←0,S←T1-I,D←T1-BSIZE);%                                            00302700 T  0029:3
               BEGIN SI←LOC N; SI←SI+6; DI←LOC L; DI←DI+7; DS←CHR;%                             00302800 T  0032:3
                    SI←S; DI←D; DS←N WDS; L(DS←32 WDS; DS←32 WDS);%                             00302900 T  0034:0
          END   END;%                                                                           00303000 T  0036:1
     SUBROUTINE REFILL;%                                                                        00303100 T  0036:3
     BEGIN FOR I←0 STEP 1 UNTIL NBUFFS-1 DO%                                                    00303200 T  0037:0
```

```
      TANK[I]←TANK[I]&1[19:47:1]&DIREC[22:47:1] OR MEM;%                    00303300 T  0041:1
      IF NBUFFS >1 THEN%                                                    00303400 T  0046:2
      BEGIN;STREAM(T←IOD,N←NBUFFS-1,D←TANK);%                               00303500 T  0047:1
            BEGIN SI←D; SI←SI+8; DS←N WDS; SI←LOC T; DS←WDS END;%            00303600 T  0049:3
            P(2&(NOT DIREC)[1:47:1],TANK,10,11,COM,DEL,DEL,DEL);%           00303700 T  0051:2
END   END REFILL;%                                                          00303800 T  0054:3
SUBROUTINE EMPTY;%                                                          00303900 T  0055:0
BEGIN FIB[17]←BSIZE-(IOD.[33:15]-(STATUS=3)-ACT2←FIB[16].[33:15]);          00304000 T  0055:0
      FIB[16]←FIB[16]&0[22:22:1]&0[24:24:1];%                               00304100 T  0059:2
      FIB[19]←FIB[19]&0[22:22:1]&0[24:24:2];%                               00304200 T  0063:3
      FIB[13]←FIB[13]&0[25:25:1]&0[27:27:1];%                               00304300 T  0067:1
      FIB[5].[43:2]←0;%                                                     00304400 T  0070:3
      TANK[NOT 1]←P(DUP,LOD)&0[22:22:1]&0[24:24:1];%                        00304500 T  0073:1
      BSIZE←IF STATUS=0 THEN MEM[ACT2-1] ELSE IOD.[8:10];%                  00304600 T  0077:1
      FOR I←0 STEP 1 UNTIL NBUFFS-1 DO%                                     00304700 T  0082:0
      BEGIN TANK[I]←TANK[I]&1[19:47:1]&0[22:22:1]&0[24:24:1]%               00304800 T  0086:1
                  &FIB[18][8:38:10] OR MEM;%                                00304900 T  0090:1
            IF I>0 THEN%                                                    00305000 T  0093:1
                  TANK[I]←TANK[I]&((STATUS=3)+ACT2)[33:33:15];%             00305100 T  0094:0
            ACT2←MEM[ACT2-2].[18:15];%                                      00305200 T  0098:1
END   END EMPTY;%                                                           00305300 T  0101:0
IF I = 6 THEN GO TO L6;                                                     00305400 T  0101:3
IF I=7 THEN GO TO MYUSERR ;                                                 00305410 T  0105:2
TANK←((I-1) INX *P(,TANK))&0[8:8:25];%                                      00305500 T  0106:3
FIB←TANK[NOT 2]; STATUS←FIB[5]; UTYPE←FIB[4].[8:4];%                        00305600 T  0109:3
IF I=4 THEN IF STATUS.[42:1]=0 THEN%                                        00305700 T  0114:0
BEGIN;STREAM(S←[FPB[FIB[4].[13:11]+2]],D←[NBUFFS]);%                        00305800 T  0116:2
            BEGIN SI←S; DS←3 OCT END;%                                      00305900 T  0119:2
      IF FIB[1]=0 THEN FIB[1]:=NBUFFS;                                      00305950 T  0120:1
      BSIZE←FIB[13].[28:10];%                                               00306000 T  0123:0
      IF (ACT1 OR 4)=6 THEN T1←@12 ELSE%                                    00306100 T  0124:2
      IF ACT1=4 THEN T1←@22 ELSE%                                           00306200 T  0127:0
      IF ACT1=8 THEN T1:=@52 ELSE                                           00306250 T  0129:2
      IF ACT1=0 THEN%                                                       00306300 T  0132:0
        IF FIB[15].[24:6] LSS  16 AND NBUFFS GTR FIB[1]                     00306350 T  0133:1
            THEN T1:=@12 ELSE T1:=IF NBUFFS EQL BSIZE                       00306400 T  0135:3
                  THEN 6 ELSE @12 ELSE                                      00306500 T  0138:2
      IF ACT1 = 1 THEN                                                      00306600 T  0141:0
        BEGIN NBUFFS:=BSIZE; T1:=7; IF UTYPE = 4 THEN                       00306610 T  0142:1
        BEGIN HEADER:=*[FIB[14]];                                           00306620 T  0145:0
        IF(DIREC:=FIB[7]-1) GTR (INOUT:=HEADER[7]) THEN                     00306630 T  0146:3
        DIREC:=INOUT;INOUT:=(DIREC DIV HEADER[0].[30:12])+1;                00306640 T  0149:2
        END END ELSE T1:=0;                                                 00306650 T  0153:1
      P(TANK&T1[18:33:15],6,11,COM,DEL,DEL);%                               00306700 T  0154:2
      FIB[13].[28:10]:=IF FIB[15].[24:6] LSS 16 AND NOT ACT1                00306800 T  0156:2
        THEN FIB[1] ELSE NBUFFS; IF ACT1 AND UTYPE =4 THEN                  00306805 T  0159:0
        BEGIN FIB[6]:=INOUT; FIB[7]:=DIREC; END; GO TO EXIT;                00306810 T  0164:0
    END ELSE GO TO EXIT ELSE%                                               00306900 T  0167:2
    IF STATUS.[41:2]≠0 THEN%                                                00307000 T  0167:2
EASY: BEGIN FIB[13]←FIB[13]&(ACT2=3)[25:47:1]&(ACT2≠0)[27:47:1];%            00307100 T  0169:1
      GO TO EXIT;%                                                          00307200 T  0174:1
    END;%                                                                   00307300 T  0174:3
    GO TO USW[UTYPE];%                                                      00307400 T  0174:3
MT:       NBUFFS←FIB[13].[10:9]; BSIZE←FIB[18].[18:15];%                    00307500 T  0182:3
      INOUT←ACT2≠0; DIREC←ACT2=3; STATUS←STATUS.[46:2];%                    00307600 T  0186:0
      GO TO CURRENT[FIB[5].[43:2]];%                                        00307700 T  0189:3
CR: LP: CP: PP: PR:  GO TO ERR;                                             00307800 T  0193:2
```

```
 $ SET OMIT = NOT(TIMESHARING)                                                  00307801 T  0194:0
DC:                                                                             00307805 T  0194:0
 $ POP OMIT                                                                     00307808 T  0194:0
DC19: POLISH(IOD, [TANK[1]], *P(DUP), TANK, ←, ←);                              00307810 T  0194:0
      FIB[5] ← (*P(DUP))&P(DUP, LNG)[43:43:1];  GO TO EASY;                     00307820 T  0196:2
 $ SET OMIT = TIMESHARING                                                       00307825 T  0199:3
WR:   IF NOT DIREC THEN                                                         00307900 T  0199:3
ERR:        P(TANK,8,11,COM);%                                                  00308000 T  0200:1
SP:   P(MKS,1,0,(NOT 2) INX TANK,4,SELECT); GO TO EASY;%                        00308100 T  0202:0
RF:   IF INOUT THEN%                                                            00308200 T  0204:3
      BEGIN T1←COUNT; P((-I)); SPACE;%                                          00308300 T  0205:0
            IF (I←MEM[FIB[16] INX NOT 0])≠BSIZE THEN%                           00308400 T  0209:0
            BEGIN;STREAM(N←I+1,NDIV64←(I+1) DIV 64,%                            00308500 T  0212:0
                         S←FIB[16] INX (NOT 0) INX I,%                          00308600 T  0214:3
                         D←FIB[16] INX (NOT 0) INX BSIZE);%                     00308700 T  0216:2
                  BEGIN SI←S; N(DS←WDS; SI←SI-16; DI←DI-16);%                   00308800 T  0218:2
                        NDIV64(2(32(DS←WDS; SI←SI-16; DI←DI-16)));%             00308900 T  0220:1
                  END;%                                                         00309000 T  0222:3
                  TANK[0]←(BSIZE-I) INX IOD;%                                   00309100 T  0223:0
            END;%                                                               00309200 T  0225:0
            FIB[17]←I-(IF FIB[17]=0 THEN I ELSE FIB[17])%                       00309300 T  0225:0
                  +(STATUS≠0)×IOD.[8:10]+(STATUS=3);%                           00309400 T  0228:2
            FIB[16]←(BSIZE-1) INX FIB[16]&1[22:47:1];%                          00309500 T  0232:2
            FIB[19]←FIB[19]&(FIB[16] INX (STATUS≠3)%                            00309600 T  0236:0
                        -(STATUS=1)×FIB[18].[33:15])[33:33:15]%                 00309700 T  0238:1
                  &(IF STATUS=0 THEN FIB[19].[3:5]+2 ELSE%                      00309800 T  0240:3
                    IF STATUS=1 THEN 5 ELSE 7)[3:43:5]&1[22:47:1];%             00309900 T  0244:1
            FIB[5].[43:2]←3; FIB[13].[25:1]←1;%                                 00310000 T  0248:3
            TANK[NOT 1]←P(DUP,LOD)&1[22:47:1];%                                 00310100 T  0253:3
            MEM[FIB[16] INX 1]←I;%                                              00310200 T  0256:3
            MEM[FIB[16] INX NOT(I-1)]←I;%                                       00310300 T  0259:0
FILL:       REFILL;%                                                            00310400 T  0262:0
            P(MKS,0,1,TANK,REED,MKS,0,0,TANK,REED);%                            00310500 T  0263:0
      END ELSE%                                                                 00310600 T  0265:2
      BEGIN IF COUNT THEN IF I=1 THEN%                                          00310700 T  0265:2
            BEGIN P(MKS,1,0,(NOT 2) INX TANK,4,SELECT);%                        00310800 T  0268:1
                  FIB[13].[25:1]←1;%                                            00310900 T  0271:0
                  P(TANK,0,11,COM,DEL,DEL);%                                    00311000 T  0273:2
                  P(MKS,ACT1,0,TANK,1,SELECT);%                                 00311100 T  0275:0
                  GO TO EXIT;%                                                  00311200 T  0276:2
            END;%                                                               00311300 T  0277:0
            P((-I)); SPACE; EMPTY;%                                             00311400 T  0277:0
      END;%                                                                     00311500 T  0280:0
      GO TO EXIT;%                                                              00311600 T  0280:0
RR:   IF INOUT THEN%                                                            00311700 T  0280:2
      BEGIN T1←COUNT; P(I); SPACE; MOVEUP;%                                     00311800 T  0280:3
      FIB[17]←I-(IF FIB[17]=0 THEN I ELSE FIB[17])+(STATUS=3)%                  00311900 T  0285:0
            +(STATUS≠0)×IOD.[8:10];%                                            00312000 T  0289:2
      FIB[16]←FIB[16]&P(DUP,1,INX,BSIZE,-)[33:33:15]&0[22:22:1];%               00312100 T  0292:2
      FIB[19]←FIB[19]&(FIB[16] INX (STATUS=3))[33:33:15]&0[22:22:1]%            00312200 T  0296:2
            &(P(DUP).[3:5]-STATUS&(NOT STATUS)[46:46:1])[3:43:5];%              00312300 T  0299:2
            TANK[NOT 1]←P(DUP,LOD)&0[22:22:1];%                                 00312400 T  0304:0
      FIB[5].[43:2]←2; FIB[13].[25:1]←0; GO FILL;%                              00312500 T  0307:0
      END;%                                                                     00312600 T  0312:2
      IF COUNT THEN IF I=1 THEN%                                                00312700 T  0312:2
      BEGIN P(MKS,0,0,(NOT 2) INX TANK,4,SELECT); GO TO EASY END;%              00312800 T  0315:1
      P(I-1); SPACE; MOVEUP;%                                                   00312900 T  0318:2
```

```
          FIB[16]←FIB[16]&P(DUP,1,INX,BSIZE,-)[33:33:15];%                                00313000 T  0321:0
          FIB[19]←(STATUS=3) INX FIB[16]&FIB[18] [8:38:10];%                              00313100 T  0324:0
          EMPTY;%                                                                         00313200 T  0327:3
          P(MKS,0,0,0,(-1),TANK, RITE,MKS,0,0,0,BSIZE,TANK, RITE);%                       00313300 T  0329:0
          GO TO EXIT;%                                                                    00313400 T  0332:3
%                                                                                         00313500 T  0333:1
DK:;      IF FIB[4].[27:3]=1 THEN%                                                        00313600 T  0333:1
          BEGIN FIB[5].[43:2]←ACT2;%                                                      00313700 T  0335:2
                FIB[16].[24:1]←ACT2←ACT2≠0;%                                              00313800 T  0338:2
                FIB[19]←FIB[19]&ACT2[24:47:1]&0[25:47:1];                                 00313900 T  0342:0
          END ELSE%                                                                       00314000 T  0345:2
          IF FIB[4].[27:3]=0 THEN%                                                        00314100 T  0345:2
REW:%                                                                                     00314200 T  0347:2
          BEGIN IF ACT2=1 THEN ACT2←FIB[5].[43:2]ELSE%                                    00314300 T  0348:0
                IF ACT2=4 THEN ACT2←0 ELSE%                                               00314400 T  0350:3
                IF ACT2=3 THEN ACT1←FIB[7]-1 ELSE%                                        00314500 T  0353:1
                IF FIB[5].[43:2]=3 THEN ACT1←FIB[7]+1 ELSE%                               00314600 T  0356:2
                ACT1←FIB[7];%                                                             00314700 T  0360:2
                P(MKS,0,0,(NOT 2)INX TANK,4,SELECT);%                                     00314800 T  0362:0
                FIB[13]←FIB[13]&(ACT2=3)[25:47:1]&(ACT2≠0)[27:47:1];%                     00314900 T  0364:1
                FIB[7]←ACT1;%                                                             00315000 T  0368:3
                P(TANK,0,11,COM,DEL,DEL);%                                                00315100 T  0370:0
          END ELSE%                                                                       00315200 T  0371:2
          BEGIN IF ACT2=3 THEN GO TO ERR;%                                                00315300 T  0371:2
                IF ACT2=1 OR ACT2=4 THEN GO TO REW;%                                      00315400 T  0373:1
                IF ACT2=0 THEN BEGIN HEADER←*[FIB[14]];%                                  00315500 T  0375:3
                                     IF FIB[7]>HEADER[7] THEN%                            00315600 T  0378:1
                                        HEADER[7]←FIB[7];%                                00315700 T  0379:2
                                     P(MKS,0,1,TANK,REED,MKS,0,0,TANK,REED);              00315800 T  0381:2
                                END ELSE%                                                 00315900 T  0384:0
                                P(MKS,1,0,0,(-1),TANK,RITE,%                              00316000 T  0384:0
                                  MKS,1,0,0,FIB[18].[33:15],TANK,RITE);%                  00316100 T  0386:2
             GO TO EXIT;                                                                  00316200 T  0389:0
          END; GO EXIT;                                                                   00316300 T  0389:2
L6:       FIB ← *TANK; TANK ←[TANK[3]];     % SORT REEL SWITCHING                        00316400 T  0390:0
          IF ACT1 = 1 THEN                                                                00316500 T  0392:1
             BEGIN % I/O COMPLETE BUT NOT PRESENT                                         00316600 T  0393:0
              IF NOT (*TANK).[27:1] THEN  % PARITY                                        00316700 T  0393:2
                   P(1,[TANK[NOT 2]],19,17,COM)  % TERMNATE ON PARITY                     00316800 T  0394:3
                 ELSE                                                                     00316900 T  0397:1
               BEGIN % EOF OR EOR                                                         00317000 T  0397:1
                 P(TANK,11,11,COM,DEL,DEL);   % READ ENDING LABEL                       00317100 T  0397:3
                 IF MEM[TANK[NOT 1] INX 4].[42:6] =0 THEN P(1,RTN); %EOF                  00317200 T  0399:1
                 T1← FIB[13].[28:10] + 1; % REEL # + 1                                    00317300 T  0403:3
                 P(MKS,4,0,[TANK[NOT 2]],4,SELECT);  % CLOSE PURGE                      00317400 T  0405:3
                 FIB[13].[28:10] ← T1;                                                    00317500 T  0408:0
                 P([TANK],0,11,COM); P(0,RTN);                                            00317600 T  0410:2
               END;                                                                       00317700 T  0412:0
             END;                                                                         00317800 T  0412:0
          IF ACT1 = 0 THEN  % REEL SWITCH ON OUTPUT                                       00317900 T  0412:0
               BEGIN                                                                      00318000 T  0412:3
                 HEADER ← TANK[NOT 1]; HEADER[4].[42:6] ← 1; % EOR FLAG                   00318100 T  0413:1
                 T1 ← FIB[13].[28:10] + 1;                                                00318200 T  0417:2
                 P(MKS,7,0,[TANK[NOT 2]],4,SELECT);                                       00318300 T  0419:2
                 FIB[13].[28:10] ← T1;                                                    00318400 T  0421:3
                 P( TANK,0,11,COM); P(XIT);                                               00318500 T  0424:1
               END;                                                                       00318600 T  0425:2
```

```
      IF ACT1 = 2 THEN    % REWIND OUTPUT                                                  00318700 T  0425:2
            BEGIN                                                                         00318800 T  0426:1
              T1 ← IF FIB[13],[28:10] = 1 THEN 0 ELSE 7;                                  00318900 T  0426:3
              P(MKS,T1,0,[TANK[NOT 2]],4,SELECT); P(XIT);                                 00319000 T  0430:1
            END; P(XIT);                                                                  00319100 T  0432:3
MYUSERR: %%% BRANCH TO HERE IF I=7;                                                       00319110 T  0433:0
    P(TANK[NOT 3]);  TANK[NOT 3]←TANK[NOT 4]←0;  P(MKS,9,JUNK,DEL) ;                      00319120 T  0433:0
    FIB←TANK[NOT 2];  HEADER←P([HEADER[1]],CFX,SFB) & 10[8:38:10] ;                       00319130 T  0438:2
    STREAM(P1←ACT2,P2←[FPB[FIB[4],[13:11]]],P3←FIB[5],[11:2],HEADER) ;                    00319140 T  0442:3
        BEGIN DS←5LIT"=FAE(";  SI←P2;  SI←SI+1;  DS←7CHR;  SI←SI+1;                       00319150 T  0446:1
        DS←LIT"/";  TALLY←0;  7(IF SC=" " THEN JUMP OUT;  SI←SI+1;                        00319152 T  0448:1
        TALLY←TALLY+1);  SI←P2;  SI←SI+9;  P2←TALLY;  DS←P2 CHR ;                         00319154 T  0450:2
        DS←7LIT",MYUSE=";  SI←LOC P3;  DS←DEC;  DS←8LIT") TRIED " ;                       00319160 T  0452:1
        SI←LOC P1;  SI←SI+2;  DS←6CHR;  DS←2LIT":←" ;                                     00319170 T  0455:1
        END OF STREAM ;                                                                   00319180 T  0456:2
    P([HEADER[0]],[33:15],34,COM) ;                                                       00319190 T  0456:3
EXIT::%                                                                                   00319200 T  0458:1
END SELECT;%                                                                              00319300 T  0459:0
                                                                                                    SIZE= 0460 WORDS
```

```
 PROCEDURE INTRINSIC(DUPE,D,NUMDIM,SIZE,TYPE);%                                           00400000 T  0000:0
                                                                                  START OF REL SEGMENT; DISK ADDRESS = 00069
                    VALUE DUPE,D,NUMDIM,SIZE,TYPE;%                                       00400100 T  0000:0
                    ARRAY DUPE[*];NAME D;%                                                00400200 T  0000:0
                   INTEGER NUMDIM,SIZE,TYPE;%                                             00400300 T  0000:0
    BEGIN%                                                                                00400400 T  0000:0
        NAME DUM=TYPE,A;%                                                                 00400500 T  0000:0
    ARRAY DOPE=-8[*];%                                                                    00400600 T  0000:0
    ARRAY PRTPOINTER=10[*];%                                                              00400700 T  0000:0
    REAL NUMBUFF=-7,IOT=-2,MODE=-6,FILENO=-9,BUFFSIZE=-5;%                                00400800 T  0000:0
    REAL DISPOSITION=-10,ROWSIZE=-11,NUMROWS=-12,RECSIZE=0;%                              00400900 T  0000:0
    NAME E;%                                                                              00401000 T  0000:0
        INTEGER I,J,K;%                                                                   00401100 T  0000:0
        REAL C;%                                                                          00401200 T  0000:0
        BOOLEAN B;%                                                                       00401300 T  0000:0
        ARRAY AIT=6[*];%                                                                  00401400 T  0000:0
     REAL RECURSE=5; INTEGER BLOCKCTR=16;%                                                00401500 T  0000:0
    NAME M=2;                                                                             00401600 T  0000:0
    ARRAY FIB[*];                                                                         00401700 T  0000:0
ARRAY FPB=3[*],SEGDICT=4[*];                                                              00401800 T  0000:0
        INTEGER TIPE=-2,CYCLE=-3,DATE=-4,REEL=-5,FID=-6,MFID=-7;                          00401900 T  0000:0
        NAME FLE=-8;                                                                      00402000 T  0000:0
    LABEL EXIT,AOK,UPDATEFPB;                                                             00402100 T  0000:0
    REAL PTR=-11,APTR=-10,LBO=-7,DIMO=-6,LBN=-5,DIMN=-4,MAXLB,MINUB,                      00402200 T  0000:0
        UBO,UBN,N,TP,H;                                                                   00402300 T  0000:0
        ARRAY ARRY = MINUB[*];                                                            00402310 T  0000:0
        INTEGER DIM1 = UBO,  DIM2 = UBN;                                                  00402320 T  0000:0
    ARRAY OAT=11[*],NEW=-8[*],OLD=-9[*];                                                  00402400 T  0000:0
    NAME MAT=-2,NAT;                                                                      00402500 T  0000:0
    BOOLEAN OWNTOG,REDECLTOG,TASKARRAYTOG,AUXTOG;                                         00402600 T  0000:0
    LABEL FOUND,ARROUNDFOUND,TY12;                                                        00402700 T  0000:0
    NAME PHILE=-10;                                                                       00402800 T  0000:0
        LABEL TY0,TY1,TY2,TY3,TY4,TY5,TY6,TY7,TY8,TY9,TY10;%                              00402900 T  0000:0
        LABEL TY11,TY13,TY14,TY15,TY16,TY17,TY18,TY19;                                    00403000 T  0000:0
```

```
          SWITCH SW←TY0,TY1,TY2,TY3,TY4,TY5,TY6,TY7,TY8,TY9,TY10,TY11,          00403100 T  0000:0
                 TY12,TY13,TY14,TY15,TY16,TY17,TY18,TY19;                       00403200 T  0000:0
          TASKARRAYTOG←TYPE.[1:1];AUXTOG←TYPE.[41:1];TYPE←TYPE AND @77;         00403250 T  0000:0
          GO TO SW[TYPE];%                                                      00403300 T  0008:3
          TY0::TY1:TY2:TY3:%                                                    00403400 T  0019:3
               OWNTOG←TYPE.[46:1];                                              00403500 T  0020:0
                    I←AIT[J←0];TYPE←TYPE+4;%                                    00403600 T  0021:1
     E←P(NUMDIM=1+OWNTOG×(NUMDIM=1),NOT,[NUMDIM],INX);                          00403700 T  0024:0
     A←P(SIZE=1+OWNTOG,NOT,[E],INX);IF P([E[0]],DUP,LOD,XCH,ISN)≤0 OR            00403800 T  0027:2
                                       E[0]>1023 THEN                           00403900 T  0031:3
                                            P(E[0],TRUE,1,29,COM);              00404000 T  0032:3
     IF OWNTOG THEN BEGIN                                                       00404100 T  0034:2
                         H←OAT[0];NAT←P(0,NOT,[E],INX);                         00404200 T  0035:1
                         FOR K←1 STEP 1 UNTIL H DO                              00404300 T  0037:3
                         IF A[J].[CF]=OAT[K].[1:15] THEN GO TO FOUND;           00404400 T  0039:0
                         FOR C←1 STEP 1 UNTIL SIZE DO                           00404500 T  0044:2
                             OAT[H←H+1]←0&A[C=1] [1:33:15];                     00404600 T  0046:0
                         FOR C←1 STEP 1 UNTIL 2×NUMDIM  DO                      00404700 T  0052:3
                             P(NAT[C=1],[OAT[H←H+1]],ISD);                      00404800 T  0057:0
                         OAT[0]←H;GO AOK;                                       00404900 T  0060:3
          FOUND:REDECLTOG←TRUE;                                                 00405000 T  0062:2
               STREAM(R←0:NUMDIM,A←[OAT[TP←K+SIZE  ]],B←[NAT]   );              00405100 T  0063:1
                    BEGIN SI←A;TALLY←1;                                         00405200 T  0066:1
                     NUMDIM(IF 16 SC≠DC THEN TALLY←0); R←TALLY ;                00405300 T  0066:3
                    END;                                                        00405400 T  0068:2
          IF P(NOP) THEN GO TO EXIT;                                            00405500 T  0068:3
     ARROUNDFOUND: A[J]←[PRTPOINTER[17]];                                       00405600 T  0069:3
                    END;                                                        00405700 T  0071:2
AOK:DO                                                                          00405800 T  0071:2
                    BEGIN%                                                      00405900 T  0071:2
               B←NUMDIM≠1;C←((IF AUXTOG THEN 3 ELSE (TYPE.[47:1] OR B))         00406000 T  0071:2
                         &A[J][CTF] &   E[0][8:38:10]);%                        00406100 T  0075:2
               IF NOT OWNTOG THEN                                               00406200 T  0078:1
                         AIT[I←I+1]←C & TYPE[2:46:1]&BLOCKCTR[8:38:10]%         00406300 T  0078:3
                         &NUMDIM[3:43:5];        P(FLAG(C),A[J],STD);%          00406400 T  0082:1
                         IF TASKARRAYTOG THEN AIT[I],[1:2] ← 3;                 00406450 T  0086:1
                         IF B THEN%                                             00406500 T  0089:2
               P(MKS,FLAG(C),[E[1+OWNTOG]],NUMDIM=1,E[0],                       00406600 T  0089:3
                         TYPE&AUXTOG[41:47:1],RECURSE) %                        00406700 T  0093:1
                    END%                                                        00406800 T  0094:3
               UNTIL ((J←J+1)=SIZE) OR REDECLTOG;                               00406900 T  0094:3
          IF REDECLTOG THEN                                                     00407000 T  0097:2
               BEGIN %REMAP                                                     00407100 T  0097:3
P(MKS,TP+2,2,M[OAT[K],[1:15]],[PRTPOINTER[17]],LOD,                             00407200 T  0098:1
                    OAT[TP],OAT[TP+1],NAT[0],NAT[1],NUMDIM,[NAT],12             00407300 T  0102:0
                    , RECURSE,[M[OAT[K],[1:15]]],LOD,NUMDIM,25,COM,DEL,         00407400 T  0105:1
                    DEL);K←K+1;                                                 00407500 T  0108:2
                    IF J<SIZE THEN GO ARROUNDFOUND;                             00407600 T  0110:0
                    STREAM(NUMDIM,A←[NAT[0]],B←[OAT[TP]]);                      00407700 T  0111:1
                     BEGIN SI←A;NUMDIM(DS←2 WDS) END;                           00407800 T  0112:3
               END; AIT[0]←I;GO TO EXIT;                                        00407900 T  0114:1
;         GO TO EXIT;%                                                          00408000 T  0116:0
          TY4::TY5:TY6:TY7:%                                                    00408100 T  0116:2
               OWNTOG←TYPE.[46:1];                                              00408200 T  0117:0
     IF P(D,DUP,LOD,XCH,ISN,DUP )≤0 OR P(XCH)>1023 THEN P(D[0],1,1,29,COM       00408300 T  0118:1
               );  DO                                                           00408400 T  0122:3
                    BEGIN%                                                      00408500 T  0123:0
```

```
                    B←NUMDIM≠1;                                                  00408510 T  0123:0
                    DUPE[K]←FLAG(C←((IF AUXTOG THEN 3 ELSE                       00408600 T  0124:1
                                    (TYPE,[47:1] OR B))%                         00408700 T  0126:1
                    &[DUPE[K]][CTF]&D[0][8:38:10]));%                            00408800 T  0127:2
                    IF B THEN%                                                   00408900 T  0130:2
               P(MKS,FLAG(C),[D[1+OWNTOG]],NUMDIM-1,D[0],                        00409000 T  0130:3
                    TYPE&AUXTOG[41:47:1],RECURSE) %                              00409100 T  0134:1
                    END%                                                         00409200 T  0135:3
              UNTIL(K←K+1)=SIZE%                                                 00409300 T  0135:3
;       GO TO EXIT;%                                                             00409400 T  0137:0
  TY12:                                                                          00409500 T  0138:2
       IF LBO<LBN THEN MAXLB←LBN                                                 00409600 T  0138:2
                  ELSE MAXLB←LBO;                                                00409700 T  0139:3
       UBO←LBO+DIMO-1;                                                           00409800 T  0141:3
       UBN←LBN+DIMN-1;                                                           00409900 T  0143:2
       IF UBO<UBN THEN MINUB←UBO                                                 00410000 T  0145:1
                  ELSE MINUB←UBN;                                                00410100 T  0146:2
       N←MINUB-MAXLB+1;                                                          00410200 T  0148:2
       IF NUMDIM=1 THEN     BEGIN                                                00410300 T  0150:1
       IF N≤0 THEN GO TO EXIT;                                                   00410400 T  0151:2
       STREAM(N,M←N,[38:4],A←[OLD[MAXLB-LBO]],B←[NEW[MAXLB-LBN]]);               00410500 T  0152:3
            BEGIN SI←A;M(DS←32 WDS;DS←32 WDS));DS←N WDS; END;                    00410600 T  0156:1
                       END                                                       00410700 T  0158:2
                  ELSE                                                           00410800 T  0158:2
       FOR I←0 STEP 1 UNTIL N-1 DO                                               00410900 T  0158:2
            P(MKS,PTR+2,APTR+2,[OLD[MAXLB-LBO+I]],LOD,                           00411000 T  0163:1
             [NEW[MAXLB-LBN+I]],LOD,OAT[PTR],OAT[PTR+1],                         00411100 T  0166:3
             MAT[APTR],MAT[APTR+1],NUMDIM-1,[MAT],12,RECURSE);                   00411200 T  0170:0
       GO TO EXIT;                                                               00411300 T  0174:2
TY8::%                                                                           00411400 T  0175:0
       P(IF NUMBUFF<1 THEN 1 ELSE NUMBUFF, ,NUMBUFF, ISD));                      00411500 T  0175:0
       P(NUMDIM, ,NUMDIM, ISD));                                                 00411600 T  0177:3
       P(RECSIZE, ,RECSIZE, ISD));                                               00411700 T  0178:2
       P(BUFFSIZE, ,BUFFSIZE, ISD));                                             00411800 T  0179:1
       P(ROWSIZE, ,ROWSIZE, ISD));                                               00411900 T  0180:0
       IF P(NUMROWS, ,NUMROWS, ISN)>20 THEN                                      00412000 T  0180:3
            P(NUMROWS, TRUE, 2, 29, COM);                                        00412100 T  0182:0
            P(MKS, *P( ,DOPE),(NUMBUFF=1)+NUMBUFF+27,                            00412200 T  0183:3
                 1,1,1,RECURSE);                                                 00412210 T  0186:1
            AIT[AIT[0]]←-AIT[AIT[0]];%                                           00412300 T  0187:1
            DOPE←*[DOPE];%                                                       00412400 T  0189:2
            DOPE[2]←[DOPE[(NUMBUFF=1)+NUMBUFF+5]]&22[8:38:10];                   00412500 T  0190:3
       DOPE[4]←[DOPE[5]];%                                                       00412600 T  0194:3
       DOPE[3]←0&10[8:38:10];%                                                   00412700 T  0196:1
                 I←0;C← @20002020000000 &(IOT≠10)[24:47:1]&MODE%                 00412800 T  0198:2
       [27:47:1]&([DOPE[6]])[CTC];%                                              00412900 T  0201:0
                 WHILE(I←I+1)≤NUMBUFF DO%                                        00413000 T  0203:1
                 DOPE[I+4]←FLAG(C);%                                             00413100 T  0205:2
                 DOPE←*[DOPE[2]];%                                               00413200 T  0209:0
       STREAM(T←[NUMDIM]); BEGIN SI←T; DS ← 8 DEC END;%                          00413300 T  0210:1
       FILENO←(FILENO-1)×ETRLNG;%                                                00413400 T  0211:3
       DOPE[4]←NUMDIM&FILENO[13:37:11]&1[12:47:1]&3[8:44:4]%                     00413500 T  0213:2
              &(IOT=11)[6:47:1]&DISPOSITION[25:46:2];%                           00413600 T  0216:2
       IF RECSIZE=0 THEN%                                                        00413700 T  0220:1
       BEGIN RECSIZE←BUFFSIZE; I←0 END ELSE%                                     00413800 T  0221:0
       IF BUFFSIZE≤RECSIZE THEN%                                                 00413900 T  0223:0
       BEGIN I←BUFFSIZE; BUFFSIZE←RECSIZE; RECSIZE←I; I←1 END%                   00414000 T  0224:1
```

```
                    ELSE I←3;%                                                    00414100 T  0227:3
                    DOPE[5]←I & (IOT≠10)[43:47:1] & 1[42:47:1] & (IF TYPE←(       00414200 T  0229:0
                          TYPE←FPB[FILENO+3].[43:5])=1 OR TYPE=4 OR TYPE=6        00414205 T  0232:1
                          OR (TYPE>14 AND TyPE<19) THEN 2 ELSE 3)[11:46:2]        00414210 T  0236:0
                          & (IF TYPE THEN 0 ELSE 3)[9:46:2] & (IF TYPE THEN       00414215 T  0240:3
                          4 ELSE 0)[13:45:3] ;                                    00414220 T  0244:1
                    STREAM(FB←[FPB[FILENO]],R←[I]);%                              00414300 T  0247:0
                    BEGIN SI←FB; SI←SI+16; DS←3 OCT END;%                         00414400 T  0248:1
                    DOPE[13]←0&NUMBUFF[1:39:9]&MODE[24:47:1]%                     00414500 T  0249:1
                          &I[28:38:10]&NUMBUFF [10:39:9]%                         00414600 T  0251:1
                          &(IOT≠10)[27:47:1];%                                    00414700 T  0253:1
                    DOPE[18]←RECSIZE&BUFFSIZE[3:33:15]&BUFFSIZE[18:33:15];%       00414800 T  0256:0
                    DOPE[8]←ROWSIZE&NUMROWS[15:38:10];%                           00414900 T  0258:3
$ SET OMIT = NOT SHAREDISK                                                        00414949 T  0261:0
              GO TO EXIT;%                                                        00415000 T  0261:0
              TY9::%                                                              00415100 T  0261:2
                    BEGIN IF NUMDIM ≠ 0 THEN                                      00415200 T  0262:0
                         IF NUMDIM ≠ 15 THEN BEGIN                                00415300 T  0262:3
                         IF (J ← (C ← NUMDIM).[8:10]) ≠ BLOCKCTR THEN%            00415400 T  0264:2
                              BEGIN BLOCKCTR ← J+1;%                              00415500 T  0266:3
                                   P(10,COM);%                                    00415600 T  0268:2
                              END;%                                               00415700 T  0269:0
                         P(SIZE,.NUMDIM,←);%                                      00415800 T  0269:0
                         IF (J ← C.[18:15]) = 0 THEN%                             00415900 T  0269:3
                         J ← PRTPOINTER.[18:15]+2;%                               00416000 T  0271:2
                    DO UNTIL (*(PRTPOINTER&(J←HUNT(J+1) INX 0)[33:33:15])).%      00416100 T  0274:0
                              [1:3]=4 AND M[J].[6:12] ≠0;                         00416200 T  0277:0
                         P((*[PRTPOINTER[C.[33:15]]])&J[18:33:15],BRT);%          00416300 T  0280:3
                    END                                        END;%              00416400 T  0282:3
                    NUMDIM←0;                                                     00416500 T  0282:3
              GO TO EXIT;%                                                        00416600 T  0283:2
              TY10::%                                                             00416700 T  0284:0
                    AIT[AIT[0]←AIT[0]+1] ←-2&1[8:38:10]&[SIZE][18:33:15];%        00416800 T  0284:0
              GO TO EXIT;%                                                        00416900 T  0288:3
TY11::        FIB←FLE[NOT 2];                                                     00417000 T  0289:1
              IF FIB[5].[41:2] = 0 THEN GO TO EXIT; % FILE OPENED                 00417100 T  0291:3
              IF FIB[5].[42:1]     THEN GO TO UPDATEFPB; % CLOSED,RELEASED        00417110 T  0293:3
              IF FIB[4].[24:2] ≠ 1 THEN GO TO EXIT; % REWOUND                     00417120 T  0295:2
              IF FIB[4].[8:4] ≠ 2 THEN GO TO EXIT;       % MUST BE TAPE          00417125 T  0297:2
              MFID ← TIPE ← -0; % PREVENT CHANGE IN MFID OR TYPE                  00417130 T  0299:2
UPDATEFPB:                                                                        00417140 T  0301:0
              BEGIN                                                               00417150 T  0301:0
                    STREAM(A←0:MFID,FID,REEL←REEL←REEL,DATE←DATE←DATE,            00417200 T  0301:0
                         CYCLE←CYCLE←CYCLE,TIPE←TIPE←TIPE,                        00417300 T  0303:3
                         F← I← [FPB[FIB[4].[13:11]]]);                            00417400 T  0305:1
                    BEGIN SI←LOC MFID;                                            00417500 T  0307:1
                    2(IF SC="+" THEN BEGIN SI←SI+8; DI←DI+8 END ELSE              00417600 T  0307:2
                      IF SC="0" THEN DS←WDS ELSE BEGIN TALLY←1; A←TALLY;          00417610 T  0309:0
                      JUMP OUT TO ERR END);                                       00417620 T  0310:2
                    IF SC="+" THEN BEGIN SI←SI+8; DI←DI+3 END ELSE DS←3 DEC;      00417700 T  0311:1
                    IF SC="+" THEN BEGIN SI←SI+8; DI←DI+5 END ELSE DS←5 DEC;      00417800 T  0312:3
                    IF SC="+" THEN BEGIN SI:=SI+8;DI:=DI+2 END ELSE               00417900 T  0314:1
                                       BEGIN DS← DEC; DI←DI+1; END;               00417910 T  0315:2
                    IF SC≠"+" THEN BEGIN SI←SI+7; DI←DI+5; DS←CHR END;            00418000 T  0316:0
                    ERR: END;                                                     00418100 T  0317:1
                    IF P THEN P((-75),34,COM); % DS = INVALD FILE NAME            00418110 T  0317:2
                    IF REEL≥0 THEN FIB[13].[28:10]←REEL END;                      00418200 T  0319:0
```

```
                I← P(.I,LOD) INX 0;              % MARK MFID OF REMOTE FILE        00418210 T  0322:3
                IF M[I INX 3].[43:5]=19 THEN     % WHICH HAS BEEN FILLED           00418220 T  0324:1
                     M[I]← P(DUP,LOD,SSN);       % SO FILE OPEN WILL KNOW           00418230 T  0326:3
           GO TO EXIT;                                                             00418300 T  0329:1
      TY13::    C←1; I←AIT[J←0];                                                   00418400 T  0329:3
                DO AIT[I←I+1]←(M[[TYPE]INX NOT C])&BLOCKCTR[8:38:10]               00418500 T  0332:1
                          &1[1:46:2] UNTIL(C←C+1)> SIZE;                           00418600 T  0335:3
                AIT[0]←I; GO EXIT;                                                 00418700 T  0340:1
      TY14::    IF TIPE<3 THEN TIPE←0 ELSE IF TIPE>5 THEN TIPE←5;        %AS     00418800 T  0342:0
                IF TIPE≠0 THEN DO % DECLARE SORT FILES                   %AS     00418900 T  0346:2
                BEGIN P(MKS,0,0,3,CYCLE+I,[D[I]],2,1,10,0,3,11,8,RECURSE           00419000 T  0347:3
                     ,D[I],5,CDC,@1612-I,←);                             %AS     00419100 T  0352:0
                END UNTIL (I←I+1)≥TIPE;                                  %AS     00419200 T  0354:3
                P(TIPE,RTN);                                             %AS     00419300 T  0357:0
      COMMENT   TY14 DECLARES SORT TAPE FILES FOR ALGOL;                 %AS     00419400 T  0357:2
                GO EXIT;                                                         00419500 T  0357:2
TY15:: PHILE[NOT 3] ← IOT;  PHILE[NOT 4] ← NUMDIM;  GO TO EXIT;                  00419600 T  0358:0
TY16:: TY17: TY18:                                                               00419700 T  0362:0
           E ← M OR (*(P(.NUMDIM)+SIZE)).[18:15];                                00419705 T  0362:0
           IF SIZE = 0 THEN     COMMENT FISH OUT OLD SIZES TO USE;               00419710 T  0364:3
           BEGIN;STREAM(A←0:S←*E);                                               00419715 T  0365:2
                BEGIN TALLY←1;  SI←S;  SI←SI-16;  SKIP 2 SB;                     00419720 T  0367:2
                     IF SB THEN TALLY←2;  A←TALLY;                               00419725 T  0368:2
                END STREAM;                                                      00419730 T  0369:2
                IF(SIZE:=P) THEN DIM2:=(*E).[8:10] ELSE                          00419735 T  0369:3
                     DIM2:=P(*E,P(DUP).[8:10],,DIM1,STD,0,CDC,LOD).[8:10];       00419740 T  0372:1
           END ELSE BEGIN DIM2 ← NUMDIM;                                         00419745 T  0376:1
                IF NOT SIZE THEN DIM1 ← RECSIZE;                                 00419750 T  0377:2
           END;                                                                  00419755 T  0379:1
           IF TYPE = 18    COMMENT "ION" FUNCTION;                               00419760 T  0379:1
             THEN IF SIZE OR (DIM1 NEQ DIM2) THEN P((-54), 26, COM);             00419770 T  0379:3
           POLISH(SIZE,E,39,COM,DEL,DEL); % RETURN OLD ARRAY                     00419780 T  0383:1
           POLISH(MKS, E);  IF NOT SIZE THEN P(DIM1);                            00419790 T  0384:3
             POLISH(DIM2, SIZE, 1, 0, RECURSE);                                  00419800 T  0386:2
           ARRY ← *E;  DIM1 ← DIM1-1;                                            00419810 T  0387:3
           IF TYPE = 17 THEN    COMMENT "CON" FUNCTION;                          00419820 T  0390:0
           BEGIN IF SIZE THEN P(*E, 2, CCX, E, ←) ELSE                           00419830 T  0390:3
                FOR I←1 STEP 1 UNTIL DIM1 DO                                     00419840 T  0393:2
                  POLISH([ARRY[I]], DUP, LOD, 2, CCX, XCH, ←);                   00419850 T  0395:0
           END;                                                                  00419860 T  0399:1
           IF TYPE = 18 THEN    COMMENT "ION" FUNCTION AGAIN;                    00419870 T  0399:1
             FOR I←1 STEP 1 UNTIL DIM1 DO                                        00419880 T  0400:0
                P(*[ARRY[I]], I, CDC, 1, XCH, ←);                                00419890 T  0402:0
           GO TO EXIT;                                                           00419900 T  0406:1
      TY19::  % IMPLEMENTED FOR COBOL 68 ARRAY DECLARATION  1 OR 2 DIM           00420000 T  0406:3
           ARRY ← *[PRTPOINTER[17]];                                             00420100 T  0407:0
           M[[ARRY[0]] INX NOT 1].[2:1] := 1;       % MARK IT SAVE               00420105 T  0408:1
           FOR I ← 1 STEP 1 UNTIL ARRY[0] DO                                     00420110 T  0412:0
                BEGIN                                                            00420120 T  0416:0
                C ← ARRY[I];                                                     00420130 T  0416:0
                P(MKS,[PRTPOINTER[C.[FF]]],                                      00420140 T  0417:0
                     P(DUP,LOD,P(DUP).[FF],P(XCH),[CF]),                         00420150 T  0418:1
                     IF C.[17:1] THEN P(XCH,DEL) ELSE P,                         00420160 T  0420:1
                     C.[16:2],1,C.[CF],RECURSE);                                 00420170 T  0422:2
                END;                                                             00420180 T  0424:2
           SEGDICT[0] ← *P(DUP)-1;                  % DELETE TEMP AIT            00420185 T  0425:0
           P([PRTPOINTER[17]] INX M,3,COM,DEL);                                  00420190 T  0427:0
```

```
 EXIT::                                                                  00429900 T  0428:3
END INTRINSIC INTRINSIC;                                                 00430000 T  0428:3
                                                                                 SIZE= 0430 WORDS



PROCEDURE FILEATTRIBUTES(TANK,ERRL,DUM1,VAL,NAM,INFO,TEN) ;              00430050 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00084
VALUE TANK,DUM1,VAL,NAM,INFO,ERRL ;                                      00430100 T  0000:0
INTEGER VAL ;                                                            00430150 T  0000:0
REAL ERRL,DUM1,NAM,INFO ;                                                00430200 T  0000:0
NAME TANK ;                                                              00430250 T  0000:0
ARRAY TEN[*] ;                                                           00430300 T  0000:0
    BEGIN                                                                00430350 T  0000:0
    % THIS PROC HANDLES FILE ATTRIBUTES (FOR MORE INFO, REFER TO THE      00430400 T  0000:0
    % ALGOL COMPILER, PROCEDURE FILEATTRIBUTEHANDLER, FOR A DESCRIPTION   00430450 T  0000:0
    % OF THE VARIOUS KINDS OF FILE ATTRIBUTE CALLS).  DUM1 IS A DUMMY     00430500 T  0000:0
    % PARAMETER RESERVED FOR POSSIBLE FUTURE USE.                         00430525 T  0000:0
    %      TO ADD A NEW ATTRIBUTE, FIRST MAKE THE APPROPIATE CHANGES IN    00430550 T  0000:0
    % THE COMPILER(S), THEN, DECLARE TWO NEW LABELS, XN & XVN -- "X" IS    00430600 T  0000:0
    % THE FILE ATTRIBUTE, E.G., ACCESS, AND "N" IS THE CORRESPONDING       00430650 T  0000:0
    % SWITCH LABEL NUMBER, E.G., THE 4 IN MFID4 -- AND ATTACH XN ONTO      00430675 T  0000:0
    % GETFILATT, AND ATTACH XVN ONTO THE SWITCH SETFILATT.  THEN INSERT    00430700 T  0000:0
    % XN: AND ITS CODE BELOW THE LAST XN-TYPE CODE, AND INSERT XVN: AND    00430750 T  0000:0
    % ITS CODE BELOW THE LAST XVN-TYPE CODE.  THE XN-TYPE CODE SETS THE    00430800 T  0000:0
    % FILE ATTRIBUTE (AFTER MUCH CHECKING TO ASSURE THAT THE FILE IS OF    00430850 T  0000:0
    % THE PROPER TYPE AND IN THE PROPER STATUS, AND THAT THE VALUE OF      00430900 T  0000:0
    % VAL IS WITHIN THE PROPER BOUNDS), AND THE XVN-TYPE CODE RETRIEVES    00430950 T  0000:0
    % AND STACKS THE FILE ATTRIBUTE.                                      00431000 T  0000:0
                                                                         00431050 T  0000:0
    ARRAY FIB=+1[*],FPB=3[*] ;                                           00431100 T  0000:0
    REAL  FIB5=17,TYPE=9,FI=ERRL,SELECT=14,INTRINSIC=5,                  00431110 T  0000:0
          OPEN=FIB+1,                                                    00431111 T  0000:0
          NOTCLOSREL=OPEN+1,                                             00431112 T  0000:0
          NOTDISK=NOTCLOSREL+1,                                          00431113 T  0000:0
          RTNVAL=NOTDISK+1,                                              00431114 T  0000:0
          TEMP=RTNVAL+1,                                                 00431115 T  0000:0
          PMET=TEMP+1,                                                   00431116 T  0000:0
          FPB3=DUM1,VALSIGN=FIB,                                         00431135 T  0000:0
          MFIDX=OPEN,FIDX=NOTCLOSREL,REELX=NOTDISK,DATEX=FPB3,           00431140 T  0000:0
          CYCLEX=FIB5,TYPEX=TYPE ;                                       00431155 T  0000:0
                                                                         00431175 T  0000:0
    LABEL QUIT,EXIT,SETUSE,VALER,TOIT,BIG,CHK1,CHK2,CHK3L,CHK3T,MYUSERR, 00431200 T  0000:0
          OPENERR,CLOSRELERR                                             00431225 T  0000:0
          ,ACCESS0,ACCESSV0                                              00431250 T  0000:0
          ,MYUSE1,MYUSEV1                                                00431260 T  0000:0
          ,SAVE2,SAVEV2                                                  00431270 T  0000:0
          ,OTHERUSE3,OTHERUSEV3                                          00431280 T  0000:0
          ,MFID4,MFIDV4                                                  00431290 T  0000:0
          ,FID5,FIDV5                                                    00431300 T  0000:0
          ,REEL6,REELV6                                                  00431310 T  0000:0
          ,DATE7,DATEV7                                                  00431320 T  0000:0
          ,CYCLE8,CYCLEV8                                                00431330 T  0000:0
          ,TYPE9,TYPEV9                                                  00431340 T  0000:0
          ,AREAS10,AREASV10                                              00431350 T  0000:0
          ,AREASIZE11,AREASIZEV11                                        00431360 T  0000:0
```

```
      ,EUNUM12,EUNUMV12 % EU NUMBER FOR DISK                                  00431370 T  0000!0
      ,DSKSPEED13,DSKSPEEDV13 % FAST/SLOW DISK (1=FAST)                       00431380 T  0000!0
      ,TIMELIMIT14,TIMELIMITV14 % WAIT TIME FOR LOCKED ADDRESS (RLL)          00431390 T  0000!0
      ,IOSTATUS15,IOSTATUSV15   % LAST IO RESULT STATUS (RLL)                00431400 T  0000!0
      ,SENSITIVE14,SENSITIVEV14 % SENSITIVE                                   00431410 T  0000!0
; %%%% ADD NEW ATTRIBUTE LABELS ON A NEW LINE ABOVE ****************          00431490 T  0000!0
  %%%% AND BE SURE TO POST-FIX THE SWITCH NUMBER FOR DOCUMENTATION.          00431495 T  0000!0
                                                                              00431500 T  0000!0
SWITCH SETFILATT :=                                                           00431550 T  0000!0
                    ACCESS0                                                   00431551 T  0000!0
                   ,MYUSE1                                                    00431552 T  0000!0
                   ,SAVE2                                                     00431553 T  0000!0
                   ,OTHERUSE3                                                 00431554 T  0000!0
                   ,MFID4                                                     00431555 T  0000!0
                   ,FID5                                                      00431556 T  0000!0
                   ,REEL6                                                     00431557 T  0000!0
                   ,DATE7                                                     00431558 T  0000!0
                   ,CYCLE8                                                    00431559 T  0000!0
                   ,TYPE9                                                     00431560 T  0000!0
                   ,AREAS10                                                   00431561 T  0000!0
                   ,AREASIZE11                                                00431562 T  0000!0
                   ,EUNUM12                                                   00431563 T  0000!0
                   ,DSKSPEED13                                                00431564 T  0000!0
                   ,TIMELIMIT14                                               00431565 T  0000!0
                   ,IOSTATUS15                                                00431566 T  0000!0
                   ,SENSITIVE14                                               00431567 T  0000!0
; %%%% ATTACH THE NEW XN-TYPE ATTRIBUTE LABEL ONTO SWITCH ABOVE ****          00431840 T  0000!0
                                                                              00431850 T  0000!0
SWITCH GETFILATT :=                                                           00431900 T  0000!0
                    ACCESSV0                                                  00431901 T  0000!0
                   ,MYUSEV1                                                   00431902 T  0000!0
                   ,SAVEV2                                                    00431903 T  0000!0
                   ,OTHERUSEV3                                                00431904 T  0000!0
                   ,MFIDV4                                                    00431905 T  0000!0
                   ,FIDV5                                                     00431906 T  0000!0
                   ,REELV6                                                    00431907 T  0000!0
                   ,DATEV7                                                    00431908 T  0000!0
                   ,CYCLEV8                                                   00431909 T  0000!0
                   ,TYPEV9                                                    00431910 T  0000!0
                   ,AREASV10                                                  00431911 T  0000!0
                   ,AREASIZEV11                                               00431912 T  0000!0
                   ,EUNUMV12                                                  00431913 T  0000!0
                   ,DSKSPEEDV13                                               00431914 T  0000!0
                   ,TIMELIMITV14                                              00431915 T  0000!0
                   ,IOSTATUSV15                                               00431916 T  0000!0
                   ,SENSITIVEV14                                              00431917 T  0000!0
; %%%% ATTACH THE NEW XVN-TYPE ATTRIBUTE LABEL ONTO SWITCH ABOVE ***          00432109 T  0000!0
                                                                              00432200 T  0000!0
DEFINE CANTUSE = 0 #,                                                         00432250 T  0000!0
       IO      = 3 #,                                                         00432259 T  0000!0
       SERIAL  = 0 #,                                                         00432262 T  0000!0
       RANDOM  = 1 #,                                                         00432265 T  0000!0
       UPDATE  = 2 #,                                                         00432268 T  0000!0
       PROTECT = 3 #,                                                         00432269 T  0000!0
       DISK    = (NOT NOTDISK) #,                                             00432271 T  0000!0
       HEADER[HEADER1] = P(HEADER1,14,,FIB,LOD,INX,LOD,INX,LOD) #,            00432274 T  0000!0
       ERM(ERM1) = BEGIN                                                      00432276 T  0000!0
```

```
                    INITERR; STREAM(FI); DS←13LIT ERM1; GO QUIT ;
                    END #,
          CHKOPEN = BEGIN IF OPEN THEN GO OPENERR END #,
          CHKCLOSREL = BEGIN IF NOTCLOSREL THEN GO CLOSRELERR END #,
          CHKMYUSE = BEGIN IF FIB5.[11:2]=0 THEN GO MYUSERR END #,
          P        = POLISH #;

SUBROUTINE INITERR ;
     BEGIN
     PMET←P;  FOR TEMP←P STEP -1 UNTIL 1 DO P(←);  P(PMET) ;
     P(TANK[NOT 4]);  TANK[NOT 4]←0;  P(MKS,9,INTRINSIC,DEL) ;
     TEN←0; TEN←P([TEN[1]],CFX,SFB)&10[8:38:10] ;
     STREAM(TEMP←NAM:A←[FPB[FPB3-3]],N←NAM,[6:6],TEN) ;
          BEGIN DS←5LIT"-FAE,";  SI←A ;
          2(SI←SI+1;  A←SI;  TALLY←0;  7(IF SC=" " THEN
          JUMP OUT;  SI←SI+1;  TALLY←TALLY+1));  SI←A;  A←TALLY ;
          DS←A CHR;  DS←ODEC;  DS←LIT"/");  DI←DI-1;  DS←LIT"," ;
          SI←LOC TEMP;  SI←SI+2;  DS←N CHR;  DS←2LIT", ";  TEMP←DI ;
          DI←DI+13;  DS←2LIT":←" ;
          END STREAM ;
     FI←P ;
     END OF INITERR ;

REAL SUBROUTINE OCTTODEC ;
     BEGIN
     STREAM(Q←0:VAL); BEGIN SI←LOC VAL; DI←LOC Q; DS←8DEC END ;
     OCTTODEC←P ;
     END OF OCTTODEC ;

REAL SUBROUTINE DECTOOCT ;
     BEGIN PMET←P(XCH) ;
     STREAM(Q←0:PMET); BEGIN SI←LOC PMET; DI←LOC Q; DS←8OCT END ;
     DECTOOCT←P ;
     END OF DECTOOCT ;

SUBROUTINE INITIALIZE ;
     BEGIN % INITIALIZES A FEW USEFUL VARIABLES.
     NOTDISK←NOT((NOTDISK←(TYPE←FPB[FPB3←FIB[4],[13:11]+3] AND 63)
               AND 31)=10 OR NOTDISK=12 OR NOTDISK=13 OR
               NOTDISK=26) ;
     NOTCLOSREL←NOT(OPEN←(FIB5←FIB[5]),[41:2]) ;
     OPEN←OPEN=0 ;
     END OF INITIALIZE ;

SUBROUTINE SCATTERFPB ;
     STREAM(F←[FPB[FPB3-3]],D←[DATEX],C←[CYCLEX],M←[MFIDX]) ;
          BEGIN
          SI←F;  2(DS←LIT"0"; SI←SI+1; DS←7CHR); DS←3OCT ;
          DI←D;  DS←5OCT; DI←C;  DS←2OCT ;
          END OF SCATTERFPB ;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%   *******   F I R S T   E X E C U T A B L E   C O D E   ********
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
00432277 T  0000:0
00432278 T  0000:0
00432280 T  0000:0
00432285 T  0000:0
00432290 T  0000:0
00432475 T  0000:0
00432480 T  0000:0
00432500 T  0000:0
00432550 T  0001:0
00432560 T  0001:0
00432600 T  0006:0
00432700 T  0010:0
00432750 T  0013:1
00432800 T  0016:2
00432850 T  0017:3
00432900 T  0019:2
00432950 T  0021:1
00432975 T  0023:2
00432980 T  0025:1
00433100 T  0026:0
00433150 T  0026:1
00433200 T  0026:3
00433220 T  0027:0
00433230 T  0027:0
00433240 T  0027:0
00433250 T  0027:0
00433260 T  0029:1
00433270 T  0029:2
00433275 T  0029:3
00433280 T  0029:3
00433290 T  0030:0
00433295 T  0030:3
00433300 T  0033:0
00433310 T  0033:1
00433325 T  0033:2
00433650 T  0033:2
00433660 T  0034:0
00433670 T  0034:0
00433675 T  0036:2
00433677 T  0040:3
00433680 T  0042:2
00433685 T  0045:1
00433690 T  0046:2
00433725 T  0046:3
00433750 T  0046:3
00433760 T  0047:0
00433775 T  0049:1
00433780 T  0049:1
00433785 T  0051:1
00433790 T  0052:1
00435445 T  0052:3
00435446 T  0052:3
00435447 T  0052:3
00435448 T  0052:3
00435449 T  0052:3
00435450 T  0052:3
00435460 T  0052:3
```

```
      P(TANK[NOT 2],0,0,0,0,0,0); TANK[NOT 4]←ERRL; INITIALIZE ;                    00435480 T  0052:3
      IF (FI←INFO AND 255)>3 AND FI<10 THEN SCATTERFPB ;                            00435550 T  0059:0
      IF NAM≥0 THEN IF ABS(VAL)>@7777777777777 THEN GO VALER ELSE VAL←VAL;          00435600 T  0063:0
      IF INFO.[39:1] THEN IF INFO←INFO.[38:1] THEN RTNVAL←VAL                       00435750 T  0065:3
                          ELSE BEGIN TANK[NOT 4]←0; GO GETFILATT[FI] END            00435800 T  0069:2
      ELSE INFO←FALSE ;                                                             00435825 T  0083:1
      GO SETFILATT[FI] ;                                                            00435850 T  0084:2
      MYUSERR:: ERM("MYUSE=CANTUSE") ;                                              00435855 T  0094:0
      CLOSRELERR:: ERM("NOT CLOSRELES") ;                                           00435860 T  0098:2
      OPENERR:: INITERR; STREAM(FI); DS←13LIT"NOT RWND/CLSD" ;                      00435870 T  0103:2
      QUIT:: P([TEN[0]],[33:15],34,COM) ;                                           00435900 T  0108:0

ACCESSO:  IF ((FI←TYPE AND 31)=12 AND VAL=SERIAL)                                   00435980 T  0109:2
          OR (FI=10 AND VAL=RANDOM) OR (FI=13 AND VAL=UPDATE)                       00436000 T  0109:2
 $ SET OMIT = NOT SHAREDISK                                                         00436025 T  0111:3
          THEN GO EXIT;                                                             00436049 T  0115:2
          P(FPB[FPB3],[FPB[FPB3]],FIB[4],[FIB[4]],FIB[13],[FIB[13]],               00436075 T  0115:2
            3) ;                                                                    00436080 T  0117:0
          FPB[FPB3].[43:5]←FI←IF VAL=0 THEN 12 ELSE IF VAL=1 THEN 10                00436085 T  0120:0
 $ SET OMIT = NOT SHAREDISK                                                         00436180 T  0120:1
                              ELSE 13 ;                                             00436184 T  0124:0
 $ SET OMIT = SHAREDISK                                                             00436190 T  0124:0
          IF FIB[4].[27:3]≠3 THEN FIB[4].[27:3]←VAL ;                               00436194 T  0127:1
          FIB[13].[39:5]←FI; P(UPDATE); GO CHK3T ;                                  00436195 T  0127:1
 $ POP OMIT                                                                         00436200 T  0131:3
 $ SET OMIT = NOT SHAREDISK                                                         00436201 T  0135:0

MYUSE1:   IF FIB5.[11:2]=VAL THEN GO EXIT ;                                         00436209 T  0135:0
          IF P([FIB[14]],LOD).[FF]=2 THEN P(MKS,"CHNGNG",TANK,7,                    00436225 T  0135:0
                                            SELECT);                                00436250 T  0135:0
          FIB[5].[11:2]←FI←VAL; TEMP←FIB5.[9:2]; P(1); GO SETUSE;                   00436260 T  0136:3
                                                                                    00436270 T  0140:0
SAVE2:    IF FIB[4].[30:18]=PMET←OCTTODEC THEN GO EXIT ;                            00436330 T  0140:1
          P(FIB[4],[FIB[4]],1) ;                                                    00436350 T  0145:1
          FIB[4].[30:18]←PMET; P(999); GO CHK2 ;                                    00436400 T  0147:0
                                                                                    00436410 T  0150:1
OTHERUSE3: IF FIB5.[9:2]=VAL THEN GO EXIT ;                                         00436520 T  0151:2
          FI←FIB5.[11:2]; FIB[5].[9:2]←TEMP←VAL; P(0) ;                             00436550 T  0154:3
    SETUSE: PMET←P; P(FIB5,[FIB[5]],1) ;                                            00436600 T  0154:3
          IF DISK THEN IF HEADER[4] THEN CHKCLOSREL ELSE CHKOPEN ;                  00436660 T  0156:2
          FIB[5].[13:3]←IF FI=0 THEN 7 ELSE IF TEMP=0 THEN 4 ELSE                   00436670 T  0161:0
                        IF TEMP=1 THEN IF FI=1 THEN 3 ELSE 2 ELSE IF                00436675 T  0162:2
                        FI=1 THEN 1 ELSE 0 ;                                        00436680 T  0168:2
          P(10); IF NOT PMET THEN GO CHK2; GO CHK1 ;                                00436685 T  0173:2
                                                                                    00436690 T  0177:2
MFID4:    P(.MFIDX,0,MFIDX); GO TOIT ;                                              00436710 T  0181:0
                                                                                    00436725 T  0182:2
FID5:     P(.FIDX,0,FIDX); GO TOIT ;                                                00436750 T  0182:2
                                                                                    00436775 T  0183:3
REEL6:    P(.REELX,VAL>999,REELX); GO TOIT ;                                        00436800 T  0183:3
                                                                                    00436825 T  0185:0
DATE7:    P(.DATEX,VAL DIV 1000>99 OR VAL MOD 1000>366,DATEX) ;                     00436850 T  0185:0
          GO TOIT ;                                                                 00436875 T  0186:3
                                                                                    00436900 T  0186:3
CYCLE8:   P(.CYCLEX,VAL>99,CYCLEX); GO TOIT ;                                       00436905 T  0190:0
                                                                                    00436925 T  0190:2
TYPE9:    P(.TYPEX,VAL>63 OR(VAL AND 31)=3 OR(VAL AND 31)>26,TYPEX);                00436950 T  0190:2
                                                                                    00436975 T  0192:1
                                                                                    00437010 T  0192:1
```

```
      TOIT:     IF P=VAL THEN BEGIN P(DEL,DEL); GO EXIT END ;                    00437030 T  0196:2
                INITIALIZE; P(0); CHKMYUSE; CHKOPEN ;                            00437040 T  0198:2
                IF FIB[4].[24:2]≠1 OR FIB[4].[8:4]≠2 THEN CHKCLOSREL             00437070 T  0203:0
                ELSE IF FI=4 OR FI=9 THEN ERM("CLS*,NOT ALTR") ;                 00437090 T  0207:3
                IF P(XCH) OR VAL.[1:5]≠0 THEN GO VALER; SCATTERFPB ;             00437105 T  0215:2
                P(DEL,VAL,XCH,←,MKS,TANK,MFIDX,FIDX,REELX,DATEX,CYCLEX,          00437110 T  0219:0
                  TYPEX,11,INTRINSIC) ;                                          00437125 T  0221:3
                GO EXIT ;                                                        00437130 T  0222:2
                                                                                 00437150 T  0223:0
      AREAS10:  IF NOTDISK OR VAL=FIB[8].[20:5] THEN GO EXIT ;                   00437200 T  0223:0
                P(FIB[8],[FIB[8]],1) ;                                           00437210 T  0225:3
                FIB[8].[20:5]←VAL; P(20); GO CHK3L ;                             00437240 T  0227:0
                                                                                 00437275 T  0230:1
      AREASIZE11: IF NOTDISK OR VAL=FIB[8].[25:23] THEN GO EXIT ;                00437300 T  0230:1
                P(FIB[8],[FIB[8]],1) ;                                           00437310 T  0233:0
                FIB[8].[25:23]←VAL; P(BIG) ;                                     00437340 T  0234:1
          CHK3L:: PMET←P; CHKCLOSREL; P(PMET); GO CHK2 ;                         00437350 T  0237:0
          CHK3T:: PMET←P; CHKOPEN; P(PMET) ;                                     00437360 T  0239:1
          CHK2:: PMET←P; CHKMYUSE; P(PMET) ;                                     00437370 T  0241:3
          CHK1:: IF P≥VAL AND NOT VAL.[1:1] THEN GO EXIT ;                       00437380 T  0244:2
          VALER:: NOTDISK←ABS(VAL) ;                                             00437400 T  0247:2
                OPEN←0; WHILE TEN[OPEN←OPEN+1]≤NOTDISK DO; INITERR ;             00437410 T  0249:0
                STREAM(N←OPEN,V←NOTDISK,T←TYPE,L←VALSIGN←VAL.[1:1],Q←NAM<0        00437440 T  0254:0
                      ,R←(OPEN+VALSIGN)>8,E←1+(TYPE>9),W←VAL,FI) ;                00437450 T  0256:2
                    BEGIN                                                        00437470 T  0260:1
                    DI←DI"2; DS←2LIT":=" ;                                       00437480 T  0260:1
                    Q(SI←LOC W; DS←8CHR; JUMP OUT TO J); L(DS←LIT"-") ;          00437490 T  0261:0
                    R(DS←7LIT"*"; JUMP OUT TO J); SI←LOC V; DS←N DEC ;            00437500 T  0264:0
                J:  SI←LOC T; DS←3LIT",T="; DS←E DEC; DS←2LIT":←" ;              00437510 T  0267:1
                    END OF STREAM ;                                              00437520 T  0269:1
                GO QUIT ;                                                        00437530 T  0269:2
                                                                                 00437550 T  0270:0
      EUNUM12:  IF NOTDISK OR OPEN OR FPB[FPB3].[18:5]=VAL+1 THEN GO EXIT;       00437560 T  0270:0
                P(FPB[FPB3],[FPB[FPB3]],1); % STORE FOR RECOVERY                 00437570 T  0273:0
                FPB[FPB3].[18:5]:=VAL+1; % EU NO.+1                             00437580 T  0275:0
                IF VAL=(-1) THEN VAL:=1; P(19); GO TO CHK2;                      00437590 T  0278:0
                                                                                 00437600 T  0281:0
      DSKSPEED13: IF NOTDISK OR OPEN OR FPB[FPB3].[16:2]=VAL THEN GO EXIT;       00437610 T  0281:0
                P(FPB[FPB3],[FPB[FPB3]],1); % STORE VALUES FOR RECOVERY          00437620 T  0284:1
                FPB[FPB3].[16:2]:=VAL; % 1=FAST,2=SLOW                          00437630 T  0285:2
                P(2); GO CHK2;                                                   00437640 T  0288:0
                                                                                 00437650 T  0288:3
      TIMELIMIT14:                                                               00437700 T  0288:3
       $ SET OMIT = NOT SHAREDISK                                                00437710 T  0288:3
                                                                                 00437760 T  0288:3
      IOSTATUS15: GO EXIT;                                                       00437770 T  0288:3
                                                                                 00437780 T  0289:1
      SENSITIVE14: IF NOTDISK OR FPB[FPB3].[15:1]=VAL THEN GO EXIT;              00437800 T  0289:1
                P(FPB[FPB3],[FPB[FPB3]],1); % STORE FOR RECOVERY                 00437810 T  0292:0
                FPB[FPB3].[15:1]:=VAL; %SENSITIVE=1                              00437820 T  0293:1
                P(1); GO CHK1;                                                   00437830 T  0295:3
            BIG:: @37777777 ;                                                    00440000 T  0296:2
                                                                                 00449999 T  0298:0
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00450000 T  0298:0
      %%%%%   INSERT NEW XN-TYPE ATTRIBUTE CODE ON NEW LINES ABOVE HERE   %%%%% 00450050 T  0298:0
      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 00450100 T  0298:0
                                                                                 00450150 T  0298:0
```

```
ACCESSV0:       P(IF (FI←TYPE AND 31)=10 THEN 1 ELSE IF FI=13 THEN 2
 $ SET OMIT = NOT SHAREDISK
                ELSE 0,RTN);

MYUSEV1:        P(FIB5,[11:2],RTN) ;

SAVEV2:         P(FIB[4],[30:18],DECTOOCT,RTN) ;

OTHERUSEV3:     P(FIB5,[9:2],RTN) ;

MFIDV4:         P(MFIDX,RTN) ;

FIDV5:          P(FIDX,RTN) ;

REELV6:         P(REELX,RTN) ;

DATEV7:         P(DATEX,RTN) ;

CYCLEV8:        P(CYCLEX,RTN) ;

TYPEV9:         P(TYPEX,RTN) ;

AREASV10: IF  FIB[8],[20:28] = 0 AND FIB[4],[8:4] = 4 AND OPEN
THEN P(HEADER[9],[43:5],RTN)
ELSE P(FIB[8],[20:5],RTN);

AREASIZEV11:
IF FIB[8],[20:28] = 0 AND FIB[4],[8:4] = 4 AND OPEN
THEN P(HEADER[8],[25:23],RTN)
ELSE P(FIB[8],[25:23],RTN);
EUNUMV12:       P(FPB[FPB3],[18:5]-1,RTN);
DSKSPEEDV13:  P((IF (TEMP:=FPB[FPB3],[16:2])=1 THEN 1 ELSE
                   IF TEMP=2 THEN 2 ELSE 0),RTN);
TIMELIMITV14:
 $ SET OMIT = NOT SHAREDISK
              P(0); P(RTN);
IOSTATUSV15:
 $ SET OMIT = NOT SHAREDISK
              P(0); P(RTN);
SENSITIVEV14: P(FPB[FPB3],[15:1],RTN);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%   INSERT NEW XVN-TYPE ATTRIBUTE CODE ON NEW LINES ABOVE HERE  %%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


 EXIT:   TANK[NOT 4]←0;  IF INFO THEN P(RTNVAL,RTN) ;
    END OF FILEATTRIBUTES ;
```

```
00450200 T  0298:0
00450250 T  0298:0
00450254 T  0301:3
00450260 T  0301:3
00450300 T  0303:2
00450400 T  0303:2
00450450 T  0304:2
00450500 T  0304:2
00450550 T  0307:1
00450600 T  0307:1
00450625 T  0308:1
00450650 T  0308:1
00450675 T  0308:3
00450700 T  0308:3
00450725 T  0309:1
00450750 T  0309:1
00450775 T  0309:3
00450800 T  0309:3
00450825 T  0310:1
00450850 T  0310:1
00450875 T  0310:3
00450900 T  0310:3
00450925 T  0311:1
00450950 P  0311:1
00450951 C  0314:2
00450952 C  0318:1
00450975 T  0320:0
00451000 P  0320:0
00451001 C  0320:0
00451002 C  0323:1
00451003 C  0327:0
00451025 T  0328:3
00451050 T  0330:2
00451055 T  0333:3
00451075 T  0336:1
00451099 T  0336:1
00451150 T  0336:1
00451175 T  0336:3
00451199 T  0336:3
00451250 T  0336:3
00451300 T  0337:1
00469999 T  0338:2
00470000 T  0338:2
00470050 T  0338:2
00470100 T  0338:2
00470150 T  0338:2
00470200 T  0338:2
00470250 T  0338:2
00470350 T  0341:2
                SIZE= 0342 WORDS
```

```
PROCEDURE ALGOLREAD(TEN, FILX, DKADD, ACT, FI, AEXP,
                ARRY, EOFL, PARL, DKADR, CODE, TANK);
```

```
                %WF          00500000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00096
                %WF          00500100 T  0000:0
```

```
    VALUE     FI, DKADR, TANK, CODE, ACT, AEXP;                          %WF     00500200 T  0000:0
    ARRAY     ARRY[*], TEN[*];                                           %WF     00500300 T  0000:0
    INTEGER ACT, FI, AEXP;                                               %WF     00500400 T  0000:0
    REAL      DKADD, PARL, EOFL, CODE, DKADR;                            %WF     00500500 T  0000:0
    NAME      FILX, TANK;                                                %WF     00500600 T  0000:0
BEGIN REAL RCW=+0,BLKCNTL=5,SELECT=14;%                                          00500700 T  0000:0
      NAME MEM=2;%                                                               00500800 T  0000:0
      ARRAY FPB=3[*];%                                                           00500900 T  0000:0
    REAL      ALGOLREAD=13;                                              %WF     00501000 T  0000:0
    ARRAY     TINK=TANK[*];                                              %WF     00501100 T  0000:0
    INTEGER   RSIZE=FI;                                                  %WF     00501200 T  0000:0
    DEFINE FNUM = FIB[4].[13:11] #;                                              00501250 T  0000:0
      DEFINE IOD=(*TANK)#;%                                                      00501300 T  0000:0
 $ SET OMIT = NOT SHAREDISK                                                      00501309 T  0000:0
      LABEL DC1,DC2;                                                             00501400 T  0000:0
      LABEL DCN1,DCN2,SPIN;                                                      00501410 T  0000:0
      LABEL CR1,MT1,CLOSED,DK1,SP1,PR1,ERR;%                                     00501500 T  0000:0
     SWITCH SW1← CR1,ERR,MT1,CLOSED,DK1,SP1,ERR,ERR,ERR,PR1,DC1,CR1,             00501600 T  0000:0
                 ERR,DCN1;                                                       00501610 T  0000:0
      LABEL CR2,MT2,DK2,SP2,PR2;%                                                00501700 T  0000:0
     SWITCH SW2← CR2,ERR,MT2,ERR,DK2,SP2,ERR,ERR,ERR,PR2,DC2,CR2,                00501800 T  0000:0
                 ERR,DCN2;                                                       00501810 T  0000:0
      LABEL SW,PBIT,DS,FIB7,DSPBIT,PAR,DSRTN,DS19,RA1,EOF,D28;%                  00501900 T  0000:0
      LABEL EMPTY,FULL,SEMPTY,SFULL;                                             00501950 T  0000:0
      REAL UNITYPE,REV,ADDRESS,BLKODE;%                                          00502000 T  0000:0
 $ SET OMIT = NOT SHAREDISK                                                      00502049 T  0000:0
      LABEL DKS,DKR,DKR1,DKU;                                                    00502100 T  0000:0
      SWITCH ASW←DKS,DKR,DKU,CLOSED;                                             00502200 T  0000:0
 $ SET OMIT = NOT(TIMESHARING)                                                   00502250 T  0000:0
SUBROUTINE WAIT; POLISH(TANK, @2000000000, 36, COM, DEL, DEL);                   00502252 T  0000:0
 $ POP OMIT                                                                      00502253 T  0004:0
 $ SET OMIT = TIMESHARING                                                        00502299 T  0004:0
      LABEL RU,RA,RC;%                                                           00502400 T  0004:0
      SWITCH RTYPE←RU,RA,ERR,RC;%                                                00502500 T  0004:0
      LABEL DKSR,DKRR,DKUR;%                                                     00502600 T  0004:0
      SWITCH ASWR←DKSR,DKRR,DKUR;%                                               00502700 T  0004:0
      ARRAY FIB[*],HEADER[*];%                                                   00502800 T  0004:0
 $ SET OMIT = NOT SHAREDISK                                                      00502809 T  0004:0
      INTEGER I;%                                                                00502900 T  0004:0
      REAL SUBROUTINE DISKADDRESS;%                                              00503000 T  0004:0
      BEGIN IF DKADR≥0 THEN%                                                     00503100 T  0004:0
      BEGIN ADDRESS←(DKADR DIV HEADER[0].[30:12])×HEADER[0].[42:6];%             00503200 T  0004:3
            IF (I←ADDRESS DIV HEADER[1]+10)≥30 THEN P(0) ELSE                    00503300 T  0008:2
            IF HEADER[I]=0 THEN P(0) ELSE                                        00503400 T  0011:3
            BEGIN ADDRESS←HEADER[I]+I←ADDRESS MOD HEADER[1];%                    00503500 T  0014:0
                  STREAM(D←[ADDRESS]); BEGIN SI←D; DS←8 DEC END;%                 00503600 T  0017:1
                  P(1);%                                                         00503700 T  0018:3
      END   END ELSE P(0);%                                                      00503800 T  0019:0
            DISKADDRESS←P;%                                                      00503900 T  0019:3
      END DISKADDRESS;%                                                          00504000 T  0020:0
 $ SET OMIT = NOT SHAREDISK                                                      00504009 T  0020:1
            IF TINK=0 THEN                                                %WF    00504200 T  0020:1
            BEGIN FIB ← FILX[NOT 2];                                      %WF    00504300 T  0023:3
                  FILX[NOT 4] ← EOFL;  FILX[NOT 3] ← PARL;                %WF    00504400 T  0026:0
            IF NOT FIB[5].[12:1] THEN P(MKS,"READNG",FILX,7,SELECT) ;             00504450 T  0029:2
                  IF FIB[5].[43:2]≠((ACT<0)+2) THEN                       %WF    00504500 T  0032:2
                    P(MKS, DKADD, (ACT<0)+2, FILX, 1, SELECT);            %WF    00504600 T  0035:0
```

```
                    RSIZE←P(MKS,(ABS(ACT)=3),DKADD,1,FILX,ALGOLREAD);                00504700 T  0038:0
                    IF ARRY≠0 THEN                                          %WF      00504800 T  0040:3
                    BEGIN IF ARRY.[8:10]>P(DUP, AEXP)                       %WF      00504900 T  0041:3
                          THEN P(DEL, AEXP);                                %WF      00505000 T  0043:3
                         IF P(DUP)≥RSIZE THEN P(DEL) ELSE RSIZE ← P;        %WF      00505100 T  0045:0
                         STREAM(P4 ← *FILX, P3 ← RSIZE,                     %WF      00505200 T  0048:2
                              P2 ← P(DUP).[36:6], P1 ← [ARRY[0]]);          %WF      00505300 T  0049:2
                         BEGIN SI ← P4;  DS ← P3 WDS;                       %WF      00505400 T  0051:0
                           P2(DS ← 32 WDS;  DS ← 32 WDS);                   %WF      00505500 T  0051:3
                         END;                                               %WF      00505600 T  0053:0
                    END;                                                    %WF      00505700 T  0053:1
                    IF ABS(ACT)≥2 THEN%                                              00505800 T  0053:1
                      P(MKS, DKADD, 0, FILX, ALGOLREAD);                    %WF      00505900 T  0054:1
                    FILX[NOT 4] ← FILX[NOT 3] ← 0;                          %WF      00506000 T  0056:0
                    P(XIT);                                                 %WF      00506100 T  0059:1
               END;                                                         %WF      00506200 T  0059:2
          FIB←TANK[NOT 2];%                                                          00506300 T  0059:2
SW:       UNITYPE←FIB[4].[8:4]; REV←FIB[5].[44:1]; BLKODE←FIB[5].[46:2];%            00506400 T  0061:1
 $ SET OMIT = TIMESHARING                                                            00506440 T  0065:3
          IF DKADR.[4:1] THEN                                                        00506460 T  0065:3
          BEGIN                                                                      00506465 T  0066:2
 $ SET OMIT = NOT SHAREDISK                                                          00506469 T  0067:0
               DKADR.[3:2]←0;                                                        00506475 T  0067:0
          END;                                                                       00506480 T  0068:3
          IF CODE THEN GO TO SW1[UNITYPE]; GO TO SW2[UNITYPE];%                      00506500 T  0068:3
MT1:%                                                                                00506600 T  0085:2
CR1:%                                                                                00506700 T  0085:2
PR1:      IF IOD.[19:1] THEN% USING CHANNEL NUMBERING SCHEME FOR BITS                00506800 T  0085:2
PBIT:     BEGIN IF IOD.[2:1] THEN%                                                   00506900 T  0086:2
               BEGIN IF FIB[17]=0 THEN FIB[17]←*((IF REV THEN 1 ELSE NOT 0)         00507000 T  0088:0
                                                  INX FLAG(FIB[16]));%              00507100 T  0092:1
                    P((IF BLKODE THEN IOD.[8:10] ELSE FIB[17]),RTN);%               00507200 T  0094:1
               END;%                                                                00507300 T  0097:1
               IF IOD.[25:1] THEN%                                                  00507400 T  0097:1
CLOSED:        BEGIN                                                                00507410 T  0098:1
                    FIB[13].[27:1]←1;                                               00507420 T  0098:3
                    IF (REV←(FPB[FNUM+3] AND 31))≠10 AND REV≠12                     00507430 T  0101:1
                    AND REV≠13 AND REV≠26 THEN FIB[5].[45:1]←0 ELSE                 00507440 T  0104:3
                    FIB[5].[45:1]←P(TANK[NOT 3],DUP)≠0 AND P(XCH)≠15;               00507450 T  0110:2
                    P(TANK,0,11,COM,DEL,DEL) ;                                      00507510 T  0116:1
                    IF NOT FIB[5].[45:1] THEN GO SW ;                               00507515 T  0117:3
                    P(TANK[NOT 3]);  TANK[NOT 3]←TANK[NOT 4]←0 ;                    00507520 T  0119:1
                    P(MKS,9,BLKCNTRL,DEL) ;% TAKE PARITY ACTION LBL BRNCH.          00507525 T  0123:3
                    CODE←1; GO TO DS;                                               00507530 T  0124:3
               END ;                                                                00507535 T  0126:0
               IF IOD.[27:1] THEN%                                                  00507600 T  0126:0
               BEGIN IF UNITYPE=2 THEN%                                             00507700 T  0127:0
                    IF FIB[4].[2:1] THEN%                                           00507800 T  0128:1
                    P(MKS,1,0,(NOT 2)INX TANK,4,SELECT) ELSE%                       00507900 T  0129:3
               BEGIN P(TANK,11,11,COM,DEL,DEL);%                                    00508000 T  0132:2
                    IF MEM[TANK[NOT 1] INX 4].[42:6]=1 THEN%                        00508100 T  0134:2
                    BEGIN UNITYPE←FIB[13].[28:10];%                                 00508200 T  0138:0
                         P(MKS,6,0,(NOT 2)INX TANK,4,SELECT);%                      00508300 T  0140:0
                         FIB[13].[28:10]←UNITYPE+1;%                                00508400 T  0142:1
                         GO TO CLOSED;%                                             00508500 T  0145:1
               END       END;%                                                      00508600 T  0145:3
EOF: IF CODE = 3 THEN P(1,SSN,RTN); CODE ← 2;                                        00508700 T  0145:3
```

see Bisco hardware reference manual page 9-47

presence bit

PARe

```
                END ELSE PAR; TANK[0]←IOD OR MEM;%                                  00508800 T  0148:2
                IF CODE = 3 THEN BEGIN P(0,[TANK[NOT 2]],19,17,COM);                00508900 T  0150:2
                                       P(0,RTN); END;                               00509000 T  0153:3
                P(TANK[NOT(CODE+2)]);                                               00509400 T  0154:1
                TANK[NOT 4]←TANK[NOT 3]←0;                                          00509500 T  0156:0
                P(MKS,9,BLKCNTL);                                                   00509600 T  0159:1
DS:             P(TANK,CODE,11,COM);%                                               00509700 T  0160:0
          END;%                                                                     00509800 T  0161:0
          P(TANK); WAIT; GO TO PBIT;%                                               00509900 T  0161:0
ERR:      CODE←3; GO DS;%                                                           00510000 T  0162:2
DK1:      HEADER←*[FIB[14]]; GO TO ASW[FIB[4],[27:3]];%                             00510100 T  0163:3
DK2:      HEADER←*[FIB[14]]; GO TO ASWR[FIB[4],[27:3]];%                            00510200 T  0168:3
CR2:%                                                                               00510300 T  0173:1
MT2:%                                                                               00510400 T  0173:1
PR2:      GO TO RTYPE[BLKODE];%                                                     00510500 T  0173:1
RU:       TANK[0]←FLAG(FIB[16]); P(FLAG(FIB[19]),TANK,PRL,DEL);%                    00510600 T  0176:1
          BLKODE←FIB[19].[33:15]←FIB[16].[33:15];%                                  00510700 T  0179:1
          FIB[16].[33:15]← CODE←(*((IF REV THEN 2 ELSE NOT 1) INX%                  00510800 T  0182:0
                                   FLAG(FIB[16]))).[18:15];%                        00510900 T  0185:0
          FIB[19].[33:15]←CODE+BLKODE;%                                             00511000 T  0188:1
          FIB[6]←(REV←1&REV[1:47:1])+FIB[6];%                                       00511100 T  0190:3
          FIB[7]←FIB[7]+REV; FIB[17]←0; P(XIT);%                                    00511200 T  0194:1
RA:       IF (FIB[17]←FIB[17]-CODE←FIB[18].[33:15])<CODE THEN GO TO RU;%            00511300 T  0197:3
RA1:      TANK[0]←(IF REV THEN NOT CODE INX 1 ELSE CODE) INX IOD;%                  00511400 T  0202:0
          GO TO FIB7;%                                                              00511500 T  0205:3
RC:       IF (FIB[17]←FIB[17]-CODE←IOD.[8:10]+1)≤1 THEN GO TO RU;%                  00511600 T  0206:1
          IF REV THEN%                                                              00511700 T  0211:0
          BEGIN;STREAM(S←IOD,D←[CODE]);%                                            00511800 T  0211:1
                BEGIN SI←S; SI←SI-8; DS←4 OCT END;%                                 00511900 T  0213:0
                TANK[0]←(NOT(CODE←CODE DIV 8 -1) INX IOD)&CODE[8:38:10];%           00512000 T  0214:0
          END ELSE%                                                                 00512100 T  0218:1
          BEGIN;STREAM(S←(TANK[0]←CODE INX IOD),D←[CODE]);%                         00512200 T  0218:1
                BEGIN SI←S; SI←SI-4; DS←4 OCT END;%                                 00512300 T  0221:0
                TANK[0]←IOD&(CODE DIV 8 -1)[8:38:10];%                              00512400 T  0222:0
          END;%                                                                     00512500 T  0225:0
FIB7:      FIB[7]←1&REV[1:47:1]+FIB[7];%                                            00512600 T  0225:0
D28:      TANK[0]←IOD&(NOT P(DUP))[2:28:1];%                                        00512700 T  0228:0
SP2:      P(XIT);%                                                                  00512800 T  0230:1
DKU:%                                                                               00512900 T  0230:2
DKS:      IF DKADR=0 THEN % NORMAL SEQUENTIAL READ--NO ADDRESS SPECIFIED.%          00513000 T  0230:2
DS19:           IF IOD.[19:1] THEN%                                                 00513100 T  0231:1
DSPBIT:         IF IOD.[2:1] THEN%                                                  00513200 T  0232:3
                IF FIB[7]>HEADER[7] THEN%                                           00513300 T  0234:1
                BEGIN TANK[0]←IOD&0[2:2:1]&1[27:47:1];%                             00513400 T  0236:0
                      GO TO EOF;%                                                   00513500 T  0239:2
                END ELSE%                                                           00513600 T  0240:0
DSRTN:          BEGIN IF FIB[17]=0 THEN FIB[17]←FIB[18].[18:15];%                   00513700 T  0240:0
                      P(FIB[18].[33:15],RTN);%                                      00513800 T  0244:1
                END ELSE%                                                           00513900 T  0245:2
                IF IOD.[25:1] THEN GO TO CLOSED ELSE%                               00514000 T  0245:2
                IF IOD.[27:1] THEN GO TO EOF ELSE GO TO PAR ELSE%                   00514100 T  0247:0
                BEGIN P(TANK); WAIT; GO TO DSPBIT END;%                             00514200 T  0249:3
%         READ OR SEEK ON A SERIAL FILE WITH ADDRESS SPECIFIED.%                    00514300 T  0252:2
          P(MKS,ABS(DKADR)-1,1,TANK,1,SELECT);%                                     00514400 T  0252:2
          IF DKADR<0 THEN GO TO DSRTN; GO TO DS19;%                                 00514500 T  0254:3
DKSR:     IF DKADR>0 THEN GO TO D28 ELSE IF DKADR=0 THEN%                           00514600 T  0256:2
          BEGIN IF (FIB[17]←FIB[17]-CODE←FIB[18].[33:15])≥CODE THEN GO RA1;         00514700 T  0258:2
```

```
                FIB[6]←(REV←1&REV[1:47:1])+FIB[6];%                                    00514800 T  0263:1
                FIB[17]←0;%                                                            00514900 T  0266:3
                DKADR←FIB[7]%                                                          00515000 T  0268:0
                      +FIB[13].[10:9]×HEADER[0].[30:12]×REV;%                          00515100 T  0268:1
                FIB[7]←FIB[7]+REV;%                                                    00515200 T  0272:0
                IF DISKADDRESS THEN%                                                   00515300 T  0274:0
                BEGIN P(TANK[0]←FLAG(FIB[16]),ADDRESS,XCH,←);%                         00515400 T  0275:0
                      P(FLAG(FIB[19]),TANK,PRL,DEL);%                                  00515500 T  0277:2
                END ELSE%                                                              00515600 T  0279:1
                BEGIN TANK[0]←FLAG(FIB[16])&2[27:46:2]&0[2:47:1];%                     00515700 T  0279:1
                      P(FIB[13].[10:9],TANK,13,11,COM,DEL,DEL,DEL);%                   00515800 T  0283:0
                END;%                                                                  00515900 T  0285:3
                      BLKODE←FIB[19].[33:15]-FIB[16].[33:15];%                         00516000 T  0285:3
                      FIB[16].[33:15]←CODE←MEM[P(DUP) INX NOT 1].[18:15];%             00516100 T  0288:2
                      FIB[19].[33:15]←CODE+BLKODE;%                                    00516200 T  0293:1
          END;%                                                                        00516300 T  0295:3
DKRR: P(XIT);%                                                                         00516400 T  0295:3
DKR:                                                                                   00516410 T  0296:0
 $ SET OMIT = NOT SHAREDISK                                                            00516419 T  0296:0
DKR1: IF DKADR GEQ 0 THEN                                                              00516500 T  0296:0
          BEGIN IF DKADR=0 THEN DKADR←FIB[7] ELSE FIB[7]←DKADR←DKADR-1;%               00516600 T  0296:3
                IF HEADER[7]≥DKADR THEN%                                               00516700 T  0302:1
                IF DISKADDRESS THEN%                                                   00516800 T  0303:1
                BEGIN CODE←FIB[16].[33:15]; UNITYPE←FIB[13].[10:9];%                   00516900 T  0305:0
                      FOR I←0 STEP 1 UNTIL UNITYPE DO%                                 00517000 T  0308:2
 $ SET OMIT = SHAREDISK                                                                00517099 T  0310:0
                      BEGIN IF NOT IOD.[19:1] THEN BEGIN P(TANK); WAIT END;            00517100 T  0310:0
 $ POP OMIT                                                                            00517101 T  0313:0
 $ SET OMIT = NOT SHAREDISK                                                            00517109 T  0313:0
                           IF IOD.[27:1] THEN GO TO EMPTY;%                            00517200 T  0313:0
 $ SET OMIT = SHAREDISK                                                                00517299 T  0314:3
                           IF (MEM[CODE] EQV ADDRESS)=NOT 0 THEN GO FULL;              00517300 T  0314:3
 $ POP OMIT                                                                            00517301 T  0317:2
 $ SET OMIT = NOT SHAREDISK                                                            00517309 T  0317:2
                           TANK[0]←IOD&1[27:47:1];%                                    00517400 T  0317:2
                           P(UNITYPE,TANK,13,11,COM,DEL,DEL,DEL);%                     00517500 T  0319:2
                           FIB[16].[33:15]←CODE←MEM[CODE-2].[18:15];%                  00517600 T  0321:2
                           FIB[19].[33:15]←CODE+1;%                                    00517700 T  0326:0
                      END;  GO TO ERR;%                                                00517800 T  0328:2
EMPTY:                                                                                 00517900 T  0331:1
 $ SET OMIT = NOT SHAREDISK                                                            00517909 T  0331:1
                FIB[13].[10:9]←1;                                                      00517980 T  0331:1
                P(TANK[0]←FLAG(FIB[16]),ADDRESS,XCH,←);%                               00518000 T  0333:3
                P(FLAG(FIB[19]),TANK,PRL,DEL);%                                        00518100 T  0335:3
                FIB[13].[10:9]←UNITYPE; P(TANK);                                       00518200 T  0337:2
 $ SET OMIT = NOT SHAREDISK                                                            00518299 T  0340:1
                WAIT;                                                                  00518350 T  0340:1
FULL:           IF NOT IOD.[2:1] OR FIB[5].[1:1] THEN                                  00518400 T  0341:0
                BEGIN                                                                  00518500 T  0343:2
 $ SET OMIT = NOT SHAREDISK                                                            00518509 T  0344:0
                      CODE←1; GO TO PAR;                                               00518550 T  0344:0
                END;                                                                   00518560 T  0345:1
 $ SET OMIT = NOT SHAREDISK                                                            00518569 T  0345:1
                IF BLKODE=0 THEN SFULL:P(FIB[18].[33:15],RTN);                         00518600 T  0345:1
                TANK[0]←IOD&((I←DKADR MOD HEADER[0].[30:12])%                          00518700 T  0347:3
                      ×(I←FIB[18].[33:15])+FIB[19].[33:15])[33:33:15];%                00518800 T  0350:1
                P(I,RTN);%                                                             00518900 T  0354:0
```

```
                END;%                                                                                                  00519000 T  0354:2
                IF NOT FIB[5].[1:1] THEN                                                                               00519100 T  0354:2
                BEGIN                                                                                                  00519110 T  0355:3
$ SET OMIT = NOT SHAREDISK                                                                                             00519119 T  0356:1
                     GO TO EOF;                                                                                        00519130 T  0356:1
                END;                                                                                                   00519140 T  0356:3
$ SET OMIT = NOT SHAREDISK                                                                                             00519149 T  0356:3
                CODE←1; GO TO PAR;                                                                                     00519160 T  0356:3
           END;%                                                                                                       00519200 T  0358:0
           DKADR←ABS(DKADR)-1;%                                                                                        00519300 T  0358:0
           IF HEADER[7]<DKADR THEN GO TO SFULL;%                                                                       00519400 T  0359:2
           IF NOT DISKADDRESS THEN GO TO SFULL;%                                                                       00519500 T  0361:0
           CODE←FIB[16].[33:15]; UNITYPE←FIB[13].[10:9]-1;%                                                            00519600 T  0362:2
           FOR I←0 STEP 1 UNTIL UNITYPE DO%                                                                            00519700 T  0366:0
$ SET OMIT = SHAREDISK                                                                                                 00519799 T  0367:0
           BEGIN IF NOT TANK[I].[19:1] THEN BEGIN P([TANK[I]]); WAIT END;%                                             00519800 T  0367:0
$ POP OMIT                                                                                                             00519801 T  0371:0
$ SET OMIT = NOT SHAREDISK                                                                                             00519809 T  0371:0
                IF TANK[I].[27:1] THEN GO TO SEMPTY;%                                                                  00519900 T  0371:0
$ SET OMIT = SHAREDISK                                                                                                 00519999 T  0373:1
                IF (MEM[CODE] EQV ADDRESS)=NOT 0 THEN GO TO SFULL;                                                     00520000 T  0373:1
$ POP OMIT                                                                                                             00520001 T  0376:0
$ SET OMIT = NOT SHAREDISK                                                                                             00520009 T  0376:0
                CODE←MEM[CODE-2].[18:15];%                                                                             00520100 T  0376:0
           END;%                                                                                                       00520200 T  0378:3
$ SET OMIT = NOT SHAREDISK                                                                                             00520209 T  0381:0
$ SET OMIT = SHAREDISK                                                                                                 00520299 T  0381:0
           P(TANK[0]←FLAG(FIB[16]),ADDRESS,XCH,←);%                                                                    00520300 T  0381:0
 $ POP OMIT                                                                                                            00520301 T  0383:0
           P(FLAG(FIB[19]),TANK,PRL,DEL);%                                                                             00520400 T  0383:0
           FIB[16].[33:15]←CODE←MEM[CODE-2].[18:15];%                                                                  00520500 T  0384:3
           FIB[19].[33:15]←CODE+1;%                                                                                    00520600 T  0389:1
           GO TO SFULL;%                                                                                               00520700 T  0391:3
SEMPTY:                                                                                                                00520710 T  0392:1
 $ SET OMIT = NOT SHAREDISK                                                                                            00520719 T  0392:1
 $ SET OMIT = SHAREDISK                                                                                                00520799 T  0392:1
                P(TANK[I]←FLAG(FIB[16])&CODE[33:33:15],ADDRESS,XCH,←);                                                 00520800 T  0392:1
 $ POP OMIT                                                                                                            00520801 T  0395:2
                FIB[13].[10:9]←1;%                                                                                     00520900 T  0395:2
                P(FLAG(FIB[19])&(CODE+1)[33:33:15],[TANK[I]],PRL,DEL);%                                                00521000 T  0398:0
                FIB[13].[10:9]←UNITYPE+1;%                                                                             00521100 T  0401:1
                GO TO SFULL;%                                                                                          00521200 T  0404:1
SP1:: STREAM(D←IOD); BEGIN DS←7 LIT "ACCEPT←" END;%                                                                    00521300 T  0404:3
       IF FPB[FNUM+3].[42:6]=43 THEN GO EOF; %DUMMY                                                                    00521310 C  0407:2
      P((NOT 1) INX IOD,16,COM,DEL); GO TO SFULL;%                                                                     00521400 T  0410:3
%                                                                                                                      00521500 T  0415:0
DKUR: FIB[5].[43:2]←0;%                                                                                                00521600 T  0415:0
%                                                                                                                      00521700 T  0417:2
                P(XIT);                                                                                                00521800 T  0417:2
 $ SET OMIT = NOT(TIMESHARING)                                                                                         00521810 T  0417:3
DC1:: P(FIB[18].[33:15] & DKADR[1:47:1] &                                                                              00521813 C  0417:3
        (DKADR.[2:1] AND (DKADR.[FF]=0) AND (TANK[NOT 3]≠0))[32:47:1],                                                 00521814 C  0420:0
        IOD,1,(-13),COM);                                                                                              00521815 P  0425:0
         I:=POLISH;                                                                                                    00521816 C  0426:2
         ADDRESS:=TANK[NOT(4-(I=2))];                                                                                  00521817 P  0427:0
         TANK[NOT 4]:=TANK[NOT 3]:=0;                                                                                  00521818 C  0429:3
         IF I THEN P(FIB[18].[33:15],RTN);                                                                             00521819 C  0433:0
```

```
        IF ADDRESS NEQ 0 THEN                                          00521820 P  0435:0
           P(ADDRESS,MKS,9,BLKCNTRL);                                  00521821 C  0435:3
        ADDRESS:=(I=0)+1;                                              00521822 C  0437:1
        P(TANK,ADDRESS,11,COM);                                        00521823 C  0439:0
 DC2::  P(XIT);                                                        00521824 C  0440:0
DCN1: DCN2: SPIN:                                                      00521825 T  0440:1
 $ SET OMIT = TIMESHARING                                              00521830 T  0441:0
END ALGOLREAD;                                                         00524700 T  0441:0
                                                                                SIZE= 0442 WORDS


PROCEDURE INPUTINT(TEN,FILX,DKADR,ACT,FI,FRMT,LISX,EOFL,PARL);%        00600000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00111
          COMMENT ESPOL VERSION OF ALGOL READ INTRINSIC%               00600100 T  0000:0
          BY L.R. GUCK  12/1/64%                                       00600200 T  0000:0
    VALUE FI,%                                                         00600300 T  0000:0
          ACT;%                                                        00600400 T  0000:0
    NAME  FILX,%                                                       00600500 T  0000:0
          LISX;%                                                       00600600 T  0000:0
    ARRAY TEN[*],%                                                     00600700 T  0000:0
          FRMT[*];%                                                    00600800 T  0000:0
    INTEGER ACT,%                                                      00600900 T  0000:0
          FI;%                                                         00601000 T  0000:0
    REAL  DKADR;%                                                      00601100 T  0000:0
    REAL  EOFL,%                                                       00601200 T  0000:0
          PARL;%                                                       00601300 T  0000:0
      BEGIN COMMENT LOCAL VARIABLES;%                                  00601400 T  0000:0
REAL      JUNK2=9,%                                                    00601500 T  0000:0
          ALGOLREAD=13,%                                               00601600 T  0000:0
          SELECT=14,%                                                  00601700 T  0000:0
          JUNK1 = 17,%                                                 00601800 T  0000:0
          LSTRN=19;%                                                   00601900 T  0000:0
REAL      BLKCNTL = 5;%                                                00602000 T  0000:0
REAL      SAVEBUFF=EOFL, CODE1=PARL ;                                  00602050 T  0000:0
INTEGER   AEXP=FRMT;%                                                  00602100 T  0000:0
ARRAY     ARRY=LISX[*];%                                               00602200 T  0000:0
ARRAY REALROW=TEN=1[*];%                                               00602300 T  0000:0
REAL      F = +0;%                                                     00602400 T  0000:0
REAL      TLSTRN=F+1;%                                                 00602500 T  0000:0
REAL      BUFF=TLSTRN+1;%                                              00602600 T  0000:0
INTEGER   BSIZE=BUFF+1;%                                               00602700 T  0000:0
ARRAY     FIB=BSIZE+1[*];%                                             00602800 T  0000:0
REAL      ADDRS=FIB+1;%                                                00602900 T  0000:0
REAL      SGN=ADDRS+1;%                                                00603000 T  0000:0
REAL      WT=SGN+1;%                                                   00603100 T  0000:0
REAL      W1=WT+1;%                                                    00603200 T  0000:0
REAL      CCR = W1, DIVR = W1;%                                        00603300 T  0000:0
REAL      W2=W1+1;%                                                    00603400 T  0000:0
REAL      TYP = W2;%                                                   00603500 T  0000:0
REAL      D=W2+1;%                                                     00603600 T  0000:0
REAL      ESIG= D;%                                                    00603700 T  0000:0
REAL      D1=D+1;%                                                     00603800 T  0000:0
REAL      D2=D1+1;%                                                    00603900 T  0000:0
REAL      W=D2+1;%                                                     00604000 T  0000:0
REAL      SKIP=W+1;%                                                   00604100 T  0000:0
```

```
REAL      CHR=SKIP+1;%                                                    00604200 T  0000:0
REAL      FAW=CHR+1;%                                                     00604300 T  0000:0
REAL      CODE=FAW+1;%                                                    00604400 T  0000:0
INTEGER   CSIZE=CODE+1;%                                                  00604500 T  0000:0
INTEGER   SCFTR = CSIZE+1;%                                               00604600 T  0000:0
REAL      FLG = SCFTR +1;%                                                00604700 T  0000:0
REAL      UDECLR=FLG+1;                                                   00604710 T  0000:0
LABEL     GA,GAC,GRTY,GTB,GTC,GTD,NUMXIT,%                                00604800 T  0000:0
          FREFLD,STRT,NMRCL,HERE,NOSIG,LPTWO,NOTNUM2,L1,L1P2,%            00604900 T  0000:0
          NFRAC1,ATS,HR1,NSG,L2P1,FINXP,NCA1,NMINUS,NOTAT,%               00605000 T  0000:0
          NCA2,NNMB,NOTNUM,RNOTNUM,INSERT,NAST,QRT,QRTN,%                 00605100 T  0000:0
          EQUT,EAT1,EATUP,NQUOT,GETO,GTRT7,ASTRX,CHKOCT,%                 00605200 T  0000:0
          GETCOMA,GETC1,MAXI,                                             00605300 T  0000:0
          START,CT,CTA,CTB,CTC,%                                          00605400 T  0000:0
          ASLST,BS,BR,AEXPL,ISA,ISB,ERROR,%                               00605500 T  0000:0
          FMOUTA,FMOUT,S1,S,LFPAR,RTPAR,SCALE,STRNG,SLASH,%               00605600 T  0000:0
          PHRAS,INLOOP,FLDW,JMP,%                                         00605700 T  0000:0
          LOGI, FLAGBIT,                                                  00605800 T  0000:0
          DTYPE,OTYPE,ALFA,XTYPE,%                                        00605900 T  0000:0
          RTYPE,RBLF,RFA,RIPART,RDONA,RDONE,RFC,REXP,%                    00606000 T  0000:0
          RIPRTN,RFPRTN,RFPART,GETNUM,GRTN,%                              00606100 T  0000:0
          ITYPE,FIN,FMOUTM1,S2,                                           00606200 T  0000:0
          FOUT,FTYPE,FA,ETYPE,%                                           00606300 T  0000:0
          COMA,COMM,COMB,COMC,RERRA;%                                     00606400 T  0000:0
          COMMENT LABELS ARE LISTED IN SAME ORDER THEY APPEAR;%           00606500 T  0000:0
DEFINE    P = POLISH#,%                                                   00606600 T  0000:0
          TEN8 = @1045753604000000#;%                                     00606700 T  0000:0
SUBROUTINE CKPB; COMMENT CHECK FOR PRESENCE BIT;%                         00606800 T  0000:0
     BEGIN%                                                               00606900 T  0001:0
          IF FILX.[18:15] ≤ 1 THEN%                                       00607000 T  0001:0
               BEGIN IF NOT FILX.[18:15] THEN%                            00607100 T  0002:1
                    BEGIN;STREAM(A←[REALROW[0]]:B←0);%                    00607200 T  0003:3
                         BEGIN SI←A; DI←A; SI←SI-16;                      00607300 T  0005:3
                              SKIP 2 SB;%                                 00607400 T  0006:2
                              IF SB THEN TALLY ← 1;%                      00607500 T  0006:3
                              A ← TALLY;%                                 00607600 T  0007:2
                         END;%                                            00607700 T  0007:3
                    IF NOT P THEN%                                        00607800 T  0008:0
                         BEGIN P(FILX,14,COM,DEL);%                       00607900 T  0008:1
                              FILX.[18:15] ← 1;%                          00608000 T  0009:3
                         END;%                                            00608100 T  0011:0
                    END;%                                                 00608200 T  0011:0
               BSIZE ← REALROW.[8:10];%                                   00608300 T  0011:0
               END ELSE%                                                  00608400 T  0012:2
          BSIZE←POLISH(MKS,DKADR,1,FILX,ALGOLREAD);%                      00608500 T  0012:2
          BUFF←(*FILX)&BSIZE[8:38:10] ;                                   00608600 T  0014:3
     END;%                                                                00608700 T  0016:3
SUBROUTINE READS;%                                                        00608800 T  0017:0
          COMMENT RELEASE BUFFER;%                                        00608900 T  0017:0
     BEGIN%                                                               00609000 T  0017:0
          P(XCH);   COMMENT FLAG TO TOP OF STACK;%                        00609100 T  0017:0
               IF ACT=2 THEN COMMENT READ RELEASE;%                       00609200 T  0017:1
                  POLISH(MKS,DKADR,0,FILX,ALGOLREAD);%                    00609300 T  0018:0
               IF P THEN%                                                 00609400 T  0019:3
                    BEGIN LSTRN ← TLSTRN;%                                00609500 T  0019:3
                         IF FILX.[18:15]>1 THEN                           00609600 T  0021:0
                            FILX[NOT 4]←FILX[NOT 3]←0 ELSE                00609700 T  0022:1
```

```
                    IF FILX.[18:15] = 1 THEN%                                   00609800 T  0026:0
                        P(FILX,14,COM);%                                        00609900 T  0027:3
                    P(XIT);%                                                    00610000 T  0029:0
                END;%                                                           00610100 T  0029:1
            CKPB;%                                                              00610200 T  0029:1
        IF SGN.[45:1] THEN %%% UTYP=SGN.[45:1]; SEE U-PHRASE DECLR              00610210 T  0030:0
            BEGIN CSIZE←8×BSIZE;  BUFF←P(0,[BUFF],0,INX) END ;                  00610220 T  0030:3
        END READS;%                                                             00610300 T  0034:0
        COMMENT SUBROUTINE  USED BY FREE FIELD;%                                00610400 T  0034:1
SUBROUTINE GNCR;%                                                               00610500 T  0034:1
    BEGIN           COMMENT THIS SUB-ROUTINE GETS CHARACTERS FOR%               00610600 T  0035:0
                            FREE FIELD, INCLUDING READING OF RECORDS            00610700 T  0035:0
                            WHEN REQUIRED;%                                     00610800 T  0035:0
GA:     IF WT > 0  THEN GO TO GAC;%                                             00610900 T  0035:0
                COMMENT BUFFER IS EMPTY-FILL IT;%                               00611000 T  0036:1
        P(0);%                                                                  00611100 T  0036:1
        READS;%                                                                 00611200 T  0036:2
        WT ← BSIZE × 8; COMMENT WT = # OF CHARACTERS IN BUFFER;%                 00611400 T  0038:0
        BUFF ← P(0,0,[BUFF],CCX);%                                              00611500 T  0039:1
            COMMENT GET CHR FROM BUFFER;%                                       00611600 T  0040:3
GAC:    STREAM(%                                                                00611700 T  0040:3
            P5 ← 0,%                                                            00611800 T  0041:0
            P4 ← BUFF,%                                                         00611900 T  0041:1
            P3 ← IF WT < 63 THEN WT ELSE 63:%                                   00612000 T  0041:2
            P1 ← TYP);%                                                         00612100 T  0044:0
        BEGIN%                                                                  00612200 T  0044:2
            SI ← P4;%                                                           00612300 T  0044:2
            CI ← CI + P1;%                                                      00612400 T  0044:3
            GO TO FGNC;     COMMENT DEBLANK-THEN GET NCR;%                      00612500 T  0045:1
            GO TO GNCHC;    COMMENT GET NCR;%                                   00612600 T  0045:2
        FGNC:   P3(IF SC ≠ " " THEN JUMP OUT TO GNCHC;%                         00612700 T  0045:3
                   SI←SI+1;%                                                    00612800 T  0047:1
                   TALLY ← TALLY +1);%                                          00612900 T  0047:2
            COMMENT RETURN A -1 IF ALL WERE BLANK;%                             00613000 T  0048:0
            DI ← LOC P5;%                                                       00613100 T  0048:0
            DS ← 8 LIT "+0000001";%                                             00613200 T  0048:1
            GO TO CRTN;%                                                        00613300 T  0049:2
        GNCHC:  TALLY ← TALLY + 1;%                                             00613400 T  0049:3
            DI ← LOC P4;%                                                       00613500 T  0050:0
            DI ← DI - 1;%                                                       00613600 T  0050:1
            DS ← CHR;%                                                          00613700 T  0050:2
        CRTN:   P3 ← TALLY;%                                                    00613800 T  0050:3
            P4 ← SI;%                                                           00613900 T  0051:0
        END;%                                                                   00614000 T  0051:1
        P(WT,XCH,SUB,,WT,STD); COMMENT WT ← WT-TALLY;%                          00614100 T  0051:2
        BUFF ← P;%                                                              00614200 T  0052:3
        IF P(DUP) < 0 THEN BEGIN P(DEL); GO TO GA END;%                         00614300 T  0053:1
        P(XCH);         COMMENT RETURN LITERAL TO TOP;%                         00614400 T  0055:1
    END GNCR;%                                                                  00614500 T  0055:2
REAL SUBROUTINE LISTELEMENT ;                                                   00614505 T  0055:3
    BEGIN                                                                       00614510 T  0056:0
    IF LSTRN<0 THEN GO TO ERROR ;                                               00614515 T  0056:0
    P(ADDRS,,ADDRS,ISN) ;                                                       00614520 T  0057:1
    ADDRS←LISX ;                                                                00614525 T  0058:0
    LISTELEMENT←P ;                                                             00614530 T  0058:3
    END OF LISTELEMENT ;                                                        00614535 T  0059:0
                                                                                00614600 T  0059:1
```

```
% * * *   D E C L A R A T I O N S   F O R   U - P H R A S E   * * *                00614605 T  0059:1
% NOTE THAT CST REFERS TO THE CONSTRUCT BEGIN SCANNED. THE CST IS                 00614610 T  0059:1
% EITHER A NUMBER, AN UNQUOTED STRING, OR A QUOTED STRING.                        00614615 T  0059:1
LABEL     UERR      ,          %%% BRNCHTO FOR DATA ERROR.                       00614620 T  0059:1
          UTYPE     ,          %%% BRNCHTO FOR U-PHRASE EDITING.                 00614625 T  0059:1
          UENDNUM   ,          %%% BRNCHTO FOR END OF NUMBER SCAN.               00614630 T  0059:1
          UL1       ,          %%% BRNCHTO FOR EFFICIENCY IN UCH.                00614635 T  0059:1
          UL2       ,          %%% BRNCHTO FOR EFFICIENCY IN UCHECKT             00614640 T  0059:1
          UL3       ,          %%% BRNCHTO FOR EFFICIENCY IN STRINGS             00614645 T  0059:1
          UL4       ,          %%% BRNCHTO FOR STRING-HANDLING LOOP.             00614650 T  0059:1
          UL5       ,          %%% BRNCHTO FOR UCHECKIT(@END-OF-CST)            00614655 T  0059:1
          UL6       ,          %%% BRNCHTO FOR NO STRING STORE.                  00614660 T  0059:1
          FMTERR    ;          %%% BRNCHTO FOR ILLEGAL FORMAT.                   00614665 T  0059:1
DEFINE    UEXP      =    WT #, %%% IS VALUE OF EXPNT(OF CST AS NUM),             00614670 T  0059:1
                               %%% OR IT IS THE SHIP-IT-ANYHOW TOGGL            00614675 T  0059:1
                               %%% FOR THE 1-ST CHR OF QUOTED STRING            00614680 T  0059:1
                               %%% OR IS USED AS TEMPORY BY UGETSGN.            00614685 T  0059:1
          UBUILD    =    W1 #, %%% IS > 0 IF HAVE NOT YET IDENTIFIED            00614690 T  0059:1
                               %%% CST & SO MUST BUILD UH INTO UBUFF            00614695 T  0059:1
                               %%% IS > 0 IF SHALL BRANCH TO ENDNUM             00614700 T  0059:1
                               %%% IF HAVE HIT END OF FIELD WIDTH,              00614705 T  0059:1
                               %%% IS ALSO USED AS TEMPORARY COUNTER            00614710 T  0059:1
                               %%% OF OCT DIGITS IN OCTAL NUM PART.             00614715 T  0059:1
          UVAL      =    W2 #, %%% IS VALUE OF CST IF CST IS A NUM,            00614720 T  0059:1
                               %%% AND IS USED AS TEMPORARY STORAGE.           00614725 T  0059:1
          UNUM      =    D1 #, %%% IS TRUE IFF CST IS NUMBER.                  00614730 T  0059:1
          UADDRS    =UDECLR #, %%% STORES LIST ADDRESS REFERRED TO             00614735 T  0059:1
                               %%% BY THE ULIST DEFINE (BELOW).                00614740 T  0059:1
          UH        =    D2 #, %%% IS CURRENT CHARACTER OF CST.                00614745 T  0059:1
          UBUFF     =   FLG #, %%% IS SIX OR LESS CHARACTERS OF CST,           00614750 T  0059:1
                               %%% ALSO IS USED AS TEMPORARY BY               00614755 T  0059:1
                               %%% SUBROUTINE UCHECKIT.                        00614760 T  0059:1
          UCHCNT    =  SKIP #, %%% IS CHARACTER COUNTER FOR CST.              00614765 T  0059:1
          USCHCNT   =   FAW #, %%% IS STRING CHR, COUNTER FOR UBUFF.          00614770 T  0059:1
          UDEC      =     D #, %%% IS VALUE OF DECIMAL PART OF CST            00614775 T  0059:1
                               %%% (IF CST IS NUMBER), AND IS ALSO            00614780 T  0059:1
                               %%% USED AS TEMPORARY STORAGE.                  00614785 T  0059:1
     USGN      = SGN.[47:1] #, %%% IS TRUE IFF CST(AS NUM) IS NEGTIV           00614790 T  0059:1
     UEXPSGN   = SGN.[46:1] #, %%% IS TRUE IFF UEXP IS NEGATIVE.              00614795 T  0059:1
     UTYP      = SGN.[45:1] #, %%% IS TRUE IFF IN OR JUST USED UPHRS          00614800 T  0059:1
     UNLOCATED = SGN.[44:1] #, %%% IS TRUE IFF HAVE NOT LOCATED CST.          00614805 T  0059:1
     UQSTRNG   = SGN.[43:1] #, %%% IS TRUE IFF CST IS QUOTED STRING.          00614810 T  0059:1
     ULIST     = SGN.[42:1] #, %%% IS TRUE IFF UCH HAS TRIED TO GET           00614815 T  0059:1
                               %%% AND/OR HAS GOTTEN A NEW LIST ADRS           00614820 T  0059:1
     UD        = SGN.[36:6] #, %%% STORES ORIGINAL VALUE OF D.                00614825 T  0059:1
     UW        = SGN.[30:6] #, %%% STORES ORIGINAL VALUE OF W.                00614830 T  0059:1
     UFREEFIELD= SGN.[29:1] #, %%% IS TRUE IFF IN SPECL // FREEFIELD          00614835 T  0059:1
          UGETRECORD          = BEGIN P(CHR←0); READS END #,                  00614840 T  0059:1
          UGOOFED(UGOOFED1) = BEGIN UEXP←UGOOFED1; GO TO UERR END #,          00614845 T  0059:1
          UEOW              = W≤UCHCNT #,                                     00614850 T  0059:1
          UALLDONE          = GO TO BR # ;                                    00614855 T  0059:1
SUBROUTINE UGNCH;    %%% UGNCH GETS THE NEXT CHARACTER FROM THE               00614860 T  0059:1
     BEGIN           %%% BUFFER, GETS A NEW BUFFER WHEN NECCESSARY,           00614865 T  0060:0
     IF CHR≥CSIZE    %%% ADJUSTS BUFF, AND BUMPS CHR BY 1.                    00614870 T  0060:0
     THEN BEGIN      %%% WE NEED A NEW BUFFER, CALL READS TO GET ONE.         00614875 T  0060:1
          UGETRECORD ;    %%% GET A NEW RECORD.                              00614880 T  0061:1
          IF UNLOCATED    %%% IF HAVE NOT YET LOCATED CST, THEN              00614885 T  0063:0
```

```
          THEN UCHCNT←0 ; %%% ZERO CHAR. CNTER. FOR NEW BUFFER SCAN.          00614890 T  0063:0
          END ;                                                               00614895 T  0065:0
     CHR←CHR+1 ; %%% INCREMENT CHARACTER COUNTER.                             00614900 T  0065:0
     STREAM(P1←0,P2←BUFF:P0←0) ; %%% NOW GET CHARACTER AND BUMP BUFF          00614905 T  0066:1
          BEGIN DI←LOC P2; DI←DI-1; SI←P2; DS←CHR; P2←SI END ;                00614910 T  0067:3
     BUFF←P ; %%% BUFF IS SET TO NEW ABS ADDRS (BUFF+1).                      00614915 T  0069:1
     UH←P ; %%% CHARACTER IS STORED IN UH.                                    00614920 T  0069:3
     END OF UGNCH ;                                                           00614925 T  0070:1
SUBROUTINE UBUFFIT ; %%% STUFFS UH INTO UBUFF FROM THE RIGHT                  00614930 T  0070:2
     BEGIN           %%% AND BUMPS THE STRING CHARACTER COUNTER.              00614935 T  0071:0
     UBUFF←UH & UBUFF[12:18:30] ;                                             00614940 T  0071:0
     USCHCNT←USCHCNT+1 ;                                                      00614945 T  0072:3
     END OF UBUFFIT ;                                                         00614950 T  0074:0
SUBROUTINE UCH ; %%% UCH IS THE CONTROL FOR UGNCH. UCH WATCHES OUT            00614955 T  0074:1
     BEGIN       %%% FOR END-OF-W AND, IF NOT IN "STRING", SCANS              00614960 T  0075:0
                 %%% OVER IN-LINE COMMENTS (/...=) AND STORES NON-            00614965 T  0075:0
                 %%% BLANK PORTION OF CST IN UBUFF FOR FUTURE USE             00614970 T  0075:0
                 %%% IF CST TURNS OUT NOT TO BE A NUMBER. UCH ALSO            00614975 T  0075:0
                 %%% HANDLES END-OF-RECORD SITUATIONS (/... OR ←).            00614980 T  0075:0
     IF UEOW THEN     %%% IF HAVE HIT END-OF-W (W=FIELD WIDTH), THEN          00614985 T  0075:0
          BEGIN                                                               00614990 T  0075:3
          IF UBUILD≠0 %%% IF WE ARE NOT IN THE STRING SECTION, WE             00614995 T  0076:1
          THEN BEGIN P(DEL); GO UENDNUM END %%% BRNCHTO END-OF-NUM.           00615000 T  0076:3
          END                                                                 00615005 T  0078:1
     ELSE %%% ELSE WE ARE STILL INSIDE FIELD-WIDTH W...                       00615010 T  0078:1
          BEGIN %%% STILL SOME FIELD WIDTH LEFT.                              00615015 T  0078:1
          UGNCH; UCHCNT←UCHCNT+1 ; %%% GET CHRTER, BUMP CHR. CNTER.           00615020 T  0078:3
          IF NOT UQSTRNG THEN %%% NOT IN A QUOTED STRING.                     00615025 T  0081:1
               BEGIN                                                          00615030 T  0082:1
               WHILE UH="/" DO %%% WE"VE HIT AN END-OF-RECORD MARK            00615035 T  0082:3
                    BEGIN      %%% ("/") OR AN INLINE CMMNT("/...=")         00615040 T  0084:0
                    DO UGNCH UNTIL CHR=1 OR UH="=" OR UH="←" ;                00615045 T  0084:0
                    IF CHR≠1 THEN                                             00615050 T  0088:1
                         BEGIN                                                00615055 T  0089:0
                         IF UH="=" THEN GO TO UL1 ;                           00615060 T  0089:2
                         IF UH="←" THEN                                       00615065 T  0090:3
                              BEGIN CHR←CSIZE; UL1: UGNCH END ;               00615070 T  0091:2
                         END ;                                                00615075 T  0094:0
                    END ;                                                     00615080 T  0094:0
               IF UH="←" THEN %%% WE SET UH=DELIMETER, AND IF THERES          00615085 T  0094:2
                    BEGIN      %%% MORE LIST, WE GET A NEW RECORD.            00615090 T  0095:1
                    UH←" "; IF LSTRN≥0 THEN UADDRS←LISX ;                     00615095 T  0095:3
                    SGN←SGN OR 32 ; %%% SETS ULIST = TRUE.                    00615100 T  0098:2
                    IF LSTRN≥0 THEN UGETRECORD ;                              00615105 T  0099:3
                    END ;                                                     00615110 T  0103:0
               IF UBUILD≠0      %%% WE HAVE NOT YET IDENTIFIED CST,           00615115 T  0103:0
               THEN UBUFFIT ;   %%% SO MUST SAVE UH IN UBUFF.                 00615120 T  0103:2
               END ;                                                          00615125 T  0105:0
          END ;                                                               00615130 T  0105:0
     END OF UCH ;                                                             00615135 T  0105:0
BOOLEAN SUBROUTINE UDELIMCHK ; %%% IS TRUE IFF HAVE ENCOUNTERED A             00615140 T  0105:1
     BEGIN                     %%% DELIMITER NOT IN A QUOTED STRING.          00615145 T  0106:0
     UDELIMCHK←UH="," OR UH=" " OR UH="*" OR UEOW ;                           00615150 T  0106:0
     END OF UDELIMCHK ;                                                       00615155 T  0110:0
DEFINE UCHECKIT =     %%% UCHECKIT IS USED WHENEVER THE SCAN HAS              00615160 T  0110:1
                      %%% TERMINATED. UCHECKIT CHECKS FOR THE PROPER          00615165 T  0110:1
                      %%% DELIMITER AND TAKES THE ASSOCIATED BRANCH.          00615170 T  0110:1
```

```
                    %%% UCHECKIT WILL POSITION UH APPROPIATELY          00615175 T  0110:1
                    %%% (PRIOR TO DELIMITER CHECKING) IF THE            00615180 T  0110:1
                    %%% MINIMUM FIELD WIDTH (UD) HAS NOT BEEN           00615185 T  0110:1
                    %%% EXHAUSTED.                                      00615190 T  0110:1
     IF UH="*" THEN UALLDONE ; %%% THE * TERMINATES THE READ STMT.       00615195 T  0110:1
     IF NOT((UBUFF←UH≠",") AND UH≠" ") THEN                              00615200 T  0110:1
          BEGIN                                                          00615205 T  0110:1
          UQSTRNG←UBUILD←0;  W←123;  D←UD ;                              00615210 T  0110:1
          WHILE UCHCNT≤D DO %%% SCAN OFF UNTIL AT LEAST                  00615215 T  0110:1
               BEGIN        %%% D CHARACTERS HAVE BEEN PASSED ;          00615220 T  0110:1
               UCH ;                                                     00615225 T  0110:1
               IF ULIST THEN GO TO UL2 ; %%% HAVE ENCOUNTERED AN ←.      00615230 T  0110:1
               IF UH="*" THEN UALLDONE ; %%% THE * TRMNTS THE READ.      00615235 T  0110:1
               IF (UBUFF OR UVAL←UH≠",") AND (UH=" " OR NOT UVAL)        00615240 T  0110:1
               THEN UBUFF←UBUFF AND UVAL ELSE UGOOFED(1) ;               00615245 T  0110:1
               END ;                                                     00615250 T  0110:1
          UL2: W←UW ;   %%% RESTORE W TO ITS ORIGINAL VALUE.             00615255 T  0110:1
          GO TO COMM  ; %%% AND MAKE NORMAL EXIT.                        00615260 T  0110:1
          END ;                                                          00615265 T  0110:1
     UGOOFED(2) #; %%% WAS NOT "*", ",", OR " " SO WE ERROR EXIT.        00615270 T  0110:1
     %%% END OF UCHECKIT.                                               00615275 T  0110:1
BOOLEAN SUBROUTINE UGETSGN ; %%% IS TRUE IF SIGN="-"; IF SIGN(+,-,&)     00615280 T  0110:1
     BEGIN                   %%% EXISTS, UGETSGN FETCHES A NEW CHAR.     00615285 T  0111:0
     IF P(UH="-",DUP) OR UH="+" OR UH="&" THEN UCH ;                     00615290 T  0111:0
     UGETSGN←POLISH ;                                                    00615295 T  0116:0
     END OF UGETSGN ;                                                    00615300 T  0116:1
BOOLEAN SUBROUTINE USDELIMCHK ; %%% IS TRUE IF CURRENT CHARACTER(UH)     00615305 T  0116:2
     BEGIN                       %%% IS A DELIMITER(* OR , OR BLANK);    00615310 T  0117:0
     IF NOT UQSTRNG THEN P(UDELIMCHK) ELSE %%% IF UH=RIGHT HAND          00615315 T  0117:0
          BEGIN                  %%% QUOTE OF QUOTED STRING,             00615320 T  0120:0
          IF UEOW THEN UGOOFED(3) ;   %%% THEN ONE AND POSSIBLY TWO      00615325 T  0120:2
          IF NOT(UH≠""" OR UEXP) THEN %%% CHAR ARE SCANNED UNTIL UH      00615330 T  0123:0
               BEGIN              %%% IS EITHER A DELIMITER OR           00615335 T  0124:2
               UQSTRNG←0;  UCH ;  %%% THE FIRST CHARACTER OF THE         00615340 T  0125:0
               IF NOT P(UDELIMCHK,DUP)%%% NEXT CONCATENATED STRING.      00615345 T  0128:0
               THEN BEGIN                                                00615350 T  0129:1
                    IF UH≠""" THEN UGOOFED(4) ;                          00615355 T  0130:0
                    SGN←SGN OR 16 ; %%% SETS UQSTRNG = TRUE.             00615360 T  0132:2
                    UCH;                                                 00615365 T  0133:3
                    END ;          %%% DO ERROR-EXIT IF WE ARE IN         00615370 T  0135:0
               END                 %%% QUOTED STRING AND: EXCEED         00615375 T  0135:0
          ELSE P(UEXP←0) ;         %%% W OR ENCOUNTER                    00615380 T  0135:0
          END ;                    %%% A NON-QUOTE, NON-DELIMITER        00615385 T  0136:1
     USDELIMCHK←POLISH ;                                                 00615390 T  0136:1
     END OF USDELIMCHK ;                                                 00615395 T  0136:2
% * * *   E N D   O F   U - P H R A S E   D E C L A R A T I O N S *      00615400 T  0136:3
                                                                         00615405 T  0136:3
LABEL C,X,A,I,R,E,O,L,Z,ZW2,ZD,SWT ;                                     00615410 T  0136:3
          GO TO START;  COMMENT GO AROUND FREE FIELD CODE;               00615415 T  0136:3
          COMMENT FREE FIELD FORMAT ;                                    00615420 T  0137:2
FREFLD::  P(0,0,0,0,0,0); COMMENT PUSH UP STACK;%                        00615500 T  0137:2
          LSTRN ← 0;%                                                    00615600 T  0139:2
          WT ← BSIZE × 8; COMMENT WT = # OF CHR IN BUFFER;%              00615700 T  0140:1
          BUFF ← P(0,0,[BUFF],CCX);%                                     00615800 T  0141:2
STRT:     ADDRS ← LISX;   COMMENT ADDRESS OF LIST ITEM;%                 00615900 T  0143:0
          IF LSTRN < 0 THEN BEGIN COMMENT CALL READ AND EXIT;%           00616000 T  0143:3
                              P(1);%                                     00616100 T  0145:0
```

```
                                READS;%                                                  00616200 T  0145:1
                              END;%                                                      00616300 T  0146:0
          GNCR;             COMMENT GET A CHARACTER TO TOP OF STACK;%                   00616400 T  0146:0
NMRCL:    ESIG ← DIVR ← 0; COMMENT SET EXPONENT AND FRACTION PART%                     00616500 T  0147:0
                                TO ZERO;%                                              00616600 T  0148:1
          IF (SGN ←    P(DUP) = "-") THEN GO TO HERE;%                                00616700 T  0148:1
          IF P(DUP) = "+" THEN GO TO HERE;%                                           00616800 T  0150:1
          IF P(DUP) ≠ "&" THEN GO TO NOSIG;%                                          00616900 T  0151:2
HERE:     P(DEL);%                                                                    00617000 T  0152:3
          GNCR;%                                                                      00617100 T  0153:0
NOSIG:    IF P(DUP) > 9 THEN GO TO NOTNUM;%                                           00617200 T  0154:0
          GNCR;%                                                                      00617300 T  0155:1
LPTWO:    IF P(DUP) > 9 THEN GO TO NOTNUM2;%                                          00617400 T  0156:0
          P(XCH,10,MUL,+);%                                                           00617500 T  0157:1
          GNCR; GO TO LPTWO;%                                                         00617600 T  0158:1
NOTNUM2:  IF P(DUP) ≠ "." THEN GO TO NFRAC1;%                                         00617700 T  0159:2
          P(DEL);%                                                                    00617800 T  0160:3
L1:       GNCR;%                                                                      00617900 T  0161:0
L1P2:     IF P(DUP) > 9 THEN GO TO NFRAC1;%                                           00618000 T  0162:0
          P(XCH,10,MUL,+);%                                                           00618100 T  0163:1
          DIVR ← DIVR +1;%                                                            00618200 T  0164:1
          GNCR; GO TO L1P2;%                                                          00618300 T  0165:2
NFRAC1:   IF P(DUP) ≠ "@" THEN GO TO NOTAT;%                                          00618400 T  0167:2
          P(DEL);%                                                                    00618500 T  0168:3
          GNCR;%                                                                      00618600 T  0169:0
ATS:      IF (ESIG ←     P(DUP) = "-") THEN GO TO HR1;%                               00618700 T  0170:0
          IF P(DUP) = "+" THEN GO TO HR1;%                                            00618800 T  0172:0
          IF P(DUP) ≠ "&" THEN GO TO NSG;%                                            00618900 T  0173:1
HR1:      P(DEL);%                                                                    00619000 T  0174:2
          GNCR;          COMMENT 1ST DIGIT;%                                          00619100 T  0174:3
NSG:      GNCR;          COMMENT 2ND DIGIT;%                                          00619200 T  0176:0
L2P1:     IF P(DUP) > 9 THEN GO TO FINXP;%                                            00619300 T  0177:0
          P(XCH,10,MUL,+);%                                                           00619400 T  0178:1
          GNCR; GO TO L2P1;%                                                          00619500 T  0179:1
FINXP:    IF P = "," THEN GO TO NCA1;%                                                00619600 T  0180:2
          GNCR; GO TO FINXP;%                                                         00619700 T  0181:2
NCA1:     IF ESIG THEN P(CHS);%                                                       00619800 T  0183:2
NMINUS:   ESIG ← P;%                                                                  00619900 T  0184:2
          P(",");%                                                                    00620000 T  0185:0
NOTAT:    IF P = "," THEN GO TO NCA2;%                                                00620100 T  0185:1
          GNCR; GO TO NOTAT;%                                                         00620200 T  0186:1
NCA2:     IF SGN THEN P(CHS);%                                                        00620300 T  0187:2
          IF P(ESIG-DIVR,DUP) = 0 THEN%                                               00620400 T  0188:2
               BEGIN%                                                                 00620500 T  0190:0
                    P(DEL);%                                                          00620600 T  0190:2
                    P([ADDRS],ISD);%                                                  00620700 T  0190:3
                    GO TO STRT%                                                       00620800 T  0191:1
               END;%                                                                  00620900 T  0191:3
          IF P(DUP) ≥ 0 THEN%                                                         00621000 T  0191:3
                    P(TEN[P],MUL)%                                                    00621100 T  0192:2
                      ELSE%                                                           00621200 T  0193:2
                    P(TEN[-P],/);%                                                    00621300 T  0193:2
NNMB:     P([ADDRS],STD);%                                                            00621400 T  0194:3
          GO TO STRT;%                                                                00621500 T  0195:1
NOTNUM:   IF P(DUP) ≠ "." THEN GO TO RNOTNUM;%                                        00621600 T  0195:3
          P(DEL);%                                                                    00621700 T  0197:0
          P(0);%                                                                      00621800 T  0197:1
```

```
                GNCR; GO TO L1P2;%                                                  00621900 T  0197:2
RNOTNUM:        IF P(DUP) ≠ "@" THEN GO TO INSERT;%                                 00622000 T  0199:2
                P(DEL);%                                                            00622100 T  0200:3
                P(1);%                                                              00622200 T  0201:0
                GNCR; GO TO ATS;%                                                   00622300 T  0201:1
INSERT:         IF P(DUP) = "," THEN%                                               00622400 T  0202:2
                     BEGIN%                                                         00622500 T  0203:1
                          P(DEL);%                                                  00622600 T  0203:3
                          GO TO STRT;%                                              00622700 T  0204:0
                     END;%                                                          00622800 T  0204:2
NAST:           IF P(DUP) ≠ """ THEN GO TO NQUOT;%                                  00622900 T  0204:2
                P(DEL);%                                                            00623000 T  0205:3
                TYP ← CCR ← 1;%                                                     00623100 T  0206:0
                GNCR;%                                                              00623200 T  0207:1
QRT:            GNCR;%                                                              00623300 T  0208:0
QRTN:           IF P(DUP) = """ THEN GO TO EQUT;%                                   00623400 T  0209:0
                P(XCH,64,MUL,+);%                                                   00623500 T  0210:1
                IF (CCR ← CCR+1) ≠ 6 THEN%                                          00623600 T  0211:1
                     BEGIN%                                                         00623700 T  0213:0
                          GNCR; GO TO QRTN;%                                        00623800 T  0213:2
                     END;%                                                          00623900 T  0215:2
                P([ADDRS],STD);%                                                    00624000 T  0215:2
                ADDRS ← LISX;%                                                      00624100 T  0216:0
                IF LSTRN < 0 THEN%                                                  00624200 T  0216:3
                     BEGIN%                                                         00624300 T  0217:2
                DO GNCR UNTIL P= """;%                                              00624400 T  0218:0
                     DO GNCR UNTIL P=",";%                                          00624500 T  0220:0
                          P(1);%                                                    00624600 T  0222:0
                          READS;%                                                   00624700 T  0222:1
                     END;%                                                          00624800 T  0223:0
                CCR ← 0;%                                                           00624900 T  0223:0
                GNCR;%                                                              00625000 T  0223:3
                IF P(DUP) = """ THEN GO TO EQUT;%                                   00625100 T  0225:0
                CCR ← 1;%                                                           00625200 T  0226:1
                GO TO QRT;%                                                         00625300 T  0227:0
EQUT:           P(DEL);%                                                            00625400 T  0227:2
                TYP ← 0;%                                                           00625500 T  0227:3
                IF CCR = 0 THEN%                                                    00625600 T  0228:2
                     BEGIN%                                                         00625700 T  0229:1
                          GNCR;%                                                    00625800 T  0229:3
                          GO TO EATUP;%                                             00625900 T  0231:0
                     END;%                                                          00626000 T  0231:2
                P([ADDRS],STD);%                                                    00626100 T  0231:2
                GNCR;%                                                              00626200 T  0232:0
EAT1:           IF P(DUP) = "," THEN GO TO STRT;%                                   00626300 T  0233:0
                GNCR; GO TO EAT1;%                                                  00626400 T  0234:1
EATUP:          IF P ≠ "," THEN BEGIN GNCR; GO TO EATUP END;%                       00626500 T  0235:2
                GNCR;%                                                              00626600 T  0238:2
                GO TO NMRCL;%                                                       00626700 T  0240:0
NQUOT:          IF P(DUP) ≠ "%" THEN GO TO ASTRX;%                                  00626800 T  0240:2
                P(DEL,0);%                                                          00626900 T  0241:3
                GNCR;%                                                              00627000 T  0242:1
GETO:           IF P(DUP) > 7 THEN GO TO GTRT7;%                                    00627100 T  0243:0
                P(XCH,DIA 4,DIB 1,TRB 44);%                                         00627200 T  0244:1
                GNCR; GO TO GETO;%                                                  00627300 T  0245:1
GTRT7:          IF P ≠ "," THEN BEGIN GNCR; GO TO GTRT7 END;%                       00627400 T  0246:2
                P([ADDRS],STD);%                                                    00627500 T  0249:2
```

```
                GO TO STRT;%
     ASTRX:     IF P(DUP) = "*" THEN%
                     BEGIN%
                          P(1);%
                          READS;%
                     END;%
     CHKOCT:    IF P(DUP) ≠ "/" THEN GO TO GETCOMA;%
                P(DEL);%
                WT ← 0;%
                GNCR;%
                GO TO NMRCL;%
  $ SET OMIT = NOT(TIMESHARING)
GETCOMA:  ESIG ← (FIB[4],[8:4]=10);  % TRUE IF REMOTE INPUT
  $ SET OMIT = TIMESHARING
GETC1:      IF P = "," OR (ESIG AND WT=0) THEN GO STRT ELSE
            GNCR;  GO GETC1;
           COMMENT START OF INPUTINT;%
     START::   P(LSTRN,0,0);%
               IF FILX.[18:15] > 1 THEN%
          BEGIN P(FILX[NOT 2]);%
               FILX[NOT 4]←EOFL; FILX[NOT 3]←PARL;
               IF NOT FIB[5],[12:1] THEN P(MKS,"READNG",FILX,7,SELECT) ;
               IF FIB[4],[27:3] ≠ 2 THEN
               IF FIB[5],[43:2]≠((ACT<0)+2) THEN%
                  POLISH(MKS,DKADR,(ACT<0) +2,FILX,1,SELECT);%
                    COMMENT CALL SELECT IF NOT READ STATUS OR%
                            WRONG DIRECTION ;%
               END ELSE P(0);%
               CKPB;%
               COMMENT  CHECK FOR TYPE OF READ STATMENT;%
          CT:  ACT←ABS(JUNK1←ACT);%
               IF ARRY ≠ 0  THEN GO TO CTB; COMMENT<LIST PART>NOT EMPTY;
               IF FRMT ≠ 0  THEN GO TO CTA; COMMENT <LIST PART>=EMPTY,%
                                    <FORMAT PART> IS NOT;%
               COMMENT BOTH <LIST PART> & <FORMAT PART> WAS EMPTY;%
               P(1);      COMMENT SET FLAG = EXIT;%
               READS;     COMMENT RELEASE BUFFER;%
          CTA: LSTRN ← -1;    COMMENT<LIST PART> = EMPTY;%
               GO TO FMOUT; COMMENT READ IS <FORMAT>,<EMPTY>;%
          CTB: IF NOT P(ARRY   ,TOP,XCH,DEL) THEN GO TO CTC;%
                    COMMENT IF LIST IS NOT A DESCRIPTOR WE HAVE%
                              A SPACE STATEMENT AND ACT = # OF%
                              RECORDS TO SPACE;%
               IF FIB[4],[8:4]=4 THEN%
               BEGIN IF FIB[4],[27:3]≠1 THEN%
                     P(MKS,FIB[7]+JUNK1,1,FILX,1,SELECT);%
               END ELSE%
               WHILE (ACT←ACT-1)≥0 DO%
               BEGIN CKPB; POLISH(MKS,DKADR,0,FILX,ALGOLREAD); END;
               LSTRN ← TLSTRN;%
               POLISH(XIT);%
          CTC: IF NOT P( FRMT,TOP) THEN GO TO FMOUTA; COMMENT WE HAVE%
                                   <FORMAT>,<LIST>;%
               IF P ≠ 0 THEN GO TO AEXPL; COMMENT WE HAVE AEXP,A[*];%
               IF FI=1 THEN GO TO FREFLD ELSE %%% / TYPE FREE-FIELD READ.
               IF FI=2 THEN GO FMOUTM1 ELSE %%% // TYPE FREE-FIELD READ.
               IF ARRY ≥ 0 THEN GO TO AEXPL;%
```

```
00627600 T  0250:0
00627700 T  0250:2
00627800 T  0251:1
00627900 T  0251:3
00628000 T  0252:0
00628100 T  0253:0
00628200 T  0253:0
00628300 T  0254:1
00628400 T  0254:2
00628500 T  0255:1
00628600 T  0256:0
00628650 T  0256:2
00628700 T  0256:2
00628701 T  0258:2
00628750 T  0258:2
00628800 T  0260:2
00628900 T  0262:2
00629000 T  0262:2
00629100 T  0263:3
00629200 T  0265:0
00629300 T  0266:3
00629330 T  0270:1
00629340 T  0273:1
00629400 T  0274:3
00629500 T  0277:3
00629600 T  0280:3
00629700 T  0280:3
00629800 T  0280:3
00629900 T  0283:1
00630000 T  0284:0
00630100 T  0284:0
00630200 T  0285:3
00630300 T  0287:1
00630400 T  0288:3
00630500 T  0288:3
00630600 T  0288:3
00630700 T  0289:0
00630800 T  0290:0
00630900 T  0291:0
00631000 T  0291:2
00631100 T  0293:1
00631200 T  0293:1
00631300 T  0293:1
00631400 T  0293:1
00631500 T  0294:3
00631600 T  0296:3
00631700 T  0299:2
00631800 T  0299:2
00631900 T  0302:2
00632000 T  0305:3
00632100 T  0306:2
00632200 T  0306:3
00632300 T  0308:0
00632400 T  0308:0
00632500 T  0309:0
00632510 T  0309:3
00632600 T  0311:0
```

```
                    COMMENT WE HAVE *,LIST;%                                    00632700 T  0313:0
    ASLST:     P(0,,LSTRN,SND); COMMENT LSTRN ← 0,   S ← I ← 0;%                00632800 T  0313:0
BS:       P(DUP,[LISX]); COMMENT S = ADDRESS OF LIST ITEM                       00632900 T  0313:3
                              S-1 = INDEX FOR BUFF;                             00633000 T  0314:1
               IF LSTRN < 0 THEN GO TO BR; COMMENT LIST IS EXAUSTED;%           00633200 T  0314:1
                    IF P(XCH,BUFF,XCH,STD,1,ADD,DUP) < BSIZE                    00633300 T  0315:2
                    THEN GO TO BS; COMMENT BUFFER ITEM TO LIST, IF I+1%         00633400 T  0317:1
                                   IS ≤ BUFFER SIZE THEN GET NEXT WORD;%        00633500 T  0318:1
    BR:        P(1);      COMMENT SET FLAG = EXIT;%                            00633600 T  0318:1
               READS;     COMMENT RELEASE BUFFER;%                             00633700 T  0318:2
         COMMENT AEXP,A[*];%                                                    00633800 T  0320:0
    AEXPL:     IF P(ARRY,[8:10],DUP) ≤ AEXP THEN GO TO ISA;%                    00633900 T  0320:0
               IF AEXP < 0 THEN ARRY[-1]←0;                                     00634000 T  0322:1
               P(DEL,AEXP);%                                                    00634100 T  0325:0
                    COMMENT STACK IS SMALLEST OF ARRAY SIZE OR AEXP;%           00634200 T  0325:2
    ISA:       IF P(DUP) ≤ BSIZE THEN GO TO ISB;%                               00634300 T  0325:2
               P(DEL,BSIZE);%                                                   00634400 T  0326:3
                    COMMENT  STACK NOW HAS SMALLEST OF BUFFER SIZE,AEXP,        00634500 T  0327:1
                      ARRAY SIZE;%                                              00634600 T  0327:1
    ISB:       BSIZE ← P;%                                                      00634700 T  0327:1
                    COMMENT BSIZE = # OF WORDS TO TRANSFER;%                    00634800 T  0327:3
               STREAM(P4 ← *FILX,P3 ← BSIZE,P2 ← BSIZE DIV 64,%                 00634900 T  0327:3
                      P1 ← [ARRY[0]]);%                                         00635000 T  0329:2
                    BEGIN%                                                      00635100 T  0330:1
                      SI←P4;%                                                   00635200 T  0330:1
                      P2(DS←32 WDS;%                                            00635300 T  0330:2
                        DS←32 WDS);%                                            00635400 T  0331:1
                      DS← P3 WDS;%                                              00635500 T  0331:3
                    END;%                                                       00635600 T  0332:1
ERROR:              P(1);    COMMENT ON ERROR, RELEASE BUFFER AND EXIT;%        00635700 T  0332:2
                    READS;%                                                     00635800 T  0333:1
         COMMENT WE HAVE  FORMAT,LIST OR FORMAT,<EMPTY>;%                       00635900 T  0334:0
FMOUTA:    P(DEL) ;                                                            00636000 T  0334:0
FMOUTM1:   LSTRN←0 ;                                                           00636100 T  0334:1
FMOUT:     P(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0) ;                          00636200 T  0335:0
                    COMMENT 19 GOOSE EGGS FOR YE OLDE STACK ;                   00636300 T  0339:3
               ADDRS ← LISX;%                                                   00636400 T  0339:3
               SAVEBUFF←BUFF←P(0,[BUFF],0,INX) ;                                00636500 T  0340:2
               CSIZE ← BSIZE × 8;%                                              00636600 T  0342:2
               IF FRMT=0 THEN %%% SPECIAL // TYPE OF FREE-FIELD READ.           00636610 T  0343:3
                    BEGIN CODE←11; UFREEFIELD←1; FAW←1&1[27:47:1] ;              00636620 T  0344:3
                    GO TO PHRAS END ELSE                                        00636630 T  0349:2
               GO TO S ;%                                                       00636700 T  0350:0
S2:            P(DEL) ;                                                        00636710 T  0350:0
S1:            FI←FI+1;              COMMENT LOOK AT NEXT EDITING PHRASE;%     00636800 T  0350:1
S:             CODE ← 0&(FAW ← FRMT[FI])[44:2:4];%                              00636900 T  0351:3
               IF FAW > 0 THEN GO TO PHRAS; COMMENT PHRAS IF S=0;%              00637000 T  0354:1
               GO TO P(CODE);%                                                  00637100 T  0355:2
               GO TO RTPAR;%                                                    00637200 T  0356:0
               GO TO STRNG;%                                                    00637300 T  0356:2
               GO TO LFPAR;%                                                    00637400 T  0357:0
               GO TO SLASH;%                                                    00637500 T  0357:2
               GO TO SCALE;%                                                    00637600 T  0358:0
         COMMENT LEFT PARENTHESIS;%                                             00637700 T  0358:2
LFPAR:     IF FAW,[12:1] THEN                                                  00637800 T  0358:2
               BEGIN IF P(LISTELEMENT,DUP)<0 THEN                               00637900 T  0359:1
                     BEGIN P(DEL); FI←FAW,[28:10]+FI END;                       00638000 T  0361:3
```

```
          END ELSE P(FAW,[38:10]);                                                00638100 T  0364:2
          GO TO S1;%                                                              00638200 T  0365:3
     COMMENT RIGHT PARENTHESIS;%                                                  00638300 T  0366:1
RTPAR:    IF P(1,SUB,DUP) =0%                                                     00638400 T  0366:1
           THEN BEGIN  COMMENT LFPAR REPEAT -1;%                                  00638500 T  0367:1
                 P(DEL);   COMMENT DELETE 0 REPEAT;%                              00638600 T  0368:0
                 GO TO S1; COMMENT PICK UP NEXT PHRASE;%                          00638700 T  0368:1
                END;%                                                             00638800 T  0368:3
          FI ← FI - (FAW AND 1023); COMMENT SET FI BACK TO LFPAR;%                 00638900 T  0368:3
          GO TO S1;%                                                              00639000 T  0370:3
     COMMENT  SCALE FACTOR;%                                                      00639100 T  0371:1
SCALE:    SCFTR←IF FAW,[12:1] THEN LISTELEMENT                                    00639200 T  0371:1
                              ELSE 0&FAW[38:38:10]&FAW[1:11:1];                   00639300 T  0373:1
          GO TO S1;%                                                              00639400 T  0377:1
     COMMENT STRINGS;%                                                            00639500 T  0377:3
STRNG:    IF  (CHR ← (W ← FAW,[6:6]) + CHR)>CSIZE%                                00639600 T  0377:3
           THEN GO TO ERROR; COMMENT BUFFER OVERFLOW;%                            00639700 T  0380:0
                                COMMENT CHR ← CHR + W;%                           00639800 T  0381:0
             STREAM(P2 ← W,P1 ← 0:P0 ← BUFF);%                                    00639900 T  0381:0
           BEGIN%                                                                 00640000 T  0382:2
                SI←P0;%                                                           00640100 T  0382:2
                DI←LOC P1;%                                                       00640200 T  0382:3
                DI←DI-P2;%                                                        00640300 T  0383:0
                DS←P2 CHR;%                                                       00640400 T  0383:2
                P1←SI;%                                                           00640500 T  0384:0
           END;%                                                                  00640600 T  0384:1
          BUFF ← P;%                                                              00640700 T  0384:2
          FRMT[FI] ← P(XCH)&FAW[1:1:11]; COMMENT DIAL S,CODE & W TO               00640800 T  0385:0
                    STRING OBTAINED FROM BUFFER AND PUT RESULT%                   00640900 T  0387:1
                    BACK INTO FORMAT ARRAY;%                                      00641000 T  0387:1
          GO TO S1;%                                                              00641100 T  0387:1
     COMMENT SLASH;%                                                              00641200 T  0387:3
    SLASH:     POLISH((LSTRN<0) AND FAW);%                                        00641300 T  0387:3
          READS;    COMMENT RELEASE BUFFER;%                                      00641400 T  0389:0
          CHR ←0;%                                                                00641500 T  0390:0
          BUFF ← P(0,[BUFF],0,INX);%                                              00641600 T  0390:3
          CSIZE ← BSIZE × 8;%                                                     00641700 T  0392:1
          GO TO S1;%                                                              00641800 T  0393:2
     COMMENT BREAK APART FORMAT WORD;%                                            00641900 T  0394:0
PHRAS:    IF FAW,[12:1] THEN P(LISTELEMENT) ELSE P(FAW,[38:10]);                  00642000 T  0394:0
          IF CODE=13 THEN CODE←IF (CODE←LISTELEMENT)="D" THEN 0 ELSE              00642100 T  0397:1
                                IF CODE="T" THEN 1 ELSE                           00642150 T  0401:3
                                IF CODE="X" THEN 2 ELSE                           00642200 T  0404:1
                                IF CODE="A" THEN 4 ELSE                           00642300 T  0406:1
                                IF CODE="I" THEN 6 ELSE                           00642400 T  0408:1
                                IF CODE="F" THEN 8 ELSE                           00642500 T  0410:1
                                IF CODE="E"THEN 10 ELSE                           00642600 T  0412:1
                                IF CODE="U" THEN 11 ELSE                          00642610 T  0414:1
                                IF CODE="O" THEN 12 ELSE                          00642700 T  0416:1
                                IF CODE="L" THEN 14 ELSE                          00642800 T  0418:1
                                IF CODE="R" THEN 15 ELSE 16;                      00642900 T  0420:1
          CODE1←CODE=1 ;                                                          00642905 T  0423:0
          IF (TYP←CODE=11) AND FAW,[31:1] THEN GO TO FMTERR ELSE                  00642910 T  0424:1
          W←IF FAW,[13:1] THEN LISTELEMENT-CODE1 ELSE                             00643000 T  0426:2
            IF TYP AND FAW,[27:1] THEN 64                                         00643010 T  0431:0
            ELSE FAW,[6:6] ;                                                      00643020 T  0432:1
          D←IF FAW,[14:1] THEN LISTELEMENT                                        00643100 T  0434:3
```

```
                ELSE IF TYP THEN FAW.[32:6]                                    00643110 T  0436:3
                ELSE (D1←FAW.[20:4])+(D2←FAW.[16:4]);                          00643200 T  0438:1
IF P(DUP)≤0 THEN GO TO S2 ;                                                    00643300 T  0442:3
IF W<0 THEN IF CODE1 AND W=(-1) THEN GO S2                                     00643320 T  0444:0
       ELSE IF NOT(CODE=0 OR CODE=12) THEN GO TO FMTERR;                       00643330 T  0446:3
IF D<0 THEN IF NOT(CODE≠15 AND CODE≠8 AND CODE≠10)                             00643340 T  0449:3
THEN GO TO FMTERR ;                                                            00643350 T  0453:1
IF W=0 THEN IF CODE≠2 AND NOT CODE1 THEN GO S2 ;                               00643360 T  0454:1
IF TYP THEN BEGIN IF D>63 THEN D←63 ;                                          00643370 T  0457:3
            IF NOT(FAW.[27:1] OR W<64) THEN GO TO FMTERR                       00643372 T  0460:2
            ELSE W←W.[42:6] ;  GO TO INLOOP END ELSE                           00643376 T  0462:2
IF FAW.[13:2]≠0 OR FAW.[2:4]=13 THEN                                           00643400 T  0464:2
BEGIN GO TO P(IF CODE=15 THEN 8 ELSE IF CODE1 THEN 2                           00643500 T  0467:3
              ELSE CODE) ;                                                     00643590 T  0470:2
      GO C; GO X; GO A; GO I; GO R; GO E; GO O; GO L;                          00643600 T  0472:1
      GO TO FMTERR ;                                                           00643700 T  0476:1
L:    W1←IF W≤5 THEN W ELSE 5; GO TO Z;                                        00643800 T  0476:3
X:    W1←W DIV 64; W←SKIP←W.[42:6];                                            00643900 T  0480:0
      GO TO ZW2;                                                               00644000 T  0483:0
A:    W1←IF W≤6 THEN W ELSE 6;                                                 00644100 T  0483:2
Z:    SKIP←W-W1; GO TO ZW2;                                                    00644200 T  0486:1
I:    W1←IF W≤8 THEN W ELSE 8;                                                 00644300 T  0488:0
      SKIP←IF W≤16 THEN 0 ELSE W-16;                                           00644400 T  0490:3
      W2←W-SKIP-W1; GO TO ZD;                                                  00644500 T  0494:0
E:    D←(FAW.[2:4]=13 OR FAW.[14:1])+D;                                        00644600 T  0496:1
      D2←D-D1←IF D≤8 THEN D ELSE 8;                                            00644700 T  0499:2
      SKIP←IF (W-D)≤5 THEN 0 ELSE W-D-5;                                       00644800 T  0503:1
      W1←W2←0; GO TO SWT;                                                      00644900 T  0507:2
R:    D2←D-D1←IF D≤8 THEN D ELSE 8;                                            00645000 T  0509:1
      SKIP←IF (W-D)≤17 THEN 0 ELSE W-D-17;                                     00645100 T  0513:0
      W1←IF (W-D)≤8 THEN W-D-1 ELSE 8;                                         00645200 T  0517:1
      W2←IF (W-D-SKIP)≤9 THEN 0 ELSE W-D-SKIP-9;                               00645300 T  0521:2
      GO TO SWT;                                                               00645400 T  0526:3
C: O: W←8; W1←SKIP←0;                                                          00645500 T  0527:1
ZW2:  W2←0;                                                                    00645600 T  0529:1
ZD:   D←D1←D2←0;                                                               00645700 T  0530:0
SWT:  WT←W1+W2;                                                                00645800 T  0531:3
      END ELSE                                                                 00645900 T  0533:0
      BEGIN WT←(W1←FAW.[28:4])+(W2←FAW.[24:4]);                                00646000 T  0533:0
            SKIP←FAW.[32:6];                                                   00646100 T  0536:3
      END;                                                                     00646200 T  0538:0
INLOOP:  IF CODE ≤ 2 THEN GO TO FLDW;%                                         00646300 T  0538:0
      IF LSTRN≥0 THEN IF CODE=11 THEN GO TO UTYPE ELSE GO FLDW                 00646400 T  0539:1
      ELSE UALLDONE ;                                                          00646500 T  0542:1
FLOW:    IF CODE1 THEN BEGIN BUFF←SAVEBUFF; CHR←W; GO XTYPE END ;              00646600 T  0542:1
      IF (CHR←W+CHR)>CSIZE                                                     00646700 T  0545:0
           THEN GO TO ERROR;   COMMENT BUFFER EXAUSTED;%                       00646800 T  0546:1
   COMMENT SELECT EDITING PHRASE;%                                             00646900 T  0547:1
JMP:     IF CODE = 15 THEN GO TO RTYPE;%                                       00647000 T  0547:1
      IF CODE THEN GO TO FMTERR ;                                              00647020 T  0548:2
      GO TO P(CODE);%                                                          00647100 T  0549:2
      GO TO DTYPE;    COMMENT CODE = 0 =D;%                                    00647200 T  0550:0
      GO TO XTYPE;    COMMENT CODE = 2 =X;%                                    00647300 T  0550:2
      GO TO ALFA;     COMMENT CODE = 4 =A;%                                    00647400 T  0551:0
      GO TO ITYPE;    COMMENT CODE = 6 =I;%                                    00647500 T  0551:2
      GO TO FTYPE;    COMMENT CODE = 8 =F;%                                    00647600 T  0552:0
      GO TO ETYPE;    COMMENT CODE = 10=E;%                                    00647700 T  0552:2
```

```
                GO TO OTYPE;    COMMENT CODE = 12=0;%                                   00647800 T  0553:0
                GO TO LOGI ;    COMMENT CODE = 14=L;%                                   00647900 T  0553:2
FMTERR:                                                                                 00647903 T  0554:0
UERR:           IF FILX.[18:15]>1 THEN                                                  00647906 T  0554:0
                     BEGIN %%% NOT ARRAYROWBUFF, SO TRY PAR LBL BRANCH.                 00647909 T  0555:1
                     P(FILX[NOT 3]);  FILX[NOT 3]←FILX[NOT 4]←0 ;                       00647912 T  0555:3
                     P(MKS,9,JUNK) ;                                                    00647915 T  0560:1
                     END ;                                                              00647918 T  0561:0
                TEN←0;  TEN←P([TEN[1]],CFX,SFB)&10[8:38:10] ;                           00647921 T  0561:0
                W2←((W1←FIB[7]+1)>9)+(W1>99)+(W1>999)+1 ;                               00647924 T  0564:1
                IF NOT UTYP THEN                                                        00647927 T  0569:1
                     BEGIN ;                                                            00647930 T  0570:1
                     STREAM(P2←W2,P1←W1,TEN) ;                                          00647933 T  0570:3
                          BEGIN DS←14LIT"-FMT ERR, REC=";  SI←LOC P1 ;                  00647936 T  0572:1
                          DS←P2 DEC;  DS←10LIT", NO LBL:←" ;                           00647939 T  0574:2
                          END ;                                                         00647942 T  0576:2
                     END                                                                00647945 T  0576:3
                ELSE                                                                    00647948 T  0576:3
                     BEGIN ;                                                            00647951 T  0576:3
                     STREAM(P9←W2,P8←(CHR>9)+(CHR>99)+1,P7←UEXP,UH,CHR,                 00647954 T  0577:1
                            W1,P6←D←UD,P5←D≠0,P4←(D>9)+1,P3←W←UW,P2←W≠0,                00647957 T  0580:3
                            P1←(W>9)+1,P0←CHR←UFREEFIELD,TEN) ;                         00647960 T  0586:1
                          BEGIN DS←2LIT"=U";  P2(SI←LOC P3; DS←P1 DEC) ;                 00647963 T  0589:2
                          P5(DS←LIT",";  SI←LOC P6; DS←P4 DEC);P0(DI←DI+5);             00647966 T  0591:2
                          DS←5LIT" ERR#";  SI←LOC P7;  DS←DEC ;                         00647969 T  0594:2
                          DS←5LIT",CHR=";  SI←SI+7;  DS←CHR ;                           00647972 T  0596:0
                          DS←5LIT",COL=";  DS←P8 DEC;  DS←3LIT",R=" ;                   00647975 T  0597:2
                          DS←P9 DEC;  DS←9LIT",NO LBL:←" ;                              00647978 T  0599:3
                          END ;                                                         00647981 T  0601:3
                     IF CHR THEN STREAM(TEN); DS←7LIT"=FREFLD" ;                        00647984 T  0602:0
                     END ;                                                              00647987 T  0605:1
                P([TEN[0]].[33:15],34,COM) ;                                            00647990 T  0605:1
           COMMENT L PHRASE;%                                                           00648000 T  0606:3
LOGI:                STREAM(P3 ← W1, P2 ← BUFF:P1 ← SKIP);%                             00648100 T  0606:3
                     BEGIN%                                                             00648200 T  0608:1
                          SI←P2;%                                                       00648300 T  0608:1
                          SI←SI+P1;       COMMENT SKIP ANY LEADING BLANKS;              00648400 T  0608:2
                          DI←LOC P1;%                                                   00648500 T  0609:0
                          DS←6 LIT "TRUE  "; COMMENT PUT COMPARE IN P1;%                00648600 T  0609:1
                          DI←DI-6;%                                                     00648700 T  0610:1
                          IF P3 SC ≠ DC%                                                00648800 T  0610:2
                             THEN GO TO BL ;                                            00648900 T  0611:0
                     LA:  TALLY ← 1 ;          COMMENT IF SAME,P3←1;%                  00649000 T  0611:2
                          GO TO LC;%                                                    00649100 T  0611:3
                     BL:  DI←LOC P1 ;                                                   00649200 T  0612:0
                          DS← 6 LIT " TRUE "; COMMENT PUT COMPARE IN P1;%               00649300 T  0612:1
                          DI←DI-6;%                                                     00649400 T  0613:1
                          SI←SI-P3;%                                                    00649500 T  0613:2
                          IF P3 SC=DC%                                                  00649600 T  0614:0
                          THEN TALLY←1;       COMMENT IF SAME, P3=1;%                   00649700 T  0614:2
                     LC:  P3 ← TALLY;%                                                  00649800 T  0615:0
                          P2 ← SI;%                                                     00649900 T  0615:1
                     END;%                                                              00650000 T  0615:2
                GO TO COMA;%                                                            00650100 T  0615:3
           COMMENT D PHRASE;%                                                           00650200 T  0616:1
DTYPE:               STREAM(P2 ← 0:P1 ← BUFF);%                                         00650300 T  0616:1
                     BEGIN%                                                             00650400 T  0617:2
```

```
                    SI←P1;%                                                    00650500 T  0617:2
                    SI←SI+8;%                                                  00650600 T  0617:3
                    P2←SI;%                                                    00650700 T  0618:0
               END;%                                                           00650800 T  0618:1
          BUFF ← P;%                                                           00650900 T  0618:2
          GO TO COMM;%                                                         00651000 T  0619:0
     COMMENT O PHRASE;%                                                        00651100 T  0619:2
OTYPE:                                                                         00651200 T  0619:2
          STREAM(P2←0: P1←BUFF);    % CHECK FOR FLAG BIT                      00651210 T  0619:2
          BEGIN                                                                00651220 T  0620:3
               SI ← P1;                                                        00651230 T  0620:3
               IF SB THEN TALLY ← 1;                                           00651240 T  0621:0
               P2 ← TALLY;                                                     00651250 T  0621:3
          END;                                                                 00651300 T  0622:0
          IF P THEN                                                            00651310 T  0622:1
          BEGIN          % DATA HAS FLAG BIT                                   00651320 T  0622:1
               COMMENT IF F-FIELD = 0 OR R THEN LIST ITEM IS                   00651330 T  0622:3
               SIMPLE VARIABLE IN STACK OR PRT;                                00651340 T  0622:3
               IF (JUNK1 ← [ADDRS].[18:15]) = 0 THEN GO FLAGBIT;               00651350 T  0622:3
               IF P(10,LOD).[18:15] = JUNK1 THEN GO FLAGBIT;                   00651400 T  0625:1
          END;                                                                 00651410 T  0627:1
          COMMENT EITHER NO FLAG BIT OR DATA GOES TO ARRAY;                    00651420 T  0627:1
          STREAM(P3←0: P2←BUFF, P1←[ADDRS]);                                   00651430 T  0627:1
          BEGIN                                                                00651440 T  0628:3
               SI ← P2;  % DI SET FROM LAST PARAMETER                          00651450 T  0628:3
               DS ← 8 CHR;                                                     00651500 T  0629:0
               P3 ← SI;                                                        00651510 T  0629:1
          END;                                                                 00651520 T  0629:2
          BUFF ← P;                                                            00651530 T  0629:3
          GO TO COMM;                                                          00651540 T  0630:1
FLAGBIT:                                                                       00651550 T  0630:3
          COMMENT FLAGGED DATA GOING TO STACK OR PRT CAN CAUSE                 00651600 T  0630:3
          BAD PROBLEMS. FORCE FLAG BIT INTERRUPT HERE;                         00651610 T  0630:3
          JUNK1 ← [JUNK1];                                                     00651620 T  0630:3
          P(JUNK1);                                                            00651630 T  0631:2
          COMMENT CONTROL CANNOT REACH THIS POINT;                             00651640 T  0631:3
     COMMENT U PHRASE;%                                                        00651715 T  0631:3
UTYPE:    IF D≥CSIZE THEN UALLDONE ; %%% EXIT IF MIN FLD-WDTH≥BUFFSZ          00651720 T  0631:3
          SGN←SGN&12[42:42:6]&D[36:42:6]&W[30:42:6] ;                          00651725 T  0633:0
          W←IF W=0 THEN TEN[60] ELSE IF W>CSIZE THEN CSIZE ELSE W+1;           00651730 T  0636:3
          UBUILD←UCHCNT←0 ;                                                    00651735 T  0641:2
          DO UCH UNTIL UEOW OR UH≠" " ; %%% SCAN UNTIL CST OR E-0-W.           00651740 T  0643:2
          USCHCNT←UBUILD←1 ;                                                   00651745 T  0647:1
          IF UDELIMCHK THEN GO TO UL5 ;  UBUFF←UH ;                            00651750 T  0648:2
          UNLOCATED←UNUM←UVAL←UDEC←UEXP←0;  USGN←UGETSGN ;                     00651755 T  0651:2
          IF UH="%" THEN %%% WE MAY HAVE AN OCTAL NUM; ERROR EXIT IF           00651760 T  0658:1
               BEGIN     %%% GTR 3777777777777777 OR HAS DIGIT GTR 7         00651765 T  0659:0
               UCH; SGN←SGN EQV (NOT UGETSGN) ; %%%USGN←USGN+UGETSGN           00651770 T  0659:2
               UNUM←UH<8 ;                                                     00651775 T  0663:0
               WHILE (UBUILD←UBUILD+1)<17 AND UH<8 DO                          00651780 T  0664:1
                    BEGIN UVAL←UH&UVAL[3:6:42];  UCH END ;                     00651785 T  0667:2
               IF UBUILD=17 AND UH<8 AND (NOT UVAL.[3:1]) THEN                 00651790 T  0672:0
                    BEGIN %%% WE NOW BUILD 16-TH OCTAL DIGIT.                  00651795 T  0675:0
                    UVAL←UH&UVAL[1:4:44];  UCH ;                               00651800 T  0675:2
                    END ;                                                      00651805 T  0678:0
               GO TO UENDNUM ;                                                 00651810 T  0678:0
               END ;                                                           00651815 T  0678:2
```

```
                    UNUM←UH<10 ;                                                                00651820 T  0678:2
                    WHILE UH<10 DO BEGIN UVAL←10×UVAL+UH;  UCH END ;                            00651825 T  0679:3
                    IF UH="." THEN                                                              00651830 T  0684:2
                         BEGIN                                                                  00651835 T  0685:1
                         UCH;  UNUM←UNUM OR UH<10 ;                                             00651840 T  0685:3
                         WHILE UH<10 DO                                                         00651845 T  0688:3
                              BEGIN UBUILD←UBUILD+1; UDEC←10×UDEC+UH; UCH END;                  00651850 T  0690:0
                         END ;                                                                  00651855 T  0694:0
                    IF UH="@" OR UH="E" THEN                                                    00651860 T  0694:2
                         BEGIN UBUILD←-UBUILD;  UCH;  UEXPSGN←UGETSGN ;                         00651865 T  0696:1
                         IF NOT UNUM THEN UVAL←1 ;                                              00651870 T  0701:1
                         IF UH<10 THEN                                                          00651875 T  0703:0
                              BEGIN UNUM←1;  UBUILD←-UBUILD ;                                   00651880 T  0703:3
                              DO BEGIN UEXP←10×UEXP+UH; UCH END UNTIL UH>9 ;                    00651885 T  0706:0
                              END ;                                                             00651890 T  0710:1
                         END ;                                                                  00651895 T  0710:1
UENDNUM:            IF UNUM THEN %%% THE CST HAS ENOUGH CHARACTERS TO UNAMBIG-                   00651900 T  0710:1
                         BEGIN   %%% UOUSLY APPEAR AS A NUMBER.                                 00651905 T  0711:1
                         IF UBUILD≤0 THEN UGOOFED(5) ;                                          00651910 T  0711:3
                         UVAL←UVAL+UDEC/TEN[UBUILD-1] ;                                         00651915 T  0714:1
                         IF UEXP≠0 THEN UVAL←P(UVAL,TEN[UEXP],IF UEXPSGN THEN                   00651920 T  0716:3
                            P(/) ELSE P(×)) ;                                                   00651925 T  0719:2
                         P(IF SGN THEN -UVAL ELSE UVAL,[ADDRS],STD) ;                           00651930 T  0721:2
                         UL5:  UCHECKIT ;                                                       00651935 T  0724:0
                         END ;                                                                  00651940 T  0750:3
                         UBUILD←0 ;                                                             00651945 T  0750:3
                         IF UH≠"""" THEN                                                        00651950 T  0751:2
                              IF UDELIMCHK THEN UBUFF←UBUFF,[12:30] ELSE GO TO UL3              00651955 T  0752:1
                         ELSE BEGIN UBUFF←USCHCNT←0; UQSTRNG←UEXP←1; UL3: UCH END ;              00651960 T  0755:3
                         IF NOT(UDEC←USDELIMCHK) AND USCHCNT<6 THEN                             00651965 T  0761:0
                              BEGIN UL4: UBUFFIT; GO TO UL3 END ;                               00651970 T  0763:3
                         IF (UVAL←0)=USCHCNT THEN GO TO UL6 ;                                   00651975 T  0765:2
                         DO IF (UVAL←"x"&UVAL[24:30:18])=UBUFF THEN GO TO UL6                   00651980 T  0767:1
                         UNTIL UVAL,[24:1] ;                                                    00651985 T  0769:2
                         P(UBUFF,[ADDRS],STD) ;                                                 00651990 T  0771:1
                         UL6:  IF UDEC THEN GO TO UL5;  IF LSTRN≥0 THEN ADDRS←LISX;             00651995 T  0772:0
                         IF LSTRN<0 THEN                                                        00652000 T  0774:1
                              BEGIN DO UCH UNTIL USDELIMCHK;  GO TO UL5 END ;                   00652005 T  0775:3
                         USCHCNT←UBUFF←0;  GO TO UL4 ;                                          00652010 T  0779:0
                    COMMENT A PHRASE ;                                                          00652015 T  0780:3
ALFA:                         STREAM(P3←W1,P2←BUFF:P1←SKIP);%                                   00652100 T  0780:3
                         BEGIN%                                                                 00652200 T  0782:1
                              SI←P2;%                                                           00652300 T  0782:1
                              SI←SI+P1; COMMENT SKIP EVERYTHING BUT LAST 6;%                    00652400 T  0782:2
                              DI←LOC P2;%                                                       00652500 T  0783:0
                              DI←DI-P3;%                                                        00652600 T  0783:1
                              DS←P3 CHR;%                                                       00652700 T  0783:3
                              P2←SI;%                                                           00652800 T  0784:1
                         END;%                                                                  00652900 T  0784:2
                         GO TO COMA;%                                                           00653000 T  0784:3
COMMENT X PHRASE AND T PHRASE ;                                                                 00653100 T  0785:1
XTYPE:              IF (CHR←CHR+W1×64)>CSIZE-CODE1                                              00653200 T  0785:1
                     THEN GO TO ERROR;   COMMENT BUFFER EXAUSTED;%                              00653300 T  0787:1
                         STREAM(P3←BUFF:P2←W1,P1←W);%                                           00653400 T  0788:2
                         BEGIN%                                                                 00653500 T  0790:0
                              SI←P3;%                                                           00653600 T  0790:0
                              SI←SI+P1;%                                                        00653700 T  0790:1
```

```
                    P2(SI←SI+32;%                                                    00653800 T  0790:3
                       SI←SI+32);%                                                   00653900 T  0791:2
                    P3←SI;%                                                          00654000 T  0792:0
                END;%                                                                00654100 T  0792:1
            BUFF ← P;%                                                               00654200 T  0792:2
            GO TO COMM;%                                                             00654300 T  0793:0
        COMMENT I PHRASE;%                                                           00654400 T  0793:2
ITYPE:  P(0);          COMMENT RLIT = FROM I (SEE FOUT);%                            00654500 T  0793:2
FIN:        COMMENT FIRST WE GET SIGN AND COUNT LEADING BLANKS;%                     00654600 T  0793:3
              STREAM(%                                                               00654700 T  0793:3
                P4←WT,     COMMENT IN=FIELD WIDTH,OUT=LEADING%                     00654800 T  0794:0
                                                      BLANKS;%                       00654900 T  0794:1
                P3←0 ,     COMMENT PLACE TO RETURN SIGN;%                          00655000 T  0794:1
                P2←BUFF:   COMMENT IN AND OUT=BUFFER ADDRESS;%                     00655100 T  0794:2
                P1←SKIP);COMMENT # OF LEADING CHARACTERS TO%                       00655200 T  0795:0
                           IGNORE;%                                                  00655300 T  0795:2
              BEGIN%                                                                 00655400 T  0795:2
                SI←P2;%                                                            00655500 T  0795:2
                SI←SI+P1;%                                                         00655600 T  0795:3
                P4(IF SC≠" " THEN%                                                 00655700 T  0796:1
                 JUMP REAL TO NTBLK;%                                              00655800 T  0797:1
                   SI←SI+1;%                                                       00655900 T  0797:3
                   TALLY←TALLY+1);%                                                00656000 T  0798:0
                         COMMENT IF WE FALL THROUGH LOOP THEN%                     00656100 T  0798:2
                             WHOLE FIELD WAS BLANK;%                               00656200 T  0798:2
                P4←TALLY;%                                                         00656300 T  0798:2
                GO TO IEXIT;%                                                      00656400 T  0798:3
NTBLK:          IF SC<"0" THEN%                                                    00656500 T  0799:0
                BEGIN COMMENT SIGN IS PRESENT;%                                    00656600 T  0799:2
                IF SC="-" THEN;            COMMENT TOGGLE← TRUE;%                  00656700 T  0799:2
                SI←SI+1;                   COMMENT SKIP SIGN;%                     00656800 T  0800:0
                TALLY←TALLY+1;%                                                    00656900 T  0800:1
                END;%                                                              00657000 T  0800:2
IMPLS:          P4←TALLY;       COMMENT LEADING BLANKS+"SIGN";%                    00657100 T  0800:2
                TALLY←0;        COMMENT  INDICATE + SIGN;%                         00657200 T  0800:3
                IF TOGGLE%                                                         00657300 T  0801:0
                 THEN TALLY←1; COMMENT TOGGLE = TRUE IF "-";%                      00657400 T  0801:0
                P3←TALLY;       COMMENT PASS BACK SIGN;%                           00657500 T  0801:2
IEXIT:          P2←SI;          COMMENT ADDRESS OF FIRST DIGIT;%                   00657600 T  0801:3
            END;%                                                                  00657700 T  0802:0
BUFF ← P;%                                                                         00657800 T  0802:1
SGN  ← P;%                                                                         00657900 T  0802:3
            COMMENT NOW TO CONVERT INTEGER;%                                       00658000 T  0803:1
STREAM(%                                                                           00658100 T  0803:1
            P5 ←(P(SSN,WT,+,DUP)),%                                                00658200 T  0803:2
            P4 ← (IF P ≤ 8 THEN 0%                                                 00658300 T  0804:2
                           ELSE P(8, - ,8,XCH)),%                                  00658400 T  0805:3
            COMMENT  IF WT-"LEADING BLANKS" > 8%                                   00658500 T  0807:1
                 THEN P5←WT-LEADING BLANKS,P4← 0%                                  00658600 T  0807:1
                 ELSE P5←8,P4←WT-LEADING BLANKS-8;%                                00658700 T  0807:1
            P3←0,      COMMENT PLACE TO RETURN LOW HALF;%                          00658800 T  0807:1
            P2←0,COMMENT PLACE TO RETURN HIGH HALF;%                               00658900 T  0807:2
            P1←0: P0 ← BUFF);%                                                     00659000 T  0807:3
            BEGIN%                                                                 00659100 T  0808:3
                SI←P0;%                                                            00659200 T  0808:3
                DI←LOC P2;%                                                        00659300 T  0809:0
                DS←P4 OCT;      COMMENT CONVERT HIGH HALF;%                        00659400 T  0809:1
```

```
                    DI←LOC P3;%                                                  00659500 T  0809:3
                    DS←P5 OCT;      COMMENT CONVERT LOW  HALF;%                  00659600 T  0810:0
                    P1←SI;%                                                      00659700 T  0810:2
                END;%                                                            00659800 T  0810:3
            BUFF ← P;               COMMENT SAVE NEXT FIELD ADDRESS;             00659900 T  0811:0
            P(TEN8,MUL,+);          COMMENT HIGH HALF × 10*8%                    00660000 T  0811:2
                                                    + LOW HALF;%                 00660100 T  0812:1
            IF SGN THEN P(CHS);%                                                 00660200 T  0812:1
            IF P(XCH,DEL,XCH,DEL,XCH,DUP) THEN P(XCH,[ADDRS],←)                  00660290 T  0813:1
            ELSE IF P(XCH,DUP)≤ P(MAXI) THEN P([ADDRS],ISD)                      00660300 T  0816:0
                ELSE P([ADDRS],←);                                               00660310 T  0820:0
% VOID                                                                          00660400 T  0821:0
    FOUT:   IF P THEN GO TO FA;%                                                 00660500 T  0821:0
            GO TO COMM;%                                                         00660600 T  0821:3
        COMMENT  F PHRASE;%                                                      00660700 T  0822:1
    FTYPE:  P(1);%                                                               00660800 T  0822:1
            GO TO FIN;      COMMENT USE ITYPE TO CONVERT INTEGER PART;           00660900 T  0823:1
    FA:     STREAM(P5← D2,%                                                      00661000 T  0823:3
                P4← D1,%                                                         00661100 T  0824:1
                P3← 0 ,             COMMENT PLACE TO RETURN LOW HALF;            00661200 T  0824:2
                P2← 0 ,         COMMENT PLACE TO RETURN HIGH HALF;%              00661300 T  0824:3
                P1 ← 0:P0 ← BUFF);%                                              00661400 T  0825:0
                BEGIN%                                                           00661500 T  0826:0
                    SI←P0 ;%                                                     00661600 T  0826:0
                    SI←SI+1;  COMMENT SKIP DECIMAL POINT;%                       00661700 T  0826:1
                    DI←LOC P2;%                                                  00661800 T  0826:2
                    DS←P4 OCT;    COMMENT CONVERT HIGH HALF;%                    00661900 T  0826:3
                    DI←LOC P3;%                                                  00662000 T  0827:1
                    DS←P5 OCT;    COMMENT CONVERT LOW HALF;%                     00662100 T  0827:2
                    DI←LOC P3;%                                                  00662000 T  0827:1
                    DS←P5 OCT;    COMMENT CONVERT LOW HALF;%                     00662100 T  0827:2
                    P1←SI;%                                                      00662200 T  0828:0
                END;%                                                            00662300 T  0828:1
            BUFF ← P;%                                                           00662400 T  0828:2
            P(TEN[D2] × P + P); COMMENT HIGH HALF × 10*D2 + LOW HALF;             00662500 T  0829:0
            P((ABS(ADDRS)× TEN[D]) + P);  COMMENT INSERT INTEGER PART;           00662600 T  0830:0
            P(TEN[D],/);            COMMENT SCALE TO PROPER DECIMAL PLACE;       00662700 T  0831:2
            IF SGN  THEN P(CHS);%                                                00662800 T  0832:1
            P([ADDRS],STD);%                                                     00662900 T  0833:1
            P(DEL,DEL) ;%                                                        00663000 T  0833:3
            GO TO COMM;%                                                         00663100 T  0834:1
        COMMENT E PHRASE;%                                                       00663200 T  0834:3
    ETYPE:  STREAM(P6← 0,           COMMENT PLACE TO RETURN EXPONENT;%           00663300 T  0834:3
              P5 ← P(D-1, DUP), COMMENT D2 IN,MANTISSA SIGN OUT;                  00663400 T  0835:2
               P4 ← (IF P ≤ 8 THEN P(0,,D2,SND,XCH)%                             00663500 T  0836:2
                        ELSE P(8,-,,D2,SND,8)),%                                 00663600 T  0838:2
            COMMENT IF (D-1) > 8 THEN P5= D-1-8,P4= 8%                           00663700 T  0840:1
                            ELSE P5= 0,     P4=D-1,%                             00663800 T  0840:1
                            D1 ← P4, ON RETURN P4=INTEGER%                       00663900 T  0840:1
                            DIGIT;%                                              00664000 T  0840:1
            P3 ← P(0),      COMMENT PLACE TO RETURN LOW HALF;%                   00664100 T  0840:1
              P2 ← 0,       COMMENT PLACE TO RETURN HIGH HALF;%                  00664200 T  0840:2
              P1 ← BUFF;%                                                        00664300 T  0840:3
              P0 ← SKIP);%                                                       00664400 T  0841:1
            BEGIN%                                                               00664500 T  0841:3
                SI←P1;%                                                          00664600 T  0841:3
                SI←SI+P0;%                                                       00664700 T  0842:0
                P0←SI; COMMENT ADDRESS OF INTEGER;%                              00664800 T  0842:2
                SI←SI+2;  COMMENT SKIP INTEGER DIGIT & ".";%                     00664900 T  0842:3
```

```
                DI← LOC P2;%                                                    00665000 T  0843:0
                DS← P4 OCT; COMMENT CONVERT HIGH HALF;%                         00665100 T  0843:1
                DI← LOC P3;%                                                    00665200 T  0843:3
                DS← P5 OCT; COMMENT CONVERT LOW  HALF;%                         00665300 T  0844:0
                SI←SI+1;  COMMENT SKIP "@";%                                    00665400 T  0844:2
                IF SC="-" THEN; COMMENT IF EXPONENT < 0%                        00665500 T  0844:3
                                THEN TOGGLE ← TRUE;%                            00665600 T  0845:1
                SI←SI+1;  COMMENT SKIP EXPONENT SIGN;%                          00665700 T  0845:1
                DI← LOC P6;%                                                    00665800 T  0845:2
                DS← 2 OCT;  COMMENT CONVERT EXPONENT;%                          00665900 T  0845:3
                P1← SI;         COMMENT RETURN ADDRESS OF NEXT%                 00666000 T  0846:0
                                FIELD;%                                         00666100 T  0846:1
                IF TOGGLE THEN%                                                 00666200 T  0846:1
                  BEGIN  DI←DI-8;%                                              00666300 T  0846:2
                    DS← LIT "+";%                                               00666400 T  0846:3
                 END;      COMMENT IF TOGGLE SET EXPONENT%                      00666500 T  0847:1
                                NEGATIVE;%                                      00666600 T  0847:1
                SI←P0;%                                                         00666700 T  0847:1
                DI←LOC P4;      COMMENT CONVERT INTEGER DIGIT;%                 00666800 T  0847:2
                DS ← OCT;%                                                      00666900 T  0847:3
                SI←SI-2;  COMMENT LOOK AT SIGN;%                                00667000 T  0848:0
                IF SC="-" THEN TALLY ←1;%                                       00667100 T  0848:1
                P5←TALLY;%                                                      00667200 T  0849:0
            END;%                                                               00667300 T  0849:1
        COMMENT ON RETURN STACK CONTAINS%                                       00667400 T  0849:2
                                BUFF%                                           00667500 T  0849:2
                                HIGH HALF%                                      00667600 T  0849:2
                                LOW  HALF%                                      00667700 T  0849:2
                                INTEGER DIGIT%                                  00667800 T  0849:2
                                MANTISSA SIGN%                                  00667900 T  0849:2
                                EXPONENT;%                                      00668000 T  0849:2
                BUFF ← P;%                                                      00668100 T  0849:2
                P(TEN[D2] × P + P); COMMENT HIGH HALF×10*D2+LOW HALF;%           00668200 T  0850:0
                P(XCH,TEN[D-1]×P+P);COMMENT SCALE INTEGER DIGIT D PLACES%        00668300 T  0851:0
                                        AND ADD FRACTION PART;%                 00668400 T  0852:3
                IF P(XCH) THEN P(CHS); COMMENT INSERT SIGN;%                    00668500 T  0852:3
                P(XCH);                COMMENT EXPONENT TO TOP;%                00668600 T  0853:3
                IF (JUNK1 ← P-(D-1)) ≥ 0%                                       00668700 T  0854:0
                     THEN P(TEN[JUNK1],MUL)%                                    00668800 T  0855:3
                     ELSE P(TEN[-JUNK1],/); COMMENT INSERT EXPONENT;%           00668900 T  0857:1
                GO TO COMB;%                                                    00669000 T  0858:3
COMA:           BUFF ← P;%                                                      00669100 T  0859:1
COMB:           P([ADDRS],STD);  COMMENT RESULT TO LIST;%                       00669200 T  0859:3
COMM:           IF CODE ≤ 2 THEN GO TO COMC; COMMENT PHRASE DIDNT USE%          00669300 T  0860:1
                                        ANYTHING FROM LIST;%                    00669400 T  0861:2
                IF LSTRN≥0 THEN ADDRS←IF NOT ULIST THEN LISX                    00669500 T  0861:2
                                                ELSE P(,UADDRS,LOD) ;           00669505 T  0864:1
                ULIST←0 ;                                                       00669510 T  0866:0
COMC:           IF P((UFREEFIELD=0),-,DUP)>0 THEN GO TO INLOOP ;                00669600 T  0867:3
                P(DEL);%                                                        00669700 T  0870:2
                GO TO S1;%                                                      00669800 T  0870:3
                COMMENT THE <REPEAT PART> OF PHRASE IS IN TOP OF STACK%         00669900 T  0873:0
                        NOW(I HOPE). IF REPEAT-1 > 0 THEN GO TO INLOOP TO       00670000 T  0873:0
                        USE SAME PHRASE OVER. IF REPETE = 0 THEN DELETE%        00670100 T  0873:0
                        THE 0 AND GO TO S1 TO PICK UP NEXT PHRASE;%             00670200 T  0873:0
            COMMENT R EDITING PHRASE;%                                          00670300 T  0873:0
RTYPE::     STREAM(P6 ←(FLG←0), COMMENT RETURNS FLAG AS TO WHAT IS%             00670400 T  0873:0
```

```
                                    IN BUFFER;%                                  00670500 T  0874:0
          P5 ← 0,       COMMENT SIGN;%                                         00670600 T  0874:0
          P4 ← W,       COMMENT FIELD WIDTH;%                                  00670700 T  0874:1
          P3 ← BUFF:    COMMENT BUFFER CHARACTER ADDRESS;%                     00670800 T  0874:2
          P1 ← 0);%                                                            00670900 T  0875:0
     BEGIN%                                                                    00671000 T  0875:2
          SI ← P3;%                                                            00671100 T  0875:2
          TALLY ← P4;%                                                         00671200 T  0875:3
          COMMENT SKIP LEADING BLANKS;%                                        00671300 T  0876:2
          P4(IF SC ≠ " " THEN JUMP OUT TO RSIGN;%                              00671400 T  0876:2
             SI←SI+1;%                                                         00671500 T  0878:0
             TALLY ← TALLY +63);COMMENT TALLY=1;%                              00671600 T  0878:1
          COMMENT FALL THRU LOOP MEANS FIELD WAS BLANK;%                       00671700 T  0878:3
          TALLY←4;      COMMENT SET FLAG TO 4;%                                00671800 T  0878:3
          GO TO RXITA;%                                                        00671900 T  0879:0
NOI:      TALLY ← TALLY + 63;%                                                 00672000 T  0879:1
          SI ← SI+1;%                                                          00672100 T  0879:2
          P4 ← TALLY;   COMMENT A "." WAS FOUND FIRST, SKIP%                   00672200 T  0879:3
                        THE "." AND SET FLAG TO 6;%                            00672300 T  0880:0
          TALLY ← 6;%                                                          00672400 T  0880:0
          GO TO RXITA;%                                                        00672500 T  0880:1
     COMMENT EXPONENT FOUND FIRST;%                                            00672600 T  0880:2
EXPFRST:  TALLY ← TALLY +63;%                                                  00672700 T  0880:2
          SI ← SI+1;%                                                          00672800 T  0880:3
          P4 ← TALLY;%                                                         00672900 T  0881:0
          TALLY ← 8;    COMMENT SET FLAG TO 8;%                                00673000 T  0881:1
          GO TO RXITA;%                                                        00673100 T  0881:2
     COMMENT LOOK AT FIRST NON-BLANK CHARACTER;%                               00673200 T  0881:3
RSIGN:    IF SC="-" THEN GO TO RMINUS;%                                        00673300 T  0881:3
          IF SC="+" THEN GO TO RPLUS;%                                         00673400 T  0882:2
          IF SC="&" THEN GO TO RPLUS;%                                         00673500 T  0883:1
RXITB:    IF SC≥"0" THEN GO TO RIMPLUS;%                                       00673600 T  0884:0
          IF SC="." THEN GO TO NOI;%                                           00673700 T  0884:3
          IF SC="E" THEN GO TO EXPFRST;%                                       00673800 T  0885:2
          IF SC="@" THEN GO TO EXPFRST;%                                       00673900 T  0886:1
     COMMENT IF NONE OF THE ABOVE THEN ERROR;%                                 00674000 T  0887:0
RERR:     TALLY ← 2;%                                                          00674100 T  0887:0
          GO TO RXITA;%                                                        00674200 T  0887:1
RMINUS:   DI ← LOC P4;%                                                        00674300 T  0887:2
          DI ← DI-1;    COMMENT PASS BACK A "1" FOR A"-";%                    00674400 T  0887:3
          DS ← LIT "1";%                                                       00674500 T  0888:0
RPLUS:    TALLY ← TALLY+63;%                                                   00674600 T  0888:2
          SI←SI+1;%                                                            00674700 T  0888:3
          P4←TALLY;%                                                           00674800 T  0889:0
     COMMENT SKIP BLANKS PAST SIGN (IF ANY) THEN LOOK AT%                      00674900 T  0889:1
             NEXT NON-BLANK;%                                                  00675000 T  0889:1
     P4(IF SC≠" " THEN JUMP OUT TO RXITB;%                                     00675100 T  0889:1
        SI←SI+1;%                                                              00675200 T  0890:3
        TALLY ← TALLY + 63);%                                                  00675300 T  0891:0
        GO TO RERR;%                                                           00675400 T  0891:2
RIMPLUS:  P4 ← TALLY;%                                                         00675500 T  0891:3
          TALLY ← 0;%                                                          00675600 T  0892:0
RXITA:    P3 ← SI;%                                                            00675700 T  0892:1
          P6 ← TALLY;%                                                         00675800 T  0892:2
     END;%                                                                     00675900 T  0892:3
     BUFF ← P;      COMMENT ADDRESS OF NEXT CHARACTER;%                       00676000 T  0893:0
     WT ←   P;      COMMENT REMAINING FIELD;%                                 00676100 T  0893:2
```

```
                SGN ← P;        COMMENT SAVE SIGN;%                                    00676200 T  0894:0
                GO TO P;        COMMENT SWITCH ON KEY;%                                00676300 T  0894:2
                GO TO RIPART;   COMMENT KEY = 0 = <SIGN><DIGIT> OR <DIGIT>;            00676400 T  0894:3
                GO TO RERRA;    COMMENT KEY = 2 = ERROR;%                              00676500 T  0895:1
                GO TO RBLF;     COMMENT KEY = 4 = BLANK FIELD;%                        00676600 T  0895:3
                GO TO RFA;      COMMENT KEY = 6 = <SIGN>"." OR ".";%                   00676700 T  0896:1
                     COMMENT FALL THRU FOR KEY = 8 =<SIGN> <EPONENT> OR%               00676800 T  0896:3
                               <EXPONENT>;%                                            00676900 T  0896:3
                JUNK1 ← 1;      COMMENT MANTISSA ← 1;%                                 00677000 T  0896:3
                W1 ← 0;         COMMENT 0 DECIMAL PLACES;%                             00677100 T  0897:2
                GO TO REXP;     COMMENT OUT TO DEVELOP EXPONENT;%                      00677200 T  0898:1
                     COMMENT BLANK FIELD;%                                             00677300 T  0898:3
RBLF:           P(0,SSN);       COMMENT SET RESULT TO =0;%                             00677400 T  0898:3
                GO TO COMB;%                                                           00677500 T  0899:1
                     COMMENT "." FOUND FIRST;%                                         00677600 T  0899:3
RFA:            JUNK1 ← 0;      COMMENT MANTISSA ← 0;%                                 00677700 T  0899:3
                FLG ← 1  ;      COMMENT SET FLG TO REMEMBER NO INTEGER%                00677800 T  0900:2
                                        PART;%                                         00677900 T  0901:1
                GO TO RFPART;%                                                         00678000 T  0901:1
                     COMMENT DIGIT FOUND FIRST;%                                       00678100 T  0901:3
RIPART:         P(1);           COMMENT CALL GETNUM TO BUILD OCTAL%                    00678200 T  0901:3
                                        INTEGER PART;%                                 00678300 T  0902:0
                GO TO GETNUM;%                                                         00678400 T  0902:0
RIPRTN:         IF NOT P THEN GO TO RFC; COMMENT BRANCH ON KEY GETNUM%                 00678500 T  0902:2
                     RETURNS. IF NO BRANCH THEN WE HAVE FIELD EXAUSTED.%               00678600 T  0903:0
                     I.E. IMPLIED DECIMAL;%                                            00678700 T  0903:0
                W1 ← D;         COMMENT  DECIMAL PLACES IN FRACTION;%                  00678800 T  0903:0
RDONA:          W2 ← 0;         COMMENT  NO EXPONENT;%                                 00678900 T  0903:3
                     COMMENT BUILD RESULT;%                                            00679000 T  0904:2
RDONE:          P(JUNK1);       COMMENT GET NUMBER;%                                   00679100 T  0904:2
                IF SGN THEN P(SSN);  COMMENT INSERT SIGN;%                             00679200 T  0904:3
                               COMMENT SCALE NUMBER;%                                  00679300 T  0905:3
                IF P(W2 + SCFTR-W1,DUP) ≥ 0%                                           00679400 T  0905:3
                     THEN P(TEN[P],MUL)%                                               00679500 T  0907:2
                     ELSE P(TEN[-P],/);%                                               00679600 T  0908:3
                GO TO COMB;%                                                           00679700 T  0910:0
MAXI::: @7777777777777;                                                                00679750 T  0910:2
                COMMENT THE FIELD IS NOT EXAUSTED;%                                    00679800 T  0912:0
RFC:            WT ← WT -1;     COMMENT WT-1 TO ACCOUNT FOR CHARACTER%                 00679900 T  0912:0
                                        ENDING INTEGER FIELD;%                         00680000 T  0913:1
                IF W2 = "." THEN GO TO RFPART;%                                        00680100 T  0913:1
                               COMMENT OUT FOR VISABLE DECIMAL POINT;%                 00680200 T  0914:2
                IF WT ≥ D THEN GO TO RERRA;%                                           00680300 T  0914:2
                               COMMENT ERROR IF IMPLIED POINT TO RIGHT%                00680400 T  0915:3
                                        OF MANTISSA FIELD;%                            00680500 T  0915:3
                W1 ← D-WT-1;    COMMENT CALCULATE DECIMAL POSITION FROM%               00680600 T  0915:3
                               DECIMAL PLACES AND POSITION OF RIGT MOST%               00680700 T  0917:2
                               DIGIT IN MANTISSA;%                                     00680800 T  0917:2
     COMMENT LOOK FOR AND CONVERT ANY EXPONENT FOUND;%                                 00680900 T  0917:2
REXP:           STREAM(%                                                               00681000 T  0917:2
                     P6 ← 0,          COMMENT PLACE TO RETURN EXPONENT;%             00681100 T  0917:3
                     P5←WT+1,         COMMENT REMAINING FIELD WIDTH;%                00681200 T  0918:0
                     P4← BUFF,%                                                        00681300 T  0918:3
                     P3←1:            COMMENT FLAG;%                                00681400 T  0919:0
                     P1←0);%                                                           00681500 T  0919:2
                BEGIN COMMENT LOOK FOR E OR @;%                                        00681600 T  0920:0
                     SI ← P4; SI ← SI -1;%                                             00681700 T  0920:0
```

```
     TALLY ← P5;%                                                    00681800 T  0920:2
     P5( IF SC≠ " " THEN JUMP OUT TO RAA;%                           00681900 T  0921:1
         SI ← SI + 1;%                                               00682000 T  0922:3
         TALLY ← TALLY + 63);COMMENT TALLY = 1 ;%                    00682100 T  0923:0
     GO TO REXTA;    COMMENT OUT IF NO EXPONENT;%                    00682200 T  0923:2
RAA: IF SC="E" THEN GO TO RAB;%                                      00682300 T  0923:3
     IF SC="@" THEN GO TO RAB;%                                      00682400 T  0924:2
RAER:TALLY ← 0;      COMMENT IMPROPER EXPONENT;%                     00682500 T  0925:1
     P3 ← TALLY;%                                                    00682600 T  0925:2
GO TO REXTA;%                                                        00682700 T  0925:3
          COMMENT LOOK FOR EXPONENT SIGN;%                           00682800 T  0926:0
RAB: TALLY ← TALLY + 63;%                                            00682900 T  0926:0
     SI ← SI + 1;%                                                   00683000 T  0926:1
     IF SC="-" THEN%                                                 00683100 T  0926:2
     BEGIN%                                                          00683200 T  0927:0
          P5 ← TALLY;%                                               00683300 T  0927:0
          TALLY ← 1;%                                                00683400 T  0927:1
          P1 ← TALLY;     COMMENT REMEMBER "-" SIGN;%                00683500 T  0927:2
          TALLY ← P5;%                                               00683600 T  0927:3
          GO TO REP;%                                                00683700 T  0928:2
     END;%                                                           00683800 T  0928:3
     IF SC="+" THEN%                                                 00683900 T  0928:3
     BEGIN%                                                          00684000 T  0929:1
     REP: TALLY ← TALLY + 63;%                                       00684100 T  0929:1
          P5←TALLY;                                                  00684200 T  0929:2
          SI ← SI + 1;    COMMENT SKIP OVER SIGN;%                   00684300 T  0929:3
          P5(JUMP OUT TO RADC); COMMENT OUT IF FIELD NOT%            00684400 T  0930:0
                                EXAUSTED (P5≠0);%                    00684500 T  0931:1
          GO TO RAER;     COMMENT OUT ON ERROR;%                     00684600 T  0931:1
     REXTA: GO TO REXT;%                                             00684700 T  0931:2
     RAERA: GO TO RAER;%                                             00684800 T  0931:3
     END;%                                                           00684900 T  0932:0
     IF SC="&" THEN GO TO REP;%                                      00685000 T  0932:0
          COMMENT LOOK FOR DIGITS IN EXPONENT;%                      00685100 T  0932:3
RADC:  IF SC < "0" THEN GO TO RAER;%                                 00685200 T  0932:3
                          COMMENT OUT IF NOT DIGIT=ERROR;%           00685300 T  0933:2
     TALLY ← TALLY +63;%                                             00685400 T  0933:2
     P5 ← TALLY;%                                                    00685500 T  0933:3
          COMMENT LOOK FOR 2ND DIGIT;%                               00685600 T  0934:0
          P5(SI←SI+1;%                                               00685700 T  0934:0
             IF SC≥"0" THEN%                                         00685800 T  0934:3
               BEGIN%                                                00685900 T  0935:1
               SI←SI-1;%                                             00686000 T  0935:1
               DI ← LOC P6;%                                         00686100 T  0935:2
               DS ← 2 OCT;%                                          00686200 T  0935:3
               TALLY ← TALLY + 63;%                                  00686300 T  0936:0
               P5 ← TALLY;%                                          00686400 T  0936:1
               JUMP OUT TO RAIS;%                                    00686500 T  0936:2
               END;%                                                 00686600 T  0937:0
             IF SC≠" " THEN JUMP OUT TO RAER;%                       00686700 T  0937:0
             SI ← SI - 1; JUMP OUT TO RAC);%                         00686800 T  0938:0
  RAC:       DI ← LOC P6;%                                           00686900 T  0939:0
             DS ← OCT;%                                              00687000 T  0939:1
     COMMENT PUT IN EXPONENT SIGN SAVED IN P1;%                      00687100 T  0939:2
RAIS:  P1(DI ← LOC P6;%                                              00687200 T  0939:2
          DS ← LIT "+");%                                            00687300 T  0940:1
     P5( IF SC ≠ " " THEN JUMP OUT TO RAERA;%                        00687400 T  0941:0
```

```
                    SI ← SI + 1);%                                                  00687500 T  0942:2
          REXT:  P4 ← SI;%                                                          00687600 T  0943:0
            END;%                                                                   00687700 T  0943:1
            IF NOT  P THEN GO TO RERRA;  COMMENT OUT ON ERROR;%                     00687800 T  0943:2
            BUFF ← P;%                                                              00687900 T  0944:0
            P(DEL);%                                                                00688000 T  0944:2
            W2 ←P;          COMMENT EXPONENT;%                                      00688100 T  0944:3
            GO TO RDONE;%                                                           00688200 T  0945:1
                 COMMENT WE COME HERE IF A "." IS FOUND IN FIELD;%                  00688300 T  0945:3
          RFPART:   P(JUNK1,[ADDRS],STD);%                                          00688400 T  0945:3
                    COMMENT SAVE INTEGER PART IN ADDRS;%                            00688500 T  0946:2
            IF (JUNK2 ← WT) ≤ 0 THEN%                                               00688600 T  0946:2
                 BEGIN COMMENT "." WAS LAST IN FIELD;%                              00688700 T  0947:3
                      IF FLG THEN GO TO RERRA;%                                     00688800 T  0948:1
                           COMMENT ERROR IF ONLY A "." WAS FOUND;%                  00688900 T  0949:1
                      W1 ← 0;  COMMENT INDICATE NO FRACTION PART;%                  00689000 T  0949:1
                      GO TO RDONA;%                                                 00689100 T  0950:0
                 END;%                                                              00689200 T  0950:2
            P(0);%                                                                  00689300 T  0950:2
            GO TO GETNUM;  COMMENT CALL GETNUM TO BUILD FRACTION;%                  00689400 T  0950:3
          RFPRTN:   IF (W1 ← JUNK2 - WT) = 0 THEN%                                  00689500 T  0951:1
            BEGIN COMMENT FRACTION PART IS BLANK;%                                  00689600 T  0953:0
                 IF FLG THEN GO TO RERRA;%                                          00689700 T  0953:2
                           COMMENT ERROR IF ONLY "." IN FIELD;%                     00689800 T  0954:2
            END;%                                                                   00689900 T  0954:2
                 COMMENT DEVELOP NUMBER;%                                           00690000 T  0954:2
            JUNK1 ← JUNK1 + ADDRS × TEN[W1];%                                       00690100 T  0954:2
                           COMMENT INTEGER PART × 10@<DECIMAL PLACES>               00690200 T  0956:2
                                + FRACTION PART;%                                   00690300 T  0956:2
            IF P THEN GO TO RDONA;%                                                 00690400 T  0956:2
                           COMMENT BRANCH ON KEY GETNUM RETURNED.%                  00690500 T  0957:1
                                IF TRUE THEN FIELD EXAUSTED;%                       00690600 T  0957:1
            WT ← WT -1;    COMMENT WT-1 TO ACCOUNT FOR CHARACTER%                   00690700 T  0957:1
                           ENDING FRACTION PART;%                                   00690800 T  0958:2
            GO TO REXP;    COMMENT CHECK FOR EXPONENT;%                             00690900 T  0958:2
        COMMENT SUB-PROGRAM USED BY R-TYPE;%                                        00691000 T  0959:0
            COMMENT GETNUM BUILDS AN OCTAL INTEGER FROM THE BCL%                    00691100 T  0959:0
            FOUND IN THE BUFFER;%                                                   00691200 T  0959:0
          GETNUM::  P(1);               COMMENT FLAG USED AT GRTN;%                 00691300 T  0959:0
          GRTY:     STREAM(%                                                        00691400 T  0959:1
                      P6←0,             COMMENT RETURN CHR ENDING INTEGER;%        00691500 T  0959:2
                      P5←[D1],          COMMENT POINTER TO BCL INTEGER;%           00691600 T  0959:3
                      P4←(IF WT > 16 THEN 16 ELSE WT),%                             00691700 T  0960:0
                                        COMMENT WT = FIELD WIDTH;%                  00691800 T  0962:1
                      P3←BUFF:%                                                     00691900 T  0962:1
                      P1← 0);%                                                      00692000 T  0962:3
                 BEGIN%                                                             00692100 T  0963:1
                      SI ← P3;%                                                     00692200 T  0963:1
                      DI ← P5;%                                                     00692300 T  0963:2
                      P4(IF SC < "0" THEN JUMP OUT  TO RENDM;%                      00692400 T  0963:3
                         DS ← CHR;%                                                 00692500 T  0965:1
                         TALLY ← TALLY +1);%                                        00692600 T  0965:2
                      GO TO RCXIT;%                                                 00692700 T  0966:0
                 RENDM:  DI ← LOC P5;%                                              00692800 T  0966:1
                      DI ← DI-1;%                                                   00692900 T  0966:2
                      DS ← CHR;         COMMENT RETURN CHARACTER ENDING%            00693000 T  0966:3
                                             INTEGER FIELD;%                        00693100 T  0967:0
```

```
RCXIT:    P3 ← SI;          COMMENT NEXT BUFF ADDRESS;%                  00693200 T  0967:0
          P4 ← TALLY;       COMMENT RETURN NUMBER OF DIGITS%             00693300 T  0967:1
                                    IN INTEGER;%                         00693400 T  0967:2
     END;%                                                               00693500 T  0967:2
          BUFF ← P;         COMMENT BUFF ← P3;%                          00693600 T  0967:3
          W1 ← P;           COMMENT W1   ← P4;%                          00693700 T  0968:1
          P(DEL);           COMMENT DELETE P5;%                          00693800 T  0968:3
          W2 ← P;           COMMENT W2 ←   P6;%                          00693900 T  0969:0
GRTN:     IF NOT P THEN GO TO GTD; COMMENT BRANCH ON FLAG PUT%           00694000 T  0969:2
                                    IN AT GRTY OR GTC;%                  00694100 T  0970:0
     STREAM(%                                                            00694200 T  0970:0
          P7 ← P(W1,DUP),%                                               00694300 T  0970:1
          P6 ← (IF P ≤ 8 THEN 0 ELSE P(8,SUB,8,XCH)),%                   00694400 T  0970:3
             COMMENT THE ABOVE IS "IF WT ≤ 8 THEN P7←WT,P6←0              00694500 T  0973:2
                                    ELSE P7←8,P6←WT-8";%                 00694600 T  0973:2
          P5 ←0,            COMMENT OCTAL OF RIGHT 8 DIGITS;%            00694700 T  0973:2
          P4 ← 0,           COMMENT OCTAL OF WHATS LEFT;%                00694800 T  0973:3
          P3 ← [D1]:        COMMENT ADDRESS OF BCL INTEGER;%             00694900 T  0974:0
          P1 ← 0);%                                                      00695000 T  0974:2
     BEGIN%                                                              00695100 T  0975:0
          SI← P3;%                                                       00695200 T  0975:0
          DI← LOC P4;%                                                   00695300 T  0975:1
          DS← P6 OCT;%                                                   00695400 T  0975:2
          DI← LOC P5;%                                                   00695500 T  0976:0
          DS ← P7 OCT;%                                                  00695600 T  0976:1
     END;%                                                               00695700 T  0976:3
          P(DEL);           COMMENT DELETE P3;%                          00695800 T  0977:0
          P(TEN8,MUL,ADD,,JUNK1,STD);%                                   00695900 T  0977:1
               COMMENT  JUNK1 ← P4 × 10*8 + P5;%                         00696000 T  0978:2
          P(DEL,DEL);    COMMENT DELETE P6 & P7;%                        00696100 T  0978:2
GTB:      IF (WT ← WT-W1) ≤ 0 THEN%                                      00696200 T  0979:0
               BEGIN COMMENT PASS BACK A KEY OF 1 TO%                    00696300 T  0980:3
                                    FLAG FIELD EXAUSTED;%                00696400 T  0981:1
               P(1,XCH);%                                                00696500 T  0981:1
               GO TO NUMXIT%                                             00696600 T  0981:3
               END;%                                                     00696700 T  0982:1
          COMMENT FIELD NOT EXAUSTED SO LOOK AT WHAT ENDED IT;           00696800 T  0984:0
          IF W2 > 9  THEN%                                               00696900 T  0984:0
               BEGIN COMMENT MANTISSA EXAUSTED BUT NOT FIELD%            00697000 T  0984:3
                               SO RETURN A FLAG OF 0;%                   00697100 T  0985:1
               P(0,XCH);%                                                00697200 T  0985:1
               GO TO NUMXIT;%                                            00697300 T  0985:3
               END;%                                                     00697400 T  0986:1
GTC:      P(0); GO TO GRTY;  COMMENT MANTISSA NOT EXAUSTED,%             00697500 T  0986:1
                               SCALE NUMBER LEFT UNTIL IT IS;%           00697600 T  0987:0
GTD:      JUNK1 ← JUNK1 × TEN[W1];%                                      00697700 T  0987:0
          GO TO GTB;%                                                    00697800 T  0988:2
NUMXIT:   IF P THEN GO TO RIPRTN ELSE GO TO RFPRTN;%                     00697900 T  0989:0
          COMMENT DATA ERROR READING R FORMAT;%                          00698000 T  0990:0
RERRA:    IF FILX.[18:15]>1 THEN                                         00698100 T  0990:0
          BEGIN PARL←FILX[NOT 3];  FILX[NOT 3]←FILX[NOT 4]←0 END         00698150 T  0991:1
          ELSE BEGIN                                                     00698200 T  0997:0
               IF FILX.[18:15]=1 THEN P(FILX,14,COM);  PARL←0;  FILX←*2 ; 00698300 T  0997:2
               END ;                                                     00698400 T  1001:1
               IF PARL = 0 THEN P(FILX.[33:15],7,11,COM)                 00698500 T  1002:0
                           ELSE P(PARL,MKS,9,BLKCNTL);%                  00698600 T  1004:3
               COMMENT IF NO PARITY ACTION LABEL PRINT "RER"%             00698700 T  1006:1
```

```
                    ERROR AND TERMINATE ELSE GO TO PARITY%                    00698800 T  1006:1
                    LABEL;%                                                   00698900 T  1006:1
          END INPUTINT;%                                                      00699000 T  1006:1
                                                                                     SIZE= 1007 WORDS


PROCEDURE DISKSORT(                                                           00700000 T  0000:0
                                                        START OF REL SEGMENT; DISK ADDRESS = 00145
                    T1,T2,RELA,                                               00700100 T  0000:0
                    ENDQ,BINGO,IPFIDX,OUTPRO,INPRO,OUTF,INF,                  00700200 T  0000:0
                    OPTOG,IPTOG,DKO,DKI,TP1,TP2,TP3,TP4,TP5,NT,               00700300 T  0000:0
                    HIVALU,EQUALS,R,ALFA,CORESIZE,DISKSIZE);                  00700400 T  0000:0
     COMMENT DISK-SORT BY L.R. GUCK DATE 9/19/1965 ;                          00700500 T  0000:0
          VALUE     OPTOG,IPTOG,NT,HIVALU,EQUALS,R,ALFA,                      00700600 T  0000:0
                    CORESIZE,DISKSIZE;                                        00700700 T  0000:0
     REAL ENDQ,                                                               00700800 T  0000:0
          BINGO,                                                              00700900 T  0000:0
          IPFIDX,                                                             00701000 T  0000:0
          OUTPRO,                                                             00701100 T  0000:0
          INPRO,                                                              00701200 T  0000:0
          OUTF,     % POINTER TO DESC WHICH DESCRIBES OUT AREA                 00701300 T  0000:0
          T1,                                                                 00701400 T  0000:0
          T2,                                                                 00701500 T  0000:0
          RELA,                                                               00701600 T  0000:0
          INF;      % POINTER TO DESC WHICH DESCRIBES INPUT AREA               00701700 T  0000:0
   BOOLEAN OPTOG,   % TRUE IF OUTPUT PROCEDURE                                00701800 T  0000:0
          IPTOG;    % TRUE IF INPUT PROCEDURE                                 00701900 T  0000:0
      REAL DKO,     % DISK OUTPUT FILE                                        00702000 T  0000:0
          DKI;      % DISK INPUT FILE                                         00702100 T  0000:0
     NAME TP1,TP2,TP3,TP4,TP5;   % SCRATCH TAPES                             00702200 T  0000:0
     REAL NT,       % FOR FURTURE USE                                         00702300 T  0000:0
          HIVALU,                                                             00702400 T  0000:0
          EQUALS;   % KEY COMPARE ROUTINE                                     00702500 T  0000:0
  INTEGER R;        % RECORD: <0 FOR ALGOL                                    00702600 T  0000:0
   BOOLEAN ALFA;    % TRUE FOR ALPHA KEYS                                     00702700 T  0000:0
     REAL CORESIZE; % CORE STORAGE AVAILABLE                                 00702800 T  0000:0
  INTEGER DISKSIZE; % DISK STORAGE AVAILABLE                                 00702900 T  0000:0
     BEGIN                                                                    00703000 T  0000:0
    LABEL GRA,RTNRD,WRTBLOC,RTNDW,SA,RTNDR,                                   00703100 T  0000:0
          IPB,IPBA,IPC,IPD,IPE,IPG,                                           00703200 T  0000:0
          MIC,MID,MIE,RTA,                                                    00703300 T  0000:0
          START,LY,LZ,LX,CALLSORT,ENDSORTPASS,                                00703400 T  0000:0
          DKC,DKD,DKE,DKF,                                                    00703500 T  0000:0
          TPA,TPB,TPC,WRAPUP,SORTDONE;                                        00703600 T  0000:0
     COMMENT GENERAL PARAMETERS;                                              00703700 T  0000:0
     REAL S,        % MATRIX SIZE FOR SORT PASS                               00703800 T  0000:0
          M,        % MATRIX SIZE FOR MERGE PASS                              00703900 T  0000:0
         MS,        % CURRENT MATRIX SIZE                                     00704000 T  0000:0
        STPP,       % INDEX OF LAST ADDRESS IN VECTOR (V) ARRAY               00704100 T  0000:0
          D,        % SEGMENTS PER DISK INPUT BLOCK                           00704200 T  0000:0
          OD,       % SEGMENTS PER DISK OUTPUT BLOCK                          00704300 T  0000:0
          BF,       % RECORDS PER DISK INPUT BLOCK                            00704400 T  0000:0
         TBO,       % RECORDS PER DISK OUTPUT BLOCK & TAPE BLOCKING           00704500 T  0000:0
          I,X,Y;    % TEMPORARY STORAGE                                       00704600 T  0000:0
     ARRAY DATA[*,*]; ARRAY DATX = DATA[*]; NAME DATN = DATA;                 00704700 T  0000:0
```

```
ARRAY V[*]; NAME VN = V;                                                    00704800 T  0000:0
DEFINE VX1=FLAG(V[X+1])#, VX=FLAG(V[X])#, VL=FLAG(V[VLOW])#;                00704900 T  0000:0
DEFINE     VA1 = FLAG(V[X+1 ]&P(0,RDS)[CTF])#,                              00704910 T  0000:0
           VA  = FLAG(V[X   ]&P(0,RDS)[CTF])#,                              00704920 T  0000:0
           XAL= *[INFIL],0,RDS,CFX#,                                        00704925 T  0000:0
           VAL = FLAG(V[VLOW]&P(0,RDS)[CTF])#;                              00704930 T  0000:0
REAL VLOW;      % INDEX OF NEXT RECORD IN SEQUENCE                          00705000 T  0000:0
ARRAY MHK[*];  % HIGH KEY FOR MERGE PHASE                                   00705100 T  0000:0
NAME  MHN=MHK;                                                              00705200 T  0000:0
BOOLEAN MOREDATA, % GOES FALSE WHEN NO MORE INPUT DATA                      00705300 T  0000:0
        FM,       % TRUE ON LAST MERGE PASS                                 00705400 T  0000:0
       EOF,       % TRUE WHEN INPUT FILE EXAUSTED                           00705500 T  0000:0
        TM,    % TRUE FOR SORT WITH BACK-UP TAPES                            00705600 T  0000:0
        MF=T1, % TRUE IF MERGE ONLY                                         00705700 T  0000:0
        DF=IPTOG,  % TRUE IF OUTPUT FILE IS A DISK                          00705800 T  0000:0
DISKFULL;       % TRUE WHEN ASSIGNED DISK SPACE IS FULL                     00705900 T  0000:0
REAL TR;        % # OF RECORDS OF DATA SAVED ON DISK                        00706000 T  0000:0
DEFINE IOC = @2000000000#,                                                  00706100 T  0000:0
     POLYMERGE = PRTBASE[RELA] ← P(DUP,LOD)&1[6:47:1]#;                     00706200 T  0000:0
COMMENT PARAMETERS RELATED TO PROGRAMMERS FILES;                            00706300 T  0000:0
NAME  INFIL = INF;     % POINTER TO TOP I/O DESC.                            00706400 T  0000:0
NAME  WAIN = IPFIDX;  % COBOL68 INFILE WORK AREA                            00706420 T  0000:0
NAME OUTFIL = OUTF;                                                         00706500 T  0000:0
NAME  WAOUT = T2;      % COBOL68 OUTFILE WORK AREA                          00706520 T  0000:0
ARRAY PRFIB[*];        % CONTAINS TAPE FILES FIB                            00706600 T  0000:0
REAL AC;        % TRUE FOR COBOL INPUT FILE                                 00706700 T  0000:0
REAL INCOUNT,  % COUNTS # OF RECORDS FROM INPUT FILE                        00706800 T  0000:0
     OUTCOUNT; % COUNTS # OF RECORDS WRITTEN ON OUTPUT FILE                 00706900 T  0000:0
COMMENT   POINTERS FOR STANDARD PROCEDURES;                                 00707000 T  0000:0
NAME MEM   = 2;                                                             00707100 T  0000:0
ARRAY FPB = 3[*];                                                           00707110 T  0000:0
REAL BLOCK = 5,                                                             00707200 T  0000:0
     ALWR  = 12,                                                            00707300 T  0000:0
     ALRD  = 13,                                                            00707400 T  0000:0
     COFCR = 12,                                                            00707500 T  0000:0
     PERFORMGEN   =   13,     % COBOL68 IN-OUT PROCEDURES                   00707510 T  0000:0
     CORW  = 14,                                                            00707600 T  0000:0
ALFCR = 14,                                                                 00707700 T  0000:0
     BLKCTR = 16;                                                           00707800 T  0000:0
ARRAY PRTBASE = 10[*];                                                      00707900 T  0000:0
COMMENT PARAMETERS RELATED TO DISK OUTPUT FILE;                             00708000 T  0000:0
NAME DOTOP = DKO;   % POINTER TO DISK OUTPUT I/O DESC.                      00708100 T  0000:0
ARRAY OUTFIB[*]; % POINTER TO FIB                                           00708200 T  0000:0
ARRAY OUTHEAD[*]; % POINTER TO FILE HEADER BLOCK                           00708300 T  0000:0
REAL  LOSA,     % DISK ADDRESS OF CURRENT STRING TAG WORD                   00708400 T  0000:0
       ONS,     % RUNNING COUNT OF STRINGS IN OUTPUT AREA                   00708500 T  0000:0
       ORC,     % RUNNING COUNT OF RECORDS IN STRING                        00708600 T  0000:0
      OCDA,     % DISK ADDRESS OF NEXT AVAILABLE OUTPUT AREA                00708700 T  0000:0
       ORL,     % RUNNING COUNT OF NUMBER OF SEGMENTS LEFT IN               00708800 T  0000:0
                % CURRENT ROW                                               00708900 T  0000:0
       ORI,     % CURRENT ROW BEING USED                                    00709000 T  0000:0
       SRI,     % ROW WHERE STRING STARTED                                  00709100 T  0000:0
       SRS,     % NUMBER OF SEGMENTS  OF STRING IN ROW SRI                  00709200 T  0000:0
       OBC;     % CURRENT # OF RECORDS IN OUTPUT BUFFER                     00709300 T  0000:0
DEFINE ORS= OUTHEAD[8]#;                                                    00709400 T  0000:0
DEFINE FNUM = OUTFIB[4].[13:11]#;                                           00709410 T  0000:0
COMMENT PARAMETERS FOR DISK INPUT FILE;                                     00709500 T  0000:0
```

```
ARRAY ITNK = DKI[*];  % POINTER TO INPUT TANK                                  00709600 T  0000:0
ARRAY INFIB[*], % POINTER TO FIB                                               00709700 T  0000:0
      INHEAD[*];  % POINTER TO FILE HEADER BLOCK                              00709800 T  0000:0
ARRAY BASE[*],    % POINTER TO CONTROL INFO IN DATA                           00709900 T  0000:0
      ITOP[*],    % POINTER TO TOP I/O DESC                                    00710000 T  0000:0
     BUFF[*] ;      % I/O DESCRIPTOR                                          00710100 T  0000:0
REAL LISA;        % HOLDS TAG ADDRESS FOR NEXT MERGE PASS                      00710200 T  0000:0
DEFINE IBC = BASE[0]#,          % RECORDS LEFT IN BLOCK                        00710300 T  0000:0
       IRL = BASE[1]#,          % RECORDS LEFT IN STRING                       00710400 T  0000:0
       ISL = BASE[2]#,          % BLOCKS LEFT IN ROW                           00710500 T  0000:0
       IDA = BASE[3]#,          % DISK ADDRESS OF NEXT BLOCK                   00710600 T  0000:0
       IRC = BASE[4]#,          % CURRENT ROW OF THIS STRING                   00710700 T  0000:0
       FCR = IF AC THEN COFCR ELSE ALFCR#;                                     00710800 T  0000:0
COMMENT PARAMETERS RELATED TO MERGE TAPES;                                     00710900 T  0000:0
INTEGER   CTRL;     % CURRENT CONTROL TAPE                                     00711000 T  0000:0
INTEGER   COT;      % CURRENT OUTPUT TAPE                                      00711100 T  0000:0
NAME      COIOD;    % LOC OF I/O D OF CURRENT OUTPUT TAPE                       00711200 T  0000:0
NAME      TP;       % BASE POINTER OF MERGE TAPES                              00711300 T  0000:0
ARRAY     TS[*];    % ARRAYS FOR CONTROLLING DISTRIBUTION                      00711400 T  0000:0
ARRAY     TC[*];    %    PATTERNS ON MERGE TAPES                              00711500 T  0000:0
ARRAY     TN[*];                                                               00711600 T  0000:0
NAME      TSN=TS; NAME TCN=TC; NAME TNN=TN;                                    00711700 T  0000:0
REAL      TM1=CORESIZE; % TAPES - 1                                            00711800 T  0000:0
NAME      CIIOD;    % LOC OF I/O D FOR CURRENT INPUT TAPE                       00711900 T  0000:0
                  %**********************************%                         00712000 T  0000:0
SUBROUTINE WAIT; COMMENT WAIT FOR I/O COMPLETE USING ADRRESS                   00712100 T  0000:0
                  ON TOP OF STACK;                                             00712200 T  0001:0
$ SET OMIT = NOT(TIMESHARING)                                                  00712250 T  0001:0
      BEGIN IF NOT (P(XCH,DUP,LOD)).[19:1] THEN P(IOC,36,COM,DEL);             00712252 T  0001:0
$ POP OMIT                                                                     00712253 T  0004:0
$ SET OMIT = TIMESHARING                                                       00712299 T  0004:0
      IF NOT P(LOD,DUP).[2:1] THEN  % CHECK FOR ERRORS                         00712340 T  0004:0
        IF NOT (P(DUP,DUP).[27:1] AND P(XCH).[7:1]) THEN                       00712350 T  0005:1
          P(1,XCH, MEM[P INX NOT 1] INX P(2,LNG,XCH), 72, 17, COM);            00712360 T  0008:0
      P(DEL); END WAIT;                                                        00712400 T  0012:1
                  %***************MM*****************%                        00712500 T  0014:0
SUBROUTINE RELEASETAPE;  % CALLS MCP TO WRITE OUT BUFFERS                      00712600 T  0014:0
      BEGIN                                                                    00712700 T  0014:0
       PRFIB[11] ← TBO;                                                        00712800 T  0014:0
       P(COIOD[0] ← FLAG(PRFIB[16]),COIOD,PRL,DEL);                            00712900 T  0015:1
RTA:   P(COIOD); WAIT;                                                         00713000 T  0017:2
       IF (*COIOD).[27:1] THEN  % REEL SWITCH                                  00713100 T  0019:0
         BEGIN                                                                 00713200 T  0020:0
           P(MKS,0,0,[COIOD[NOT 2]],6,FCR);                                    00713300 T  0020:2
           GO TO RTA;                                                          00713400 T  0024:1
         END;                                                                  00713500 T  0024:3
       COIOD[0] ← 1 INX FLAG(PRFIB[16] ← NFLAG(*COIOD));                       00713600 T  0024:3
      END RELEASETAPE;                                                         00713700 T  0027:3
SUBROUTINE TAPEWRITE;   % BLOCKS OUTPUT TAPES                                  00713800 T  0028:0
      BEGIN                                                                    00713900 T  0028:0
       PRFIB ← *[COIOD[NOT 2]];                                                00714000 T  0028:0
       PRFIB[9] ← PRFIB[9] + 1;  % RECORD COUNTER + 1                          00714100 T  0029:3
       IF (PRFIB[11] ← PRFIB[11] - 1) > 0 THEN   % BLOCK COUNTER               00714200 T  0031:3
           COIOD[0] ← R INX *COIOD                                             00714300 T  0034:1
        ELSE                                                                   00714400 T  0035:0
      BEGIN     % TIME FOR RELEASE                                             00714500 T  0036:1
        P(0,PRFIB[16] INX MEM,STD);  % ZERO CONTROL WORD IN BUFF[2]             00714600 T  0036:3
```

```
      RELEASETAPE;                                                                    00714700 T  0038:1
    END;                                                                              00714800 T  0039:0
    END TAPEWRITE;                                                                    00714900 T  0039:0
SUBROUTINE WRITESTOPPER;                                                              00715000 T  0039:1
    BEGIN   % WRITES END OF STRING OR DUMMY STRINGS                                   00715100 T  0040:0
     PRFIB ← *[COIOD[NOT 2]];                                                         00715200 T  0040:0
     X ← PRFIB[9]&(TBO-PRFIB[11])[18:33:15]&("DS")[3:33:15];                          00715300 T  0041:3
     P(X,PRFIB[16] INX MEM,STD);                                                      00715400 T  0045:0
     TN[COT] ← TN[COT] + 1;   % COUNT UP STRINGS FOR THIS TAPE                       00715500 T  0046:2
     RELEASETAPE;                                                                     00715600 T  0048:2
    PRFIB [9] ← 0 ;                              % ZERO OUT STRING CTR               00715610 T  0050:0
    END WRITESTOPPER;                                                                 00715700 T  0051:1
                  %************************************%                            00715800 T  0053:0
SUBROUTINE OPENOUT; % OPENS PROGRAMMERS OUTPUT TAPE                                   00715900 T  0053:0
    BEGIN                                                                             00716000 T  0053:0
    IF OPTOG THEN                                                                     00716100 T  0053:0
      BEGIN  P(MKS,[OUTFIL],R,1,1,1,BLOCK);                                           00716200 T  0053:1
      IF AC THEN                                                                      00716300 T  0055:2
      BEGIN  BINGO ← OUTPRO; ENDQ ← 0;                                                00716400 T  0055:3
      IF AC.[46:1] THEN          % COBOL68                                           00716430 T  0058:1
         BEGIN P(MKS,BINGO,0,PERFORMGEN);                                             00716440 T  0059:0
               COIOD ← [WAOUT];                                                       00716445 T  0060:2
               WAOUT ← P(*[OUTFIL],0,CDC);                                            00716450 T  0061:1
         END ELSE                                                                     00716460 T  0062:3
      P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC);                                  00716500 T  0062:3
    END END ELSE                                                                      00716600 T  0065:0
      BEGIN DF ← FALSE; PRFIB ← OUTFIL[NOT 2];                                        00716700 T  0065:0
      PRFIB[13].[27:1] ← 0;                                                           00716800 T  0068:0
      IF AC THEN                                                                      00716900 T  0070:2
      BEGIN                                                                           00717000 T  0070:3
         P(MKS,[OUTFIL[NOT 2]],3,COFCR);                                              00717100 T  0071:1
         DF ← PRFIB[4].[8:4] = 4;                                                     00717200 T  0073:0
         IF AC.[46:1] THEN              % COBOL68                                    00717240 T  0075:0
              BEGIN  COIOD ← [WAOUT];                                                 00717250 T  0075:3
                     WAOUT ← P(PRFIB[20].[FF],DUP,DIB 0,LOD,0,                        00717260 T  0077:0
                               CDC,DEL,DIB 0,LOD);                                    00717270 T  0079:1
              END;                                                                    00717280 T  0080:3
      END                                                                             00717300 T  0080:3
      ELSE BEGIN  P([OUTFIL],0,11,COM,DEL,DEL);                                       00717400 T  0080:3
              P(MKS,1,0,0,("R),[OUTFIL],ALWR,DEL); END;                               00717500 T  0082:3
      END                                                                             00717600 T  0085:0
    END OPENOUT;                                                                      00717700 T  0085:0
                  %************************************%                            00717800 T  0085:1
SUBROUTINE SETUPTAPES;   %  INITIALIZES TAPES FOR DISTRIBUTION PASS                  00717900 T  0085:1
    BEGIN                                                                             00718000 T  0086:0
    FOR I ← 0 STEP 1 UNTIL NT DO TS[I] ← TC[I] ← TN[I] ← 0;                           00718100 T  0086:0
    CTRL ← 0; TC[1] ← COT ← 1; TM1 ← NT-1;                                            00718200 T  0092:2
    TP ← ((NOT 5) INX [NT]); X ← TBO×R+1;                                             00718300 T  0096:1
    FOR I ← 1 STEP 1 UNTIL NT  DO                                                     00718400 T  0099:2
    BEGIN                                                                             00718500 T  0101:0
         COIOD ←  TP[I]; PRFIB ← *[COIOD[NOT 2]];                                     00718600 T  0101:0
         PRFIB[18] ← X&X[3:33:15]&X[18:33:15]&(I≠1)[1:47:1];                          00718700 T  0104:1
         PRFIB[13].[27:1] ← 0;     % SET TO OUTPUT                                   00718800 T  0108:2
         PRFIB[4].[7:1] ← ((AC AND 3) = 1); %COBOL61 TAPE SORT FLG                    00718900 T  0111:0
         PRFIB[9] ← 0; PRFIB[11] ← TBO;                                               00719000 T  0114:2
         IF (Y←PRFIB[4].[12:12]) < 1023 THEN                                          00719100 T  0117:0
    PRFIB[4] ← PRFIB[4]&((Y-1)×ETRLNG)[13:37:11]&1[12:47:1];                          00719200 T  0119:0
```

```
            IF I ≠ NT THEN P([COIOD],0,11,COM,DEL,DEL)  % OPEN TAPES          00719300 T  0124:0
                      ELSE PRFIB[18] ← ABS(PRFIB[18]);                        00719400 T  0126:3
            IF I ≠ NT THEN BEGIN P(COIOD); WAIT; END;                         00719500 T  0129:0
            IF I = 1 THEN COIOD[0] ← 1 INX *COIOD;                            00719600 T  0132:0
        END;                                                                  00719700 T  0134:3
        COIOD ←   IP[COT] ;                                                   00719800 T  0137:0
        END SETUPTAPES;                                                       00719900 T  0138:2
                        %**********************************%                  00720000 T  0138:3
    SUBROUTINE GETROW;  % GETS DISK SPACE FOR NEXT  ROW IN OUTPUT AREA        00720100 T  0138:3
        BEGIN ORL ← ORS;                                                      00720200 T  0139:0
          IF (ORI ← ORI + 1) ≥ 30 THEN   % DISK SCRATCH FILE IS FULL          00720300 T  0140:0
            BEGIN                                                             00720400 T  0141:3
    GRA:    IF NT < 3 THEN P(1,[DOTOP[NOT 2]],84,17,COM);                     00720500 T  0142:1
            IF NOT TM THEN SETUPTAPES;                                        00720600 T  0145:2
            TM ← DISKFULL ← TRUE;                                             00720700 T  0148:0
            END                                                               00720800 T  0149:1
                        ELSE                                                  00720900 T  0149:1
          IF (OCDA ← OUTHEAD[ORI]) = 0 THEN  % GET DISK  SPACE                00721000 T  0149:1
            BEGIN                                                             00721100 T  0151:1
            P(FPB[FNUM+3],FPB[FNUM],FPB[FNUM+1],ORI,                          00721110 T  0151:3
                 ,OUTHEAD,LOD,4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL);              00721120 T  0156:3
              IF (OCDA ← OUTHEAD[ORI]) = 0 THEN GO TO GRA; % NO DISK          00721200 T  0159:2
            END                                                               00721300 T  0161:2
        END GETROW;                                                           00721400 T  0161:2
    SUBROUTINE FORGETDISK;    % RETURNS DISK NO LONGER NEEDED                  00721500 T  0161:3
        BEGIN                                                                 00721600 T  0162:0
        PRFIB ← P(XCH); I ← 9;                                                00721700 T  0162:0
        WHILE (I←I+1) ≤ 29 DO IF PRFIB[I] ≠ 0                                 00721800 T  0163:2
         THEN P(I ,,PRFIB,LOD,24,COM,DEL,DEL);                               00721900 T  0166:1
        END FORGETDISK;                                                       00722000 T  0169:2
                        %XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX%                  00722100 T  0169:3
    SUBROUTINE INROWCHK;                                                      00722200 T  0169:3
        BEGIN                                                                 00722300 T  0170:0
         IF (ISL ← ISL - 1) ≤ 0 THEN                                          00722400 T  0170:0
           BEGIN                                                              00722500 T  0172:2
             ISL ← (ORS DIV OD ) × (OD  DIV  D);  % BLOCKS IN ROW             00722600 T  0173:0
             IF INHEAD[(IRC ← IRC + (IRC < 29))] ≠ 0                          00722700 T  0176:0
                THEN IDA ← INHEAD[IRC];                                       00722800 T  0179:0
           END                                                                00722900 T  0181:3
         ELSE IDA ← IDA + D;                                                  00723000 T  0181:3
        END;                                                                  00723100 T  0184:1
                        %**********************************%                  00723200 T  0184:2
    SUBROUTINE INREAD; COMMENT POINT INPUT BUFFER AT NEXT RECORD;             00723300 T  0184:2
        BEGIN                                                                 00723400 T  0185:0
        IF EOF THEN GO TO RTNRD;                                              00723500 T  0185:0
        INCOUNT ← INCOUNT + 1;                                                00723600 T  0186:0
        IF IPTOG THEN                                                         00723700 T  0187:1
           BEGIN IF AC THEN                                                   00723800 T  0187:2
          BEGIN  COMMENT CALL INPUT PROCEDURE;                                00723900 T  0188:1
             IF AC.[46:1] THEN P(MKS,BINGO,0,PERFORMGEN)  ELSE                00723980 T  0188:3
             P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC);                   00724000 T  0191:0
             EOF ← ENDQ;                                                      00724100 T  0193:1
          END ELSE EOF ← P(MKS,*[INFIL],0,INPRO);                             00724200 T  0194:0
           END                                                                00724300 T  0196:1
                   ELSE                                                       00724400 T  0196:1
          BEGIN                                                               00724500 T  0196:1
                   IF AC THEN  EOF ← P(MKS,R,[INFIL],0,CORW) % COBOL          00724600 T  0196:3
```

```
                    ELSE                                                    00724700 T  0198:3
                BEGIN   COMMENT ALGOL READ;                                 00724800 T  0199:1
                  P(MKS,0,0,[INFIL],ALRD);  % READ FILE                     00724900 T  0199:3
                  EOF ← P(MKS,0,3,[INFIL],ALRD) < 0;  % WAIT FOR I/O        00725000 T  0201:0
                END;                                                        00725100 T  0203:1
        END;                                                                00725200 T  0203:1
RTNRD:  IF EOF THEN V[VLOW] ← NFLAG(*[DATX[S]])&                            00725300 T  0203:1
                    S[18:33:15]&1[5:47:1];                                  00725400 T  0205:2
        END INREAD;                                                         00725500 T  0207:2
                    %**********************************%                   00725600 T  0207:3
SUBROUTINE DISKWRITE;  COMMENT BLOCKS OUTPUT BUFFER AND WRITES IT;          00725700 T  0207:3
     BEGIN                                                                  00725800 T  0208:0
     IF NOT P(XCH) THEN GO TO WRTBLOC;  % WRITE BLOCK IF TOS = FALSE        00725900 T  0208:0
     ORC ← ORC+1;   % RECORD COUNT +1;                                      00726000 T  0208:3
     IF (OBC ← OBC-1) = 0 THEN      % IF BUFFER EXAUSTED                   00726100 T  0210:0
     BEGIN COMMENT BUFFER IS FULL SO WRITE IT OUT;                          00726200 T  0211:3
WRTBLOC:  OBC ← TBO;                                                        00726300 T  0212:1
          STREAM(P1←OCDA,                                                   00726400 T  0213:0
             P2←FLAG(OUTFIB[16]));  % DISK ADDRESS TO BUFFER                00726500 T  0213:2
          BEGIN SI←LOC P1; DS ← 8 DEC END;                                  00726600 T  0214:2
          P(1 INX FLAG(OUTFIB[16]),[DOTOP],DUP,OUTFIB[16],SFB,XCH,          00726700 T  0215:1
                STD,PRL,DEL);     % CALL MCP TO REFILL BUFFER               00726800 T  0218:0
          IF (ORL←ORL-OD) ≥ OD THEN                                         00726900 T  0219:0
          OCDA ←OCDA+OD                                                     00727000 T  0220:3
               ELSE                                                         00727100 T  0221:2
          GETROW;  % GET SPACE AND ADDRESS OF NEXT ROW                      00727200 T  0222:2
          P([DOTOP]); WAIT;  % WAIT FOR I/O COMPLETE                        00727300 T  0224:0
          COMMENT ON I/O COMPLETE SAVE ORGINIAL I/O DESC. IN FIB;           00727400 T  0225:0
          OUTFIB[16].[33:15] ← (NOT 0) INX NFLAG(*[DOTOP]);                 00727500 T  0225:0
     END                                                                    00727600 T  0228:1
          ELSE                                                              00727700 T  0228:1
     DOTOP[0] ← R INX *[DOTOP];     % POINT AT NEXT RECORD                  00727800 T  0228:1
RTNDW:                                                                      00727900 T  0230:1
     END DISKWRITE;                                                         00728000 T  0230:1
                    %**********************************%                   00728100 T  0230:2
SUBROUTINE DIST;    % CALCULATES DISTRIBUTION PATTERNS FOR                  00728200 T  0230:2
     BEGIN          %     MERGE TAPES                                      00728300 T  0231:0
          CTRL ← (CTRL MOD TM1) + 1;                                        00728400 T  0231:0
          FOR I ← 1 STEP 1 UNTIL TM1 DO                                     00728500 T  0232:3
          BEGIN TS[I] ← TC[I];                                              00728600 T  0234:0
                IF I ≠ CTRL THEN TC[I] ← TC[I] + TC[CTRL];                  00728700 T  0235:2
          END;                                                              00728800 T  0239:0
     END DIST;                                                              00728900 T  0241:1
SUBROUTINE SELECT;       % SELECTS A MERGE TAPE TO WRITE A STRING ON         00729000 T  0241:2
     BEGIN                                                                  00729100 T  0242:0
     X ← COT;  % SAVE INDEX OF PRIOR TAPE                                   00729200 T  0242:0
SA:  COT ← COT + 1;                                                         00729300 T  0242:3
     IF COT = NT THEN                                                       00729400 T  0244:0
     BEGIN COT ← 1; DIST; END;                                              00729500 T  0244:3
     IF COT = CTRL THEN GO TO SA;                                           00729600 T  0247:0
     PRFIB ←  COIOD[NOT 2];                                                 00729700 T  0248:1
     COIOD[0] ← FLAG(PRFIB[16]);                                            00729800 T  0250:0
     IF COT ≠ X THEN P( ((NOT 2) INX TP[COT]) ,[COIOD[NOT 2]],              00729900 T  0251:1
              20,COM,DEL,DEL);                                              00730000 T  0255:1
     COIOD ←   TP[COT] ;                                                    00730100 T  0256:1
     COIOD[0] ← 1 INX *COIOD;                                               00730200 T  0257:3
     END SELECT;                                                            00730300 T  0259:1
```

```
                    %************************************%                00730400 T  0259:2
SUBROUTINE WRITETAG;                        % WRITE FRONT OF STRING TAG,   00730500 T  0259:2
     BEGIN                                  % DEVELOP ADDRESS OF NEXT     00730600 T  0260:0
     IF OBC ≠ TBO THEN                      %  STRING                      00730700 T  0260:0
     BEGIN P(0); DISKWRITE; END;            % WRITE OUT BUFFER             00730800 T  0260:3
     BUFF ← FLAG(OUTFIB[16])                                             00730900 T  0263:0
                    &30[8:38:10]            % SET TOP I/O DESCRIPTOR TO    00731000 T  0263:1
                    &1[27:42:6];            % 30 WORD, 1 SEGMENT WRITE.    00731100 T  0264:0
     STREAM(P1←LOSA,P2←  [BUFF[0]]);        % DISK ADDRESS OF TAG TO       00731200 T  0266:1
     BEGIN SI ←LOC P1; DS ← 8 DEC END;      % BUFFER[0].                   00731300 T  0267:2
     IF NOT DISKFULL THEN                                                00731400 T  0268:1
     IF ORL < OD + 1 THEN GETROW;           % GET SPACE FOR NEXT ROW       00731500 T  0268:3
     BUFF[1] ← SRI;                         % ROW WHERE STRING STSRTED     00731600 T  0272:0
     BUFF[2] ← ORC;                         % RECORDS/STRING.              00731700 T  0273:1
     BUFF[3] ← SRS;                         % AMOUNT OF STRING IN          00731800 T  0274:2
                                            % ROW WHERE STRING STARTED.   00731900 T  0275:3
     BUFF[4] ← LOSA ← IF MOREDATA           % ADDRESS OF NEXT TAG.        00732000 T  0275:3
       AND NOT DISKFULL THEN OCDA ELSE 0;    %   OR EOF FLAG              00732100 T  0276:1
     OCDA ← OCDA + 1;                       % SKIP OVER TAG ADDRESS,       00732200 T  0279:3
     DOTOP[0] ← P(.BUFF,LOD);               % WRITE TAG ON DISK            00732300 T  0281:0
     P(1 INX FLAG(OUTFIB[16]),[DOTOP],PRL,DEL);                          00732400 T  0282:0
     ONS ← ONS+1;                           % STRING COUNTER + 1.          00732500 T  0284:1
     SRI ← ORI;                             % SAVE WHERE NEXT ROW STARTS   00732600 T  0285:2
     SRS ← (ORL←ORL-1);                     % AMOUNT OF ROW LEFT           00732700 T  0286:1
     ORC ← 0;                               % RECORDS/STRING ← 0.          00732800 T  0288:0
     P([DOTOP]); WAIT;                      % WAIT FOR I/O COMPLETE        00732900 T  0288:3
     OUTFIB[16] ← (NOT 0) INX NFLAG(*[DOTOP]); % SAVE IOD IN FIB            00733000 T  0290:0
END WRITETAG;                                                            00733100 T  0292:2
                    %************************************%                00733200 T  0292:3
SUBROUTINE DISKREAD;                        % READS DISK ON DISK-TO-       00733300 T  0292:3
     BEGIN                                  % DISK MERGE PASSES.          00733400 T  0293:0
     BASE ← *[DATX[VLOW]];                  % POINT AT CURRENT STRING      00733500 T  0293:0
     IF IRL ≤ 0 THEN                        % IF IRL ≤ 0,ALL RECORDS       00733600 T  0294:1
          BEGIN V[VLOW] ← NFLAG(MHK)&MS     % IN THIS STRING HAVE BEEN    00733700 T  0295:1
                              [18:33:15];                                00733800 T  0297:0
               GO TO RTNDR;                 % READ SO POINT CORRESPON-     00733900 T  0298:0
          END;                              % ING V AT HK1.                00734000 T  0298:2
     IRL←IRL-1;                             % RECORDS LEFT - 1.            00734100 T  0298:2
     IF (IBC←IBC-1)≠0 THEN BEGIN            % IF BUFFER NOT EXAUSTED       00734200 T  0300:2
     V[VLOW] ← R INX V[VLOW];               % THEN INDEX TO NEXT RECORD    00734300 T  0303:2
     GO TO RTNDR END;                                                    00734400 T  0305:2
     IBC ← BF;                              % BLOCK COUNTER ← BLOCKING     00734500 T  0306:0
     Y ← P(VLOW,DUP,ADD);                   % FACTOR.                      00734600 T  0307:1
     STREAM(P1←IDA,P2←*[ITOP[Y]]);          % CONVERT DISK ADDRESS         00734700 T  0308:2
     BEGIN SI ← LOC P1;DS←8 DEC END;        %  INTO BUFFER.                00734800 T  0310:1
     P([ITOP[Y]],DUP,LOD,XCH,PRL,DEL);      % READ NEXT BLOCK              00734900 T  0311:0
     INROWCHK;                              % GET ADDRESS OF NEXT BLOCK    00735000 T  0313:0
     P([ITOP[Y]]); WAIT;                    % WAIT FOR I/O COMPLETE.       00735100 T  0314:0
          COMMENT POINT I/O D PAST DISK ADDRESS;                          00735400 T  0316:0
     V[VLOW] ← V[VLOW]&(1 INX (*[ITOP[Y]]))[33:33:15];                   00735500 T  0316:0
RTNDR: END DISK READ;                                                    00735600 T  0319:0
                    %************************************%                00735700 T  0319:1
SUBROUTINE WRITEOUT;                                                     00735800 T  0319:1
     BEGIN  %  SELECTS FILE TO BE WRITTEN DURING MERGE                   00735900 T  0320:0
      IF NOT FM THEN                                                     00736000 T  0320:0
        IF TM THEN TAPEWRITE ELSE BEGIN P(1); DISKWRITE END              00736100 T  0320:2
              ELSE                                                       00736200 T  0325:0
```

```
      BEGIN COMMENT CALL OUTPUT PROCEDURE OR WRITE INTRINSIC;                  00736300 T  0325:0
      OUTCOUNT ← OUTCOUNT +1;                                                  00736400 T  0325:2
      IF OPTOG THEN                                                            00736500 T  0326:3
         BEGIN                                                                 00736600 T  0327:0
         IF AC THEN                                                            00736700 T  0327:2
          BEGIN  ENDQ ← 0;  IF AC.[46:1] THEN                                  00736800 T  0327:3
                    P(MKS,BINGO,0,PERFORMGEN)     % COBOL68                  00736850 T  0330:0
               ELSE P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC)              00736900 T  0331:2
          END ELSE P(MKS,0,*[OUTFIL],0,OUTPRO);                                00737000 T  0333:3
         END                                                                   00737100 T  0335:3
              ELSE                                                             00737200 T  0335:3
         BEGIN COMMENT OUTPUT FILE RATHER THAN OUTPUT PROCEDURE;               00737300 T  0335:3
      IF AC THEN P(MKS,0,1,1,0,R,[OUTFIL],1,CORW)                              00737400 T  0336:1
              ELSE BEGIN                                                       00737500 T  0339:0
                   P(MKS,1,0,0,R,[OUTFIL],ALWR);                               00737600 T  0339:3
                   P(MKS,1,0,0,(-R),[OUTFIL],ALWR,DEL); END;                   00737700 T  0341:2
          IF DF THEN IF P THEN P(1,[OUTFIL[NOT 2]],83,17,COM);                 00737800 T  0343:3
         END;                                                                  00737900 T  0347:0
    END;                                                                       00738000 T  0347:0
  END WRITEOUT;                                                                00738100 T  0347:0
                  %*********************************%                         00738200 T  0347:1
SUBROUTINE SUBMERGE;     % SETS UP DISK INPUT TO START A MERGE PASS            00738300 T  0347:1
    BEGIN                                                                      00738400 T  0348:0
    FOR I ← 0 STEP 1 UNTIL (MS-1) DO                                           00738500 T  0348:0
     BEGIN                                                                     00738600 T  0352:1
      IF EOF THEN BEGIN                                                        00738700 T  0352:1
                   V[I] ← NFLAG(MHK)&MS[18:33:15];                             00738800 T  0353:0
                   P(0,*[DATX[I]],1,CDC,STD);   % IRL ← 0                    00738900 T  0355:1
                  END                                                          00739000 T  0357:0
           ELSE                                                                00739100 T  0357:0
BEGIN                                                                          00739200 T  0357:0
  BASE ← *[DATX[I]];  % POINT AT CONTROL INFO                                  00739300 T  0357:2
  Y ← P(I,DUP,ADD);    % Y ← 2×I                                               00739400 T  0358:3
  BUFF ← *[ITOP[Y]];                                                           00739500 T  0360:0
       COMMENT SET I/O D = 30 WORD, 1 SEGMENT READ;                            00739600 T  0361:1
  ITOP[Y] ← (*[ITOP[Y]])&30[8:38:10]&1[27:42:6];                               00739700 T  0361:1
       COMMENT PUT ADDRESS OF STRING TAG WORD IN BUFFER(0);                    00739800 T  0365:0
  STREAM(P1←( IDA←LISA),P2←*[ITOP[Y]]);                                        00739900 T  0365:0
       BEGIN SI ← LOC P1; DS ← 8 DEC END;                                      00740000 T  0367:2
  P(.BUFF,LOD, [ITOP[Y]],PRL,DEL); % READ TAG TO BUFFER 2                      00740100 T  0368:1
       COMMENT READ 1ST DATA RECORD TO BUFFER #2, TAG GETS                     00740200 T  0370:0
               ROTATED TO BUFFER #1;                                           00740300 T  0370:0
  P([ITOP[Y]]); WAIT;  % WAIT FOR I/O COMPLETE                                 00740400 T  0370:0
  STREAM(P1←( IDA←IDA +1),P2←*[ITOP[Y]]); % BLOCK #1 ADDRESS                   00740500 T  0372:0
       BEGIN SI ← LOC P1; DS ← 8 DEC END;                                      00740600 T  0375:1
  P(*[ITOP[Y]],[ITOP[Y]],PRL,DEL); % READ BLOCK #1                            00740700 T  0376:0
  P([ITOP[Y]]); WAIT;  % WAIT FOR I/O  ON READING TAG                          00740800 T  0378:0
  BUFF ← *[ITOP[Y]];                                                           00740900 T  0380:0
  IBC ← BF;                                                                    00741000 T  0381:1
  IRC ← BUFF[1];                                                               00741100 T  0382:2
  IRL ← BUFF[2] - 1;                                                           00741200 T  0384:0
  ISL ← (BUFF[3] DIV OD) × (OD DIV D);                                         00741300 T  0386:0
  IF (LISA ← BUFF[4]) ≤ 0 THEN EOF ← TRUE;                                     00741400 T  0389:0
  INROWCHK;   % GET ADDRESS OF BLOCK #2                                        00741500 T  0391:3
  STREAM(P1←IDA ,P2←[BUFF[0]]);                                                00741600 T  0393:0
       BEGIN SI←LOC P1; DS ← 8 DEC END;                                        00741700 T  0394:2
  P(.BUFF,LOD,[ITOP[Y]],PRL,DEL); % READ BLOCK #2                              00741800 T  0395:1
```

```
          INROWCHK;    % GET ADDRESS OF BLOCK #3                                    00741900 T  0397:0
            P([ITOP[Y]]); WAIT;  % WAIT FOR I/O COMPLETE ON BLOCK #1                00742000 T  0398:0
            V[I] ← NFLAG((1 INX *[ITOP[Y]])&I[18:33:15]);                           00742100 T  0400:0
          END;                                                                      00742200 T  0403:0
        END;                                                                        00742300 T  0403:0
      END SUBMERGE;                                                                 00742400 T  0403:2
                        %*********************************%                       00742500 T  0403:3
SUBROUTINE FIRSTSELECT;   % INITIAL SELECTION OF LOW RECORD                        00742600 T  0403:3
      BEGIN                                                                         00742700 T  0404:0
      X ← 0; I←MS-1;                                                                00742800 T  0404:0
      DO BEGIN                                                                      00742900 T  0406:0
      I←I+1;                                                                        00743000 T  0406:0
      V[I] ← V[X+((IF ALFA THEN P(0,MKS,0,VX1,VX,EQUALS)                            00743100 T  0407:1
              ELSE IF AC THEN  P(0,MKS,VX1,VX,EQUALS)                           00743200 T  0411:2
          ELSE P(MKS,VA1,0,VA,0,EQUALS)) AND TRUE)];                            00743300 T  0415:2
        END UNTIL (X←X+2) = STPP; VLOW ← V[I].[18:15];                          00743400 T  0422:3
      END FIRSTSELECT;                                                          00743500 T  0426:3
SUBROUTINE LOWSELECT;                                                           00743600 T  0427:0
      BEGIN                                                                     00743700 T  0427:0
      X ← VLOW AND 1022;                                                        00743800 T  0427:0
      DO BEGIN                                                                  00743900 T  0428:1
      I ← MS + X.[38:9];  % I ] MS + (X/2)                                      00744000 T  0428:1
      V[I] ← V[X+((IF ALFA THEN P(0,MKS,0,VX1,VX,EQUALS)                        00744100 T  0430:0
              ELSE IF AC THEN P(0,MKS,VX1,VX,EQUALS)                            00744200 T  0434:1
          ELSE P(MKS,VA1,0,VA,0,EQUALS)) AND TRUE)];                            00744300 T  0438:1
      X ← I AND 1022;                                                           00744400 T  0445:2
        END UNTIL I = STPP; VLOW ← V[I].[18:15];                                00744500 T  0446:3
      END LOWSELECT;                                                            00744600 T  0449:3
                        %*********************************%                       00744700 T  0450:0
SUBROUTINE SORTIT;  % DEVELOPS STRINGS FROM INPUT TAPE                          00744800 T  0450:0
      BEGIN                                                                     00744900 T  0450:0
      COMMENT USE SPECIAL SORT COMMUNICATE TO GET STORAGE FOR                   00745000 T  0450:0
          DATA AND VECTOR ARRAYS;                                               00745100 T  0450:0
      P(R,S+1,[DATN],21,COM,DEL,DEL,DEL);                                       00745200 T  0450:0
      P(MKS,[VN],(2×S)-1,1,1,1,BLOCK);                                          00745300 T  0452:2
      STPP ← P(MS←S,DUP,ADD,2,SUB);                                             00745400 T  0455:1
      COMMENT CALL HIVALU TO SET UP HK1 ROW OF DATA;                             00745500 T  0457:2
      STREAM(A←*[DATX[S]],B←R-1,C←P(DUP).[36:6]);                               00745600 T  0457:2
      BEGIN DI←A;SI←LOC C;DS←WDS;                                               00745700 T  0460:1
            SI←A;C(DS←32 WDS;DS←32 WDS);DS←B WDS;                               00745800 T  0461:0
      END;                                                                      00745900 T  0463:0
      P(MKS,*[DATX[S]]); IF NOT AC THEN P(0,RDS,CFX,0); P(HIVALU);              00746000 T  0463:1
      COMMENT INITIAL FILL OF DATA ARRAY FROM INPUT SOURCE;                     00746100 T  0466:3
      IF TR ≥ 0 THEN                                                            00746200 T  0466:3
      BEGIN COMMENT NOT 1 ST CALL ON SORT SO FILL DATA FROM DISK;               00746300 T  0467:2
          OUTFIB[13].[27:1] ← 1;                  % SET FILE TO READ           00746400 T  0468:0
          P([DOTOP],0,11,COM,DEL,DEL);  % OPEN DISK FILE                      00746500 T  0470:2
          P([DOTOP]); WAIT;  % SLEEP UNTIL FILE IS OPENED                      00746600 T  0472:0
          OCDA ← OUTHEAD[10] + P(OD,DUP,ADD);                                   00746700 T  0473:0
          I ← 0; ORL ← ORS; OBC ← TBO;                                          00746800 T  0475:0
          WHILE I < S DO                                                        00746900 T  0477:2
          BEGIN                                                                 00747000 T  0478:3
            IF I < TR THEN                                                      00747100 T  0478:3
             BEGIN;COMMENT MOVE RECORD TO DATA;                                 00747200 T  0479:2
              STREAM(P1←*[DOTOP],P2←R,P3←(P(DUP)).[36:6],                       00747300 T  0480:0
                  P4←*[DATX[I]]);                                               00747400 T  0481:3
                BEGIN SI←P1;P3(DS←32 WDS;DS←32 WDS); DS←P2 WDS END;             00747500 T  0482:3
```

```
                P(1); DISKWRITE;                                                    00747600 T  0485:0
                V[I] ← NFLAG((*[DATX[I]])&I[18:33:15]);                             00747700 T  0486:0
               END                                                                  00747800 T  0488:2
              ELSE V[I] ← NFLAG((*[DATX[S]])&S[18:33:15]&1[5:47:1]);                00747900 T  0488:2
              I ← I+1;                                                              00748000 T  0492:2
             END;                                                                   00748100 T  0493:3
             P(MKS,0,0,[DOTOP[NOT 2]],4,FCR);   % REWIND                           00748200 T  0494:1
             OUTFIB[13].[27:1] ← 0;   % SET TO OUTPUT                              00748300 T  0498:0
             P([DOTOP],0,11,COM,DEL,DEL);   % OPEN FILE OUTPUT                    00748400 T  0500:2
             P([DOTOP[1]]); WAIT;                                                   00748500 T  0502:0
          GO TO IPB;                                                                00748600 T  0504:0
        END FILL OF DATA FROM DISK;                                                 00748700 T  0504:2
        IF IPTOG OR AC THEN BEGIN INREAD; INCOUNT ← 0 END;                          00748800 T  0504:2
        FOR VLOW ← 0 STEP 1 UNTIL S-1 DO                                            00748900 T  0507:3
        BEGIN COMMENT FILL DATA FROM INPUT FILE;                                    00749000 T  0512:0
        IF VLOW ≠ 0 THEN INREAD;   % POINT AT NEXT RECORD                           00749100 T  0512:0
        IF NOT EOF THEN                                                             00749200 T  0514:0
         BEGIN; COMMENT MOVE RECORD FROM FROM BUFFER TO DATA[VLOW,0];                00749300 T  0514:2
             STREAM(P1←*[CIIOD],P2←R,P3←P(DUP).[36:6],                              00749400 T  0515:0
                    P4← *[DATX[VLOW]]);                                             00749500 T  0516:3
             BEGIN SI ← P1; P3(DS←32WDS;DS←32WDS);DS←P2 WDS END;                    00749600 T  0517:3
             V[VLOW] ← NFLAG((*[DATX[VLOW]])&VLOW[18:33:15]);                       00749700 T  0520:0
         END;                                                                       00749800 T  0522:2
        END INITIAL FILL LOOP;                                                      00749900 T  0522:2
IPB:    ORI ← 10; GETROW;                                                           00750000 T  0523:0
        IF DISKFULL THEN P(1,[DOTOP[NOT 2]],81,17,COM);                             00750100 T  0525:0
        OCDA ← (LOSA ← OUTHEAD[ORI]) + 1;                                           00750200 T  0527:3
        SRS←ORL←ORS-1; SRI←ORI;     ONS←ORC←0;OBC←TBO;                              00750300 T  0529:3
IPBA:   FIRSTSELECT;     % INITIAL COMPARE                                          00750400 T  0534:2
        GO TO IPD;                                                                  00750500 T  0536:0
IPC:    LOWSELECT;    % INTERM COMPARE                                             00750600 T  0536:2
IPD:    IF VLOW < MS THEN                                                           00750700 T  0538:0
        BEGIN;COMMENT MOVE NEXT RECORD TO OUTPUT AREA;                              00750800 T  0538:3
        STREAM(P1←VL,P2←R,P3←(P(DUP)).[36:6],P4←*[DOTOP]);                          00750900 T  0539:1
        BEGIN SI←P1;P3(DS←32WDS;DS←32WDS);DS←P2 WDS END;                            00751000 T  0542:0
        P(1); DISKWRITE; % WRITE ON DISK THE RECORD FROM DATA[VLOW,0]               00751100 T  0544:1
        INREAD;           % POINT AT NEXT RECORD                                     00751200 T  0546:0
        IF NOT EOF THEN                                                             00751300 T  0547:0
        BEGIN COMMENT CHECK IF NEXT RECORD IS SMALLER;                              00751400 T  0547:2
        IF ( IF ALFA THEN P(0,MKS,0,*[INFIL],VL,EQUALS)                             00751500 T  0548:0
               ELSE IF AC THEN P(0,MKS,*[CIIOD],VL,EQUALS)                          00751600 T  0550:3
             ELSE P(MKS,XAL,0,VAL,0,EQUALS))                                        00751700 T  0554:0
         THEN V[VLOW] ← NFLAG((*[DATX[MS]])&MS[18:33:15]);                          00751800 T  0559:0
        STREAM(P1←*[CIIOD],P2←R,P3←P(DUP).[36:6],P4←*[DATX[VLOW]]);                 00751900 T  0562:0
        BEGIN SI←P1;P3(DS←32WDS;DS←32WDS);DS←P2 WDS END;                            00752000 T  0564:3
                    % MOVE NEXT RECORD TO DATA                                     00752100 T  0567:0
        END;                                                                        00752200 T  0567:0
        IF NOT DISKFULL THEN GO TO IPC;                                             00752300 T  0567:0
        END;                                                                        00752400 T  0567:3
             COMMENT END OF STRINGING PASS OR NO MORE DATA;                         00752500 T  0567:3
        IF NOT DISKFULL THEN  % CHECK FOR  RECORD = HIGH KEY                        00752600 T  0567:3
          FOR I ← 0 STEP 1 UNTIL MS-1 DO                                            00752700 T  0568:1
            IF (VLOW ← V[I].[18:15]) < MS THEN GO TO IPD;                           00752800 T  0573:0
             MOREDATA ← FALSE;                                                      00752900 T  0576:1
             FOR I← 0 STEP 1 UNTIL MS-1 DO                                          00753000 T  0577:0
             IF NOT V[I].[5:1] THEN                                                 00753100 T  0581:1
             BEGIN V[I] ← NFLAG((*[DATX[I]])&I[18:33:15]);                          00753200 T  0582:2
```

```
                    MOREDATA ← TRUE END;                                    00753300 T  0585:2
          DISKFULL ← TM AND ONS ≥ M-1 OR DISKFULL;                          00753400 T  0586:3
IPE:      WRITETAG; % WRITE STRING TAG WORD IN FRONT OF STRING               00753500 T  0589:2
          IF DISKFULL THEN GO TO IPG;                                       00753600 T  0591:0
          IF MOREDATA THEN                                                  00753700 T  0592:0
                IF NOT TM OR ONS < M    THEN GO TO IPBA                     00753800 T  0592:1
                   ELSE GO TO IPG;                                          00753900 T  0594:1
          FM ← NOT TM AND ONS ≤ M;                                          00754000 T  0595:1
IPG::END SORTIT;                                                            00754100 T  0597:1
                    %**********************************%                    00754200 T  0598:1
SUBROUTINE MERGEIT; % MERGES M STRINGS TO 1 STRING                          00754300 T  0598:1
     BEGIN                                                                  00754400 T  0599:0
MIC: FIRSTSELECT;                                                           00754500 T  0599:0
     GO TO MIE;                                                             00754600 T  0600:0
MID: LOWSELECT;                                                             00754700 T  0600:2
MIE: IF VLOW < MS THEN                                                      00754800 T  0602:0
        BEGIN; % MOVE LOW RECORD TO OUTPUT FILE                             00754900 T  0602:3
          STREAM(P1←VL,P2←R,P3←(P(DUP)),[36:6],P4←*[COIOD]);                00755000 T  0603:1
          BEGIN SI←P1;P3(DS←32WDS;DS←32WDS);DS← P2 WDS END;                 00755100 T  0606:0
          WRITEOUT; DISKREAD;                                               00755200 T  0608:1
          GO TO MID;                                                        00755300 T  0610:0
        END;                                                                00755400 T  0610:2
     FOR I ← 0 STEP 1 UNTIL MS-1 DO % CHECK FOR RECORD = HIGH KEY           00755500 T  0610:2
       IF (VLOW ← V[I],[18:15]) < MS THEN GO TO MIE;                        00755600 T  0614:3
     IF NOT TM AND NOT EOF THEN                                             00755700 T  0618:0
                    BEGIN  COMMENT HAVE MERGED M STRINGS FROM INPUT;        00755800 T  0619:1
                    WRITETAG;   %  TO ONE STRING IN OUTPUT, SELECT          00755900 T  0619:3
                    SUBMERGE;   %  M MORE STRINGS TO MERGE                  00756000 T  0621:0
                    GO TO MIC;                                              00756100 T  0622:0
                    END;                                                    00756200 T  0622:2
     END MERGEIT;                                                           00756300 T  0622:2
                    %**********************************%                    00756400 T  0622:3
START:              % INITIALIZE SORT PASS                                  00756500 T  0622:3
     BLKCTR ← BLKCTR + ((AC ← R>0) OR MF); R ← ABS(R);                      00756600 T  0635:2
     IF AC THEN IF CORESIZE.[1:1] THEN     % IDENTIFY COBOL68               00756610 T  0639:1
          BEGIN  AC←3; CORESIZE←ABS(CORESIZE);                              00756620 T  0640:3
                 BLKCTR ← BLKCTR - 1;                                       00756630 T  0643:0
                 END;                                                       00756650 T  0644:1
     IF NOT OPTOG THEN                                                      00756700 T  0644:1
BEGIN                                                                       00756710 T  0644:3
          PRFIB ← OUTFIL[NOT 2];                                            00756720 T  0645:1
          IF R > (PRFIB[18],[33:15]) THEN                                   00756730 T  0647:0
          P(1,[OUTFIL[NOT 2]],87,17,COM);                                   00756740 T  0648:2
END;                                                                        00756750 T  0651:0
     P(MKS,[TSN],[TCN],[TNN],  8 ,1,3,1,BLOCK);                             00756800 T  0651:0
     IF MF THEN GO TO P(POLYMERGE);                                         00756900 T  0653:1
     LISA ← IF IPTOG THEN    0 ELSE  % SIZE OF INPUT BUFFER                 00757000 T  0656:3
     2×P(*[INFIL[NOT 2]],18,COC).[3:15];                                    00757100 T  0658:1
     CORESIZE ← (IF CORESIZE = 0 THEN 12000 ELSE CORESIZE)                  00757200 T  0661:2
                    - 2000 - LISA;                                          00757300 T  0663:2
          IF CORESIZE < 2500 THEN CORESIZE ← 2500;                          00757310 T  0665:1
     ONS ← R ← ABS(R); S ← M ← 512;                                         00757400 T  0667:1
     WHILE ONS < 30 DO ONS ← ONS + R;                                       00757500 T  0670:0
LY:  IF ONS > 1023 THEN BEGIN ONS ← ONS - R; GO TO LZ END;                  00757600 T  0676:0
     IF ONS MOD 30 ≠ 0 THEN BEGIN ONS ← ONS + R; GO TO LY END;              00757700 T  0679:0
          COMMENT ONS NOW MINIMUM BUFFER SIZE;                              00757800 T  0682:2
LZ:  ORC ← ONS;                                                             00757900 T  0682:2
```

```
      WHILE (ORC + ONS) ≤ 150 DO ORC ← ORC + ONS ; %DSK INPT BUFF SZ          00758000 T  0683:1
      ORL ← ORC;                                                             00758100 T  0686:3
      WHILE (ORC + ORL) ≤ 450 DO ORL ← ORL + ORC ; %DSK OTPT BUFF SZ         00758200 T  0687:2
      OCDA ← CORESIZE - 2×ORL;                                               00758300 T  0691:0
      LOSA ← (OCDA-R) DIV (2×ORC);                                           00758400 T  0692:3
      IF LOSA ≤ 2 THEN M ← 2 ELSE WHILE M > LOSA DO M←M DIV 2;               00758500 T  0695:0
      LOSA ← (OCDA-R) DIV (R+3);                                             00758600 T  0700:2
      IF LOSA ≤ 2 THEN S ← 2 ELSE WHILE S > LOSA DO S←S DIV 2;               00758700 T  0702:3
      SRS ← ORL; SRI ← ORC;                                                  00758800 T  0708:1
LX:   ORI ← 2×ORL + 2×M×ORC;                                                 00758900 T  0709:3
      IF ORI < 1.1×CORESIZE AND ORC ≤ 1023 THEN                              00759000 T  0712:2
       BEGIN                                                                 00759100 T  0714:3
           SRS ← ORL; SRI ← ORC;                                             00759200 T  0715:1
           ORC ← ORC + ONS;                                                  00759300 T  0716:3
           ORL ← ORC; WHILE ORL < 300 DO ORL ← ORL + ORC;                    00759400 T  0718:0
           GO TO LX ;                                                        00759500 T  0723:0
       END;                                                                  00759600 T  0723:2
      D ← SRI DIV 30; IF SRI MOD 30 ≠ 0 THEN D ← D + 1;                      00759700 T  0723:2
      BF ← SRI DIV R; TBO ← SRS DIV R;                                       00759800 T  0727:3
           COMMENT COMPUTE DISK ROW SIZE, # ROWS ALWAYS = 20;                00759900 T  0730:1
      DISKSIZE ← ( IF DISKSIZE = 0 THEN 1000×600 ELSE DISKSIZE)              00760000 T  0730:1
                  / (BF×19×R);                                               00760100 T  0732:3
      IF DISKSIZE ≤ (Y ← TBO DIV BF) THEN                                    00760200 T  0735:0
           DISKSIZE ← Y + Y   ELSE                                           00760300 T  0736:3
           WHILE (DISKSIZE MOD Y)≠0 DO DISKSIZE←DISKSIZE + 1;                00760400 T  0738:2
      OD ← D × Y ;                                                           00760500 T  0742:2
      COMMENT SET UP DISK OUTPUT FILE AS ALGOL FILE;                         00760600 T  0743:3
      OUTFIB ← *[DOTOP[NOT 2]];          % GET FIB DESCRIPTOR                00760700 T  0743:3
      OUTFIB[ 8] ← (DISKSIZE DIV Y)&20[15:38:10]; % # ROWS, ROW SIZE.        00760800 T  0745:2
      OUTFIB[13].[10:9] ← OUTFIB[13].[1:9];                                  00760900 T  0748:1
      OUTFIB[4].[7:1] ← ((AC AND 3) = 1);      %COBOL61 DISK SORT FLG        00761000 T  0751:2
      OUTFIB[18] ← (X←TBO×R)&X[3:33:15]&X[18:33:15]; % DISK BLOCK             00761100 T  0755:0
      IF (Y←OUTFIB[4].[12:12]) < 1023 THEN  % FILE # TO FILE INDEX           00761200 T  0758:3
        OUTFIB[4] ← OUTFIB[4]&((Y-1)×ETRLNG)[13:37:11]&1[12:47:1];            00761300 T  0760:3
      IF FPB[FNUM+3].[16:7] = 0 THEN % NOT LABEL EQUATED                     00761310 T  0765:3
         FPB[FNUM+3].[16:2] := 1; % USE FAST DISK                            00761320 T  0768:2
      COMMENT OPEN DISK OUTPUT FILE, WILL SET UP FIB[16] AND                 00761400 T  0772:3
              WILL POINT TOP I/O DESC, PAST DISK ADDRESS;                    00761500 T  0772:3
      P([DOTOP],0,11,COM,DEL,DEL);                                           00761600 T  0772:3
      OUTHEAD ← *[OUTFIB[14]];                                               00761700 T  0774:1
      IF NT > 2 THEN                                                         00761800 T  0775:2
      OUTHEAD[8]←OUTHEAD[8] OR MEM;                                          00761900 T  0776:1
           COMMENT GET DISK SPACE FOR 1 ROW;                                 00762000 T  0778:3
      P([DOTOP[1]]); WAIT;       % WAIT FOR FILE TO BE OPENED               00762100 T  0778:3
      ORI ← 9; GETROW; MOREDATA ← TR ← -1;                                   00762200 T  0781:0
      IF DISKFULL THEN                                                       00762300 T  0784:2
      P(1,[DOTOP[NOT 2]],81,17,COM);      % IOR 81                          00762400 T  0784:3
      COMMENT IF INPUT FILE THEN OPEN IT, IF PROCEDURE THEN                  00762500 T  0787:1
           INITILIZE LINKAGE TO CALL IT;                                     00762600 T  0787:1
      IF IPTOG THEN BEGIN IF AC THEN                                         00762700 T  0787:1
            BEGIN ENDQ ← 0; BINGO ← INPRO; END;                              00762800 T  0788:1
            P(MKS,[INFIL],R,1,1,1,BLOCK);                                    00763000 T  0790:3
            IF(AC AND 3)=3 THEN          % COBOL68                           00763050 T  0792:2
                BEGIN  CIIOD ← [WAIN];                                       00763100 T  0793:3
                       WAIN ← P([INFIL],DUP,LOD,0,CDC,DEL,LOD);              00763150 T  0795:0
            END END ELSE                                                     00763200 T  0797:1
           BEGIN COMMENT CHECK FOR ALGOL OR COBOL;                           00763300 T  0797:1
```

```
      IF AC THEN BEGIN P(MKS,(NOT 2) INX [INFIL],1,COFCR);                    00763400 T  0797:3
                 IF AC.[46:1] THEN      % COBOL68                             00763440 T  0800:1
                    BEGIN CIIOD ← [WAIN];                                     00763450 T  0801:0
                          WAIN ← P(*[INFIL[NOT 2]],20,COC,0,XCH,FCX,          00763460 T  0802:1
                                DUP,DIB 0,LOD,0,CDC,DEL,DIB 0,LOD);           00763470 T  0804:3
                 END END ELSE BEGIN    % OPEN ALGOL INPUT FILE                00763500 T  0807:1
                     PRFIB ← *[INFIL[NOT 2]];                                 00763600 T  0807:3
                     PRFIB[13].[27:1] ← 1;                                    00763700 T  0809:2
                    P(MKS,0,3,[INFIL],ALRD,DEL);                              00763800 T  0812:0
                  END;                                                        00764200 T  0813:2
        END;                                                                  00764300 T  0813:2
     IF(AC AND 3)≠3 THEN CIIOD ← [INFIL];                                     00764350 T  0813:2
CALLSORT:                                                                     00764400 T  0816:0
     SORTIT;  % SORT INPUT FILE INTO STRINGS                                  00764500 T  0816:0
                %**********************************%                         00764600 T  0817:0
ENDSORTPASS:   COMMENT TURN BACK WHATS NO LONGER NEEDED AND                   00764700 T  0817:0
                INITILIZE MERGE PASS;                                         00764800 T  0817:0
     IF EOF THEN COMMENT CLOSE INPUT TAPE;                                    00764900 T  0817:0
        IF NOT MOREDATA THEN                                                  00764950 T  0817:1
     BEGIN                                                                    00765000 T  0818:1
       IF IPTOG THEN P([INFIL],3,COM,DEL)                                     00765100 T  0818:3
   ELSE P(MKS,2,0,[INFIL[NOT 2]],                                             00765200 P  0820:2
        IF (INFIL=OUTFIL) AND AC THEN 18 ELSE 4,FCR);                         00765210 C  0822:3
       IF INCOUNT = 0 THEN P(1,[DOTOP[NOT 2]],86,17,COM);                     00765300 T  0827:1
     END;                                                                     00765400 T  0830:2
     IF MOREDATA THEN                                                         00765500 T  0830:2
      BEGIN COMMENT SAVE CONTENTS OF DATA ON DISK;                            00765600 T  0830:3
        TR ← 0; OCDA ← OUTHEAD[ORI ← 10]; I ← 0;                              00765700 T  0831:1
        OBC ← TB0; ORL ← ORS;                                                 00765800 T  0834:1
        WHILE I < S DO                                                        00765900 T  0836:0
        BEGIN                                                                 00766000 T  0837:1
        IF NOT V[I].[5:1] THEN                                                00766100 T  0837:1
         BEGIN                                                                00766200 T  0838:2
            TR ← TR +1;                                                       00766300 T  0839:0
            STREAM(P1←*[DATX[I]],P2←R,P3←(P(DUP)).[36:6],                     00766400 T  0840:1
                 P4←*[DOTOP]);                                                00766500 T  0842:1
            BEGIN SI←P1;P3(OS←32WDS;DS←32WDS);DS←P2 WDS END;                  00766600 T  0843:0
            P(1); DISKWRITE;                                                  00766700 T  0845:1
         END;                                                                 00766800 T  0847:0
        I ← I + 1;                                                            00766900 T  0847:0
        END;                                                                  00767000 T  0848:1
        P(0); DISKWRITE; % WRITE BLOCK                                        00767100 T  0848:3
 END;                                                                         00767200 T  0850:0
AC ← AC&EOF[2:47:1]&MOREDATA[1:47:1];                                         00767300 T  0850:0
     COMMENT TURN BACK DATA & VECTOR ARRAYS;                                  00767400 T  0852:3
P(.DATA,LOD,RFB,.DATA,STD,[DATN],22,COM,DEL);                                 00767500 T  0852:3
P(.V,LOD,RFB,.V,STD,[VN],3,COM,DEL);                                          00767600 T  0855:0
STPP ← P(MS ← M,DUP,ADD,2,SUB);                                               00767700 T  0857:1
BLKCTR ← BLKCTR + 1;                                                          00767800 T  0859:2
     COMMENT DECLARE DISK OUTPUT FILE;                                        00767900 T  0860:3
ITNK ← 0;                                                                     00768000 T  0860:3
P(MKS,20,DISKSIZE,3,OUTFIB[4].[13:11]DIV ETRLNG +2,[DKI],                     00768100 T  0861:2
      (Y←2×M),1,BF×R,0,0,10,8,BLOCK);                                         00768200 T  0864:3
INFIB ← *[ITNK[2]];                                                           00768300 T  0868:1
ITOP ←  [ITNK[5]]&Y[8:38:10] ;      % POINT ITOP AT TOP I/O D                00768400 T  0869:2
     COMMENT OPEN FILE;                                                       00768500 T  0871:2
P([ITNK[5]],0,11,COM,DEL,DEL);                                                00768600 T  0871:2
```

```
     INHEAD ← *[INFIB[14]];                                                    00768700 T  0873:1
     IF NT > 2 THEN                                                            00768800 T  0874:2
     INHEAD[8] ← INHEAD[8]&1[2:47:1];   % FLAG SORT DISK                       00768900 T  0875:1
     P([ITOP[Y←Y-1]]); WAIT;  % WAIT FOR FILE TO BE OPENED                     00769000 T  0878:1
     IF INHEAD[10] ≠0 THEN                                                     00769100 T  0881:0
     P(10,.INHEAD,LOD,  24,COM,DEL,DEL);  % RETURN 1 ST ROW                   00769200 T  0882:0
          COMMENT SET FILE TO READ & PERMUTE 2 BUFFERS;                        00769300 T  0884:1
     INFIB[13] ← INFIB[13]&2[10:39:9]&1[27:47:1];                              00769400 T  0884:1
     INFIB[16] ← (*[INFIB[16]])&1[24:47:1];                                    00769500 T  0887:3
          COMMENT GET DATA AND VECTOR ARRAYS;                                  00769600 T  0890:2
     P(MKS,[VN],(2×M)-1,1,1,1,BLOCK);                                          00769700 T  0890:2
     P(5,M+1,[DATN],21,COM,DEL,DEL,DEL);                                       00769800 T  0893:1
          COMMENT GENERATE HIGH KEY RECORD;                                    00769900 T  0895:3
     MHN ← 0; P(MKS,[MHN],R,1,1,1,BLOCK);                                      00770000 T  0895:3
     P(MKS,MHK);  IF NOT AC THEN P(0,CDC,MHK,XCH,RDS,CFX,0);                   00770100 T  0898:1
          P(HIVALU);                                                           00770120 T  0902:1
     FOR I ← 0 STEP 1 UNTIL Y DO                                               00770200 T  0902:2
     ITOP[I] ← (NOT 0) INX (*[ITOP[I]])&1[24:47:1]&D[27:42:6];                  00770300 T  0904:0
DKC: IF NOT TM THEN                                                            00770400 T  0910:3
     BEGIN % DISK ONLY SORT COMPLETED                                          00770500 T  0911:1
       IF FM THEN GO TO DKF;                                                   00770600 T  0911:3
       P(10,,OUTHEAD,LOD,24,COM,DEL,DEL); % RETURN OVERLAY SPACE               00770700 T  0912:3
       P(,INHEAD,LOD,,OUTHEAD,LOD,,INHEAD,STD,,OUTHEAD,STD);                   00770800 T  0914:2
       SRI ← ORI; ORI ← 10;  % SRI= AMOUNT  OF DISK USED TO NOW                00770900 T  0916:2
       WHILE ORI < SRI DO      % GET ANOTHER AREA OF DISK = SRI                00771000 T  0918:0
     BEGIN                                                                     00771100 T  0919:1
          GETROW;                                                              00771200 T  0919:1
          IF DISKFULL THEN                                                     00771300 T  0920:0
           BEGIN P(.OUTHEAD,LOD); FORGETDISK; GO TO TPA; END;                  00771400 T  0920:1
     END;                                                                      00771500 T  0922:2
       COIOD ← DOTOP;                                                          00771600 T  0923:0
DKD: OCDA ← (LOSA ← OUTHEAD[(SRI ← ORI ← 11)]) + 1;                            00771700 T  0923:3
     SRS ← ORL ← ORS - 1;                                                      00771800 T  0926:3
DKE: LISA ← INHEAD[11];  % LOCATION OF FIRST TAG                               00771900 T  0928:3
     MOREDATA ← NOT(EOF ← DISKFULL ← ONS ← 0);                                 00772000 T  0929:3
     SUBMERGE; MERGEIT;                                                        00772100 T  0932:1
     IF FM THEN BEGIN P(*[INFIB[14]]); FORGETDISK; GO TO TPC END;              00772200 T  0934:0
     MOREDATA ← FALSE; WRITETAG;                                               00772300 T  0937:2
DKF: P(,INHEAD,LOD,,OUTHEAD,LOD,,INHEAD,STD,,OUTHEAD,STD);                     00772400 T  0939:0
     IF ONS > M THEN GO TO DKD;                                                00772500 T  0941:0
     FM ← TRUE; MS ← 2; WHILE ONS > MS DO MS←MS×2;                             00772600 T  0942:1
     STPP ← 2×MS-2;                                                            00772700 T  0946:3
          COMMENT REPLACE DISK OUTPUT BY PROGRAMMERS OUTPUT;                    00772800 T  0948:2
     P(MKS,0,0,[DOTOP[NOT 2]],4,FCR);  % RETURN BUFFERS                        00772900 T  0948:2
     OPENOUT;  IF(AC AND 3)≠3 THEN COIOD ← [OUTFIL];                           00773000 T  0952:1
     GO TO DKE;                                                                00773100 T  0955:2
     END;                                                                      00773200 T  0956:0
          COMMENT DISK-TAPE MERGE;                                             00773300 T  0956:0
     P(,INHEAD,LOD,,OUTHEAD,LOD,,INHEAD,STD,,OUTHEAD,STD);                     00773400 T  0956:0
TPA: P(MKS,0,0,[DOTOP[NOT 2]],4,FCR); % RETURN BUFFERS                         00773500 T  0958:0
     LISA ← INHEAD[11]; MOREDATA ← NOT(EOF←DISKFULL←ONS←0);                    00773600 T  0961:3
TPB: IF TN[COT] ≥ TC[COT] THEN BEGIN SELECT; GO TO TPB END;                    00773700 T  0965:1
     SUBMERGE; MERGEIT; WRITESTOPPER;                                          00773800 T  0968:2
     IF NOT EOF THEN GO TO TPB;                                                00773900 T  0972:0
     P(,INHEAD,LOD,,OUTHEAD,LOD,,INHEAD,STD,,OUTHEAD,STD);                     00774000 T  0972:3
TPC: P(,DATA,LOD,RFB,,DATA,STD,[DATN],22,COM,DEL); % RTN DATA                  00774100 T  0974:3
     INHEAD ← INFIB ← BASE ← ITOP ← BUFF ← 0;                                  00774300 T  0977:0
```

```
     IF FM OR AC.[1:2] =1 THEN GO TO WRAPUP;                                    00774500 T  0979:3
     P (10,COM);                    %RETURN MERGE MATRIX      %TR=117         00774550 T  0982:1
     MOREDATA ← AC.[1:1];  EOF ← AC.[2:1];                                      00774600 T  0982:3
     GO TO CALLSORT;                                                            00774700 T  0985:1
WRAPUP:                                                                         00774800 T  0985:3
     P(*[OUTFIB[14]]); FORGETDISK;                                              00774900 T  0985:3
     OUTFIB ← OUTHEAD ← 0;                                                      00775000 T  0988:0
     IF TM THEN                     %ITD SORT MERGE           %TR=117         00775100 T  0989:1
       BEGIN P (10,COM);            %RETURN MERGE MATRIX      %TR=117         00775120 T  0989:2
          GO TO P(POLYMERGE);% GO TO ITD MERGE                 %TR=117         00775130 T  0990:2
       END;                                                    %TR=117         00775140 T  0993:1
SORTDONE:                                                                       00775200 T  0993:1
                  COMMENT JUST DID FINAL PASS;                                  00775300 T  0993:1
       COMMENT RETURN EVERYTHING;                                               00775400 T  0993:1
     P([DOTOP]&0[18:18:15],6,11,COM,DEL,DEL);                                   00775500 T  0993:1
     IF NOT OPTOG THEN BEGIN                                                    00775600 T  0995:1
     P(MKS,2,0,[OUTFIL[NOT 2]],4,FCR)                                           00775700 T  0996:1
;    IF NOT AC THEN P(0,OUTFIL[NOT 2],8,CDC,STD);                               00775800 T  0999:3
                 END ELSE                                                       00775900 T  1003:1
     BEGIN COMMENT CALL OUTPUT PROCEDURE PASSING END-OF-SORT FLAG;              00776000 T  1003:1
       IF AC THEN                                                               00776100 T  1003:3
        BEGIN  ENDQ ← 1;  IF AC.[46:1] THEN                                     00776200 T  1004:0
                  P(MKS,BINGO,0,PERFORMGEN)    % COBOL68                     00776250 T  1006:1
               ELSE P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC)               00776300 T  1007:3
        END ELSE P(MKS,1,MEM,0,OUTPRO);                                         00776400 T  1010:0
     END;                                                                       00776500 T  1011:3
     P( 10,COM);                    % RETURN MERGE MATRIX       %TR=117       00776550 T  1011:3
     IF OUTCOUNT≠INCOUNT THEN P(INCOUNT,OUTCOUNT,0,                             00776600 T  1012:1
       [DOTOP[NOT 2]],82,17,COM);                                               00776700 T  1014:1
     P(10,COM);      % FALL OUT OF BLOCK COM WILL RETURN EVERYTHING             00776800 T  1016:0
END DISKSORT;                                                                   00776900 T  1016:2
                                                                                      SIZE= 1017 WORDS
```

```
PROCEDURE POLYMERGE(                                                            00800000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00179
                  T1,T2,T3,                                                     00800100 T  0000:0
                  ENDQ,BINGO,IPFIDX,OUTPRO,INPRO,OUTF,INF,                      00800200 T  0000:0
                  OPTOG,IPTOG,DKO,DKI,TP1,TP2,TP3,TP4,TP5,NT,                   00800300 T  0000:0
                  HIVALU,EQUALS,R,ALFA,CORESIZE,DISKSIZE);                      00800400 T  0000:0
    COMMENT DISK-SORT BY L.R. GUCK DATE 9/19/1965 ;                             00800500 T  0000:0
       VALUE      OPTOG,IPTOG,NT,HIVALU,EQUALS,R,ALFA,                         00800600 T  0000:0
                  CORESIZE,DISKSIZE;                                            00800700 T  0000:0
    REAL ENDQ,BINGO,IPFIDX,OUTPRO,INPRO,OUTF,T1,T2,T3,INF;                      00800800 T  0000:0
  BOOLEAN OPTOG,IPTOG;                                                          00800900 T  0000:0
    REAL DKO,DKI;                                                               00801000 T  0000:0
    NAME TP1,TP2,TP3,TP4,TP5;   % SCRATCH TAPES                                 00801100 T  0000:0
    REAL NT,HIVALU,EQUALS,R;                                                    00801200 T  0000:0
BOOLEAN ALFA;       % TRUE FOR ALPHA KEYS                                       00801300 T  0000:0
    REAL CORESIZE; % CORE STORAGE AVAILABLE                                     00801400 T  0000:0
 INTEGER DISKSIZE; % DISK STORAGE AVAILABLE                                     00801500 T  0000:0
    BEGIN                                                                       00801600 T  0000:0
   LABEL MIC,MID,MIE,START,TPD,TPE,TPF,RTNTR,TRA,SORTDONE,RTA,TRX;              00801700 T  0000:0
    REAL S,M,MS,STPP,D,OD,BF,TBO,I,X,Y,DN;                                      00801800 T  0000:0
    ARRAY V[*]; NAME VN = V;                                                    00801900 T  0000:0
```

```
      DEFINE VX1=FLAG(V[X+1])#, VX=FLAG(V[X])#, VL=FLAG(V[VLOW])#;
      DEFINE    VA1 = FLAG(V[X+1 ]&P(0,RDS)[CTF])#,
                VA  = FLAG(V[X   ]&P(0,RDS)[CTF])#,
                VAL = FLAG(V[VLOW]&P(0,RDS)[CTF])#;
      REAL VLOW;      % INDEX OF NEXT RECORD IN SEQUENCE
      ARRAY MHK[*];  % HIGH KEY FOR MERGE PHASE
      NAME  MHN=MHK;
      NAME DOTOP = DKO;
      BOOLEAN MOREDATA,FM,EOF,TM,DF,TR;
      BOOLEAN MF= T1;
      DEFINE IOC = @2000000000#,
             P = POLISH#;
      COMMENT PARAMETERS RELATED TO PROGRAMMERS FILES;
      NAME  INFIL = INF;     % POINTER TO TOP I/O DESC.
      NAME OUTFIL = OUTF;
      NAME  WAOUT = T2;      % COBOL68 OUTFILE WORK AREA
      ARRAY PRFIB[*];        % CONTAINS TAPE FILES FIB
      REAL AC;        % TRUE FOR COBOL INPUT FILE
      REAL INCOUNT,OUTCOUNT;
      NAME MEM    = 2;
      REAL BLOCK = 5,ALWR=12,ALRD=13,COFCR=12,CORW=14,ALFCR=14,
           PERFORMGEN = 13,     % COBOL68 IN-OUT PROCEDURES
           BLKCTR = 16;
     ARRAY PRTBASE = 10[*];
      REAL OF,OH,LO,ONS,ORC,OCDA,ORL,ORI,SRI,SRS,OBC,IFB,IFH;
      ARRAY BASE[*];     % POINTER TO CONTROL INFO IN DATA
REAL  ITP,DONTDONOTHINIEOPEN,LISA;
      DEFINE FCR = IF AC THEN COFCR ELSE ALFCR#;
      COMMENT PARAMETERS RELATED TO MERGE TAPES;
      INTEGER    CTRL;      % CURRENT CONTROL TAPE
      INTEGER    COT;       % CURRENT OUTPUT TAPE
      NAME       COIOD;     % LOC OF I/O D OF CURRENT OUTPUT TAPE
      NAME       TP;        % BASE POINTER OF MERGE TAPES
      ARRAY      TS[*];     % ARRAYS FOR CONTROLLING DISTRIBUTION
      ARRAY      TC[*];     %    PATTERNS ON MERGE TAPES
      ARRAY      TN[*];
      REAL       TM1=CORESIZE; % TAPES - 1
      NAME       CIIOD;     % LOC OF I/O D FOR CURRENT INPUT TAPE
                   %**********************************%
   SUBROUTINE WAIT; COMMENT WAIT FOR I/O COMPLETE USING ADRRESS
                   ON TOP OF STACK;
$ SET OMIT = NOT(TIMESHARING)
      BEGIN IF NOT (P(XCH,DUP,LOD)).[19:1] THEN P(IOC,36,COM,DEL);
$ POP OMIT
$ SET OMIT = TIMESHARING
      P(DEL); END WAIT;
                   %***************MM*****************%
   SUBROUTINE RELEASETAPE;  % CALLS MCP TO WRITE OUT BUFFERS
      BEGIN
       PRFIB[11] ← TBO;
       P(COIOD[0] ← FLAG(PRFIB[16]),COIOD,PRL,DEL);
   RTA:  P(GOIOD); WAIT;
       IF (*COIOD).[27:1] THEN  % REEL SWITCH
         BEGIN
           P(MKS,0,0,[COIOD[NOT 2]],6,FCR);
           GO TO RTA;
         END;
```

```
00802000 T  0000:0
00802010 T  0000:0
00802020 T  0000:0
00802030 T  0000:0
00802100 T  0000:0
00802200 T  0000:0
00802300 T  0000:0
00802400 T  0000:0
00802500 T  0000:0
00802600 T  0000:0
00802700 T  0000:0
00802800 T  0000:0
00802900 T  0000:0
00803000 T  0000:0
00803100 T  0000:0
00803120 T  0000:0
00803200 T  0000:0
00803300 T  0000:0
00803400 T  0000:0
00803500 T  0000:0
00803600 T  0000:0
00803610 T  0000:0
00803700 T  0000:0
00803800 T  0000:0
00803900 T  0000:0
00804000 T  0000:0
00804100 T  0000:0
00804200 T  0000:0
00804300 T  0000:0
00804400 T  0000:0
00804500 T  0000:0
00804600 T  0000:0
00804700 T  0000:0
00804800 T  0000:0
00804900 T  0000:0
00805000 T  0000:0
00805100 T  0000:0
00805200 T  0000:0
00805300 T  0000:0
00805400 T  0000:0
00805500 T  0001:0
00805550 T  0001:0
00805552 T  0001:0
00805553 T  0004:0
00805599 T  0004:0
00805700 T  0004:0
00805800 T  0006:0
00805900 T  0006:0
00806000 T  0006:0
00806100 T  0006:0
00806200 T  0007:1
00806300 T  0009:2
00806400 T  0011:0
00806500 T  0012:0
00806600 T  0012:2
00806700 T  0016:1
00806800 T  0016:3
```

```
      IF NOT(*COIOD).[2:1] THEN P(1,[COIOD[NOT 2]],74,17,COM);                  00806850 T  0016:3
      COIOD[0] ← 1 INX FLAG(PRFIB[16] ← NFLAG(*COIOD));                         00806900 T  0020:2
    END RELEASETAPE;                                                            00807000 T  0023:2
SUBROUTINE TAPEWRITE;   % BLOCKS OUTPUT TAPES                                   00807100 T  0023:3
    BEGIN                                                                       00807200 T  0024:0
      PRFIB ← *[COIOD[NOT 2]];                                                  00807300 T  0024:0
      PRFIB[9] ← PRFIB[9] + 1;  % RECORD COUNTER + 1                            00807400 T  0025:3
      IF (PRFIB[11] ← PRFIB[11] - 1) > 0 THEN   % BLOCK COUNTER                 00807500 T  0027:3
          COIOD[0] ← R INX *COIOD                                               00807600 T  0030:1
        ELSE                                                                    00807700 T  0031:0
      BEGIN     % TIME FOR RELEASE                                            00807800 T  0032:1
        P(0,PRFIB[16] INX MEM,STD);  % ZERO CONTROL WORD IN BUFF[2]              00807900 T  0032:3
        RELEASETAPE;                                                            00808000 T  0034:1
      END;                                                                      00808100 T  0035:0
    END TAPEWRITE;                                                              00808200 T  0035:0
SUBROUTINE WRITESTOPPER;                                                        00808300 T  0035:1
    BEGIN   % WRITES END OF STRING OR DUMMY STRINGS                             00808400 T  0036:0
      PRFIB ← *[COIOD[NOT 2]];                                                  00808500 T  0036:0
      X ← PRFIB[9]&(TBO-PRFIB[11])[18:33:15]&("DS")[3:33:15];                   00808600 T  0037:3
      P(X,PRFIB[16] INX MEM,STD);                                               00808700 T  0041:0
      TN[COT] ← TN[COT] + 1;   % COUNT UP STRINGS ON OUTPUT TAPE                00808800 T  0042:2
      RELEASETAPE;                                                              00808900 T  0044:2
    PRFIB [9] ← 0 ;                              % ZERO OUT STRING CTR          00808910 T  0046:0
    END WRITESTOPPER;                                                           00809000 T  0047:1
                  %***********************************%                        00809100 T  0049:0
SUBROUTINE OPENOUT; % OPENS PROGRAMMERS OUTPUT TAPE                             00809200 T  0049:0
    BEGIN                                                                       00809300 T  0049:0
    IF OPTOG THEN                                                               00809400 T  0049:0
      BEGIN  P(MKS,[OUTFIL],R,1,1,1,BLOCK);                                     00809500 T  0049:1
      IF AC THEN                                                                00809600 T  0051:2
      BEGIN  BINGO ← OUTPRO; ENDQ ← 0;                                          00809700 T  0051:3
      IF AC.[46:1] THEN          % COBOL68                                     00809730 T  0054:1
         BEGIN P(MKS,BINGO,0,PERFORMGEN);                                       00809740 T  0055:0
               COIOD ← [WAOUT];                                                 00809745 T  0056:2
               WAOUT ← P(*[OUTFIL],0,CDC);                                      00809750 T  0057:1
         END ELSE                                                               00809760 T  0058:3
      P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC);                            00809800 T  0058:3
    END END ELSE                                                                00809900 T  0061:0
      BEGIN  TR ← FALSE; PRFIB ← OUTFIL[NOT 2];                                 00810000 T  0061:0
         PRFIB[13].[27:1] ←0;                                                   00810100 T  0064:0
         IF AC THEN                                                             00810200 T  0066:2
         BEGIN                                                                  00810300 T  0066:3
         P(MKS,[OUTFIL[NOT 2]],3,COFCR);                                        00810400 T  0067:1
         IF AC.[46:1] THEN                  % COBOL68                           00810440 T  0069:0
              BEGIN  COIOD ← [WAOUT];                                           00810450 T  0069:3
                     WAOUT ← P(PRFIB[20].[FF],DUP,DIB 0,LOD,0,                  00810460 T  0071:0
                               CDC,DEL,DIB 0,LOD);                              00810470 T  0073:1
              END;                                                              00810480 T  0074:3
         TR ← PRFIB[4].[8:4] = 4;                                               00810600 T  0074:3
         END                                                                    00810700 T  0076:3
      ELSE BEGIN  P([OUTFIL],0,11,COM,DEL,DEL);                                 00810800 T  0076:3
              P(MKS,1,0,0,("R"),[OUTFIL],ALWR,DEL); END;                        00810900 T  0078:3
      END                                                                       00811000 T  0081:0
    END OPENOUT;                                                                00811100 T  0081:0
                  %***********************************%                        00811200 T  0081:1
SUBROUTINE TAPEREAD;  % READS TAPES ON POLYPHASE MERGE                          00811300 T  0081:1
    BEGIN                                                                       00811400 T  0082:0
```

```
      CIIOD ← TP[VLOW + 1 ];  PRFIB ← CIIOD[NOT 2];                                00811500 T  0082:0
      PRFIB[9] ← PRFIB[9] + 1;  % RECORD COUNTER + 1                               00811600 T  0085:3
      IF (PRFIB[11] ← PRFIB[11] -1) > 0 THEN     % BLOCK COUNTER - 1               00811700 T  0087:3
        V[VLOW] ← R INX V[VLOW]                                                    00811800 T  0090:1
          ELSE                                                                     00811900 T  0091:3
      BEGIN  % TIME FOR RELEASE                                                    00812000 T  0092:3
      IF (Y ← P(FLAG(PRFIB[16]),LOD))  ≠ 0 THEN                                    00812100 T  0093:1
        BEGIN  % CONTROL WORD ≠ 0 SO END OF STRING                                 00812200 T  0095:1
TRA:       DF ← TRUE;                                                              00812300 T  0095:3
           IF Y.[33:15] ≠ PRFIB[9].[33:15] THEN                                    00812400 T  0096:2
          P(0,[CIIOD[NOT 2]],85,17,COM,DEL,DEL,DEL);                               00812500 T  0098:2
        END;                                                                       00812600 T  0101:3
      P(CIIOD[0] ← FLAG(PRFIB[16]),CIIOD,PRL,DEL);                                 00812700 T  0101:3
TRX:  P(CIIOD); WAIT;                                                              00812800 T  0104:0
      IF NOT (*CIIOD).[2:1] THEN     % ERROR OR EOF OR EOR                        00812900 T  0105:0
        BEGIN                                                                      00813000 T  0106:1
         EOF ← P(MKS,1,0,[CIIOD[NOT 2]],6,FCR);                                    00813100 T  0106:3
         IF NOT EOF THEN GO TO TRX;                                                00813200 T  0111:0
        END;                                                                       00813300 T  0111:3
      IF EOF THEN GO TO RTNTR;                                                     00813400 T  0111:3
      PRFIB[11] ← IF P(Y← P(*CIIOD,LOD)) = 0 THEN TBO                              00813500 T  0112:3
                                             ELSE Y.[18:15];                       00813600 T  0115:2
      CIIOD[0] ← 1 INX FLAG(PRFIB[16] ← NFLAG(*CIIOD));                            00813700 T  0117:2
      IF DF THEN GO TO RTNTR;                                                      00813800 T  0120:2
      IF PRFIB[11] ≠ 0 THEN V[VLOW] ← V[VLOW]&(*CIIOD)[33:33:15]                   00813900 T  0121:2
                       ELSE GO TO TRA;                                           00814000 T  0124:2
      END;                                                                         00814100 T  0125:1
RTNTR: IF EOF OR DF THEN BEGIN                                                     00814200 T  0125:1
                         V[VLOW] ← NFLAG(MHK)&MS[18:33:15];                        00814300 T  0126:2
                         IF FM AND NOT MF THEN % REL TAPE LST PASS                 00814340 T  0128:3
                         P(MKS,4,0,[CIIOD[NOT 2]],4,FCR); % FOR SRT                00814350 T  0129:3
                         EOF←DF←FALSE;                                             00814400 T  0134:0
                         END;                                                      00814500 T  0135:1
    END TAPEREAD;                                                                  00814600 T  0135:1
                        %**********************************%                      00814700 T  0135:2
SUBROUTINE INREAD;  % READS PROGRAMMERS MERGE FILES                                00814800 T  0135:2
    BEGIN                                                                          00814900 T  0136:0
      CIIOD ← TP[VLOW+1]; PRFIB ← CIIOD[NOT 2];                                    00815000 T  0136:0
      BEGIN                                                                        00815100 T  0139:3
         IF AC THEN TC[VLOW] ← P(MKS,R,CIIOD,0,CORW)                               00815200 T  0139:3
               ELSE                                                                00815300 T  0142:0
              BEGIN                                                                00815400 T  0142:3
               P(MKS,0,0,CIIOD,ALRD);                                              00815500 T  0143:1
               TC[VLOW] ← P(MKS,0,3,CIIOD,ALRD) < 0;                               00815600 T  0144:2
              END;                                                                 00815700 T  0147:1
         IF TC[VLOW] THEN V[VLOW] ← NFLAG(MHK)&MS[18:33:15]                        00815800 T  0147:1
            ELSE IF (AC AND 3) ≠ 3 THEN  % NOT COBOL68                             00815900 T  0149:3
                 V[VLOW] ← (*P(DUP)) & (*[CIIOD])[CTC];                            00815910 T  0152:1
      END;                                                                         00816000 T  0155:0
    END  INREAD;                                                                   00816100 T  0155:0
                       %**********************************%                       00816200 T  0155:1
SUBROUTINE WRITEOUT;                                                               00816300 T  0155:1
    BEGIN  %  SELECTS FILE TO BE WRITTEN DURING MERGE                              00816400 T  0156:0
     IF NOT FM THEN TAPEWRITE                                                      00816500 T  0156:0
               ELSE                                                                00816600 T  0157:3
      BEGIN COMMENT CALL OUTPUT PROCEDURE OR WRITE INTRINSIC;                      00816700 T  0158:0
      OUTCOUNT ← OUTCOUNT +1;                                                      00816800 T  0158:2
```

```
          IF OPTOG THEN                                                                                  00816900 T  0159:3
             BEGIN                                                                                       00817000 T  0160:0
             IF AC THEN BEGIN   ENDQ ← 0;                                                                00817100 T  0160:2
                       IF AC.[46:1] THEN P(MKS,BINGO,0,PERFORMGEN)                                       00817120 T  0162:1
                  ELSE P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC)                                     00817150 T  0164:1
                  END ELSE P(MKS,0,*[OUTFIL],0,OUTPRO);                                                  00817200 T  0166:3
             END                                                                                         00817300 T  0168:3
                   ELSE                                                                                  00817400 T  0168:3
             BEGIN COMMENT OUTPUT FILE RATHER THAN OUTPUT PROCEDURE;                                     00817500 T  0168:3
          IF AC THEN P(MKS,0,1,1,0,R,[OUTFIL],1,CORW)                                                    00817600 T  0169:1
                   ELSE BEGIN                                                                            00817700 T  0172:0
                        P(MKS,1,0,0,R,[OUTFIL],ALWR);                                                    00817800 T  0172:3
                        P(MKS,1,0,0,(=R),[OUTFIL],ALWR,DEL); END;                                        00817900 T  0174:2
            IF TR THEN IF P THEN P(1,[OUTFIL[NOT 2]],83,17,COM);                                         00818000 T  0176:3
             END;                                                                                        00818100 T  0180:0
          END;                                                                                           00818200 T  0180:0
       END WRITEOUT;                                                                                     00818300 T  0180:0
SUBROUTINE FIRSTSELECT;   % INITIAL SELECTION OF LOW RECORD                                              00818400 T  0180:1
       BEGIN                                                                                             00818500 T  0181:0
       X ← 0; I←MS-1;                                                                                    00818600 T  0181:0
       DO BEGIN                                                                                          00818700 T  0183:0
       I←I+1;                                                                                            00818800 T  0183:0
       V[I] ← V[X+((IF ALFA THEN P(0,MKS,0,VX1,VX,EQUALS)                                                00818900 T  0184:1
                 ELSE IF AC THEN  P(0,MKS,VX1,VX,EQUALS)                                                 00819000 T  0188:2
            ELSE P(MKS,VA1,0,VA,0,EQUALS)) AND TRUE)];                                                   00819100 T  0192:2
        END UNTIL (X←X+2) = STPP; VLOW ← V[I].[18:15];                                                   00819200 T  0199:3
       END FIRSTSELECT;                                                                                  00819300 T  0203:3
SUBROUTINE LOWSELECT;                                                                                    00819400 T  0204:0
       BEGIN                                                                                             00819500 T  0204:0
       X ← VLOW AND 1022;                                                                                00819600 T  0204:0
       DO BEGIN                                                                                          00819700 T  0205:1
       I ← MS + X.[38:9];  % I ] MS + (X/2)                                                              00819800 T  0205:1
       V[I] ← V[X+((IF ALFA THEN P(0,MKS,0,VX1,VX,EQUALS)                                                00819900 T  0207:0
                 ELSE IF AC THEN P(0,MKS,VX1,VX,EQUALS)                                                  00820000 T  0211:1
            ELSE P(MKS,VA1,0,VA,0,EQUALS)) AND TRUE)];                                                   00820100 T  0215:1
       X ← I AND 1022;                                                                                   00820200 T  0222:2
          END UNTIL I = STPP; VLOW ← V[I].[18:15];                                                       00820300 T  0223:3
       END LOWSELECT;                                                                                    00820400 T  0226:3
                        %**********************************%                                             00820500 T  0227:0
SUBROUTINE MERGEIT; % MERGES M STRINGS TO 1 STRING                                                       00820600 T  0227:0
       BEGIN                                                                                             00820700 T  0227:0
MIC:   FIRSTSELECT;                                                                                      00820800 T  0227:0
       GO TO MIE;                                                                                        00820900 T  0228:0
MID:   LOWSELECT;                                                                                        00821000 T  0228:2
MIE:   IF VLOW < MS THEN                                                                                 00821100 T  0230:0
          BEGIN; % MOVE LOW RECORD TO OUTPUT FILE                                                        00821200 T  0230:3
            STREAM(P1←VL,P2←R,P3←(P(DUP)).[36:6],P4←*[COIOD]);                                           00821300 T  0231:1
            BEGIN SI←P1;P3(DS←32WDS;DS←32WDS);DS← P2 WDS END;                                            00821400 T  0234:0
            WRITEOUT; IF MF THEN INREAD ELSE TAPEREAD;                                                   00821500 T  0236:1
            GO TO MID;                                                                                   00821600 T  0241:0
          END;                                                                                           00821700 T  0241:2
       FOR I ← 0 STEP 1 UNTIL MS-1 DO % CHECK FOR RECORD = HIGH KEY                                      00821800 T  0241:2
         IF (VLOW ← V[I].[18:15]) < MS THEN GO TO MIE;                                                   00821900 T  0245:3
       END MERGEIT;                                                                                      00822000 T  0249:0
START:                                                                                                   00822100 T  0249:1
       CIIOD ← 0; P((.[CIIOD].[33:15]+2),STS);                                                           00822200 T  0262:2
       MS ← 2; TM1←NT-1; WHILE MS<(TM1+MF) DO MS←MS×2;                                                   00822300 T  0265:0
```

```
IF MF THEN                                                                    00822400 T  0270:2
  BEGIN   % MERGE ONLY                                                        00822500 T  0270:3
  TP ← ((NOT 7) INX [NT]); FM ← TRUE;                                         00822600 T  0271:1
  P(MKS,[VN],(2×MS)-1,1,1,1,BLOCK); P(MKS,[MHN],R,1,1,1,BLOCK);               00822700 T  0273:2
  P(MKS,MHK);  IF NOT AC THEN P(0,CDC,MHK,XCH,RDS,CFX,0);                     00822800 T  0278:0
     P(HIVALU);                                                               00822820 T  0282:0
  FOR VLOW ← 0 STEP 1 UNTIL TM1 DO                                            00822900 T  0282:1
  BEGIN         % OPEN TAPES                                                  00823000 T  0283:0
     CIIOD ← TP[VLOW+1]; PRFIB ← CIIOD[NOT 2];                                00823100 T  0283:0
     PRFIB[13].[27:1] ← 1;     % SET TO OPEN INPUT                            00823200 T  0286:3
     IF AC THEN P(MKS,[CIIOD[NOT 2]],1,COFCR)                                 00823300 T  0289:1
           ELSE TC[VLOW]←P(MKS,0,3,CIIOD,ALRD)<0;                             00823400 T  0291:2
     IF PRFIB[5].[39:1] THEN TC[VLOW] ← 1 ELSE %OPTIONAL                      00823500 T  0295:0
BEGIN                                                                         00823600 T  0297:3
     P(CIIOD); WAIT; IF AC THEN INREAD;                                       00823700 T  0298:1
END;                                                                          00823800 T  0302:0
  END;                                                                        00823900 T  0302:0
  FOR   I  ← 0 STEP 1 UNTIL MS-1 DO                                           00824000 T  0304:1
  BEGIN                                                                       00824100 T  0308:2
     IF I > TM1 OR TC[I] THEN V[I] ← NFLAG(MHK)&MS[18:33:15]                  00824200 T  0308:2
        ELSE V[I]←NFLAG(P(TP[I+1])&(IF(AC AND 3)=3 THEN                       00824300 T  0312:0
             P(2,NOT,XCH,INX,LOD,20,COC,0,XCH,FCX,DIB 0,LOD,I)                00824310 T  0316:2
             ELSE P(LOD,I))[CTF]);                                            00824320 T  0320:0
  END;                                                                        00824400 T  0322:1
  OPENOUT;  IF(AC AND 3)≠3 THEN COIOD ← OUTFIL;                               00824500 T  0322:3
  STPP ← 2 × MS - 2;                                                          00824550 T  0326:2
  MERGEIT;                                                                    00824600 T  0328:1
     FOR I ←1 STEP 1 UNTIL TM1 + 1 DO   %CLOSE LOCK ALL TAPES                 00824610 T  0329:0
     BEGIN CIIOD ←TP [I] ;                                          % PG       00824630 T  0333:1
           P (MKS,2,0,[CIIOD[NOT 2 ]],4,FCR);                       % PG       00824660 T  0334:3
     END; %                                                         % PG       00824670 T  0338:2
   GO TO SORTDONE;                                                            00824700 T  0339:0
 END;                                                                         00824800 T  0339:2
FOR I ← COT STEP 1 UNTIL TM1 DO % WRITE OUT DUMMY STRINGS                     00824900 T  0339:2
IF TN[I] < TC[I] THEN            % PERFECT DISTRIBUTION                      00825000 T  0341:0
BEGIN                                                                         00825100 T  0342:1
  IF COT ≠ I THEN                                                             00825200 T  0342:3
     BEGIN                                                                    00825300 T  0343:2
     PRFIB ← COIOD[NOT 2]; COIOD[0] ← FLAG(PRFIB[16]);                        00825400 T  0344:0
     P(((NOT 2) INX TP[I]),((NOT 2) INX TP[COT]),                             00825500 T  0347:0
                20,COM,DEL,DEL);                                              00825600 T  0350:2
     COIOD ← TP[I]; COIOD[0] ← 1 INX *COIOD;                                  00825700 T  0351:2
     COT ← I;                                                                 00825800 T  0354:2
     END;                                                                     00825900 T  0355:1
 WHILE TN[I] < TC[I] DO        % PERFECT DISTRIBUTION PATTERN                 00826000 T  0355:1
  BEGIN COIOD ←   TP[I] ; PRFIB ← COIOD[NOT 2];                               00826100 T  0357:0
        PRFIB[11] ← TBO; PRFIB[9] ← 0; WRITESTOPPER;                          00826200 T  0360:1
  END;                                                                        00826300 T  0364:0
END;                                                                          00826400 T  0364:2
FOR I ← 1 STEP 1 UNTIL TM1 DO                                                 00826500 T  0366:3
BEGIN % SET UP TO DO POLYPHASE MERGE                                          00826600 T  0368:0
 CIIOD ←   TP[I] ;                                                            00826700 T  0368:0
 P(MKS, 2 ,0,[CIIOD[NOT 2]],6,FCR);  % REWIND OR RELEASE                      00826800 T  0369:2
END;                                                                          00826900 T  0373:1
P(MKS,[VN],(2×MS)-1,1,1,1,BLOCK); STPP ← 2×MS-2; MHN←0;                       00827000 T  0375:2
P(MKS,[MHN],R,1,1,1,BLOCK);  % HI-KEY                                         00827100 T  0380:3
P(MKS,MHK);  IF NOT AC THEN P(0,CDC,MHK,XCH,RDS,CFX,0);                       00827200 T  0382:2
```

```
          P(HIVALU);
     FOR I ← 1 STEP 1 UNTIL TM1 DO
     BEGIN   % OPEN INPUT TAPES
     CIIOD ←   TP[I] ;  PRFIB ← CIIOD[NOT 2];
     PRFIB[13].[27:1] ← 1; P(CIIOD,0,11,COM,DEL,DEL);
     P(CIIOD); WAIT; PRFIB[11] ← TBO; PRFIB[9] ← 0;
     CII@D[0] ← 1 INX *CIIOD;
     END;
TPD: FM ← TRUE;
     FOR I ← 1 STEP 1 UNTIL TM1 DO IF TN[I] > 1 THEN FM ← FALSE;
     IF FM THEN
          BEGIN OPENOUT; IF(AC AND 3)≠3 THEN COIOD ← OUTFIL END
           ELSE  % OPEN SCRATCH OUTPUT TAPE
        BEGIN COIOD ←  TP[NT];       PRFIB ← *[COIOD[NOT 2]];
          PRFIB[13].[27:1] ← 0;  % SET TO OUTPUT
          P(COIOD,0,11,COM,DEL,DEL);
          PRFIB[11] ← TBO; PRFIB[9] ← 0; COT ← NT;
          P(COIOD); WAIT; COIOD[0] ← 1 INX FLAG(PRFIB[16]);
        END;
          COMMENT SET UP VECTOR ROW 0;
TPE: FOR I ← 0 STEP 1 UNTIL MS - 1 DO
     BEGIN
     IF I ≥ TM1 THEN BEGIN EOF ← TRUE; GO TO TPF END;
     CIIOD ←   TP[I+1] ;  PRFIB ← CIIOD[NOT 2];
     IF (EOF ← (*CIIOD).[27:1]) THEN GO TO TPF;
     IF ( X ← (*(FLAG(PRFIB[16])))) ≠0 THEN                    %TR=142
     IF NOT ( EOF ←X.[33:15] = 0 ) THEN PRFIB[11] ←X.[18:15] %TR=142
          ELSE ELSE PRFIB[11] ← TBO ; PRFIB[9] ←0;             %TR=142
TPF: IF TN[I+1] = 0 OR EOF
          THEN BEGIN                                          % PG
               V[I] ←NFLAG(MHK)&MS[18:33:15];                 % PG
            IF FM AND I LSS TM1 THEN P(MKS,4,0,[CIIOD[NOT 2]],4,FCR);
             END                                              % PG
        ELSE V[I] ← NFLAG(P(*TP[I+1])&I[18:33:15]);
     EOF ← FALSE;
     END;
     MERGEIT;
     IF FM THEN GO TO SORTDONE ELSE WRITESTOPPER;
          COMMENT HAVE MERGED A STRING OFF EACH TAPE;
          COMMENT CHECK IF REWIND NEEDED;
     FOR I ← 1 STEP 1 UNTIL TM1 DO TN[I] ← TN[I] - 1;
     FOR I ← 1 STEP 1 UNTIL TM1 DO IF TN[I] ≤ 0 THEN
      BEGIN   % REWIND IS NEEDED
       PRFIB ← COIOD[NOT 2];
       P(MKS, 2 ,0,[COIOD[NOT 2]],6,FCR); % REWIND OR RELEASE
       CIIOD←TP[I];
       P(MKS,4,0,[CIIOD[NOT 2]],4,FCR);  % CLOSE PURGE
       TN[I] ← TN[NT]; TN[NT]← 0;
       PRFIB[13].[27:1] ← 1; % SET FORMER OUTPUT TO INPUT
     %                                                       %TR=142
       P(COIOD,0,11,COM,DEL,DEL); % OPEN FOR INPUT
       P(COIOD); WAIT; COIOD[0] ← 1 INX FLAG(PRFIB[16]);
     P((TP[NT])); TP[NT] ← TP[I]; TP[I] ← P(XCH);
       GO TO TPD;
      END;
     GO TO TPE;
SORTDONE:
```

```
00827220 T  0386:2
00827300 T  0386:3
00827400 T  0388:0
00827500 T  0388:0
00827600 T  0391:1
00827700 T  0395:1
00827800 T  0399:2
00827900 T  0401:0
00828000 T  0403:1
00828100 T  0404:0
00828200 T  0409:2
00828250 T  0409:3
00828300 T  0413:2
00828400 T  0413:2
00828500 T  0417:1
00828600 T  0419:3
00828700 T  0421:1
00828800 T  0424:2
00828900 T  0427:3
00829000 T  0427:3
00829100 T  0427:3
00829200 T  0432:0
00829300 T  0432:0
00829400 T  0434:2
00829500 T  0438:1
00829600 T  0440:2
00829610 T  0442:2
00829620 T  0446:0
00829700 T  0450:3
00829800 T  0452:1
00829825 T  0453:1
00829850 T  0455:2
00829875 T  0461:0
00829900 T  0461:0
00830000 T  0465:0
00830100 T  0465:3
00830200 T  0466:1
00830300 T  0467:0
00830400 T  0469:0
00830500 T  0469:0
00830600 T  0469:0
00830700 T  0474:1
00830800 T  0476:0
00830900 T  0476:2
00831000 T  0478:1
00831100 T  0482:0
00831200 T  0483:2
00831300 T  0487:1
00831400 T  0490:0
00831500 T  0492:2
00831600 T  0492:2
00831700 T  0494:0
00831800 T  0496:3
00831900 T  0501:2
00832000 T  0502:0
00832100 T  0504:1
00832200 T  0504:3
```

```
                    COMMENT JUST DID FINAL PASS;                                    00832300 T  0504:3
          COMMENT RETURN EVERYTHING;                                                00832400 T  0504:3
     IF NOT MF THEN                                                                 00833000 T  0504:3
     P([DOTOP]&0[18:18:15],6,11,COM,DEL,DEL);                                       00833100 T  0505:1
     IF NOT OPTOG THEN   BEGIN                                                      00833200 T  0507:3
     P(MKS,2,0,[OUTFIL[NOT 2]],4,FCR)                                               00833300 T  0508:3
;    IF NOT AC THEN P(0,OUTFIL[NOT 2],8,CDC,STD);                                   00833400 T  0512:1
                 END    ELSE                                                       00833500 T  0515:3
     BEGIN COMMENT CALL OUTPUT PROCEDURE PASSING END-OF-SORT FLAG;                  00833600 T  0515:3
       IF AC THEN                                                                   00833700 T  0516:1
          BEGIN  ENDQ ← 1;  IF AC.[46:1] THEN                                       00833800 T  0516:2
                   P(MKS,BINGO,0,PERFORMGEN)    % COBOL68                         00833850 T  0518:3
               ELSE P(MKS,BINGO,[PRTBASE[P(DUP)]],LOD,IPFIDX,COC)                   00833900 T  0520:1
          END ELSE P(MKS,1,MEM,0,OUTPRO);                                           00834000 T  0522:2
     END;                                                                           00834100 T  0524:1
     IF NOT MF THEN                                                                 00834200 T  0524:1
     IF OUTCOUNT≠INCOUNT THEN P(INCOUNT,OUTCOUNT,0,                                 00834300 T  0524:3
       [DOTOP[NOT 2]],82,17,COM);                                                   00834400 T  0527:1
     P(10,COM);     % FALL OUT OF BLOCK COM WILL RETURN EVERYTHING                  00834500 T  0529:0
     END POLYMERGE;                                                                 00834600 T  0529:2
                                                                                          SIZE= 0530 WORDS


REAL PROCEDURE DUMPINT(SN,CV,BV,  TIPE,TENS,ALFA,CHAR,FIEL,FORMT);%                 00900000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00197
              VALUE SN,CV,BV,  TIPE,TENS,ALFA,CHAR,FORMT;%                          00900100 T  0000:0
                REAL SN,CV,BV,  TIPE,TENS,ALFA,CHAR,FORMT;%                         00900200 T  0000:0
               NAME FIEL;%                                                          00900300 T  0000:0
                    BEGIN%                                                          00900400 T  0000:0
     REAL E=+1,%                                                                    00900500 T  0000:0
          VALUEE=+2,%                                                               00900600 T  0000:0
          DH1=+3,%                                                                  00900700 T  0000:0
          DH2=+4,%                                                                  00900800 T  0000:0
          LNGTH=+5,%                                                                00900900 T  0000:0
          CSIZE=+6,%                                                                00901000 T  0000:0
          BCTR=+7%                                                                  00901100 T  0000:0
     ,    TEMP=+8,%                                                                 00901200 T  0000:0
          NL  =+9%                                                                  00901300 T  0000:0
     ,    J   =+10,%                                                                00901400 T  0000:0
          TROW=+11,%                                                                00901500 T  0000:0
          COUNT=+12,%                                                               00901600 T  0000:0
          TARRY=+13,%                                                               00901700 T  0000:0
     N=BV,%                                                                         00901800 T  0000:0
          SINN=9;%                                                                  00901900 T  0000:0
     LABEL%                                                                         00902000 T  0000:0
     PRINT,%                                                                        00902100 T  0000:0
     PR3,%                                                                          00902200 T  0000:0
     BRTN,%                                                                         00902300 T  0000:0
     TA,%                                                                           00902400 T  0000:0
     TCP,%                                                                          00902500 T  0000:0
     TC,%                                                                           00902600 T  0000:0
     IRTN,%                                                                         00902700 T  0000:0
     P2,%                                                                           00902800 T  0000:0
     P1,%                                                                           00902900 T  0000:0
     TB,%                                                                           00903000 T  0000:0
```

```
TD,%
P3E,%
TF,%
TP2,%
TP3,%
TP22,%
TP1,%
TP8,%
TP5,%
TP11,%
TD1,%
TD2,%
TD3,%
TP10,%
TP9,%
TP6,%
TP7,%
TP71,%
P3,%
P3A,%
P3L,%
P3I,%
EA,%
EB,%
EC,%
ED,%
P5,%
EE,%
EFAA,%
EFA,%
ERTN,%
EFB,%
EFC;%
SWITCH OCSWITCH←TA,TB,TC,TCP,TD;%
SWITCH TIPESW←P3L,P3E,P3A,P3I;%
REAL RITEINT=12;%
REAL SELECT=14;%
NAME M=2;%
DEFINE I=ALFA#;%
    SUBROUTINE RITE;%
            BEGIN%
            P(MKS,1,0,0,LNGTH,FIEL,RITEINT,
                MKS,1,0,0,(-1),FIEL,RITEINT,DEL);
            END;%
    SUBROUTINE FINDE ;%
        BEGIN     IF P(VALUEE×@1141000000000000,DUP)≠0 THEN%
                  BEGIN%
                  SINN←P(DUP,0,<); P(SSP,,VALUEE,SND);%
                IF  P(0,XCH,DIA 3,DIB 42,TRB 6,VALUEE,DIA 2%
              ,       DIB 1,TRB 1,12,+,@1157163034761674,×,%
                      @1154000000000000,+,,E,ISN,DUP)<0 THEN GO TO EFB;%
            P(TENS);%
            EFAA:IF P ≤ VALUEE THEN GO TO ERTN;GO TO EFC;%
                  END;%
                     P(DEL);%
                     E←SINN←0;GO TO ERTN;%
    EFB:    P(CHS,TENS,1,XCH,/); GO TO EFAA;
```

```
00903100 T  0000:0
00903200 T  0000:0
00903300 T  0000:0
00903400 T  0000:0
00903500 T  0000:0
00903600 T  0000:0
00903700 T  0000:0
00903800 T  0000:0
00903900 T  0000:0
00904000 T  0000:0
00904100 T  0000:0
00904200 T  0000:0
00904300 T  0000:0
00904400 T  0000:0
00904500 T  0000:0
00904600 T  0000:0
00904700 T  0000:0
00904800 T  0000:0
00904900 T  0000:0
00905000 T  0000:0
00905100 T  0000:0
00905200 T  0000:0
00905300 T  0000:0
00905400 T  0000:0
00905500 T  0000:0
00905600 T  0000:0
00905700 T  0000:0
00905800 T  0000:0
00905900 T  0000:0
00906000 T  0000:0
00906100 T  0000:0
00906200 T  0000:0
00906300 T  0000:0
00906400 T  0000:0
00906500 T  0000:0
00906600 T  0000:0
00906700 T  0000:0
00906800 T  0000:0
00906900 T  0000:0
00907000 T  0000:0
00907100 T  0001:0
00907200 T  0001:0
00907300 T  0002:3
00907400 T  0005:0
00907500 T  0005:1
00907600 T  0006:0
00907700 T  0007:2
00907800 T  0008:0
00907900 T  0010:0
00908000 T  0011:3
00908100 T  0013:1
00908200 T  0015:2
00908300 T  0015:3
00908400 T  0021:0
00908500 T  0021:0
00908600 T  0021:1
00908700 T  0023:0
```

```
            EFC: E←E-1;%                                                                00908800 T  0024:3
            ERTN:%                                                                      00908900 T  0026:0
        END;%                                                                           00909000 T  0026:0
SUBROUTINE OUTI;%                                                                       00909100 T  0026:1
            BEGIN FINDE;%                                                               00909200 T  0027:0
                P(VALUEE,,DH1,ISD);%                                                    00909300 T  0028:0
                STREAM(P8←0:P7←SINN,P6←[VALUEE],P5←SINN,P4←0,P3←%                       00909400 T  0028:3
                       E+1←SINN,P2←0,P1←BCTR);%                                         00909500 T  0030:2
                      BEGIN%                                                            00909600 T  0032:2
                           P2(DS←LIT" ");%                                              00909700 T  0032:2
                           P1←DI;SI←P6;DS←P4 DEC; SI←P6;SI←SI+8;%                       00909800 T  0033:3
                           DS←P3 DEC; P8←DI;SI←P1;DI←P1;%                               00909900 T  0035:1
                           P7(IF SC≠"0" THEN JUMP REAL TO IA;%                          00910000 T  0036:2
                              DS←LIT " ";SI←SI+1);%                                     00910100 T  0038:0
                       IA:SI←LOC P4;SI←SI-1;IF SC="1" THEN%                             00910200 T  0039:0
                            BEGIN%                                                      00910300 T  0040:0
                                 DI←DI-1;%                                              00910400 T  0040:0
                                 DS←LIT"-";%                                            00910500 T  0040:1
                            END;%                                                       00910600 T  0040:3
                      END;BCTR←P;%                                                      00910700 T  0040:3
            END;%                                                                       00910800 T  0041:2
SUBROUTINE BLNK;%                                                                       00910900 T  0041:3
            BEGIN;%                                                                     00911000 T  0042:0
                STREAM(P3←CSIZE,P2←CSIZE DIV 64,P1←(BCTR←*FIEL INX 0)                   00911100 T  0042:0
                       );%                                                              00911200 T  0044:0
                       BEGIN%                                                           00911300 T  0045:0
                            P2(32(DS←2 LIT" "));%                                       00911400 T  0045:0
                            P3(DS←LIT" ");%                                             00911500 T  0046:3
                       END;%                                                            00911600 T  0048:0
                END;%                                                                   00911700 T  0048:1
                IF FORMT=5 THEN DUMPINT←FIEL ELSE%                                      00911800 T  0048:2
                BEGIN  P(0,0,0     );%                                                  00911900 T  0051:1
                IF M[M[FIEL INX NOT 2] INX 5],[43:1] THEN%                              00912000 T  0052:2
                     P(MKS,0,0,FIEL,1,SELECT);%                                         00912100 T  0056:0
                IF P(MKS,1,0,0,(-1),FIEL,RITEINT,DUP)>16 THEN P(DEL,16) ;               00912200 T  0058:0
                     IF P(DUP) <4 THEN P(XIT);%                                         00912300 T  0061:3
                     P(DUP,8,X,0,0  );BLNK;%                                            00912400 T  0063:1
                BRTN:     GO TO OCSWITCH[FORMT];%                                       00912500 T  0066:0
                TA:  STREAM(BCTR:A←ALFA ); BEGIN DI←BCTR;%                              00912600 T  0069:2
                           SI←LOC A;SI←SI+1; DS←7 CHR END;%                             00912700 T  0071:0
                     RITE;P(XIT);%                                                      00912800 T  0072:0
                TCP: IF (TEMP←TEMP+1)≤N THEN%                                           00912900 T  0073:1
IF P(TEMP-1,NOT,[CV],INX,LOD)≠P(TEMP+N-1,NOT,[CV],%                                     00913000 T  0075:0
                     INX,LOD) THEN P(XIT) ELSE GO TO TCP;%                              00913100 T  0079:0
                TC:  STREAM(BCTR:A←ALFA           ); BEGIN DI←BCTR;%                    00913200 T  0080:2
                           SI←LOC A;SI←SI+1; DS←6 CHR;DS←%                              00913300 T  0082:0
                            1 LIT"[";       BCTR←DI;%                                   00913400 T  0082:3
                            END; BCTR←P;%                                               00913500 T  0083:2
VALUEE←P(N-1,NOT,[CV],INX,LOD);%                                                        00913600 T  0084:1
                     OUTI;%                                                             00913700 T  0086:2
                IRTN: I←0;%                                                             00913800 T  0088:0
                P2:  IF (I←I+1)<N THEN%                                                 00913900 T  0088:3
                     BEGIN;STREAM(B←0   :  BCTR);%                                      00914000 T  0090:2
                           BEGIN DS← 1 LIT",";B←DI    END;BCTR←P;%                     00914100 T  0092:1
                     VALUEE←P(N-I-1,NOT,[CV],INX,LOD);%                                 00914200 T  0093:3
                    OUTI;GO TO P2; END;%                                                00914300 T  0096:2
                P1:  VALUEE←CV;%                                                        00914400 T  0098:2
```

```
                    STREAM(B←0:     BCTR);%                                  00914500 T  0099:1
                    BEGIN DS←2 LIT"]=";B←DI END; BCTR←P;%                    00914600 T  0100:2
                    GO TO P3;%                                               00914700 T  0102:0
                TB: VALUEE←BV;%                                              00914800 T  0102:2
                    STREAM(BCTR:ALFA          ); BEGIN DI←BCTR;%             00914900 T  0103:1
                        SI←LOC ALFA;SI←SI+1; DS←6 CHR;DS←1 LIT"=";%           00915000 T  0104:3
                        BCTR←DI%                                             00915100 T  0106:0
                    END; BCTR←P;     GO TO P3;%                              00915200 T  0106:1
TD:                 STREAM(BCTR:ALFA          ); BEGIN DI←BCTR;%             00915300 T  0107:2
                        SI←LOC ALFA;SI←SI+1; DS←6 CHR;DS←1 LIT"=";%           00915400 T  0109:0
                        BCTR←DI%                                             00915500 T  0110:1
                    END; BCTR←P;%                                            00915600 T  0110:2
                    RITE; BLNK;%                                             00915700 T  0111:1
                TF: P((LNGTH×8) DIV(IF TIPE=0 THEN 6 ELSE IF TIPE=1%         00915800 T  0113:0
                                    THEN 19 ELSE IF TIPE= 2 THEN 9%          00915900 T  0116:1
                                    ELSE 14),0,0,0,0,0,DEL,DEL);%            00916000 T  0119:1
          P([TARRY]&(2×N+1)[8:38:10]);I←0;%                                  00916100 T  0122:0
               TP2: P(0);%                                                   00916200 T  0125:0
               TP3: IF (I←I+1)<N THEN GO TO TP2;%                            00916300 T  0125:1
                    I←0;P(0,,CV,LOD);GO TO TP1;%                             00916400 T  0127:2
               TP22:P(0,,TEMP,LOD,LOD);%                                     00916500 T  0129:2
TP1:P(,TEMP,SND,DIA 8,DIB 38,TRB 10);IF (I←I+1)<N THEN%                      00916600 T  0130:2
                    GO TO TP22;%                                             00916700 T  0133:2
               TP8: I←0;P(,CV,LOD);%                                         00916800 T  0134:0
               TP5: IF(I←I+1)<N THEN%                                        00916900 T  0135:1
                        BEGIN%                                               00917000 T  0137:0
                            P(I,TARRY,CDC,LOD);GO TO TP5%                    00917100 T  0137:2
                        END;%                                                00917200 T  0139:0
                    TROW←P; J←0;%                                            00917300 T  0139:0
               TP11:P(J);VALUEE←TROW;GO TO P3;%                              00917400 T  0140:1
               TD1: IF(J←J+1)<P(N,DUP,+,TARRY) THEN GO TO TP7;%              00917500 T  0141:3
                    RITE;BLNK;%                                              00917600 T  0144:3
               TD2: RITE;BLNK;%                                              00917700 T  0147:0
               TD3: P(0,,COUNT,SND);GO TO TP6;%                              00917800 T  0149:0
TP10:             IF P(N-I,TARRY,2×N-I,TARRY,1,-,=) THEN GO TO TP9;%        00917900 T  0150:1
                    P(N-I,[TARRY],DUP,COC,1,+,XCH,←);GO TO TP8;%             00918000 T  0154:1
               TP9: P(0,N-I,[TARRY],←,I);                                    00918100 T  0157:1
               TP6: IF(I←P(1,+))=N THEN P(XIT) ELSE GO TO TP10;%             00918200 T  0159:0
               TP7: IF(COUNT←COUNT+1)≠NL THEN GO TO TP11;%                   00918300 T  0161:1
                    RITE;BLNK;%                                              00918400 T  0163:2
               TP71:COUNT←0;GO TO TP11;%                                     00918500 T  0166:0
               P3:  GO TO TIPESW[TIPE];%                                     00918600 T  0167:1
               P3A: STREAM(BCTR:VALUEE); BEGIN DI←BCTR;%                     00918700 T  0170:1
                            SI←LOC VALUEE;SI←SI+1;DS←2 LIT" ";DS←7%          00918800 T  0171:3
                        CHR;BCTR←DI;%                                        00918900 T  0172:3
                        END;BCTR←P;GO TO P5;%                                00919000 T  0173:1
               P3L: STREAM(V←VALUEE AND 1:BCTR);                             00919100 T  0174:2
                        BEGIN DS←6 LIT " FALSE";                             00919200 T  0176:1
                            V(DI←DI-5; DS←5 LIT "TRUE ");                    00919300 T  0177:1
                            V←DI;                                            00919400 T  0179:1
                        END; BCTR←P;GO TO P5;%                               00919500 T  0179:2
               P3I: IF VALUEE ≤ @7777777777777 THEN%                         00919600 T  0180:3
                     BEGIN P(VALUEE,,VALUEE,ISN,DUP,0,<,,SINN,←,%            00919700 T  0181:2
                            SSP,,DH2,SND,@1045753604000000,DIV,%             00919800 T  0184:0
                            ,DH1,←);%                                        00919900 T  0185:1
                    STREAM(P8←0:P7←11,P6←[DH1],P5←SINN,P4←4,P3←8,P2←2,%      00920000 T  0185:3
                        P1←BCTR);%                                           00920100 T  0188:0
```

```
                    BEGIN%                                                                00920200 T  0188:2
                         P2(DS←LIT" ");%                                                  00920300 T  0188:2
                         P1←DI;%                                                          00920400 T  0189:3
                         SI←P6;%                                                          00920500 T  0190:0
                         DS←P4 DEC; SI←P6;SI←SI+8;DS←P3 DEC;%                             00920600 T  0190:1
                         P8←DI;SI←P1;DI←P1;%                                              00920700 T  0191:3
                         P7(IF SC≠"0" THEN JUMP REAL TO IA;%                              00920800 T  0192:2
                            DS←LIT " "; SI←SI+1);%                                        00920900 T  0194:0
                      IA:SI←LOC P4;SI←SI-1;IF SC="1" THEN%                                00921000 T  0195:0
                         BEGIN%                                                           00921100 T  0196:0
                            DI←DI-1;DS←LIT "-";%                                          00921200 T  0196:0
                         END;%                                                            00921300 T  0196:3
                    END;BCTR←P; GO TO P5;%                                                00921400 T  0196:3
                   END;%                                                                  00921500 T  0200:0
P3E:               FINDE; DH2←0;%                                                         00921600 T  0200:0
              EB:  IF P(VALUEE,E,11,-,DH2,+,DUP)<0 THEN%                                  00921700 T  0201:3
              BEGIN P(CHS,TENS,MUL); GO TO ED END;%                                       00921800 T  0204:0
         EC:  P(TENS,/);                                                                  00921900 T  0205:3
         ED:  IF P(DUP) ≤ @7777777777777 THEN%                                           00922000 T  0206:1
                BEGIN%                                                                    00922100 T  0207:0
                   P(.DH1,ISN);%                                                          00922200 T  0207:2
                   IF P(DUP) ≥ P(12-DH2,TENS) THEN%                                       00922300 T  0208:0
                     BEGIN%                                                               00922400 T  0209:2
                        P(DEL);%                                                          00922500 T  0210:0
                        P(11-DH2,TENS,.DH1,ISN);                                          00922600 T  0210:1
                        E ← E + 1;%                                                       00922700 T  0211:3
                     END;%                                                                00922800 T  0213:0
                   P(@1045753604000000,IDV,.VALUEE,←);                                    00922900 T  0213:0
                   STREAM(P10←0;P9←ABS(E),P8←(E<0),P7←SINN,P6←DH2,%                        00923000 T  0214:0
                         P5←[VALUEE],P4←4-DH2,P3←8,P2←2,P1←BCTR);%                        00923100 T  0216:2
                        BEGIN%                                                            00923200 T  0218:2
                             P2(DS←LIT" ");P1←DI;SI←LOC P6;SI←SI-1;%                       00923300 T  0218:2
                             IF SC="1"%                                                   00923400 T  0220:2
                              THEN BEGIN%                                                 00923500 T  0220:3
                                      DI←DI-1;DS←LIT"-"%                                  00923600 T  0221:0
                                 END;%                                                    00923700 T  0221:3
                             DI←DI+1;SI←P5;DS←P4 DEC;SI←P5;SI←SI+8;%                      00923800 T  0221:3
                             DS←P3 DEC;%                                                  00923900 T  0223:1
                             P6(DS←LIT"0");%                                              00924000 T  0223:3
                             DS←LIT"@";%                                                  00924100 T  0225:0
                             SI←LOC P7;SI←SI-1;%                                          00924200 T  0225:2
                             IF SC="1" THEN DS←LIT"-" ELSE DS←LIT"+";%                    00924300 T  0226:0
                             SI←LOC P9;DS←2 DEC; P10←DI;SI←P1;SI←SI+1;                    00924400 T  0227:3
                             DI←P1;DS←CHR;DS←LIT".";%                                     00924500 T  0229:0
                        END;BCTR←P;%                                                      00924600 T  0230:0
              P5:  IF FORMT=4 THEN GO TO TD1;%                                            00924700 T  0230:3
                   RITE; P(XIT);                    END;%                                00924800 T  0232:0
                             P(DEL);DH2←DH2+1;GO TO EB;%                                  00924900 T  0233:1
                END;%                                                                     00925000 T  0238:0
END DUMPINT;%                                                                             00925100 T  0238:0
                                                                                                 SIZE= 0239 WORDS


 PROCEDURE XTOTHEIINT(BASE,EXPON,M,LOG,EXP);%                                                    01000000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00205
```

```
                    VALUE BASE,EXPON,M,LOG,EXP;%                                                                    01001000 T  0000:0
                    REAL BASE,EXPON,M,LOG,EXP;%                                                                     01002000 T  0000:0
                    BEGIN LABEL ROWS,MORE,EXIT;%                                                                    01003000 T  0000:0
                         REAL CTR=+1,F2=+2;%                                                                        01004000 T  0000:0
                         IF EXPON = 0 THEN%                                                                         01005000 T  0000:0
                              BEGIN BASE ← 1; P(XIT) END;%                                                          01006000 T  0000:3
                         IF BASE = 0 THEN P(XIT);%                                                                  01007000 T  0002:1
                         IF EXPON.[3:35] ≠ 0 THEN%                                                                  01008000 T  0003:3
                              BEGIN BASE ← P(MKS,BASE,LOG,MKS,CTR,EXPON,×,EXP);%                                    01009000 T  0005:0
                                   P(XIT);%                                                                         01010000 T  0008:0
                              END;%                                                                                 01011000 T  0008:1
                         P(1,EXPON,BASE,DIA 38, DIB 39,EXPON);%                                                     01012000 T  0008:1
ROWS::                   IF P(0,XCH,FCE 9) THEN%                                                                    01013000 T  0009:3
                              BEGIN P(DEL);%                                                                        01014000 T  0010:3
MORE::                   IF (CTR ← CTR-1) = 0 THEN GO TO EXIT;%                                                     01015000 T  0011:2
                                   P(MUL);%                                                                         01016000 T  0014:1
                                   GO TO MORE;%                                                                     01017000 T  0014:2
                         ::%                                                                                        01018000 T  0015:0
                              END;%                                                                                 01019000 T  0015:0
                         P(DEL);%                                                                                   01020000 T  0015:0
                         IF EXPON THEN%                                                                             01021000 T  0015:1
                              BEGIN CTR ← CTR+1;%                                                                   01022000 T  0015:2
                                   P(DUP);%                                                                         01023000 T  0017:1
                              END;%                                                                                 01024000 T  0017:2
                         P(DUP,MUL,0,EXPON,TRB 9,,EXPON,SND);%                                                      01025000 T  0017:2
                         GO TO ROWS;%                                                                               01026000 T  0019:1
EXIT:                    IF F2 < 0 THEN P(1,XCH,/);%                                                                01027000 T  0019:3
                         BASE ← P;%                                                                                 01028000 T  0021:3
                    END;%                                                                                           01029000 T  0022:1
                                                                                                                              SIZE= 0023 WORDS
```

```
PROCEDURE STATUSINT(T,C); VALUE T,C; REAL T; INTEGER C;                                 01100000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00206
     BEGIN P(T,C,28,COM,DEL,RTN) END;                                                   01101000 T  0000:0
                                                                                              SIZE= 0002 WORDS
```

```
REAL PROCEDURE ABSINT(X); VALUE X; REAL X;%                                             01200000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00207
     BEGIN P(ABS(X),RTN) END;%                                                          01201000 T  0000:0
                                                                                              SIZE= 0002 WORDS
```

```
REAL PROCEDURE SIGNINT(X); VALUE X; REAL X;%                                            01300000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00208
     BEGIN P(SIGN(X),RTN) END;%                                                         01301000 T  0000:0
                                                                                              SIZE= 0003 WORDS
```

```
INTEGER PROCEDURE ENTIERINT(X); VALUE X; REAL X;%                                    01400000 T  0000:0
                                                                          START OF REL SEGMENT; DISK ADDRESS = 00209
        BEGIN ENTIERINT ← X-.5 END;%                                                 01401000 T  0000:0
                                                                                            SIZE= 0003 WORDS


REAL PROCEDURE TIMEINT (X); VALUE X; REAL X;%                                       01500000 T  0000:0
                                                                          START OF REL SEGMENT; DISK ADDRESS = 00210
        BEGIN P(X,1, COM,RTN) END;%                                                  01501000 T  0000:0
                                                                                            SIZE= 0002 WORDS


PROCEDURE DELAYINT(ARRY, MASK, TIME);%                                      %WF      01600000 T  0000:0
                                                                          START OF REL SEGMENT; DISK ADDRESS = 00211
   VALUE ARRY, MASK, TIME;%                                                 %WF      01601000 T  0000:0
   ARRAY ARRY[*]; REAL MASK; INTEGER TIME;%                                 %WF      01602000 T  0000:0
   BEGIN POLISH(ARRY, MASK, TIME, 31, COM, DEL, DEL, RTN) END;%             %WF      01603000 T  0000:0
                                                                                            SIZE= 0003 WORDS


PROCEDURE SQRTINT(X); VALUE X; REAL X;%                                             01700000 T  0000:0
                                                                          START OF REL SEGMENT; DISK ADDRESS = 00212
    BEGIN REAL Y=+1,Z=+2;%                                                       01701000 T  0000:0
        LABEL P5,ONE;%                                                           01702000 T  0000:0
        DEFINE INNER = XCH,MUL,DUP,Y,XCH,/#,%                                    01703000 T  0000:0
            ITER = P(+,P5,INNER)#;%                                              01704000 T  0000:0
        IF X<0 THEN P(1,26,COM); % ARGUMENT CHECK                       %IA      01705000 T  0000:0
        IF P(ABS(X),DUP) ≠ 0 THEN%                                               01706000 T  0002:0
            BEGIN P(ONE,+,DUP,0,DEL,%                                            01707000 T  0003:1
                    DIA 7, DIB 45, VFI 3 7, Y,%                                  01708000 T  0005:0
                    DIA 2, TRB 1,,ONE,+,LOD,%                                    01709000 T  0006:0
                    Y,DUP,DIB 3,TRB 6,XCH,INNER);%                               01710000 T  0007:1
                ITER;ITER;ITER;%                                                 01711000 T  0010:0
                P(Z,-,P5,×,+);%                                                  01712000 T  0016:0
            END ;%                                                               01713000 T  0017:1
        P(RTN);%                                                                 01714000 T  0017:1
    P5:::   @1154000000000000;%                                                  01715000 T  0017:2
ONE:::  @1770000000000001,%                                                      01716000 T  0019:0
            @1235560000000000,%                                                  01717000 T  0020:0
            @1233250000000000,%                                                  01718000 T  0021:0
            @1222000000000000,%                                                  01719000 T  0022:0
            @1221150000000000,%                                                  01720000 T  0023:0
            @0155560000000000,%                                                  01721000 T  0024:0
            @0153250000000000,%                                                  01722000 T  0025:0
            @0152000000000000,%                                                  01723000 T  0026:0
            @0151150000000000;%                                                  01724000 T  0027:0
    END;%                                                                        01725000 T  0028:0
                                                                                            SIZE= 0029 WORDS
```

```
DEFINE SINCOSBODY =%                                                                    01800000 T  0000!0
                P(1);%                                                                  01801000 T  0000!0
                IF X < 0 THEN%                                                          01802000 T  0000!0
                     BEGIN X ← -X; P(CHS) END;%                                         01803000 T  0000!0
                IF X ≥ P(PI) THEN%                                                      01804000 T  0000!0
                     BEGIN  P(NOP);%                                                    01805000 T  0000!0
                          IF P(X/P(PI)-P(HALF), DUP)>P(MAXI)              %WF       01806000 T  0000!0
                            THEN P(LITERAL, 26, COM);                     %WF       01807000 T  0000!0
                          IF I ← POLISH THEN P(CHS);                      %WF       01808000 T  0000!0
                          X ← X MOD P(PI);%                                         01809000 T  0000!0
                     END;%                                                          01810000 T  0000!0
                IF X ≥ P(PIHAF) THEN%                                               01811000 T  0000!0
                     BEGIN P(CHS);%                                                 01812000 T  0000!0
                          X ← X-P(PI);%                                             01813000 T  0000!0
                     END;%                                                          01814000 T  0000!0
                IF ABS(X) < .000001 THEN P(ZxX,RTN);%                               01815000 T  0000!0
                P(X,DUP,%                                                           01816000 T  0000!0
                x,DUP,K1,NOP,x,K2,-,T,x,K3,+,T,x,K4,-,T,x,K5,+,T,x,K6,-,T,          01817000 T  0000!0
                x,1.0,+,X,x,Z,XCH,x,RTN);%                                          01818000 T  0000!0
           PI   :::@ 114311037524210;%                                              01819000 T  0000!0
           PIHAF:::@ 114144417652104;%                                              01820000 T  0000!0
           HALF :::@ 115400000000000;%                                              01821000 T  0000!0
           K1   :::@ 127124523443111 3;%                                            01822000 T  0000!0
           K2   :::@ 125327000532062 4;%                                            01823000 T  0000!0
           K3   :::@ 123561671620117 7;%                                            01824000 T  0000!0
           K4   :::@ 121640063763415 0;%                                            01825000 T  0000!0
           K5   :::@ 117421042104110 2;%                                            01826000 T  0000!0
           K6   :::@ 115125252525252 4;%                                            01827000 T  0000!0
           MAXI :::@ 000777777777777 7;%                                  %WF       01828000 T  0000!0
      #;%                                                                           01829000 T  0000!0
PROCEDURE SININT(X); VALUE X; REAL X;%                                              01830000 T  0000!0
START OF REL SEGMENT; DISK ADDRESS = 00213
                BEGIN REAL T=+2,Z=+1;%                                              01831000 T  0000!0
                     INTEGER I=T;%                                                  01832000 T  0000!0
                     LABEL PI,PIHAF,HALF;%                                          01833000 T  0000!0
                     LABEL K1,K2,K3,K4,K5,K6;%                                      01834000 T  0000!0
                     LABEL MAXI;%                                       %WF         01835000 T  0000!0
                     DEFINE LITERAL = 4#;                               %WF         01836000 T  0000!0
                     ;SINCOSBODY;%                                                  01837000 T  0000!0
                END;%                                                               01838000 T  0034!0
                                                                                SIZE= 0037 WORDS


PROCEDURE COSINT(X); VALUE X; REAL X;%                                              01839000 T  0000!0
START OF REL SEGMENT; DISK ADDRESS = 00215
                BEGIN REAL T=+2,Z=+1;%                                              01840000 T  0000!0
                     INTEGER I=T;%                                                  01841000 T  0000!0
                     LABEL PI,PIHAF ,HALF;%                                         01842000 T  0000!0
                     LABEL K1,K2,K3,K4,K5,K6;%                                      01843000 T  0000!0
                     LABEL MAXI;                                        %WF         01844000 T  0000!0
                     DEFINE LITERAL = 5#;                               %WF         01845000 T  0000!0
                     X ← X+P(PIHAF ,NOP,NOP,NOP);%                                  01846000 T  0000!0
                     SINCOSBODY;%                                                   01847000 T  0002!0
                END;%                                                               01848000 T  0036!0
                                                                                SIZE= 0039 WORDS
```

```
COMMENT ARCTAN INTRINSIC FOR ESPOL;%                                                                    01900000 T  0000:0
REAL PROCEDURE ARCTANINT(X1);%                                                                          01901000 T  0000:0
                                                                                    START OF REL SEGMENT; DISK ADDRESS = 00217
     VALUE X1; REAL X1;%                                                                                01902000 T  0000:0
BEGIN REAL T=+1,D,PI2,ARCY;%                                                                            01903000 T  0000:0
     LABEL L1,ONEL,PIHAF,A,B,ARCA,ARCB,TENM6;%                                                          01904000 T  0000:0
     LABEL K1,K2,K3,K4,K5,K6,K7;%                                                                       01905000 T  0000:0
     DEFINE ONE = P(ONEL)#;%                                                                            01906000 T  0000:0
     REAL X=X1;%                                                                                        01907000 T  0000:0
     P(DIA 1,DIB 1);%                                                                                   01908000 T  0000:0
     IF (T ← ABS(X)) > ONE THEN%                                                                        01909000 T  0001:2
          BEGIN PI2 ← P(PIHAF,X,TRB 1);%                                                                01910000 T  0003:0
               IF T ≥ P(L1) THEN P(X←0)%                                                                01911000 T  0004:3
               ELSE P(ABS(X←-(ONE/X)));%                                                                01912000 T  0006:3
               T ← P;%                                                                                  01913000 T  0009:0
          END;%                                                                                         01914000 T  0009:2
     IF T < P(TENM6) THEN P(X+PI2,RTN);%                                                                01915000 T  0009:2
     IF T > P(K1) THEN%                                                                                 01916000 T  0011:3
          BEGIN IF T < P(K2) THEN P(A,ARCA) ELSE P(B,ARCB);%                                            01917000 T  0012:2
               D ← P(X,TRB 1,,ARCY,SND,TRB 1);%                                                         01918000 T  0015:3
               X ← (X-D)/(D×X+ONE);%                                                                    01919000 T  0017:2
          END;%                                                                                         01920000 T  0020:1
     P(X,DUP,%                                                                                          01921000 T  0020:1
     X,,T,SND,K3,×,K4,+,T,×,K5,-,T,×,K6,+,T,×,K7,-,T,×,ONE,+,%                                          01922000 T  0020:3
     X,×,+,+,RTN);%                                                                                     01923000 T  0026:2
ONEL :::@ 1141000000000000;%                                                                            01924000 T  0027:3
L1   :::@ 0631000000000000;%                                                                            01925000 T  0029:0
K1   :::@ 1151210574175662;%                                                                            01926000 T  0030:0
K2   :::@ 1154047010241407;%                                                                            01927000 T  0031:0
K3   :::@ 3165354424670553;%                                                                            01928000 T  0032:0
K4   :::@ 1167063634367006;%                                                                            01929000 T  0033:0
K5   :::@ 1151111104736450;%                                                                            01930000 T  0034:0
K6   :::@ 1151463146300126;%                                                                            01931000 T  0035:0
K7   :::@ 1152525252525235;%                                                                            01932000 T  0036:0
PIHAF:::@ 1141444176652104;%                                                                            01933000 T  0037:0
A    :::@ 1152462675773223;%                                                                            01934000 T  0038:0
B    :::@ 1155637726073171;%                                                                            01935000 T  0039:0
ARCA :::@ 1152406627566472;%                                                                            01936000 T  0040:0
ARCB :::@ 1155015457355165;%                                                                            01937000 T  0041:0
TENM6:::@ 1232061573640554;%                                                                            01938000 T  0042:0
END;%                                                                                                   01939000 T  0043:0
                                                                                                              SIZE= 0044 WORDS


     COMMENT  LN INTRINSIC FOR ESPOL;%                                                                  02000000 T  0000:0
PROCEDURE LNINT(X); VALUE X; REAL X;%                                                                   02001000 T  0000:0
                                                                                    START OF REL SEGMENT; DISK ADDRESS = 00219
     BEGIN LABEL L1,L2,L3,K15,K16,K17,K18,K19;%                                                         02002000 T  0000:0
          LABEL KON,K1,K2,K3,K4,K5,K6,K7,K8,K10,K11,K12,K13,K14;%                                       02003000 T  0000:0
          LABEL MIN;%                                                                                   02004000 T  0000:0
          DEFINE ONE = P(KON)#;%                                                                        02005000 T  0000:0
```

```
     IF X≤0 THEN P(X,0,=,DUP,+,26,COM); % ARGUMENT CHECK     %IA
     IF (X ← ABS(X+P(MIN)))> P(K1) THEN%
     IF X < P( K2) THEN%
          BEGIN P(0,0);%
L3:            P(X);%
               GO TO L2;%
          END;%
     P(X,[3:6]&X[1:2:1]+12,DUP,K3,×,XCH,DUP,+);%
     IF (X←X&76[2:41:7]) ≥ P(K7) THEN BEGIN%
     IF X ≥P(K10) THEN BEGIN P(K14,+,K13); GO TO L1 END;%
          BEGIN P(K12,+,K11); GO TO L1 END; END;%
     IF X ≥ P(K4) THEN%
          BEGIN P(ONE,+,K8); GO TO L1 END;%
     IF X ≤ P(K2) THEN GO TO L3;%
     P(K6,+,K5);%
L1:: P(X,×);%
L2:  P(ONE,-,DUP,K14,+,%
     /,DUP,DUP,NOP,%
     ×,,X,SND,K15,×,K16,+,X,×,K17,+,X,×,K18,+,X,×,K19,+,%
     X,×,K14,+,XCH,×,+,+,RTN);%
MIN:::@1770000000000001;%
KON:::@1141000000000000;%
K1 :::@1156165757475261;%
K2 :::@1141221327436077;%
K4 :::@1142073716664320;%
K3 :::@1165053107716726;%
K5 :::@1154664262770676;%
K6 :::@1154000000000000;%
K7 :::@1143373034355542;%
K8 :::@1152742653066132;%
K10:::@1145602266440557;%
K11:::@1151621741671113;%
K12:::@1141400000000000;%
K13:::@1151052252521677;%
K14:::@1142000000000000;%
K15:::@1151406657727033;%
K16:::@1151615542107107;%
K17:::@1152222224366610;%
K18:::@1153146314625377;%
K19:::@1155252525252530;%
END;%
```

```
                                                                02006000 T  0000:0
                                                                02007000 T  0003:0
                                                                02008000 T  0005:0
                                                                02009000 T  0006:1
                                                                02010000 T  0007:1
                                                                02011000 T  0007:2
                                                                02012000 T  0008:0
                                                                02013000 T  0008:0
                                                                02014000 T  0011:3
                                                                02015000 T  0014:2
                                                                02016000 T  0017:0
                                                                02017000 T  0018:1
                                                                02018000 T  0019:0
                                                                02019000 T  0020:3
                                                                02020000 T  0022:0
                                                                02021000 T  0022:3
                                                                02022000 T  0023:2
                                                                02023000 T  0024:3
                                                                02024000 T  0025:3
                                                                02025000 T  0030:2
                                                                02026000 T  0032:3
                                                                02027000 T  0034:0
                                                                02028000 T  0035:0
                                                                02029000 T  0036:0
                                                                02030000 T  0037:0
                                                                02031000 T  0038:0
                                                                02032000 T  0039:0
                                                                02033000 T  0040:0
                                                                02034000 T  0041:0
                                                                02035000 T  0042:0
                                                                02036000 T  0043:0
                                                                02037000 T  0044:0
                                                                02038000 T  0045:0
                                                                02039000 T  0046:0
                                                                02040000 T  0047:0
                                                                02041000 T  0048:0
                                                                02042000 T  0049:0
                                                                02043000 T  0050:0
                                                                02044000 T  0051:0
                                                                02045000 T  0052:0
                                                                02046000 T  0053:0
                                                                        SIZE= 0054 WORDS
```

```
     COMMENT EXP INTRINSIC FOR ESPOL;%
REAL  PROCEDURE  EXPINT(X) ; VALUE X  ; REAL X;%
     BEGIN%
     REAL   Q = +4,  Z =+1, EX = +2, B=+3, Y=+5, T = +2;%
     LABEL  K0,K1,K2,K3,K4,K5,K6,HALF;%
     LABEL MAX;
     IF  X < P(K0) THEN  P(RTN);%
     IF X>P(MAX) THEN P(3, 26, COM);                            %WF
     P( X,K1,×,,X,SND, HALF,-,,Z, ISN,CHS,X,+,,X,SND,%
     DUP,NOP,×,DUP,K2,NOP,×,K3,+,T,×,K4,+,T,K5,+,T,NOP,×,%
```

```
                                                                02100000 T  0000:0
                                                                02101000 T  0000:0
                                                  START OF REL SEGMENT; DISK ADDRESS = 00221
                                                                02102000 T  0000:0
                                                                02103000 T  0000:0
                                                                02104000 T  0000:0
                                                                02105000 T  0000:0
                                                                02106000 T  0000:0
                                                                02107000 T  0001:3
                                                                02108000 T  0003:3
                                                                02109000 T  0007:1
```

```
      K6,+,X,x,DUP,B,+,B,Q,=,NOP,/,DUP,0,DEL,P,[3:6]&Y[1:2:1],%
        Z,3,DIV,+, ,EX , SND, P & P[2:1:1]&EX[3:42:6],%
        Z,3, MOD,DUP,+, ,EX, SND,0,≠  ) ;%
      IF P THEN%
           BEGIN  IF Z < 0 THEN P( EX,/,CHS,RTN);%
      P  ( EX, x ); END;   P (RTN);%
KC:::           @ 3121520000000000  ;%
K1:::           @ 1141342521662454  ;%
K2:::           @ 1135326737175655  ;%
K3:::           @ 1102360633500106  ;%
K4:::           @ 1075621717466364  ;%
K5:::           @ 1111554324131444  ;%
K6:::           @ 1072002411247315  ;%
HALF:::         @ 1154000000000000  ;%
MAX:::          @ 1122360000000000  ;
      END   EXP   INT  ;%
```

```
02110000 T  0012:0
02111000 T  0017:1
02112000 T  0020:2
02113000 T  0022:3
02114000 T  0022:3
02115000 T  0025:2
02116000 T  0026:1
02117000 T  0028:0
02118000 T  0029:0
02119000 T  0030:0
02120000 T  0031:0
02121000 T  0032:0
02122000 T  0033:0
02123000 T  0034:0
02124000 T  0035:0
02125000 T  0036:0
                SIZE= 0037 WORDS
```

```
PROCEDURE GOTOSOLVERINT(L,X,F,B);%

      VALUE L,X,F,B;REAL L,X,B;ARRAY F[*];%
      BEGIN IF L ≠ 15 THEN
           L  ←  L&(F)[18:33:15]&B[8:38:10];%
      END;%
```

```
02200000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00223
02201000 T  0000:0
02202000 T  0000:0
02203000 T  0000:3
02204000 T  0003:3
                SIZE= 0004 WORDS
```

```
REAL PROCEDURE MAXINT(X);%

   VALUE X;  REAL X;%
   BEGIN REAL RCW=+0, SIZE=+1, JUNK=+2;%
        POLISH(RCW, FCX, [RCW] INX NOT 1 INX 0, XCH, SUB, 0, X);%
        WHILE SIZE>0 DO BEGIN P(DUP);%
             JUNK ← *(P(,X)+SIZE);%
             IF POLISH<(JUNK ← JUNK) THEN P(DEL, DUP);%
             SIZE ← SIZE-1;%
        END;%
        POLISH(RTN);%
   END MAXINT;%
```

```
%WF         02300000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00224
%WF         02301000 T  0000:0
%WF         02302000 T  0000:0
%WF         02303000 T  0000:0
%WF         02304000 T  0003:1
%WF         02305000 T  0004:3
%WF         02306000 T  0006:1
%WF         02307000 T  0008:1
%WF         02308000 T  0009:2
%WF         02309000 T  0010:0
%WF         02310000 T  0010:1
                SIZE= 0011 WORDS
```

```
REAL PROCEDURE MININT(X);%

   VALUE X;  REAL X;%
   BEGIN REAL RCW=+0, SIZE=+1, JUNK=+2;%
        POLISH(RCW, FCX, [RCW] INX NOT 1 INX 0, XCH, SUB, 0, X);%
        WHILE SIZE>0 DO BEGIN P(DUP);%
             JUNK ← *(P(,X)+SIZE);%
             IF POLISH>(JUNK ← JUNK) THEN P(DEL, DUP);%
```

```
%WF         02400000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00225
%WF         02401000 T  0000:0
%WF         02402000 T  0000:0
%WF         02403000 T  0000:0
%WF         02404000 T  0003:1
%WF         02405000 T  0004:3
%WF         02406000 T  0006:1
```

```
            SIZE ← SIZE-1;%                                                    %WF          02407000 T  0008:1
        END;%                                                                  %WF          02408000 T  0009:2
        POLISH(RTN);%                                                          %WF          02409000 T  0010:0
    END MININT;%                                                               %WF          02410000 T  0010:1
                                                                                                    SIZE= 0011 WORDS



PROCEDURE SUPERMOVERINT(SORCE, DEST, AEXP);%                                   %WF          02500000 T  0000:0
                                                                                      START OF REL SEGMENT; DISK ADDRESS = 00226
    VALUE AEXP;  INTEGER AEXP;  ARRAY SORCE[*], DEST[*];%                      %WF          02501000 T  0000:0
    BEGIN INTEGER T=+1;%                                                       %WF          02502000 T  0000:0
        POLISH(SORCE,[8:10], DEST,[8:10]);%                                    %WF          02503000 T  0000:0
        IF P(DUP)<T THEN P(XCH);%                                              %WF          02504000 T  0002:0
        IF P(DEL, DUP)>AEXP THEN T ← AEXP;%                                    %WF          02505000 T  0003:2
        IF T>0 THEN                                                            %WF          02506000 T  0005:3
        STREAM(P4←P, P3←P(DUP),[36:6], P2←[SORCE[0]]:P1←[DEST[0]]);%           %WF          02507000 T  0006:2
        BEGIN SI←P2;  P3(DS←32 WDS;  DS←32 WDS);  DS←P4 WDS;  END;%            %WF          02508000 T  0009:2
    END SUPERMOVERINT;%                                                        %WF          02509000 T  0011:3
                                                                                                    SIZE= 0012 WORDS



PROCEDURE COBOLFCR;                                                                      02600000 T  0000:0
                                                                                      START OF REL SEGMENT; DISK ADDRESS = 00227
BEGIN                                                                                    02600100 T  0000:0
    REAL CODE       ==1;      % 0=INVALID,1=OPEN INPUT,2=OPEN REV IN                    02600110 T  0000:0
                              % 3=OPEN OUT,4=CLOSE,5=OPEN I-O,6=SORT                    02600120 T  0000:0
                              % 7=CLOSE CRUNCH,16=OPEN1,17=CLOSE1                       02600200 T  0000:0
    NAME FLOC       ==2;      % POINTER TO FIB DESCRIPTOR                              02600300 T  0000:0
    REAL MKSCW      ==3;      % = MKSCW :NO REEL,= 1 FOR REEL CLOSE                    02600400 T  0000:0
                              % = REEL # FOR REEL OPEN.                                02600410 T  0000:0
    REAL CLOSELOCK  ==4;      % HOW TO CLOSE THE FILE                                  02600420 T  0000:0
                              %   0 = REWIND (RETAIN)                                  02600430 T  0000:0
                              %   1 = NO REWIND (RETAIN)                               02600450 T  0000:0
                              %   2 = LOCK (SAVE)                                      02600470 T  0000:0
                              %   4 = PURGE  LOCK (RELEASE + PURGE)                    02600480 T  0000:0
                              %   6 = RELEASE LOCK (RELEASE + LOCK)                    02600500 T  0000:0
                              %   7 = RELEASE (LOOK AT SAVE FACTOR)                    02600510 T  0000:0
                              %  64 = CRUNCH                                           02600515 T  0000:0
% PRT DESCRIPTORS                                                                      02600600 T  0000:0
    REAL COBOLCONTROL =23,    % COBOL 61: FOR CALLING USE ROUTINES                     02600650 T  0000:0
         COBOLINDEX   =22,    % COBOL 61: FOR CALLING USE ROUTINES                     02600700 T  0000:0
         COBOLIO      =14,    % COBOL READ WRITE                                       02600800 T  0000:0
         FCR          =12,    % COBOL FCR                                              02600900 T  0000:0
         PERFORMGEN   =13;    % COBOL 68: FOR PERFORMING USE ROUTNS                    02600990 T  0000:0
    REAL INTINT       =5;     % ARRAY DEC INTRINSIC                                    02600994 T  0000:0
    NAME  MEM         =2;     % DUMMY DATA DESCRIPTOR                                  02600995 T  0000:0
    ARRAY FPB         =3[*],  % FILE PARAMETER BLOCK                                   02601000 T  0000:0
         PGUSE        =24[*]; % USE ROUTINES ARRAY - COB61:13 WDS                      02601100 T  0000:0
                              %                      COB68: 6 WDS                      02601400 T  0000:0
% LOCALS                                                                              02601500 T  0000:0
    REAL REEL;                % MUST BE HERE FOR MKSCW DIDDLE                          02601600 T  0000:0
    ARRAY FIB[*];             % FILE INFO BLOCK                                        02601700 T  0000:0
    REAL I;                   % INDEX + TEMPORARY                                      02601800 T  0000:0
```

```
      NAME IOD=I;                 % IO DESCRIPTORS FOR CLOSE                    02601900 T  0000:0
      REAL IX;                    % INDEX TO FPB                                02601950 T  0000:0
      ARRAY LBL[*];               %LABEL FOR BUILDLABEL+HEADER FOR CLOSE       02602000 T  0000:0
      REAL NOTSERL;               % SET TRUE FOR RANDOM & IO FILES             02602050 T  0000:0
      INTEGER PU1,PU2,            % USED BY BUILDLABEL + USERS.DONT MOVE       02602100 T  0000:0
              FU1,FU2;            % USED BY BUILDLABEL + USERS.DONT MOVE       02602150 T  0000:0
      REAL RPU1 = PU1,            % USED BY BUILDLABEL                          02602199 T  0000:0
           RPU2 = PU2;            % USED BY BUILDLABEL                          02602200 T  0000:0
      REAL T,                     % TEMPORARY                                   02602250 T  0000:0
           TEST,                  % TRUE WHEN CALLING USERS,USERS68 SAYS        02602300 T  0000:0
                                  % TEST FOR BEG OR END FILE USE ROUTINE        02602350 T  0000:0
      SVFIB,                % TRUE IF COBOL68 FIB IS TO BE SAVED               02602360 C  0000:0
           COB68;                 % TRUE IF THIS IS COBOL 68                    02602400 T  0000:0
DEFINE                                                                         02602410 T  0000:0
      AF                = [12:12]#,        % COB68: FILE USE ROUTINE           02602430 T  0000:0
      ALGOLIO(ALGOLIO1)=P([IOD],ALGOLIO1,11,COM,DEL,DEL)#,                     02602440 T  0000:0
      ARR               = [36:12]#,        % COB68: REEL USE ROUTINE           02602450 T  0000:0
      BACKSPACE         = P((-I),[FLOC[3]] INX 0,9,11,COM)#,                   02602455 T  0000:0
      BCOUNT            = FIB[6]#,         % BLOCK COUNT                       02602460 T  0000:0
      BF                = [1:11]#,         % COB68: FILE USE ROUTINE           02602470 T  0000:0
      BOUNDED           = FIB[9].[2:1]#,   % TRUE IF BOUNDED FROM ABOVE         02602480 T  0000:0
      BREAKFAIL         = P(FIB[15].[25:5],(T=1)×4,12,COM)#,%BR OUT FAIL         02602490 T  0000:0
      BUFFERSIZE        = FIB[18].[3:15]#, % BUFFER SIZE REQUESTED             02602500 T  0000:0
      BUFREQ            = FIB[13].[1:9]#,  % NO. OF BUFFERS REQUESTED           02602510 T  0000:0
      BUFTOP            = FIB[16]#,% COPY OF TOP IOD:POINTS TO BEG BUFFR       02602520 T  0000:0
      BRR               = [24:12]#,        % COB68: REEL USE ROUTINE           02602530 T  0000:0
      CALLHASH(CALLHASH1)=P(MKS,FLOC,*FIB[8],CALLHASH1,COC)#,                  02602550 T  0000:0
      CLOSE             = 4#,                                                  02602560 T  0000:0
      CLOSED            =(FIB[5].[41:2]≠0)#,%FILE CLOSED                       02602570 T  0000:0
      CLOSEDHERE        = FIB[8].[1:1]#,   % COB68 CLOSE HERE WAS DONE         02602575 T  0000:0
      CLOSEDRET         = @20#,            % CLOSED RETAINED                   02602580 T  0000:0
      COBOLCLOSE        = P(CLOSELOCK&REEL [2:47:1],FLOC,CODE,13,COM,          02602585 T  0000:0
                          DEL,DEL,DEL)#,                                       02602586 T  0000:0
      COBOLFILE         = FIB[13]#,        % ON SAYS FILE IS COBOL             02602590 T  0000:0
      COBOLFILBIT       = FIB[13].[47:1]#,                                     02602600 T  0000:0
      COBOLOPENIN       = P(REEL,FLOC,CODE,13,COM,,I,←,DEL,DEL)#,              02602610 T  0000:0
      COBOLOPENOUT      = P(REEL,FLOC,CODE,13,COM,DEL,DEL,DEL)#,               02602620 T  0000:0
      COUNT             = FIB[12]#,  % NOTSERL : NO. OF CURRENT BLOCK          02602630 T  0000:0
                          % SERIAL IN(OUT): RECORD COUNT WITHIN BLOCK           02602640 T  0000:0
      CURRENTREEL       = FIB[13].[28:10]#,% CURRENT REEL NUMBER               02602650 T  0000:0
      DIRECTION         = (FIB[13].[25:1])#,% 1=REVERSE,0=FORWARD              02602660 T  0000:0
      DISCARDWA         = P(MEM OR ((*RCPRT).[FF]),3,COM,DEL)#,                02602670 T  0000:0
      DISK              = FIB[4].[8:4]=4#,                                     02602680 T  0000:0
      DISKR             = 10#,             % DISK RANDOM (FPB)                 02602690 T  0000:0
      DISKS             = 12#,             % DISK SERIAL (FPB)                 02602700 T  0000:0
      DISKP             = 26#,             % DISK PROTECT(FPB)                 02602705 T  0000:0
      ENDFILE           = FIB[5].[40:1]#,  % RECOGNIZED END OF FILE             02602710 T  0000:0
      EOF               = [27:1]#,         % EOF BIT IN IOD                    02602720 T  0000:0
      EORF              = [42:6]#,         % SENTINEL: 1=EOR  0=EOF            02602730 T  0000:0
      EORRERUN          = FIB[4].[3:2]#,%EOR RERUN:1=OUTPUT TAPE,2=SCRCH       02602740 T  0000:0
      FPBXDONE          = FIB[4].[12:1]#,  % [13:11] IS FPB INDEX              02602770 T  0000:0
      FCRCLOSE(FCRCLOSE1)=P(MKS,FCRCLOSE1,0,[FLOC],4,FCR)#,                    02602780 T  0000:0
      FCROPENOUT        = P(MKS,T,[FLOC],3,FCR)#,                              02602790 T  0000:0
      FILIO             = FIB[13].[22:1]#, % FILE OPEN IO                      02602795 T  0000:0
      FPBTYPE           = FPB[IX+3].[43:5]#,% FPB FILE TYPE                    02602798 T  0000:0
      GETDISKROW        = P(FPB[IX+3],FPB[IX],FPB[IX+1],10,LBL,                02602800 T  0000:0
                              4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL)#,              02602805 T  0000:0
```

```
HASH            = (FIB[8]≠0)#,      % COB61: HASH ROUTINES PRESNT        02602810 T  0000:0
HEADERPTR       = FIB[14]#,         % DESC. FOR DISK FILE HEADER         02602820 T  0000:0
HNMROWS         = LBL[9]#,          % HEADER: NUMBER OF ROWS             02602830 T  0000:0
                                    % (DO NOT CHANGE)                   02602831 T  0000:0
HNMSZRS         = NOTSERL#,         % HEADER: SIZE OF ROWS               02602840 T  0000:0
INFILE          = FIB[13].[27:1]#,  % FILE OPEN INPUT                    02602850 T  0000:0
IODONE          = FLOC[I+2].[19:1]#,% DONE BIT ON IN IOD                 02602860 T  0000:0
IOERR(IOERR1)   = P(0,FLOC,IOERR1,17,COM)#, % CALL IOERR-DONT DS         02602865 T  0000:0
LABELED         = NOT UNLABELED#,                                        02602870 T  0000:0
LABEQ           = FIB[5].[17:1]#,   % LABEL EQUATED FROM DISK           02602880 T  0000:0
LASTIO          = FIB[13].[46:1]#,  % 1=LAST WAS PHYSICAL READ           02602885 T  0000:0
LBLPTR          = FLOC[1]#,         % LABEL DESCRIPTOR                   02602890 T  0000:0
LOCK            = 2#,                                                    02602900 T  0000:0
LSUBL           = FIB[1]#,          % DISK: LOWER BOUND RECORD NO        02602910 T  0000:0
LSUBU           = FIB[3]#,          % DISK: UPPER BOUND RECORD NO        02602920 T  0000:0
MABUSE          = FIB[4].[1:1]#,    % USE ROUTINES PRESENT               02602930 T  0000:0
MAXR            = FIB[18][8:38:10]#, % MAX REC SZ FOR CONCATS            02602945 T  0000:0
MAXREC          = FIB[18].[CF]#,    % MAXIMUM RECORD LENGTH              02602950 T  0000:0
MINREC          = FIB[18].[FF]#,    % MINIMUM RECORD SIZE                02602952 T  0000:0
MT              = 2#,               % MAGNETIC TAPE                     02602955 T  0000:0
NMSZROWS        = FIB[8].[20:28]#,  % DISK: NM=[20:5],SZ=[25:23]          02602960 T  0000:0
NOAIT           = FIB[20].[3:1]#,   % AIT FOR WA WAS DESTROYED           02602965 T  0000:0
NOREW           = 1#,               % NO REWIND                         02602970 T  0000:0
NOTCLOSED       = FIB[5].[41:2]=0#,% FILE NOT CLOSED                     02602980 T  0000:0
NOTFIRSTREEL    = FIB[5].[38:1]#,   % =1 IFF CURRENTREEL≠1ST REEL        02602985 T  0000:0
NOTINANDOPEN    = FIB[5].[41:3]≠1#,% FILE NOT(INPUT & OPEN)              02602990 T  0000:0
NUMBUFF         = FIB[13].[10:9]#,  % NO.OF BUFFERS ASSIGNED             02603000 T  0000:0
NUMREC          = FIB[11]#,         % NO.OF RECORDS PER BLOCK           02603010 T  0000:0
OPENIN          = 1#,                                                    02603020 T  0000:0
OPENIO          = 5#,                                                    02603025 T  0000:0
OPENOUT         = 3#,                                                    02603030 T  0000:0
OPTIONAL        = FIB[5].[39:1]#,   % REEL OPTIONAL AND ABSENT          02603040 T  0000:0
OUTAP           = T#,               % EOR RERUN ON OUTPUT TAPE          02603050 T  0000:0
PBIT            = [2:1]#,           % PRESENCE BIT                      02603051 T  0000:0
PBT             = 7#,                                                    02603055 T  0000:0
PERFORMUSE      = P(MKS,[FIB],T,0,PERFORMGEN)#,                          02603060 T  0000:0
PRINTFILE       = FIB[20] #,        % CF=1 IS PRINTFILE                 02603070 T  0000:0
PURGEREEL       = P([[FLOC[3]]&@23[CTF],20,11,COM,DEL,DEL,DEL)#,         02603080 T  0000:0
PURGE           = 4#,                                                    02603090 T  0000:0
RANDOM          = FIB[4].[29:1]#,   % RANDOM ACCESS IS THE ORDER         02603100 T  0000:0
RCOUNT          = FIB[7]#,          % NO.OF RECORDS INTO FILE           02603110 T  0000:0
RCPRT           = FIB[20].[FF]#,    % PRT OF DESC POINTING TO REC       02603115 T  0000:0
RECSPERBLK      = LBL[0].[30:12]#,  % HEADER: RECORDS PER BLOCK          02603120 T  0000:0
REDECWA         = P(MKS,RCPRT,MAXREC,1,1,1,INTINT)#,  % DECLARE          02603130 T  0000:0
                                    % SAVE ARRAY FOR WORK AREA          02603140 T  0000:0
RELEASE         = 7#,                                                    02603200 T  0000:0
RESETPARITY     = FLOC[3]← (*P(DUP))&0[28:28:1]#, % RESET PARITY         02603202 T  0000:0
RESETREADBIT    = 0[24:24:1]#,      % USED TO TURN OFF READ BIT         02603205 T  0000:0
REWIND          = 0#,                                                    02603210 T  0000:0
SEGSPEROW       = LBL[8]#,          % HEADER:SEGMENTS PER ROW           02603220 T  0000:0
SEGSPBLK        = LBL[0].[42:6]#,   % HEADER:SEGMENTS PER BLOCK          02603230 T  0000:0
$ SET OMIT = NOT(TIMESHARING)                                            02603232 T  0000:0
SLEEPCM         = 36,COM#,          % SLEEP COMUNICATE                  02603235 T  0000:0
$ POP OMIT                                                               02603236 T  0000:0
$ SET OMIT = TIMESHARING                                                 02603237 T  0000:0
SORT            = 6#,                                                    02603240 T  0000:0
SORTFILE        = (FIB[4].[7:1] OR FIB[18].[1:1])#,                      02603250 T  0000:0
```

```
          SZF             = [8:10]#,                                          02603260 T  0000:0
          TECH            = FIB[5].[46:2]#,                                   02603270 T  0000:0
          TECHB           = 2#,               % TECHNIQUE B                   02603280 T  0000:0
          TECHC           = 3#,               % TECHNIQUE C                   02603290 T  0000:0
          TERM(TERM1)     = P(1,FLOC,TERM1,17,COM)#, % TERMINATE ON IO ERR    02603300 T  0000:0
          TIP             = FLOC[3]#,         % TOP IOD                       02603310 T  0000:0
          UNITYPE         = (FIB[4],[8:4])#, %ASSND INTERNAL HARDWARE TYPE    02603330 T  0000:0
          UNLABELED       = FIB[4].[2:1]#,    % UNLABELED FILE                02603340 T  0000:0
          WAITIO          = P([FLOC[I+2]],@2000000000,SLEEPCM,DEL,DEL)#,      02603360 T  0000:0
          WORDSLEFT       = FIB[17]#,         % NO.OF WORDS LEFT IN BLOCK     02603370 T  0000:0
          WRITEPARITY     = FIB[5].[3:1]#,    % INDICATES FORCED REELSWITCH   02603375 T  0000:0
          WRITBACK        = FIB[13],[23:1]#, % WRITE BLOCK BACK ON IO          02603380 T  0000:0
          WRITEAFTEREOF   = FIB[13].[44:2]#;                                  02603385 T  0000:0
     LABEL LINVALID,LOPENIN,LOPREVIN,LOPENOUT,LCLOSE,LOPENIO,LSORT,           02603390 T  0000:0
           LOPEN1,LCLOSE1,STARTL,EXIT,TSTBRK,BSTP;%                           02603395 T  0000:0
     SWITCH TYPE ← LINVALID,LOPENIN,LOPREVIN,LOPENOUT,LCLOSE,LOPENIO,         02603400 T  0000:0
                   LSORT;%                                                    02603450 T  0000:0
SUBROUTINE BUILDLABEL;%                                                       02603500 T  0000:0
          BEGIN%                                                             02603600 T  0001:0
          I←IX;%                                                             02603700 T  0001:0
          FLOC[1]←FLAG(1&(IF FPB[I+3].[43:5]=1 THEN 19 ELSE FLOC[1].[8:10    02603800 T  0001:3
               ]+4)[8:38:10]&(FLOC INX 1)[18:33:15]);%                       02603900 T  0006:1
          P(FLOC[1],0,COC,DEL);%                                             02604000 T  0010:2
          FLOC[1]←(2 INX FLOC[1])&(FLOC[1].[8:10]-4)[8:38:10];%              02604100 T  0012:1
          STREAM( A ← FU2←P(0,1,COM),B←FIB[4],C←[PU1]);%                     02604200 T  0017:3
               BEGIN SI← LOC A;SI←SI+3;DS←20CT;DS←30CT;%                     02604300 T  0020:1
                     SI←LOC B; SI←SI+5; DS←30CT;  END;%                      02604400 T  0021:1
      FU1←(PU2←PU2+FU1+3649)MOD 365+(PU2 DIV 365+PU1-10)×1000+1;%            02604500 T  0022:1
%    AT THIS POINT FU1 CONTAINS PURGE DATE(BINARY) AN FU2=DATE(DECIMAL)%      02604600 T  0028:0
          STREAM(K←0:A←CURRENTREEL,B←FPB[I+2]);   BEGIN%                     02604700 T  0028:0
          DI←LOC K;SI←LOC A;DS←3DEC;SI←LOC B;SI←SI+3;DS←5CHR END;%           02604800 T  0031:0
          IF (RPU1←P).[1:17]=0 THEN                                          02604900 T  0032:3
          IF (RPU1←FPB[I+2]).[1:17]=0 THEN RPU1.[17:1]←1;                    02605000 T  0034:1
          IF RPU1.[18:30]=0 THEN RPU1.[18:30]←FU2;                           02605100 T  0039:2
          IF (RPU2:=FPB[I+3]).[1:5]=0 THEN RPU2.[5:1]:=1;                    02605200 T  0043:0
          STREAM(K←0:PU1);BEGIN DI←LOC K;SI←LOC PU1;DS←30CT END;%            02605300 T  0047:3
          REEL←P;       CURRENTREEL←REEL;                                    02605400 T  0050:0
          STREAM(A←[FPB[I]],B←PU1,C←PU2,D←FU1,%                              02605500 T  0053:0
               Q←IF (T←FPB[I+3].[43:5])=10 OR T=12 OR T=26 THEN              02605600 T  0054:2
                  P([HEADERPTR],LOD,7,COC,1,+) ELSE 0,                       02605700 T  0059:0
               G←FLOC[1].[8:10]-8,%                                          02605900 T  0062:0
               F←IF REEL≠1 THEN FIB[4].[4:1] ELSE 0,E←FLOC[1]);              02606000 T  0064:0
               BEGIN  DS←8LIT" LABEL  ";%                                    02606100 T  0068:1
               SI←A; DS←2 WDS;  SI←LOC B; DS←WDS; SI←LOC C;%                 02606200 T  0069:2
              DI:=DI+1;DS:= CHR; SI:=LOC D; DS:=5 DEC;%                      02606300 T  0070:3
               DS←LIT"0";      %SENTINAL%                                    02606400 T  0071:3
               DS←5LIT"0";%  BLOCK-COUNT%                                    02606500 T  0072:1
          SI←LOC Q;DS←7DEC;    %REC-COUNT%                                   02606600 T  0073:1
               SI←LOC F;SI←SI+7;DS←CHR;     %  MEM-DUMP KEY%                02606700 T  0073:3
               DS←5LIT"0";    % PHYSICAL TAPE NO.%                           02606800 T  0074:2
               DS←6LIT"0";%                                                  02606900 T  0075:2
               G(DS←8LIT" ");%                                               02607000 T  0076:2
          END  END;%                                                         02607100 T  0078:2
          SUBROUTINE  GOUSE;%                                                02607200 T  0080:0
               BEGIN%                                                        02607300 T  0080:0
               COBOLINDEX ← T.[26:10];%                                      02607400 T  0080:0
               P( MKS, T.[38:10], [COBOLCONTROL]);%                          02607500 T  0081:1
```

```
     END;%
SUBROUTINE CALLGOUSE;
     BEGIN
          IF I OR TEST THEN
     BEGIN     IF(T←PGUSE[I].[ 1:23])≠ 0 THEN GOUSE;
               IF(T←PGUSE[I].[24:24])≠ 0 THEN GOUSE;
     END;
     END CALLGOUSE;
SUBROUTINE CALLGOUSER;
     BEGIN
          IF I OR TEST THEN
     BEGIN     IF (T←FIB[I].[ 1:23])≠ 0 THEN GOUSE;
               IF (T←FIB[I].[24:24])≠ 0 THEN GOUSE;
     END;
     END CALLGOUSER;
SUBROUTINE USERS;
     BEGIN
          I ← PU1;          CALLGOUSE;
          IF (I←PU2)>0 THEN CALLGOUSE;
          I ← FU1;          CALLGOUSER;
          IF (I←FU2)>0 THEN CALLGOUSER;
     END  USERS;
SUBROUTINE GOUSE68;
     BEGIN PERFORMUSE; END;
SUBROUTINE USERS68;
     BEGIN
          IF TEST THEN
     BEGIN               % CHECK FOR FILE USE ROUTINES
               IF (T←FIB[FU1].BF)≠0 THEN GOUSE68;
               IF (T←FIB[FU1].AF)≠0 THEN GOUSE68;
               IF (T←PGUSE[PU1].BF)≠0 THEN GOUSE68;
               IF (T←PGUSE[PU1].AF)≠0 THEN GOUSE68;
     END;
               IF PU2>0 THEN
               BEGIN               % NOT DISK: CHECK FOR REEL USE ROUTINES
               IF (T←FIB[FU1].BRR)≠0 THEN GOUSE68;
               IF (T←FIB[FU1].ARR)≠0 THEN GOUSE68;
               IF (T←PGUSE[PU1].BRR)≠0 THEN GOUSE68;
               IF (T←PGUSE[PU1].ARR)≠0 THEN GOUSE68;
               END;
               END USERS68;
% * * * * * * * * * * * * * * * S T A R T  H E R E * * * * * * * * * * * * *
          REEL ← IF P(MKSCW,TOP,XCH,DEL) THEN MKSCW ELSE 0;%
          COB68 ← (FIB←*FLOC).SZF=22;%
     IF CODE=18 THEN BEGIN CODE:=4; SVFIB:=1; END;
          IF NOT FPBXDONE THEN FIB[4].[12:12] ← %
               ((FIB[4].[12:12]=1)×ETRLNG)&1 [36:47:1];%
          IF NOT COB68 THEN
          IF REEL>9 THEN%
               BEGIN                              % CONVERT REEL NO.TO OCTAL
                    STREAM(K←0:L←REEL);%
                    BEGIN SI←LOC L; SI←SI+5;%
                         DI←LOC K; DS←3 OCT;%
                    END;%
                    REEL ← P;%
               END;%
          IX ← FIB[4].[13:11];% INDEX TO FPB
```

```
02607600 T  0082:2
02607700 T  0082:3
02607800 T  0083:0
02607900 T  0083:0
02608000 T  0083:3
02608100 T  0088:0
02608200 T  0092:0
02608300 T  0092:0
02608310 T  0092:1
02608320 T  0093:0
02608330 T  0093:0
02608340 T  0093:3
02608350 T  0098:0
02608360 T  0102:0
02608370 T  0102:0
02608400 T  0102:1
02608500 T  0103:0
02608600 T  0103:0
02608700 T  0105:0
02608800 T  0108:0
02608900 T  0110:0
02609000 T  0113:0
02609005 T  0113:1
02609006 T  0114:0
02609010 T  0115:3
02609020 T  0116:0
02609030 T  0116:0
02609040 T  0116:1
02609050 T  0116:3
02609060 T  0120:0
02609070 T  0124:0
02609080 T  0128:0
02609090 T  0132:0
02609100 T  0132:0
02609110 T  0132:3
02609120 T  0133:1
02609130 T  0137:0
02609140 T  0141:0
02609150 T  0145:0
02609160 T  0149:0
02609170 T  0149:0
02610150 T  0149:1
02610200 T  0149:1
02610300 T  0156:2
02610310 C  0159:0
02610400 T  0161:3
02610500 T  0164:2
02610550 T  0168:3
02610600 T  0169:1
02610700 T  0170:2
02610800 T  0171:0
02610900 T  0172:1
02611000 T  0172:3
02611100 T  0173:1
02611200 T  0173:2
02611300 T  0174:0
02611370 T  0174:0
```

```
     IF CODE≠CLOSE THEN%                                                                02611390 T  0175:2
          IF (T←FPBTYPE)=DISKR OR T=DISKP OR CODE=OPENIO THEN                           02611400 T  0176:1
               BEGIN IF (T=DISKR OR T=DISKP) AND NOT COB68 THEN                         02611410 T  0181:1
                    BUFREQ←1; NOTSERL←TRUE;                                             02611430 T  0184:1
               END ELSE                                                                 02611440 T  0188:0
               IF T<3 OR T=11 OR (T GEQ 7 AND T<10) THEN                                02611450 T  0188:0
               IF FIB[8].[20:5]>0 THEN    % HAS BEEN LABEQ FRM DISK                     02611480 T  0192:1
                    BEGIN NMSZROWS←0; LABEQ ← TRUE; END;%                               02611490 T  0194:1
     IF CODE=SORT THEN GO TO LSORT;%                                                    02611495 T  0199:3
     IF CODE≠CLOSE THEN%                                                                02611500 T  0201:0
          BEGIN                                                                         02611510 T  0201:3
               FIB[13].[19:5] ←0;%                                                      02611610 T  0202:1
               IF T=DISKR OR T=DISKS OR T=DISKP THEN% TECH B & C NOT                    02611620 T  0204:3
               IF TECH>1 THEN TERM(30);   % VALID ON DISK  %CJC 103I                    02611630 T  0207:2
               IF COB68 THEN IF TECH≠TECHC THEN MINREC←MAXREC;                          02611675 T  0211:1
          END;%                                                                         02611680 T  0216:3
          NUMBUFF ← BUFREQ;%                                                            02611700 T  0216:3
STARTL:%                                                                                02611800 T  0220:0
     IF CODE>5 THEN IF CODE=16 THEN GO TO LOPEN1 ELSE%                                  02611900 T  0220:0
                    IF CODE=17 THEN GO TO LCLOSE1 ELSE TERM(25);%                       02612000 T  0222:0
     GO TO TYPE[CODE];%                                                                 02612100 T  0225:0
LOPENIO:%                                                                               02612200 T  0229:2
     CODE ← OPENIN;%                                                                    02612300 T  0229:2
     FILIO ← 1;%                                                                        02612400 T  0230:1
     GO TO LOPENIN;%                                                                    02612500 T  0232:3
LOPREVIN:%                                                                              02612600 T  0233:1
     IF ((T←TECH)=TECHC) OR (T=TECHB AND NUMREC≠1) THEN TERM(5);                        02612700 T  0233:1
LOPENIN:%                                                                               02612800 T  0239:1
     IF NOTCLOSED THEN TERM(2×CODE-1);%                                                 02612900 T  0239:1
     IF REEL=0 THEN REEL ← CURRENTREEL ELSE CURRENTREEL ← REEL;                         02613600 T  0243:2
     IF (T←FPBTYPE)=DISKR OR T=DISKS OR T=DISKP THEN                                    02614200 T  0249:1
          BEGIN%                                                                        02614300 T  0253:3
               NMSZROWS ← 0; % SINCE ITS INPUT                                          02614400 T  0254:1
               IF LSUBU≠0 THEN     % UPPER BOUND                                       02614500 T  0256:3
                    BEGIN LSUBU ← *P(DUP)-1; BOUNDED ← TRUE; END;                       02614600 T  0257:3
               IF LSUBL≠0 THEN LSUBL ← *P(DUP)-1;%                                      02614700 T  0262:3
               WRITEAFTEREOF ← 0;%                                                      02614750 T  0266:1
               BCOUNT ← IF (RCOUNT←LSUBL)=0 THEN 0 ELSE % STARTING                      02614800 T  0268:3
                    (RCOUNT-1) DIV NUMREC + 1;          % BLOCK                       02614900 T  0272:2
          END;%                                                                         02615000 T  0275:1
     COBOLOPENIN;       % STORE BOOLEAN RESULT IN I: TRUE FOR                         02615100 T  0275:1
                        % LABELED AND NOT SORT FILE ON OPEN IN                        02615105 T  0277:2
     IF COB68 THEN%                                                                     02615110 T  0277:2
          BEGIN%                                                                        02615115 T  0277:3
               IF NOAIT THEN%                                                           02615120 T  0278:1
                    BEGIN%                                                              02615125 T  0279:1
                         REDECWA;%                                                      02615130 T  0279:3
                         NOAIT ← 0;%                                                    02615133 T  0283:1
                    END;%                                                               02615135 T  0285:3
          END;%                                                                         02615140 T  0285:3
     IF DISK THEN%                                                                      02615170 T  0285:3
          BEGIN%                                                                        02615200 T  0287:1
               IF NOT COB68 THEN                                                        02615250 T  0287:3
               IF RANDOM THEN TIP ← 1 INX TIP;% DISK ADDR IN WRD 1                      02615300 T  0288:1
               BUILDLABEL;%                                                             02615400 T  0293:0
               IF MABUSE OR NOT COB68 THEN%                                             02615500 T  0294:0
                    BEGIN     % BEGINNING INPUT/IO FILE                               02615600 T  0295:3
```

```
                    FU1← 0; TEST ← 1; PU2 ← FU2 ← -1;                          02615700 T  0296:1
                    IF COB68 THEN BEGIN PU1 ← 4×FILIO;%                        02615800 T  0299:1
                        USERS68; END%                                          02615900 T  0302:0
                    ELSE BEGIN PU1 ← 10×FILIO; USERS; END;%                    02616000 T  0303:0
               END;%                                                           02616100 T  0307:0
          IF COB68 THEN                                                        02616140 T  0307:0
               BEGIN                                                           02616160 T  0307:1
                    BCOUNT ← (IF RANDOM THEN NOT 0                             02616180 T  0307:3
                                  ELSE RCOUNT DIV NUMREC);                     02616200 T  0309:3
                    COUNT←BCOUNT + (NUMBUFF-1)&FIB[5][1:44:1];                 02616220 T  0312:2
               END ELSE                                                        02616240 T  0317:0
BEGIN IF NOTSERL THEN                                        %TR 1476          02616260 T  0317:0
          TIP ← (BUFFERSIZE + 1) INX TIP & MAXR;             %TR 1476          02616300 T  0317:3
     COUNT ← IF NOTSERL THEN -1 ELSE 0;                      %TR 1476          02616400 T  0323:2
     RESETPARITY;                                            %TR 1476          02616450 T  0326:2
END ;                                                        %TR 1476          02616470 T  0329:1
          GO TO EXIT;%                                                         02616500 T  0329:1
     END DISK;%                                                                02616600 T  0329:3
IF HASH THEN IF NOT WRITEPARITY THEN CALLHASH(2);                              02616700 T  0329:3
IF NOT OPTIONAL THEN                                                           02616750 T  0334:3
IF (MABUSE OR NOT COB68) AND I THEN  % LABELED AND NOT SORT                    02616800 T  0336:0
IF NOT WRITEPARITY THEN                                                        02616850 T  0338:3
     BEGIN                    % BEGINNING INPUT FILE/REEL                      02616900 T  0340:2
          PU1 ← FU1 ← 0; TEST ← CURRENTREEL=1;%                                02617000 T  0341:0
          PU2 ← FU2 ← 1;%                                                      02617100 T  0344:1
          IF COB68 THEN USERS68 ELSE USERS;%                                   02617200 T  0345:2
     END;%                                                                     02617300 T  0349:0
GO TO TSTBRK;%                                                                 02617400 T  0349:0
LOPENOUT:%                                                                     02617500 T  0349:2
     IF NOTCLOSED THEN TERM(6);%                                               02617600 T  0349:2
     IF CLOSEDHERE THEN BEGIN CLOSEDHERE←0; GO LOPEN1; END;%                   02617700 T  0352:3
     IF REEL≠0 THEN CURRENTREEL←REEL;% FIXES OPEN OUT REEL DATA-NAME           02618500 T  0357:1
     IF (T←FPBTYPE)=5 OR T=8 OR T=9 % UNLABELED SPEC UNIT, PT, OR MT           02618550 T  0361:0
          THEN UNLABELED ← 1;%                                                 02618600 T  0365:0
     IF T=DISKR OR T=DISKS OR T=DISKP THEN                                     02618650 T  0368:2
          BEGIN%                                                               02618700 T  0371:1
               IF LSUBU≠0 THEN BEGIN LSUBU ← *P(DUP)-1;%                       02618750 T  0371:3
                    BOUNDED ← TRUE; END;%                                      02618800 T  0375:1
               IF LSUBL≠0 THEN LSUBL ← *P(DUP)-1;%                             02618900 T  0377:3
               BCOUNT ← (RCOUNT←LSUBL) DIV NUMREC;%                            02619000 T  0381:1
               IF COB68 AND NMSZROWS = 0 THEN      % DISK DEFAULT IS            02619004 T  0384:2
                    IF NOT DISK THEN               % (LBL EQU ONLY)             02619005 T  0386:2
                    NMSZROWS ← 100&20[20:43:5];    % 20 ROWS 100 RECS            02619006 T  0388:3
          END%                                                                 02619010 T  0392:3
     ELSE IF LABELED THEN%                                                     02619020 T  0392:3
          BEGIN%                                                               02619100 T  0394:2
               BUILDLABEL;%                                                    02619200 T  0395:0
               IF NOT SORTFILE THEN%                                           02619300 T  0396:0
                    BEGIN IF HASH THEN CALLHASH(2);%                           02619400 T  0398:2
                         IF MABUSE OR NOT COB68 THEN%BEG OUT FILE/RL           02619410 T  0402:1
                              BEGIN TEST ← REEL=1; FU1 ← 0; PU2 ← 5;           02619420 T  0404:0
                                   IF COB68 THEN%                              02619430 T  0406:3
                                        BEGIN PU1←2; USERS68; END%             02619500 T  0407:2
                                   ELSE BEGIN PU1←4;FU2←1;USERS;END;           02619600 T  0410:0
                              END;%                                            02619700 T  0413:0
                    END;%                                                      02619800 T  0413:0
               END;%                                                           02619900 T  0413:0
```

```
COBOLOPENOUT;%                                                                  02620000 T  0413:0
IF COB68 THEN  % MOVE WA TO BUF.SAVE WA ADDR. POINT PRT TO BUFF                 02620010 T  0415:0
     BEGIN%                                                                     02620015 T  0415:1
          IF NOAIT THEN%                                                        02620020 T  0415:3
               BEGIN%                                                           02620030 T  0416:3
                    REDECWA;%                                                   02620040 T  0417:1
                    NOAIT ← 0;%                                                 02620050 T  0420:3
               END;%                                                            02620060 T  0423:1
          IF NOT (DISK) THEN                                                    02620065 T  0423:1
               BEGIN   WORDSLEFT ← BUFFERSIZE;                                  02620070 T  0424:3
                    PRINTFILE ← P(DUP,LOD,(FIB[4],[8:4]),                       02620075 T  0427:1
                    P(DUP)=1,P(XCH,DUP)=7,P(XCH)=12,OR,OR,CCX);                 02620080 T  0429:1
               END UNDISK;    % 1=LP,  7=PBT,  12=PBD                         02620085 T  0432:2
     END COB68ING;                                                              02620090 T  0433:0
IF DISK THEN%                                                                   02620100 T  0433:0
     BEGIN%                                                                     02620200 T  0434:2
          IF RANDOM THEN                                                        02620250 T  0435:0
               IF COB68 THEN BCOUNT ← NOT 0                                     02620300 T  0436:0
               ELSE TIP ← 1 INX TIP;                                            02620350 T  0437:3
          BUILDLABEL;%                                                          02620400 T  0442:0
          LBL ← *[HEADERPTR];%                                                  02620500 T  0443:0
          LBL[7] ← -1;%                                                         02620600 T  0444:1
          IF MABUSE OR NOT COB68 THEN%                                          02620800 T  0445:3
               BEGIN     % BEGINNING OUTPUT FILE                              02620810 T  0447:2
                   FU1 ← 0; TEST ← 1; PU2 ← FU2 ← -1;                           02620900 T  0448:0
                   IF COB68 THEN BEGIN PU1 ← 2; USERS68; END                    02621000 T  0451:0
                   ELSE BEGIN PU1 ← 4; USERS; END;                              02621100 T  0454:0
               END;%                                                            02621200 T  0456:0
          IF NOT COB68 THEN                                                     02621220 T  0456:0
               BEGIN                                                            02621230 T  0456:2
          RESETPARITY;%                                                         02621250 T  0457:0
          IF NOTSERL THEN%                                                      02621300 T  0459:3
               BEGIN%                                                           02621310 T  0460:0
     IF UNITYPE =4 AND NOT UNLABELED AND NOT SORTFILE THEN     %TR-90           02621320 T  0460:2
               TIP ← (BUFFERSIZE + 1) INX TIP & MAXR;        %TR 1476         02621325 T  0466:1
          BUFTOP ← (*P(DUP)) & 1[24:47:1];                   %TR 1476         02621330 T  0472:0
               END   END;                                                       02621335 T  0474:2
          END DISK;%                                                            02621340 T  0474:2
     IF NOT COB68 THEN COUNT ← IF NOTSERL THEN -1 ELSE NUMREC;                  02621350 T  0474:2
TSTBRK:%                                                                        02621360 T  0478:3
     IF (T←EORRERUN)≠0 AND CURRENTREEL≠1 THEN IF BREAKFAIL AND OUTAP            02621364 T  0478:3
          THEN BEGIN PURGEREEL; GO TO STARTL; END % TRY BREAK AGAIN             02621365 T  0485:3
          ELSE P(DEL);%                                                         02621370 T  0490:0
     GO TO EXIT;%                                                               02621371 T  0490:3
LCLOSE:%                                                                        02621375 T  0491:1
     IF OPTIONAL THEN % EOF ON ABSENT OPTIONAL FILE                             02621380 T  0491:1
          BEGIN FIB[5] ← (*P(DUP))&4 [39:42:6]; % MARK CLOSED RLSD              02621400 T  0493:0
           P(XIT);%                                                             02621500 T  0496:0
          END;%                                                                 02621600 T  0496:1
     IF NOT SORTFILE AND CLOSED THEN BEGIN IOERR(12-FIB[5],[43:1]);             02621700 T  0496:1
          GO TO EXIT; END;                                                      02621750 T  0503:2
     IF INFILE THEN%                                                            02621800 T  0504:0
          IF (COB68 AND MABUSE AND DISK) THEN  % END INPUT/IO FILE              02621810 T  0505:0
               BEGIN     FU1←2; PU2←-1; TEST←1; PU1←1+4×FILIO;%                 02621820 T  0508:3
                         USERS68;%                                              02621830 T  0514:1
               END ELSE%                                                        02621840 T  0515:0
     ELSE IF (LABELED AND NOT SORTFILE) THEN%                                   02621900 T  0515:0
```

```
IF DISK THEN%
     BEGIN      % MOVE RECORD COUNT FROM HEADER TO LABEL
          STREAM(A←P([HEADERPTR],LOD,7,COC,1,+),
                 B← 5 INX LBLPTR);%
              BEGIN SI←LOC A; DI←DI+5; DS←7 DEC; END;%
          IF MABUSE OR NOT COB68 THEN% END OUTPUT FILE
              BEGIN FU1←2; TEST←1; PU2←FU2←-1;
                   IF COB68 THEN BEGIN PU1←3; USERS68; END
                   ELSE BEGIN PU1←6; USERS; END;%
              END;%
     END        % NOT DISK
ELSE BEGIN      % MOVE BLK & RECORD COUNTS FROM FIB TO LBL
          IF HASH THEN CALLHASH(1);%
          STREAM(A←BCOUNT,B←RCOUNT,C←5 INX LBLPTR);%
              BEGIN SI←LOC A; DS←5 DEC; DS←7 DEC; END;%
          LBL ← LBLPTR;%
          LBL[4].EORF← REEL≠0;%
          LBL ← 0; %FILE CLOSE FORGETS LABELS-SO CLEAR PTR
          IF REEL THEN CLOSELOCK ← LOCK;%
          IF MABUSE OR NOT COB68 THEN% END OUTPUT FILE/RL
              BEGIN FU1←2; TEST ← REEL=0; PU2 ← 7;
                   IF COB68 THEN BEGIN PU1←3; USERS68; END
                   ELSE BEGIN PU1←6; FU2←3; USERS; END;
              END;%
     END; % OF NONDISK
IF DISK AND LABELED AND NOT SORTFILE THEN%
     BEGIN%
          LSUBL ← *P(DUP)+1;%
          IF BOUNDED THEN  % IF UPPER BOUND
              BEGIN LSUBU ← *P(DUP)+1; BOUNDED←FALSE; END
             ELSE IF COB68 THEN LSUBU ← 0;
          LBL ← *[HEADERPTR];%
          HNMSZRS ←-(((SEGSPEROW×RECSPERBLK) DIV SEGSPBLK)
               & HNMROWS [20:43:5]);% NM,SZ ROWS FR HEADER
          NMSZROWS ← 0; % ZERO FIB NM,SZ ROWS
          IF NOT COB68 THEN IF RANDOM THEN WORDSLEFT←0;
     END;%
IF UNITYPE=MT AND NOT REEL THEN%
     BEGIN%
          IF CLOSELOCK=REWIND AND NOTFIRSTREEL THEN%
               CLOSELOCK ← LOCK;%
          NOTFIRSTREEL ← FALSE;%
     END;%
T ← CURRENTREEL;%
IF REEL AND UNITYPE=PBT THEN FIB[9].[1:1] ← 1;
COBOLCLOSE;%
IF HNMSZRS.[1:1] THEN
     BEGIN%
          NMSZROWS ← ABS(HNMSZRS);%
          HNMSZRS ← 0;%
          WRITBACK ← FALSE; % RANDOM OUTPUT AND I=0
     END;%
IF REEL THEN%
     BEGIN % REEL SWITCH
          REEL ← T+1;%
          CODE ← 3-(2×INFILE)+DIRECTION;%
          IF CODE=OPENOUT THEN CURRENTREEL ← REEL;%
```

```
02622000 T  0520:0
02622010 T  0522:0
02622020 T  0522:2
02622030 T  0524:2
02622040 T  0526:1
02622050 T  0527:1
02622070 T  0529:0
02622100 T  0532:2
02622200 T  0535:0
02622300 T  0537:0
02622400 T  0537:0
02622500 T  0537:0
02622600 T  0537:2
02622700 T  0540:3
02622800 T  0543:3
02622900 T  0544:3
02623000 T  0546:1
02623100 T  0549:1
02623200 T  0550:0
02623300 T  0551:2
02623400 T  0553:1
02623500 T  0556:2
02623600 T  0559:0
02623610 T  0562:0
02623620 T  0562:0
02623700 T  0562:0
02623800 T  0567:3
02623900 T  0568:1
02624000 T  0570:1
02624100 T  0571:1
02624200 T  0576:1
02624300 T  0578:3
02624400 T  0580:0
02624500 T  0582:0
02624600 T  0585:0
02624610 T  0587:2
02624700 T  0591:1
02624800 T  0591:1
02624900 T  0593:2
02624910 T  0594:0
02624920 T  0596:0
02625000 T  0597:1
02625100 T  0599:3
02625105 T  0599:3
02625110 T  0601:1
02625200 T  0606:1
02625210 T  0609:1
02625220 T  0610:0
02625225 T  0610:2
02625230 T  0613:1
02625240 T  0614:0
02625250 T  0616:2
02625255 T  0616:2
02625260 T  0616:3
02625265 T  0617:1
02625268 T  0618:2
02625270 T  0622:1
```

```
                    NOTFIRSTREEL ← TRUE;                                                02625280 T  0626:0
                    GO TO STARTL;%                                                      02625300 T  0628:2
                END%                                                                    02625400 T  0629:0
            ELSE CURRENTREEL ← 0;                                                       02625500 T  0629:0
        IF NOT SVFIB THEN                                                               02625510 C  0632:0
            IF COB68 AND CLOSELOCK>1 THEN IF (T←FIB[20])<0 THEN                         02625600 T  0632:2
          % THROW AWAY FILE TANK, RE-INITIALIZE TYPE=2 SEGMENT DESC                     02625650 T  0636:1
                P(FLOC,DUP,FCX,3,COM,FLAG(0 & T [23:8:10] & T                           02625700 T  0636:1
                  [8:8:10] & 9 [3:44:4]),SSN,XCH,←);                                    02625800 T  0639:1
            P(XIT);                                                                     02625950 T  0642:1
LINVALID:%                                                                              02626000 T  0642:2
        TERM(25);%                                                                      02626050 T  0642:2
LCLOSE1:%                                                                               02626100 T  0644:1
        IF NOTINANDOPEN THEN TERM(12-FIB[5],[43:1]);%                                   02626150 T  0644:1
        IF ENDFILE THEN BEGIN I ← 1; GO TO BSTP; END;%                                  02626200 T  0649:2
        FOR I ← 1 STEP 1 UNTIL NUMBUFF DO%                                              02626250 T  0652:1
            BEGIN     % WAIT UNTIL ALL IO-S ARE DONE%                                   02626300 T  0656:3
                IF NOT IODONE THEN WAITIO;%                                             02626350 T  0656:3
                IF FLOC[I+2].EOF THEN GO BSTP;%                                         02626400 T  0662:0
            END;%                                                                       02626450 T  0664:3
        I ← NUMBUFF;%                                                                   02626500 T  0667:0
BSTP:%                                                                                  02626550 T  0668:2
        BACKSPACE;   % BACKSPACE I BLOCKS                                               02626600 T  0668:2
        TIP.EOF ← ENDFILE;%                                                             02626650 T  0671:0
        CLOSEDHERE ← COB68;%                                                            02626660 T  0674:2
        FIB[5].[40:6] ← CLOSEDRET;%                                                     02626700 T  0677:0
        GO TO EXIT;%                                                                    02626750 T  0679:2
LOPEN1:%                                                                                02626800 T  0680:0
        IF FIB[5].[40:6]≠CLOSEDRET THEN TERM(6);%                                       02626850 T  0680:0
        FIB[5].[40:6] ← 0;%                                                             02626900 T  0683:1
        BUFTOP ←(*P(DUP))& RESETREADBIT;%                                               02626950 T  0685:3
        INFILE ← 0; % MAKE IT OUTPUT                                                    02627000 T  0688:1
        LBLPTR ←(*P(DUP))& RESETREADBIT;%                                               02627050 T  0690:3
        IF TIP.EOF THEN%                                                                02627100 T  0693:2
            BEGIN    % HAD  READ EOF BEFORE BACKSPACE                                   02627150 T  0695:0
                WORDSLEFT ← BUFFERSIZE;%                                                02627200 T  0695:2
                TIP.EOF ← 0; % RESET EOF                                                02627250 T  0697:2
                COUNT ← NUMREC;  % # RECS LEFT IN BUFF =  WHOLE BUFF                    02627300 T  0700:1
                BUFTOP.[CF] ← TIP.[CF];%                                                02627340 T  0701:3
            END%                                                                        02627350 T  0705:0
        ELSE BEGIN % NO EOF - OPEN IN PLACE                                             02627400 T  0705:0
                RCOUNT ← *P(DUP)-1;% BACK UP BECAUSE WE                                 02627450 T  0705:2
                BCOUNT ← *P(DUP)-1;% WERE READING                                       02627500 T  0707:2
                WORDSLEFT ← BUFFERSIZE-(TIP.[CF]-BUFTOP.[CF]);%                         02627510 T  0709:2
                COUNT ← WORDSLEFT DIV MAXREC;   % # RECS LEFT IN BUFF                  02627600 T  0714:2
            END;%                                                                       02627650 T  0717:1
        FOR T ← 1 STEP 1 UNTIL NUMBUFF DO%                                              02627700 T  0717:1
            FLOC[T+2] ← FLAG(BUFTOP&FLOC[T+2][CTC]); % CHANGE TO WRITE                  02627750 T  0721:3
        GO TO EXIT;%                                                                    02627800 T  0726:2
LSORT:%                                                                                 02627850 T  0727:0
        IOD ← [TIP];%                                                                   02627860 T  0727:0
        IF CLOSELOCK=NOREW THEN          % FOR CALLED WITH THESE PARAMS                02627870 T  0728:1
            BEGIN                        %IF IO COMPLETE BUT NOT PRESENT              02627880 T  0729:0
                IF NOT (*IOD).EOF        % NOT EOF:MUST HAVE BEEN PARITY               02627890 T  0729:2
                    THEN TERM(19)        % TERMINATE ON PARITY                         02627900 T  0730:2
                ELSE                     % MUST HAVE BEEN EOF OR EOR                   02627910 T  0732:2
                    BEGIN ALGOLIO(11);% READLABEL                                       02627920 T  0732:2
```

```
                    LBL ← LBLPTR;%                                          02627930 T  0734:2
                    IF LBL[4].EORF=0 THEN P(1,RTN);%RETURN EOF              02627940 T  0736:0
                    REEL ← CURRENTREEL+1;%REEL SWITCH ON INPUT              02627950 T  0738:2
                    T ← COBOLFILBIT; % REMEMBER IF COBOL FILE               02627960 T  0740:2
                    FORCLOSE(PURGE);                                        02627970 T  0742:0
                    FIB[13]←(*P(DUP))&REEL[28:38:10]% NXT REEL              02628000 T  0743:2
                          &0 [47:47:1]; % MAKE IT LOOK ALGOL                02628050 T  0744:3
                    ALGOLIO(0);% OPEN INPUT NEXT REEL                       02628100 T  0747:0
                    FIB[13] ←(*P(DUP))OR T;% RESTORE COBOL BIT              02628150 T  0748:2
                    P(0,RTN); % RETURN EOR                                  02628200 T  0750:2
               END;%                                                        02628250 T  0751:0
     END NOREW;%                                                            02628300 T  0751:0
IF CLOSELOCK=REWIND THEN%                                                   02628310 T  0751:0
     BEGIN     % REEL SWITCH ON OUTPUT                                      02628320 T  0751:3
          LBL ← LBLPTR;%                                                    02628330 T  0752:1
          LBL[4].EORF ← 1;  % EOR                                           02628340 T  0753:3
          LBL ← 0;%FILE CLOSE FOGETS LABEL SO PTR MUST BE CLRD              02628350 T  0756:1
          T ← CURRENTREEL+1;%                                               02628360 T  0757:0
          FORCLOSE(RELEASE);     % CLOSE RELEASE CURRENT REEL               02628370 T  0759:0
          CURRENTREEL ← REEL ← T;%WITH NO REEL SWITCH-DONE HERE             02628380 T  0760:2
          IF COBOLFILE THEN FOROPENOUT ELSE ALGOLIO(0);%NXT RL              02628390 T  0763:2
          P(XIT);           % OPEN OUT (ALGOL OR COBOL)NXT REEL             02628400 T  0767:3
     END;%                                                                  02628450 T  0768:0
IF CLOSELOCK=LOCK THEN                                                      02628500 T  0768:0
     BEGIN%                                                                 02628510 T  0768:3
          T ← IF CURRENTREEL=1 THEN REWIND ELSE RELEASE;                    02628520 T  0769:1
          FORCLOSE(T);      % CLOSE REWIND FIRST REEL.                     02628550 T  0772:3
          P(XIT);           % CLOSE RELEASE ALL OTHERS                     02628600 T  0774:1
     END;%                                                                  02628700 T  0774:2
EXIT:;%                                                                     02628800 T  0774:2
     END COBOLFOR;%                                                         02628900 T  0775:0
                                                                                  SIZE= 0776 WORDS
```

```
PROCEDURE COBOLATT;     BEGIN     % INT # = @ 165          %CJC 103I        02650000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00253
     COMMENT   INTRINSIC FOR COBOL ATTRIBUTES             %CJC 103I        02650100 T  0000:0
          CALLING SEQUENCE IS                             %CJC 103I        02650110 T  0000:0
               MKS                                        %CJC 103I        02650120 T  0000:0
               LITC OPERATION                             %CJC 103I        02650130 T  0000:0
               LITC FILEFIB                               %CJC 103I        02650140 T  0000:0
               DESC 10                                    %CJC 103I        02650145 T  0000:0
               LITC ATTRIBUTENUM                          %CJC 103I        02650150 T  0000:0
               LITC WORD-OFSET                            %CJC 103I        02650160 T  0000:0
               DESC DATAWORD                              %CJC 103I        02650170 T  0000:0
                                                          %CJC 103I        02650180 T  0000:0
          OPERATIONS ARE:                                 %CJC 103I        02650190 T  0000:0
               1 = MOVE                                   %CJC 103I        02650200 T  0000:0
               0 = SET (OR CHANGE ATTRIBUTE VALUE)        %CJC 103I        02650210 T  0000:0
                                                          %CJC 103I        02650220 T  0000:0
          ATTRIBUTENUM HAS THE FOLLOWING VALUES:          %CJC 103I        02650230 T  0000:0
               0 = EOF                                    %CJC 103I        02650240 T  0000:0
               1 = DO NOT USE                             %CJC 103I        02650250 T  0000:0
               2 = DO NOT USE                             %CJC 103I        02650260 T  0000:0
               3 = SAVEFACTOR                             %CJC 103I        02650270 T  0000:0
```

```
                    4 = AREAS
                    5 = AREASIZE
                    6 = MFID
                    7 = FID
                    8 = REEL
                    9 = DATE
                   10 = BUFFERS
                   11 = TYPE
                   12 = BLOCKSIZE
                   13 = MAXRECSIZE
                   14 = FILE INFORMATION BLOCK
                   15 = FILE PARAMETER BLOCK
                   16 = LABEL (8 WORDS ONLY)
                   17 = EU NUMBER (0 THRU 19)
                   18 = DISK SPEED (1=FAST,2=SLOW)
                   19 = TIMELIMIT (PROTECT FILES)
                   20 = IOSTATUS (PROTECT FILES)
                   21 = SENSITIVE
END-OF-COMMENTS;

    NAME        ITEM        = -1;     % COMP AREA FOR VALUE
    REAL        ATTNUM      = -2;     % ATTRIBUTE NUMBER
    NAME        FLOC        = -3;     % POINTER TO FIB
    REAL        OPCODE      = -4;     % OPERATION 0=SET 1=MOVE

    ARRAY       FPB         = 3[*];   % FILE PARAMETER BLOCK

    ARRAY       FIB[*];
    ARRAY       LBL[*];
    LABEL                   EOF,IOERR,          SAVEFACTOR,AREAS,
                            AREASIZE,MFID,FID,REEL,DATE,BUFFERS,
                            TYPE,BLOCKSIZE,MAXRECSIZE,ATTEXIT,
                           FIBWORDS,FPBWORDS,LABELWORDS,
                           EUNUM,DSKSPEED,
                           TIMELIMIT,IOSTATUS,
                            SENSITIVE,
                           DUMMY;

    SWITCH      ROUTINE    ← EOF,IOERR,IOERR   ,SAVEFACTOR,AREAS,
                            AREASIZE,MFID,FID,REEL,DATE,BUFFERS,
                            TYPE,BLOCKSIZE,MAXRECSIZE,
                           FIBWORDS,FPBWORDS,LABELWORDS,
                           EUNUM,DSKSPEED,
                           TIMELIMIT,IOSTATUS,
                            SENSITIVE,
                           IOERR;

    REAL        XI,TEMP,UNITYPE;
    DEFINE      GETFROMITEM = P(*[ITEM])#,
                STOREINTOITEM(STOREINTOITEM1) =
                     P(STOREINTOITEM1,[ITEM],←)#,
                IOERROR(IOERROR1) =
                     P(1,FLOC,IOERROR1,17,COM)#;

    COMMENT     I/O ERRORS ARE AS FOLLOWS:
               40 = FILE WAS OPEN WHEN SETTING THE ATTRIBUTE
               41 = SETTING A READ ONLY ATTRIBUTE
```

```
%CJC 103I       02650280 T  0000:0
%CJC 103I       02650290 T  0000:0
%CJC 103I       02650300 T  0000:0
%CJC 103I       02650310 T  0000:0
%CJC 103I       02650320 T  0000:0
%CJC 103I       02650330 T  0000:0
%CJC 103I       02650340 T  0000:0
%CJC 103I       02650350 T  0000:0
%CJC 103I       02650360 T  0000:0
%CJC 103I       02650370 T  0000:0
%CJC 103I       02650371 T  0000:0
%CJC 103I       02650372 T  0000:0
%CJC 103I       02650373 T  0000:0
                02650374 T  0000:0
                02650375 T  0000:0
                02650376 T  0000:0
                02650377 T  0000:0
                02650378 T  0000:0
%CJC 103I       02650380 T  0000:0
%CJC 103I       02650390 T  0000:0
%CJC 103I       02650400 T  0000:0
%CJC 103I       02650500 T  0000:0
%CJC 103I       02650600 T  0000:0
%CJC 103I       02650700 T  0000:0
%CJC 103I       02650800 T  0000:0
%CJC 103I       02650900 T  0000:0
%CJC 103I       02651000 T  0000:0
%CJC 103I       02651100 T  0000:0
%CJC 103I       02651110 T  0000:0
%CJC 103I       02651300 T  0000:0
%CJC 103I       02651400 T  0000:0
%CJC 103I       02651410 T  0000:0
                02651500 T  0000:0
                02651510 T  0000:0
                02651520 T  0000:0
                02651530 T  0000:0
                02651590 T  0000:0
                02651595 T  0000:0
%CJC 103I       02651600 T  0000:0
%CJC 103I       02651610 T  0000:0
%CJC 103I       02651700 T  0000:0
                02651710 T  0000:0
                02651720 T  0000:0
                02651730 T  0000:0
                02651740 T  0000:0
                02651799 T  0000:0
%CJC 103I       02651800 T  0000:0
%CJC 103I       02651900 T  0000:0
%CJC 103I       02652000 T  0000:0
%CJC 103I       02652100 T  0000:0
%CJC 103I       02652150 T  0000:0
%CJC 103I       02652200 T  0000:0
%CJC 103I       02652250 T  0000:0
%CJC 103I       02652251 T  0000:0
%CJC 103I       02652252 T  0000:0
%CJC 103I       02652253 T  0000:0
%CJC 103I       02652254 T  0000:0
```

```
                42 = SETTING AN ATTRIBUTE TO AN ILLEGAL VALUE
                43 = CHANGING # OF BUFFERS OF A NON-SERIAL FILE
                44 = INCREASING # OF BUFFERS
                45 = CHANGING BLOCKSIZE TO A VALUE WHICH IS NOT
                     A MULTIPLE OF RECORD SIZE
                46 = CHANGE TO BLOCKSIZE WHEN FILE IS OTHER THAN
                     TAPE, PAPER TAPE OR SERIAL DISK
                47 = ACCESSING "LABEL" WHEN FILE IS NOT OPEN
                48 = THIS FILE MAY NOT HAVE "TYPE" CHANGED
                49 = ILLEGAL ATTNUM VALUE
    END OF I/O ERRORS;

% S T A R T   H E R E
                FIB ← *FLOC;
                IF FIB[5].[41:2] = 0 THEN LBL ←  FLOC[1];
                IF NOT FIB[4].[12:1] THEN FIB[4].[12:12] ←
                     ((FIB[4].[12:12] -1) × ETRLNG) & 1[36:47:1];
                XI ← FIB[4].[13:11];
                IF OPCODE = 0 AND FIB[5].[41:2] = 0 THEN
                     IOERROR(40);  % SET AN ATTRIBUTE ON A FILE
                                   % WHICH IS OPEN.
                GO TO ROUTINE[ATTNUM];
IOERR::         IOERROR(49); % ILLEGAL ATTNUM

EOF::
                IF OPCODE = 0 THEN IOERROR(41);
                                   % EOF IS READ ONLY
                STOREINTOITEM(FIB[5].[40:1]);
                GO TO ATTEXIT;

AREAS::
                IF OPCODE = 0 THEN
                IF (TEMP ← GETFROMITEM) < 1 OR TEMP > 20 THEN
                     IOERROR(42) ELSE  % OK VALUES 1-20
                     FIB[8].[20:5] ← TEMP ELSE
                STOREINTOITEM(FIB[8].[20:5]);
                GO TO ATTEXIT;

AREASIZE::
                IF OPCODE = 0 THEN
                IF (TEMP ← GETFROMITEM) < 1 THEN
                     IOERROR(42) ELSE  % MUST HAVE 1 OR MORE
                     FIB[8].[25:23] ← TEMP ELSE
                STOREINTOITEM(FIB[8].[25:23]);
                GO TO ATTEXIT;

MFID::
FID::
   IF OPCODE = 0 AND FIB [5].[42:1] = 0 THEN
      IF ((UNITYPE←FPB[XI+3].[43:5])= 10 OR UNITYPE = 12
         OR UNITYPE = 13) THEN % DISK FILE IS NOT CLOSED WITH RELEASE
            IOERROR(40); % CANT CHANGE MFID/FID
              ATTNUM ← ATTNUM - 6;
              IF FIB[4].[2:1] = 0 THEN  % IF LABELED
              IF OPCODE = 1 AND FIB[5].[41:3] = 1 THEN
         BEGIN % IF "MOVE" AND FILE OPEN INPUT PICKUP
               % MFID AND ID FROM LABEL IN CASE OF "IL",
```

```
%CJC 103I        02652255 T  0000:0
%CJC 103I        02652256 T  0000:0
%CJC 103I        02652257 T  0000:0
%CJC 103I        02652258 T  0000:0
%CJC 103I        02652259 T  0000:0
%CJC 103I        02652260 T  0000:0
%CJC 103I        02652261 T  0000:0
%CJC 103I        02652262 T  0000:0
%CJC 103I        02652263 T  0000:0
%CJC 103I        02652264 T  0000:0
%CJC 103I        02652290 T  0000:0
%CJC 103I        02652300 T  0000:0
%CJC 103I        02652400 T  0000:0
%CJC 103I        02652500 T  0000:0
%CJC 103I        02652550 T  0002:1
%CJC 103I        02652600 T  0005:3
%CJC 103I        02652700 T  0008:2
%CJC 103I        02652800 T  0012:3
%CJC 103I        02652900 T  0014:1
%CJC 103I        02653000 T  0016:3
%CJC 103I        02653100 T  0018:2
%CJC 103I        02653200 T  0018:2
%CJC 103I        02653250 T  0031:0
%CJC 103I        02653300 T  0032:1
%CJC 103I        02653400 T  0032:1
%CJC 103I        02653500 T  0033:0
%CJC 103I        02653600 T  0035:2
%CJC 103I        02653700 T  0035:2
%CJC 103I        02653800 T  0037:0
%CJC 103I        02655500 T  0037:2
%CJC 103I        02655600 T  0037:2
%CJC 103I        02655700 T  0038:0
%CJC 103I        02655800 T  0038:3
%CJC 103I        02655900 T  0041:3
%CJC 103I        02656000 T  0043:2
%CJC 103I        02656100 T  0046:2
%CJC 103I        02656200 T  0048:2
%CJC 103I        02656300 T  0049:0
%CJC 103I        02656400 T  0049:0
%CJC 103I        02656500 T  0049:0
%CJC 103I        02656600 T  0049:3
%CJC 103I        02656700 T  0051:3
%CJC 103I        02656800 T  0053:2
%CJC 103I        02656900 T  0056:2
%CJC 103I        02657000 T  0058:2
%CJC 103I        02657100 T  0059:0
%CJC 103I        02657200 T  0059:0
%CJC 103I        02657300 T  0059:0
                 02657310 C  0059:0
                 02657320 C  0061:2
                 02657330 C  0064:3
                 02657340 C  0066:2
%CJC 103I        02657400 T  0068:1
%CJC 103I        02657405 T  0069:2
%CJC 103I        02657410 T  0071:0
%CJC 103I        02657420 T  0074:0
%CJC 103I        02657430 T  0074:2
```

```
                STOREINTOITEM(LBL[ATTNUM + 1],[6:42]);
                GO TO ATTEXIT;
            END;
                IF OPCODE = 0 THEN
                FPB[XI + ATTNUM],[6:42] ← GETFROMITEM ELSE
                STOREINTOITEM(FPB[XI + ATTNUM],[6:42]);
                GO TO ATTEXIT;
            % NOTE THAT MFID MUST BE ATTRIBUTE 6 AND FID MUST
            % BE ATTRIBUTE 7 TO MAKE THE ABOVE WORK.

BUFFERS::
                IF OPCODE = 0 THEN
                IF FIB[4],[27:3] ≠ 0 THEN IOERROR(43) ELSE
                     % CHANGING # OF BUFFERS ON NON-SERIAL
                IF (TEMP ← GETFROMITEM) > FIB[13],[1:9] THEN
                     IOERROR(44) ELSE % INCREASING # OF BUFFERS
                IF TEMP < 1 THEN IOERROR(42) ELSE
                     FIB[13],[1:9] ← TEMP ELSE
                STOREINTOITEM(FIB[13],[1:9]);
                GO TO ATTEXIT;

BLOCKSIZE::
                IF OPCODE = 1 THEN
            BEGIN STOREINTOITEM(FIB[18],[3:15]);
                GO TO ATTEXIT;
            END;
            % THE FOLLOWING WILL "SET" BLOCKSIZE:
                IF (TEMP ← GETFROMITEM) MOD FIB[18],[33:15] ≠ 0
                     THEN IOERROR(45);
            % I/O ERROR 45 IF NOT MULTIPLE OF RECORD LENGTH
                IF NOT ((UNITYPE ← FPB[XI + 3],[43:5]) = 2
                     OR UNITYPE = 7 OR UNITYPE = 8
                     OR UNITYPE = 9 OR UNITYPE = 12) THEN
                IOERROR(46);
            % I/O ERROR 46 UNLESS FILETYPE IS MAGTAPE, PAPERTAPE
            % OR SERIAL DISK.
            % AT THIS POINT THE CHANGE TO BLOCKSIZE IS VALID
            % A CHANGE TO TECHNIQUE (FIB[5],[46:2]) AND RECORDS
            % PER BLOCK (FIB[11]) IS TAKEN INTO CONSIDERATION.
                FIB[18],[3:15] ← TEMP;
                FIB[11] ← TEMP DIV FIB[18],[33:15];
                FIB[5],[46:2] ← (IF FIB[11] = 1 THEN 0 ELSE 1);
                GO TO ATTEXIT;

MAXRECSIZE::
                IF OPCODE = 0 THEN IOERROR(41);
                                % MAXRECSIZE IS READ ONLY
                STOREINTOITEM(FIB[18],[33:15]);
                GO TO ATTEXIT;

TYPE::
                IF OPCODE = 1 THEN
            BEGIN STOREINTOITEM(FPB[XI + 3],[43:5]);
                GO TO ATTEXIT;
            END;
                IF (TEMP ← FPB[XI+3],[43:5]) > 9 AND TEMP < 15
                    OR TEMP = 19 OR TEMP = 26
```

```
%CJC 103I       02657440 T  0074:2
%CJC 103I       02657450 T  0077:1
%CJC 103I       02657460 T  0077:3
%CJC 103I       02657500 T  0077:3
%CJC 103I       02657600 T  0078:2
%CJC 103I       02657700 T  0082:1
%CJC 103I       02657800 T  0085:2
%CJC 103I       02657900 T  0086:0
%CJC 103I       02658000 T  0086:0
%CJC 103I       02658100 T  0086:0
%CJC 103I       02658200 T  0086:0
%CJC 103I       02658300 T  0086:0
%CJC 103I       02658400 T  0086:3
%CJC 103I       02658500 T  0090:2
%CJC 103I       02658600 T  0090:2
%CJC 103I       02658700 T  0093:1
%CJC 103I       02658800 T  0095:0
%CJC 103I       02658900 T  0098:0
%CJC 103I       02659000 T  0101:0
%CJC 103I       02659100 T  0103:0
%CJC 103I       02659200 T  0103:2
%CJC 103I       02659300 T  0103:2
%CJC 103I       02659400 T  0104:0
%CJC 103I       02659500 T  0104:3
%CJC 103I       02659600 T  0106:3
%CJC 103I       02659700 T  0107:1
%CJC 103I       02659800 T  0107:1
%CJC 103I       02659900 T  0107:1
%CJC 103I       02660000 T  0109:3
%CJC 103I       02660100 T  0111:3
%CJC 103I       02660200 T  0111:3
%CJC 103I       02660300 T  0114:0
%CJC 103I       02660400 T  0115:3
%CJC 103I       02660500 T  0118:2
%CJC 103I       02660600 T  0120:1
%CJC 103I       02660700 T  0120:1
%CJC 103I       02660800 T  0120:1
%CJC 103I       02660900 T  0120:1
%CJC 103I       02661000 T  0120:1
%CJC 103I       02661050 T  0120:1
%CJC 103I       02661100 T  0122:3
%CJC 103I       02661200 T  0125:1
%CJC 103I       02661300 T  0130:0
%CJC 103I       02661400 T  0130:2
%CJC 103I       02661500 T  0130:2
%CJC 103I       02661600 T  0131:0
%CJC 103I       02661700 T  0133:2
%CJC 103I       02661800 T  0133:2
%CJC 103I       02661850 T  0135:0
%CJC 103I       02661900 T  0135:2
%CJC 103I       02661990 T  0135:2
%CJC 103I       02662000 T  0136:0
%CJC 103I       02662010 T  0136:3
%CJC 103I       02662020 T  0139:1
                02662030 T  0139:3
                02662040 T  0139:3
                02662050 T  0142:2
```

```
                THEN IOERROR(48);
          % I/O ERROR 48 = FILE TYPE NOT ALTERABLE
               FPB[XI + 3].[43:5] ← TEMP ← GETFROMITEM;
               IF TEMP = 0                    % CARD
               OR TEMP > 19 AND TEMP < 26 THEN %PUNCH BACKUP
               IF FIB[18].[3:15] < 11 THEN GO TO ATTEXIT
               ELSE IOERROR(42);  % AND BLOCK < 11 WORDS
               IF TEMP = 1 OR TEMP = 4 OR TEMP = 6
               OR TEMP > 14 AND TEMP < 19 THEN % PRINTERS
               IF FIB[18].[3:15] < 18 THEN GO TO ATTEXIT
               ELSE IOERROR(42);  % BLOCK < 18 WORDS
               IF TEMP = 2            % MAG TAPE
               OR TEMP = 7            % PAPER TAPE
               OR TEMP = 8            % PT UNLABELED
               OR TEMP = 9 THEN       % MT UNLABELED
               GO TO ATTEXIT;
               IOERROR(42);

SAVEFACTOR:: REEL:: DATE::
               IF OPCODE = 0 THEN  % "SET" ATTRIBUTE
          BEGIN STREAM(K←[TEMP], L←[ITEM]);
          BEGIN SI ← L;  DI ← K;  DS ← 8 DEC;
          END;
               IF ATTNUM = 9 THEN % "DATE"
               FPB[XI + 2].[18:30] ← TEMP     ELSE
               IF ATTNUM = 8 THEN  % "REEL"
               FPB[XI + 2].[1:17] ← TEMP      ELSE
               FIB[4].[30:18] ← TEMP;
               GO TO ATTEXIT;
          END;
               IF FIB[4].[2:1] = 1 THEN IOERROR(47);
                    STREAM(K ← IF FIB[5].[41:2] ≠ 0 THEN
                    IF ATTNUM = 9 THEN FPB[XI+2].[18:30] ELSE
                    IF ATTNUM = 8 THEN FPB[XI+2].[1:17] ELSE
                         FIB[4].[30:18]  ELSE
                         IF ATTNUM = 9 THEN LBL[3].[18:30] ELSE
                         IF ATTNUM = 8 THEN LBL[3].[1:17]  ELSE
                    FIB[4].[30:18], L ← [ITEM]);
               BEGIN SI←LOC K;  DI←L;  DS←8 OCT;
               END;
               GO TO ATTEXIT;

FIBWORDS::
               IF OPCODE = 0 THEN IOERROR(41);
                              % FIB IS READ ONLY
               STREAM(A←[FIB[0]], B←[ITEM]);
               BEGIN SI ← A;  DI ← B;  DS ← 20 WDS;
               END;
               GO TO ATTEXIT;

FPBWORDS::
               IF OPCODE = 0 THEN IOERROR(41);
                              % FPB IS READ ONLY
               STREAM(A←[FPB[XI]], B←[ITEM]);
               BEGIN SI ← A;  DI ← B;  DS ← 5 WDS;
               END;
               GO TO ATTEXIT;
```

```
%CJC 103I        02662060 T  0144:2
                 02662080 T  0147:0
%CJC 103I        02662100 T  0147:0
%CJC 103I        02662200 T  0150:3
%CJC 103I        02662300 T  0151:1
%CJC 103I        02662400 T  0153:2
%CJC 103I        02662500 T  0155:2
%CJC 103I        02662600 T  0157:1
%CJC 103I        02662700 T  0159:2
%CJC 103I        02662800 T  0162:0
%CJC 103I        02662900 T  0164:0
%CJC 103I        02663000 T  0165:3
%CJC 103I        02663100 T  0166:1
%CJC 103I        02663200 T  0167:0
%CJC 103I        02663300 T  0168:0
%CJC 103I        02663500 T  0169:2
%CJC 103I        02663600 T  0170:1
%CJC 103I        02663700 T  0171:2
%CJC 103I        02663800 T  0171:2
%CJC 103I        02663900 T  0172:0
%CJC 103I        02664000 T  0172:3
%CJC 103I        02664100 T  0174:1
%CJC 103I        02664200 T  0175:0
%CJC 103I        02664300 T  0175:1
%CJC 103I        02664400 T  0176:0
%CJC 103I        02664500 T  0179:2
%CJC 103I        02664600 T  0180:3
%CJC 103I        02664800 T  0184:1
%CJC 103I        02664900 T  0187:1
%CJC 103I        02665000 T  0187:3
%CJC 103I        02665050 T  0187:3
%CJC 103I        02665100 T  0191:0
%CJC 103I        02665200 T  0192:3
%CJC 103I        02665300 T  0196:2
%CJC 103I        02665350 T  0199:3
%CJC 103I        02665360 T  0201:1
%CJC 103I        02665370 T  0204:0
%CJC 103I        02665400 T  0206:3
%CJC 103I        02665500 T  0208:1
%CJC 103I        02665600 T  0209:0
%CJC 103I        02665700 T  0209:1
%CJC 103I        02665800 T  0209:3
%CJC 103I        02665900 T  0209:3
%CJC 103I        02666000 T  0210:0
%CJC 103I        02666100 T  0212:2
%CJC 103I        02666200 T  0212:2
%CJC 103I        02666300 T  0213:3
%CJC 103I        02666400 T  0214:2
%CJC 103I        02666500 T  0214:3
%CJC 103I        02666600 T  0215:1
%CJC 103I        02666700 T  0215:1
%CJC 103I        02666800 T  0216:0
%CJC 103I        02666900 T  0218:2
%CJC 103I        02667000 T  0218:2
%CJC 103I        02667100 T  0219:3
%CJC 103I        02667200 T  0220:2
%CJC 103I        02667300 T  0220:3
```

```
                                                                %CJC 103I       02667400 T  0221:1
LABELWORDS::                                                    %CJC 103I       02667500 T  0221:1
            IF OPCODE = 0 THEN IOERROR(41);                     %CJC 103I       02667600 T  0222:0
                               % LABEL IS READ ONLY             %CJC 103I       02667700 T  0224:2
            IF FIB[5].[41:2] ≠ 0 THEN IOERROR(47);              %CJC 103I       02667710 T  0224:2
        % I/O ERROR 47 = ACCESS TO LABEL WHEN FILE NOT OPEN     %CJC 103I       02667720 T  0227:3
            IF FIB[4].[2:1] = 1 THEN IOERROR(47);               %CJC 103I       02667730 T  0227:3
            STREAM(A←[LBL[0]], B←[ITEM]);                       %CJC 103I       02667800 T  0231:0
            BEGIN SI ← A;  DI ← B;  DS ← 8 WDS;                 %CJC 103I       02667900 T  0232:1
            END;                                                %CJC 103I       02668000 T  0233:0
            GO TO ATTEXIT;                                      %CJC 103I       02668100 T  0233:1
EUNUM::     IF FIB[5].[41:2] = 0 AND OPCODE=0 THEN GO TO ATTEXIT;               02668200 T  0233:3
            IF OPCODE = 0 THEN                                                  02668300 T  0237:1
            FPB[XI+3].[18:5]:=GETFROMITEM+1 ELSE                                02668400 T  0238:0
            STOREINTOITEM(FPB[XI+3].[18:5]-1);                                  02668500 T  0242:1
            GO TO ATTEXIT;                                                      02668600 T  0245:1
DSKSPEED::  IF FIB[5].[41:2] = 0 AND OPCODE=0 THEN GO TO ATTEXIT;               02668700 T  0245:3
            IF OPCODE = 0 THEN                                                  02668800 T  0249:1
            FPB[XI+3].[16:2]:=GETFROMITEM ELSE                                  02668900 T  0250:0
               BEGIN                                                            02669000 T  0253:3
               TEMP := IF (TEMP:=FPB[XI+3].[16:2])=1 THEN                       02669100 T  0254:1
                       1 ELSE IF TEMP=2 THEN 2 ELSE 0;                          02669200 T  0256:3
               STOREINTOITEM(TEMP);                                             02669400 T  0260:3
               END;                                                             02669500 T  0261:2
            GO TO ATTEXIT;                                                      02669600 T  0261:2
                                                                                02680000 T  0262:0
TIMELIMIT::                                                                     02680100 T  0262:0
            IF OPCODE = 0 THEN                                                  02680200 T  0262:0
 $ SET OMIT = NOT SHAREDISK                                                     02680299 T  0262:3
            ELSE                                                                02680500 T  0262:3
            BEGIN                                                               02680600 T  0263:1
 $ SET OMIT = NOT SHAREDISK                                                     02680699 T  0263:3
               STOREINTOITEM(TEMP);                                             02680900 T  0263:3
            END;                                                                02681000 T  0264:2
            GO TO ATTEXIT;                                                      02681100 T  0264:2
                                                                                02681200 T  0265:0
IOSTATUS::                                                                      02681300 T  0265:0
            IF OPCODE = 0 THEN IOERROR(41);                                     02681400 T  0265:0
                               % IOSTATUS IS READ ONLY                          02681500 T  0267:2
 $ SET OMIT = NOT SHAREDISK                                                     02681599 T  0267:2
            STOREINTOITEM(TEMP);                                                02681800 T  0267:2
            GO TO ATTEXIT;                                                      02681900 T  0268:1
                                                                                02682000 T  0268:3
SENSITIVE::                                                                     02683000 T  0268:3
            IF OPCODE=0 THEN                                                    02683100 T  0268:3
            FPB[XI+3].[15:1]:=GETFROMITEM ELSE                                  02683200 T  0269:3
               BEGIN                                                            02683300 T  0273:2
               TEMP:=FPB[XI+3].[15:1];                                          02683400 T  0274:0
               STOREINTOITEM(TEMP);                                             02683500 T  0276:0
               END;                                                             02683600 T  0276:3
            GO ATTEXIT;                                                         02683700 T  0276:3
ATTEXIT::                                                                       02685000 T  0277:1
            P(XIT);                                                             02686000 T  0277:1
        END OF COBOLATT;                                                        02687000 T  0278:1
                                                                                      SIZE= 0279 WORDS
```

```
PROCEDURE COBOLDC;       % INTRINSIC NUMBER 167                                  02690000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00263
        BEGIN                                                                    02690020 T  0000:0
        REAL CODE       = -1;    % 0=READ,1=WRITE,2=SEEK,6=WRTBLK,               02690040 T  0000:0
        NAME DLOC       = -2;    % POINTS TO BUFFER I/O DESC                      02690060 T  0000:0
        REAL NUMWDS     = -3,    % # WDS TO BE WRITTEN                            02690080 T  0000:0
             KEY        = -4,    % RANDOM RECORD ADDRESS OR CARRAGE RTN           02690100 T  0000:0
             EXPSTATAR  = -4,    % AREA TO EXPAND STATUS INTO                     02690120 T  0000:0
             CHNNL      = -4,    % LP CHANNEL SKIP                                02690140 T  0000:0
             LINES      = -5,    % # LINES TO BE SPACED                           02690160 T  0000:0
             TIMEOUT    = -5,    % UNTIL PORTION OF DATA COM                      02690180 T  0000:0
             SKIPAFT    = -6     % 1=SPACE AFTER PRINT                           02690200 T  0000:0
        ;INTEGER                                                                 02690220 T  0000:0
             STATN      = -6,    % DATA COMM STATION (BUFFER)                     02690240 T  0000:0
             TUNR       = -7;    % DATA COMM TERMINAL UNIT                        02690260 T  0000:0
        %LOCALS                                                                  02690280 T  0000:0
          REAL  COBOLCONTROL=23;  % FOR LINKAGE BY USE ROUTINES                   02690300 T  0000:0
          REAL  COBOLINDEX  =22;  % FOR LINKAGE BY USE ROUTINES                   02690320 T  0000:0
          REAL  DEST    ;         % DESTINATION IN RANDOM MOVE                    02690340 T  0000:0
          ARRAY FIB [*];          % FIB ARRAY                                     02690360 T  0000:0
          REAL  FILECTRL =12  ;   % USED TO CALL COBOLFCR                         02690380 T  0000:0
          NAME  FLOC;             % POINTER TO FIB                                02690400 T  0000:0
          ARRAY FPB = 3[*];       % FILE PARAMETER BLOCK                          02690420 T  0000:0
          ARRAY H[*];             % DISK FILE HEADER                              02690440 T  0000:0
          NAME  MEM = 2;          % DUMMY DATA DESC                               02690460 T  0000:0
          ARRAY PGUSE=24[*];      % PROGRAM USE ROUTINES                          02690480 T  0000:0
          REAL  RTOG;             % 1=I/O DONE THIS ROUND                         02690500 T  0000:0
          REAL  T;                % TEMPORARY                                     02690520 T  0000:0
          REAL  TECHCOFLO;        % USED FOR TECH-C OVER FLOWS                    02690540 T  0000:0
          REAL  UNITYPE;          % STORE UNIT TYPE FOR MANY TESTS                02690560 T  0000:0
          REAL  X1;               % *DO*NOT*SEPARATE X1 & X2 THEY ARE             02690580 T  0000:0
          REAL  X2;               % USED IN CONJUNCTION FOR TECHC OFLOWS          02690600 T  0000:0
          INTEGER  BS = X1;       % USED IN COMPUTING DISK ADDR                   02690620 T  0000:0
          INTEGER  RT = X2;       % USED IN COMPUTING DISK ADDR                   02690640 T  0000:0
DEFINE                                                                           02690660 T  0000:0
          ARROW             = P(0,NOT,NUMWDS,TIP,INX,+)#,                        02690680 T  0000:0
                                % THIS INSERTS THE GROUP MARK                    02690700 T  0000:0
          BADKEY            = FIB[13].[19:1]#,  % BAD KEY RANDOM DISK           02690720 T  0000:0
          BCOUNT            = FIB[6]#,          % BLOCK COUNT                   02690740 T  0000:0
          BINARY            = FIB[13].[24:1]#,  % 1=BINARY,0=ALPHA              02690760 T  0000:0
          BOUNDED           = FIB[9].[2:1]#,    % TRUE IF BOUNED FROM ABOVE     02690780 T  0000:0
          BREAK             = FIB[9] # 0 #   ,  % BREAKOUT RESTART POINT        02690800 T  0000:0
          BREAKOUT          = IF(RCOUNT MOD FIB[9])=0 THEN                       02690820 T  0000:0
                                P(0,0,12,COM,DEL,DEL)#,% CALL BREAKOUT           02690840 T  0000:0
          BUFFNUM           = FIB[13].[1:9] #,  % # OF BUFFS REQUSTED          02690860 T  0000:0
          BUFFSIZE          = FIB[18].[3:15]#,  % BUFFER SIZE  (REQUESTED)      02690880 T  0000:0
          BUFFSZ            = FIB[18][8:8:10]#, % SIZE FOR CONCATINATES         02690900 T  0000:0
          BUFSTATUS         = FIB[14] #,        % STATUS AFTER SEEKDC           02690920 T  0000:0
          BUFTOP            = FIB [16]#,        % USED ON I-O AND RANDOM        02690940 T  0000:0
          BUILDSTATNWD      =P((STATN+SKIPAFT)& % BUILD STATION WORD FOR DC      02690960 T  0000:0
                             P(DUP)[14:44:4]&(TUNR+TUNR)[9:44:4])#,              02690980 T  0000:0
          CHECK(CHECK1)     = IF P(DUP)#(CHECK1) THEN P(CHECK1,0,FLOC,#,         02691000 T  0000:0
           ONERR(ONERR1)    =   ONERR1,17,COM,DEL,DEL,DEL,DEL);    P(DEL)#,      02691020 T  0000:0
                                % THE ABOVE ARE USED ON BLOCK+REC CHKS          02691040 T  0000:0
          CLEARSTATUS       =P(0,TIP,+)#,       % CLEAR BUFF[0] FOR WRITE      02691060 T  0000:0
          CLOSEANDOPEN      =P(MKS,1,0,FLOC,4,FILECTRL, %CLOSE NO RWD          02691080 T  0000:0
                               MKS,FLOC,1,FILECTRL)#,   % OPEN INPUT            02691100 T  0000:0
```

```
COUNT           = FIB[12] #,          % USED FOR BLOCKING TECH=A,B          02691120 T  0000:0
DCBUFRLS        = P(NUMBUF,DLOC,16,   % DATA COMM BUFFER RELEASE           02691140 T  0000:0
                    11,COM,DEL,DEL,DEL)#,                                  02691160 T  0000:0
DELAY           = TIP.[20:1] #,       % THIS ALLOWS ONE CYCLE DELY          02691180 T  0000:0
DONE            = TIP.[19:1] #,       % 1= IO COMPLETED                     02691200 T  0000:0
DISK            = (UNITYPE=4) #,      % DISK IS UNIT TYPE OF 4              02691220 T  0000:0
FNAM            = FIB[4].[13:11]#,    % FILE NAME INDEX IN FPB              02691240 T  0000:0
ENDFILE         = FIB[5].[40:1] #,    % ALREADY PASSED EOF                  02691260 T  0000:0
ENDPROCESS      = FIB[5].[39:2]#,     % SEE OPTIONAL AND ENDFILE            02691280 T  0000:0
ENDREEL         = X2 #,               % USED ONLY ON READ                   02691300 T  0000:0
EOF             =((*DLOC).[27:1])#,   % FIRST EOF OR EOT                    02691320 T  0000:0
FOREVER         =(NOT 0).[9:39] #,    % UNTIL END TIME                      02691340 T  0000:0
EXPAND          = *P(.EXPSTATAR) #,   % EXPAND CELL CHECK                   02691360 T  0000:0
EXPANDSTATUS    = P(TIP,0,0,EXPAND,   % EXPAND STATUS WORD                  02691380 T  0000:0
                    27,COM,DEL,DEL,DEL,DEL)#,                              02691400 T  0000:0
GETSEG          =  P(FPB[(BS+FNAM)+3],FPB[BS],FPB[BS+1],                   02691420 T  0000:0
                    T,H,4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL)#,                02691440 T  0000:0
HASH            =IF NOT DISK THEN IF FIB[8]>0 THEN                         02691460 T  0000:0
                                P(MKS,FLOC,*FIB[8],3,COC)#,                02691480 T  0000:0
                    % ABOVE CALLS ROUTINES FOR HASH ACCUMULATON            02691500 T  0000:0
HASHTOT         =IF FIB[8] >0 THEN IF P(MKS,FLOC,*FIB[8],0,COC)            02691520 T  0000:0
                    THEN IOERR(18)#,% CHECKS HASH TOTALS                   02691540 T  0000:0
HOWOPEN         = FIB[5].[41:3]#,     % 1=OPEN INPUT,0= OPEN OUTPT          02691560 T  0000:0
                                      % 1 > CLOSED                         02691580 T  0000:0
INVALIDUSER     = FIB[5]<0#,          % INVALID USER NOT PARITY             02691600 T  0000:0
IOERR(IOERR1)   = P(0,FLOC,IOERR1,17,COM,DEL,DEL,DEL)#,                    02691620 T  0000:0
                    % ABOVE CALLS IOERROR ROUTINE                          02691640 T  0000:0
IOMASK          = DEST #,             % HAS IOMASK TO SAVE C=REL            02691660 T  0000:0
LABEQ           = FIB[5].[17:1] #,    % LABEL EQUATED FROM DISK             02691680 T  0000:0
LASTDONE        = FIB[13].[21:1] #,   % NOT OF LAST OPERATION DONE          02691700 T  0000:0
LASTIO          = FIB[13].[46:1]#,    %LAST WAS PHYSICAL READ              02691720 T  0000:0
LBLPTR          = FLOC[1] #,          % LABEL POINTER                       02691740 T  0000:0
LINEPRINT       = UNITYPE=1 OR UNITYPE=7 OR UNITYPE=12 #,                  02691760 T  0000:0
                      % 1= LP , 7 = PBT , 12 = PBD                         02691780 T  0000:0
LSUBL           = FIB [1]   #,        % LOWER BOUND FOR RANDOM              02691800 T  0000:0
LSUBU           = FIB [3] #,          % UPPER BOUND FOR DISK REC            02691820 T  0000:0
MAXR            = FIB[18][8:38:10]#,% MAX REC SZ FOR CONCATS               02691840 T  0000:0
MAXREC          = FIB[18].[33:15]#,   % MAX REC SZ                          02691860 T  0000:0
NONSTD          = FIB [5].[16: 1]#,   % NON=STANDARD LABELS                 02691880 T  0000:0
NUMBUF          = FIB[13].[10: 9]#,   % NUMBER OF BUFFERS ASSIGNED          02691900 T  0000:0
NUMBSPC         = H[9].[43:5]#,       % ROWS SPECIFIED   %CJC 020           02691920 T  0000:0
NUMREC          = FIB[11] #,          % RECORDS PER BLOCK                   02691940 T  0000:0
NXTREEL         = P(MKS,2,1,FLOC,4,   % THIS DOES REEL SWITCHING            02691960 T  0000:0
                    FILECTRL)#,       %                                    02691980 T  0000:0
OPENIO          = FIB[13].[22:1]#,    % 1= OPEN INPUT=OUPUT (DISK)          02692000 T  0000:0
OPTIONAL        = FIB[ 5].[39:1]#,    % OPTIONAL FILE NOT PRESENT           02692020 T  0000:0
PARITY          = TIP.[28:1]#,        % PARITY BIT ON DESC                  02692040 T  0000:0
PRESENT         =((*DLOC).[2:1])#,    % CHECKS PRESENTSBIT                  02692060 T  0000:0
PROPER          =21+CODE+CODE+REVERSE#,% GENERATES PROPER IOERR             02692080 T  0000:0
PUNCH           = UNITYPE=6#,  % UNIT IS CARD PUNCH     %TR 830 I         02692100 T  0000:0
PURGE           = TIMEOUT.[FF]#0#,    % TRUE IF LINE TO BE PURGED           02692120 T  0000:0
RANDOM          = TECHCOFLO#,         % 1 = RANDOM DISK                     02692140 T  0000:0
RCOUNT          = FIB[7] #,           % RECORD COUNT                        02692160 T  0000:0
READER          =(UNITYPE MOD 11=0)#,% 0=READER 11=PSUDOREADER             02692180 T  0000:0
READLBL         =P(DLOC INX 0,11,11   % THIS READS THE LABEL,              02692200 T  0000:0
                    ,COM,DEL,DEL)#,   %                                    02692220 T  0000:0
RECPERBLK       = H[0].[30:12] #,     % RECORDS PER BLOCK                   02692240 T  0000:0
```

```
        REMOTEIO        =P(BUFFSIZE,DLOC,   % READ & WRITE BATCH SYSTEM          02692260 T  0000:0
                         FOREVER,(IF CODE THEN LINES ELSE 1),      %FOR        02692280 T  0000:0
                         KEY=0,CFX,TIP,CODE,36,COM,          %REMOTE OR      02692300 T  0000:0
                         DEL,DEL,DEL,DEL,DEL,1,SUB,RTN)#,%TYPE 19 FILES      02692320 T  0000:0
        REMOTEREAD      =P(BUFFSIZE,TIP,0,  % READ FOR TSS                    02692340 T  0000:0
                         (-13),COM,0,RTN)#, %                                 02692360 T  0000:0
        REMOTEWRIT      =P(TIP,NUMWDS  x8,  % WRITE FOR TSS                   02692380 T  0000:0
                         LINES,KEY,CFX,0,(-11),COM,DEL,RTN)#,                 02692400 T  0000:0
        RESETPARITY     = DLOC[0]←TIP&0[28:28:1]#,%RESET PARITY BIT DISK      02692420 T  0000:0
        RESETREADBIT    = 0[24:24:1]#,      % USED TO TURN OFF READ BIT        02692440 T  0000:0
        REVERSE         = FIB[5].[44:1] #,  % 1=REVERSE                       02692460 T  0000:0
        ROTATEBUF       =P(NUMBUF,DLOC,13,11 % ROTATES BUFFERS WITH           02692480 T  0000:0
                          ,COM,DEL,DEL,DEL)#,%   NO I/O                       02692500 T  0000:0
        ROWLGTH         = H[1]#,            % ROW LGTH FROM HEADER             02692520 T  0000:0
        SANDBKEY        = FIB[13].[19:2] #, % SEEK AND BADKEY                 02692540 T  0000:0
        SEEKDC          =P(0&NUMWDS[14:44:4] % DATA COM SEEK AND XIT          02692560 T  0000:0
                         &CHNNL  [9:44:4],DLOC,5,11,COM,XIT)#,                02692580 T  0000:0
        SEEKEY          = FIB[13].[20:1]#,  % SEEK WAS DONE                   02692600 T  0000:0
        SERIAL          = FIB[4].[27:3]=0 #, % FILE ACCESS = SERIAL           02692620 T  0000:0
        SEGPERBLK       = H[0].[42:6] #,    % SEGMENTS PER BLOCK               02692640 T  0000:0
        SETPRESENTSBIT  =P(TIP OR MEM ,DLOC,←)#,% SET PRESENCE BIT            02692660 T  0000:0
$ SET OMIT = NOT(TIMESHARING)                                                 02692680 T  0000:0
        SLEEP           = 36 #,                                               02692700 T  0000:0
$ POP OMIT                                                                    02692720 T  0000:0
$ SET OMIT = TIMESHARING                                                      02692740 T  0000:0
        TAPEE           = TIP.[7:1] #,      % 1= TAPES   0=ALL ELSE            02692800 T  0000:0
        TECH            = FIB[5].[46:2] #,  %TECHNIQUE                        02692820 T  0000:0
        TECHA           =(FIB[5].[46:2]=1) #,% TECHNIQUE=A                    02692840 T  0000:0
        TECHC           =(FIB[5].[46:2]=3) #,% TECHNIQUE=C                    02692860 T  0000:0
        TERM(TERM1)     = P(1,FLOC,TERM1,17,COM)#,%TERMINATE I/O ERROR        02692880 T  0000:0
        TIP             = (*DLOC) #,        % LOAD I/O DESC                   02692900 T  0000:0
        TOTREC          = H[7]    #,        % TOTAL RECORDS ON FILE            02692920 T  0000:0
        UNLABELED       = (FIB[4].[2:1])#,  % UNLABELED FILE                   02692940 T  0000:0
        UT              = (FIB[4].[8:4])#,  % HARDWARE TYPE                    02692960 T  0000:0
        WAITDC          = P(DLOC,IOMASK,    % THIS SLEEPS ON I/O COMPLE        02692980 T  0000:0
                            SLEEP,COM,=)#,  % AND LEAVES A FALSE ON STK        02693000 T  0000:0
        WAITIO          = P(DLOC,IOMASK,    % THIS SLEEPS ON I/O               02693020 T  0000:0
                          SLEEP,COM,DEL,DEL)#,% WAITING FOR A COMPLETE        02693040 T  0000:0
        WORDSLEFT       = FIB[17]#,         % WORDS LEFT IN BUFFER             02693060 T  0000:0
        WRITEAFTEREOF   = FIB[13].[44:2]#,  %                                 02693080 T  0000:0
        WRITBACK        = FIB[13].[23:1]#;  % FLAG TO SAY WRITE BACK           02693100 T  0000:0
LABEL   LPRETURN,IOUT,START,IODONE,RANDOMLBL,SEEKRTN,SETUP;                   02693120 T  0000:0
LABEL   IMPROPER,DCPRL,FIXSTATNWD,DIDDLE,DIDDLEWRT,SERIALIO,EOFSETCK;         02693140 T  0000:0
LABEL   DATACOM,RANDOMIO;                                     %CUBE XIX I     02693160 T  0000:0
 START :                                                                      02693180 T  0000:0
            FIB ← *(FLOC ← (NOT 2) INX DLOC);                                 02693200 T  0002:2
            IOMASK ← @2000000000;                                             02693220 T  0004:3
            IF CODE THEN            % DC WRITE                               02693240 T  0005:2
       BEGIN    RTOG ← (-4);        % SET ALGOLIO FOR COBOLDCWR               02693260 T  0005:3
                CLEARSTATUS;                                                  02693280 T  0007:1
                GO TO FIXSTATNWD;                                             02693300 T  0008:1
       END;                                                                   02693320 T  0010:0
            IF  BUFSTATUS=0 THEN                                              02693340 T  0010:0
       BEGIN    RTOG ← 1;           % SET ALGOLIO FOR READC                  02693360 T  0011:0
FIXSTATNWD:     BUILDSTATNWD;                                                 02693380 T  0012:1
                GO TO DCPRL;                                                  02693400 T  0015:2
       END;                                                                   02693420 T  0016:0
```

```
                IF DELAY THEN       % THIS IS USED TO INHIBIT BUFFER
                    DCBUFRLS;       % ROTATION ON 1ST READ
                IF TIMEOUT < 0 THEN % UNTIL END READ
            BEGIN   WAITDC;         % THIS LEAVES 0 ON STACK
                    DLOC[0]← TIP&1[20:47:1]; % SET DELAY
            END  ELSE BEGIN
                    P(BUFSTATUS);       % SET ALGOLIO FOR READSOUGHT
        DCPRL:      P(IF TIMEOUT < 0 THEN FOREVER ELSE TIMEOUT,[CF]
                      ×60&(PURGE)[1:47:1],XCH,DLOC,15-RTOG,11,COM,
                      DEL,DEL,DEL,1,≠); %THIS LEAVES 1 OR 0 ON STACK
            END;                        %DEPENDING ON HOW IO WAS.
                IF EXPAND ≠ 0 THEN EXPANDSTATUS;
                P(PRESENT,NOT,OR); %THIS ORS RESULTS OF ABOVE WITH
                SETPRESENTSBIT;    % PRESENTS BIT AND IS RETURNED TO
                                   % PROGRAM.
                P(RTN);
        END COBOLDC;
```

```
02693440 T  0016:0
02693460 T  0017:0
02693480 T  0020:1
02693500 T  0021:0
02693520 T  0022:3
02693540 T  0024:3
02693560 T  0025:1
02693580 T  0025:3
02693600 T  0028:2
02693620 T  0033:2
02693640 T  0034:3
02693660 T  0034:3
02693680 T  0039:1
02693700 T  0040:3
02693720 T  0042:1
02693740 T  0042:1
02693760 T  0042:2
                SIZE= 0043 WORDS
```

```
PROCEDURE COBOLIO;
```

```
02700000 T  0000:0
                START OF REL SEGMENT; DISK ADDRESS = 00265
```

```
        BEGIN
        REAL CODE       = -1;     % 0=READ,1=WRITE,2=SEEK,6=WRTBLK,
        NAME DLOC       = -2;     % POINTS TO BUFFER I/O DESC
        REAL NUMWDS     = -3,     % # WDS TO BE WRITTEN
             KEY        = -4,     % RANDOM RECORD ADDRESS OR CARRAGE RTN
             EXPSTATAR  = -4,     % AREA TO EXPAND STATUS INTO
             CHNNL      = -4,     % LP CHANNEL SKIP
             LINES      = -5,     % # LINES TO BE SPACED
             TIMEOUT    = -5,     % UNTIL PORTION OF DATA COM
             SKIPAFT    = -6      % 1=SPACE AFTER PRINT
        ;INTEGER
             STATN      = -6,     % DATA COMM STATION (BUFFER)
             TUNR       = -7;     % DATA COMM TERMINAL UNIT
          ARRAY MKSCW=-4[*];
        %LOCALS
          REAL  COBOLCONTROL=23;   % FOR LINKAGE BY USE ROUTINES
          REAL  COBOLINDEX  =22;   % FOR LINKAGE BY USE ROUTINES
          REAL  DEST     ;         % DESTINATION IN RANDOM MOVE
          ARRAY FIB [*];           % FIB ARRAY
          REAL  FILECTRL =12  ;    % USED TO CALL COBOLFCR
          NAME  FLOC;              % POINTER TO FIB
          ARRAY FPB = 3[*];        % FILE PARAMETER BLOCK
          ARRAY H[*];              % DISK FILE HEADER
          REAL  IOMASK;            % TO SAVE C-REL CALL
          NAME  MEM = 2;           % DUMMY DATA DESC
          ARRAY PGUSE=24[*];       % PROGRAM USE ROUTINES
          REAL  RTOG;              % 1=I/O DONE THIS ROUND
          REAL  T;                 % TEMPORARY
          REAL  TECHCOFLO;         % USED FOR TECH-C OVER FLOWS
          REAL  UNITYPE;           % STORE UNIT TYPE FOR MANY TESTS
          REAL  X1;                % *DO*NOT*SEPARATE X1 & X2 THEY ARE
          REAL  X2;                % USED IN CONJUNCTION FOR TECHC OFLOWS
          INTEGER  BS = X1;        % USED IN COMPUTING DISK ADDR
```

```
02700100 T  0000:0
02700200 T  0000:0
02700300 T  0000:0
02700400 T  0000:0
02700500 T  0000:0
02700600 T  0000:0
02700700 T  0000:0
02700800 T  0000:0
02700900 T  0000:0
02701000 T  0000:0
02701100 T  0000:0
02701200 T  0000:0
02701300 T  0000:0
02701310 C  0000:0
02701400 T  0000:0
02701500 T  0000:0
02701600 T  0000:0
02701700 T  0000:0
02701800 T  0000:0
02701900 T  0000:0
02702000 T  0000:0
02702100 T  0000:0
02702200 T  0000:0
02702250 T  0000:0
02702300 T  0000:0
02702400 T  0000:0
02702500 T  0000:0
02702600 T  0000:0
02702700 T  0000:0
02702800 T  0000:0
02702900 T  0000:0
02703000 T  0000:0
02703100 T  0000:0
```

```
        INTEGER  RT = X2;          % USED IN COMPUTING DISK ADDR                    02703200 T  0000:0
 $ SET OMIT = NOT SHAREDISK                                                        02703204 T  0000:0
DEFINE                                                                             02703300 T  0000:0
        ARROW           = P(0,NOT,NUMWDS,TIP,INX,←)#,                              02703400 T  0000:0
                            % THIS INSERTS THE GROUP MARK                          02703500 T  0000:0
        BADKEY          = FIB[13].[19:1]#,  % BAD KEY RANDOM DISK                  02703600 T  0000:0
        BCOUNT          = FIB[6]#,          % BLOCK COUNT                           02703700 T  0000:0
        BINARY          = FIB[13].[24:1]#,  % 1=BINARY,0=ALPHA                     02703800 T  0000:0
        BOUNDED         = FIB[9].[2:1]#,    % TRUE IF BOUNED FROM ABOVE             02703900 T  0000:0
        BREAK           = FIB[9] ≠ 0 #   ,  % BREAKOUT RESTART POINT                02704000 T  0000:0
        BREAKOUT        = IF(RCOUNT MOD FIB[9])=0 THEN                             02704100 T  0000:0
                            P(0,0,12,COM,DEL,DEL)#,% CALL BREAKOUT                 02704200 T  0000:0
        BUFFNUM         = FIB[13].[1:9] #,  % # OF BUFFS REQUSTED                  02704300 T  0000:0
        BUFFSIZE        = FIB[18].[3:15]#,  % BUFFER SIZE  (REQUESTED)             02704400 T  0000:0
        BUFFSZ          = FIB[18][8:8:10]#, % SIZE FOR CONCATINATES                02704500 T  0000:0
        BUFSTATUS       = FIB[14] #,        % STATUS AFTER SEEKDC                   02704600 T  0000:0
        BUFTOP          = FIB [16]#,        % USED ON I=0 AND RANDOM                02704700 T  0000:0
        BUILDSTATNWD    =P((STATN←SKIPAFT)& % BUILD STATION WORD FOR DC            02704800 T  0000:0
                         P(DUP)[14:44:4]&(TUNR←TUNR)[9:44:4])#,                    02704900 T  0000:0
        CHECK(CHECK1)   = IF P(DUP)≠(CHECK1) THEN P(CHECK1,0,FLOC,#,               02705000 T  0000:0
         ONERR(ONERR1)  =   ONERR1,17,COM,DEL,DEL,DEL,DEL);    P(DEL)#,            02705100 T  0000:0
                            % THE ABOVE ARE USED ON BLOCK+REC CHKS                 02705200 T  0000:0
        CLEARSTATUS     =P(0,TIP,←)#,       % CLEAR BUFF[0] FOR WRITE            02705300 T  0000:0
        CLOSEANDOPEN    =P(MKS,1,0,FLOC,4,FILECTRL, %CLOSE NO RWD                  02705350 T  0000:0
                          MKS,FLOC,1,FILECTRL)#,   % OPEN INPUT                   02705351 T  0000:0
        COUNT           = FIB[12] #,        % USED FOR BLOCKING TECH=A,B           02705400 T  0000:0
        DCBUFRLS        = P(NUMBUF,DLOC,16, % DATA COMM BUFFER RELEASE             02705500 T  0000:0
                            11,COM,DEL,DEL,DEL)#,                                  02705600 T  0000:0
        DELAY           = TIP.[20:1] #,     % THIS ALLOWS ONE CYCLE DELY          02705700 T  0000:0
        DONE            = TIP.[19:1] #,     % 1= IO COMPLETED                     02705800 T  0000:0
        DISK            = (UNITYPE=4) #,    % DISK IS UNIT TYPE OF 4              02705900 T  0000:0
        FNAM            = FIB[4].[13:11]#,  % FILE NAME INDEX IN FPB               02706000 T  0000:0
        ENDFILE         = FIB[5].[40:1] #,  % ALREADY PASSED EOF                   02706100 T  0000:0
        ENDPROCESS      = FIB[5].[39:2]#,   % SEE OPTIONAL AND ENDFILE             02706200 T  0000:0
        ENDREEL         = X2 #,             % USED ONLY ON READ                   02706300 T  0000:0
        EOF             =((*DLOC).[27:1])#, % FIRST EOF OR EOT                    02706400 T  0000:0
        FOREVER         =(NOT 0).[9:39] #,  % UNTIL END TIME                      02706500 T  0000:0
        EXPAND          = *P(.EXPSTATAR) #, % EXPAND CELL CHECK                    02706600 T  0000:0
        EXPANDSTATUS    = P(TIP,0,0,EXPAND, % EXPAND STATUS WORD                   02706700 T  0000:0
                            27,COM,DEL,DEL,DEL,DEL)#,                              02706800 T  0000:0
        GETSEG          =  P(FPB[(BS:=FNAM)+3],FPB[BS],FPB[BS+1],                  02706900 T  0000:0
                           T,H,4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL)#,                 02707000 T  0000:0
        HASH            =IF NOT DISK THEN IF FIB[8]>0 THEN                         02707100 T  0000:0
                                P(MKS,FLOC,*FIB[8],3,COC)#,                        02707200 T  0000:0
                         % ABOVE CALLS ROUTINES FOR HASH ACCUMULATON               02707300 T  0000:0
        HASHTOT         =IF FIB[8] >0 THEN IF P(MKS,FLOC,*FIB[8],0,COC)            02707400 T  0000:0
                            THEN IOERR(18)#,% CHECKS HASH TOTALS                   02707500 T  0000:0
        HOWOPEN         = FIB[5].[41:3]#,   % 1=OPEN INPUT,0= OPEN OUTPT           02707600 T  0000:0
                                            % 1 > CLOSED                          02707700 T  0000:0
        INVALIDUSER     = FIB[5]<0#,        % INVALID USER NOT PARITY             02707800 T  0000:0
        IOERR(IOERR1)   = P(0,FLOC,IOERR1,17,COM,DEL,DEL,DEL)#,                    02707900 T  0000:0
                            % ABOVE CALLS IOERROR ROUTINE                         02708000 T  0000:0
        LABEQ           = FIB[5].[17:1] #,  % LABEL EQUATED FROM DISK             02708200 T  0000:0
        LASTDONE        = FIB[13].[21:1] #, % NOT OF LAST OPERATION DONE           02708300 T  0000:0
        LASTIO          = FIB[13].[46:1]#,  %LAST WAS PHYSICAL READ                02708350 T  0000:0
        LBLPTR          = FLOC[1] #,        % LABEL POINTER                       02708400 T  0000:0
        LINEPRINT       = UNITYPE=1 OR UNITYPE=7 OR UNITYPE=12 #,                  02708500 T  0000:0
```

```
                    % 1= LP , 7 = PBT , 12 = PBD                          02708600 T  0000:0
LSUBL           = FIB [1]   #,          % LOWER BOUND FOR RANDOM           02708700 T  0000:0
LSUBU           = FIB [3] #,            % UPPER BOUND FOR DISK REC         02708800 T  0000:0
MAXR            = FIB[18][8:38:10]#,% MAX REC SZ FOR CONCATS              02708900 T  0000:0
MAXREC          = FIB[18].[33:15]#, % MAX REC SZ                          02709000 T  0000:0
NONSTD          = FIB [5].[16: 1]#, % NON-STANDARD LABELS                 02709100 T  0000:0
NUMBUF          = FIB[13].[10: 9]#, % NUMBER OF BUFFERS ASSIGNED          02709200 T  0000:0
NUMBSPC         = H[9].[43:5]#,         % ROWS SPECIFIED   %CJC 020        02709300 T  0000:0
NUMREC          = FIB[11] #,            % RECORDS PER BLOCK                02709400 T  0000:0
NXTREEL         = P(MKS,2,1,FLOC,4, % THIS DOES REEL SWITCHING            02709500 T  0000:0
                  FILECTRL)#,           %                                 02709600 T  0000:0
OPENIO          = FIB[13].[22:1]#,  % 1= OPEN INPUT-OUPUT (DISK)          02709700 T  0000:0
OPTIONAL        = FIB[ 5].[39:1]#,  % OPTIONAL FILE NOT PRESENT          02709800 T  0000:0
PARITY          = TIP.[28:1]#,          % PARITY BIT ON DESC               02709900 T  0000:0
PRESENT         =((*DLOC).[2:1])#,  % CHECKS PRESENTSBIT                  02710000 T  0000:0
PROPER          =21+CODE+CODE+REVERSE#,% GENERATES PROPER IOERR           02710100 T  0000:0
PUNCH           = UNITYPE=6#,  % UNIT IS CARD PUNCH    %TR 830 I        02710150 T  0000:0
PURGE           = TIMEOUT.[FF]#0#,  % TRUE IF LINE TO BE PURGED          02710200 T  0000:0
RANDOM          = TECHCOFLO#,           % 1 = RANDOM DISK                 02710300 T  0000:0
RCOUNT          = FIB[7] #,             % RECORD COUNT                    02710400 T  0000:0
READER          =(UNITYPE MOD 11=0)#,% 0=READER 11=PSUDOREADER            02710500 T  0000:0
READLBL         =P(DLOC INX 0,11,11  % THIS READS THE LABEL.              02710600 T  0000:0
                  ,COM,DEL,DEL)#,       %                                 02710700 T  0000:0
RECPERBLK       = H[0].[30:12] #,       % RECORDS PER BLOCK                02710800 T  0000:0
REMOTEIO        =P(BUFFSIZE,DLOC,   % READ & WRITE BATCH SYSTEM          02710900 T  0000:0
                 FOREVER,(IF CODE THEN LINES ELSE 1),      %FOR           02710950 T  0000:0
                 KEY=0,CFX,TIP,CODE,36,COM,          %REMOTE OR           02711000 T  0000:0
                .DEL,DEL,DEL,DEL,DEL,1,SUB,RTN)#,%TYPE 19 FILES           02711100 T  0000:0
REMOTEREAD      =P(BUFFSIZE,TIP,1,      % READ FROM TSS                   02711200 P  0000:0
                  (-13),COM,1,SUB,RTN)#,  %                               02711300 P  0000:0
REMOTEWRIT      =P(TIP,NUMWDS  x8,  % WRITE FOR TSS                       02711400 T  0000:0
                 LINES,KEY,CFX,0,(-11),COM,DEL,RTN)#,                     02711500 T  0000:0
RESETPARITY     = DLOC[0]←TIP&0[28:28:1]#,%RESET PARITY BIT DISK          02711550 T  0000:0
RESETREADBIT    = 0[24:24:1]#,          % USED TO TURN OFF READ BIT       02711600 T  0000:0
REVERSE         = FIB[5].[44:1] #,  % 1=REVERSE                           02711700 T  0000:0
ROTATEBUF       =P(NUMBUF,DLOC,13,11 % ROTATES BUFFERS WITH               02711800 T  0000:0
                  ,COM,DEL,DEL,DEL)#,%   NO I/O                           02711900 T  0000:0
ROWLGTH         = H[1]#,                % ROW LGTH FROM HEADER            02712000 T  0000:0
SANDBKEY        = FIB[13].[19:2] #,  % SEEK AND BADKEY                    02712100 T  0000:0
SEEKDC          =P(0&NUMWDS[14:44:4] % DATA COM SEEK AND XIT              02712200 T  0000:0
                 &CHNNL  [9:44:4],DLOC,5,11,COM,XIT)#,                    02712300 T  0000:0
SEEKEY          = FIB[13].[20:1]#,   % SEEK WAS DONE                      02712400 T  0000:0
SERIAL          = FIB[4].[27:3]=0 #, % FILE ACCESS = SERIAL               02712500 T  0000:0
SEGPERBLK       = H[0].[42:6] #,        % SEGMENTS PER BLOCK              02712600 T  0000:0
SETPRESENTSBIT  =P(TIP OR MEM ,DLOC,←)#,% SET PRESENCE BIT                02712700 T  0000:0
$ SET OMIT = NOT(TIMESHARING)                                             02712800 T  0000:0
SLEEP           = 36 #,                                                   02712900 T  0000:0
$ POP OMIT                                                                02712950 T  0000:0
$ SET OMIT = TIMESHARING                                                  02713000 T  0000:0
TAPEE           = TIP.[7:1] #,          % 1= TAPES   0=ALL ELSE           02713200 T  0000:0
TECH            = FIB[5].[46:2] #,  %TECHNIQUE                            02713250 T  0000:0
TECHA           =(FIB[5].[46:2]=1) #,% TECHNIQUE-A                        02713300 T  0000:0
TECHC           =(FIB[5].[46:2]=3) #,% TECHNIQUE-C                        02713400 T  0000:0
TERM(TERM1)     = P(1,FLOC,TERM1,17,COM)#,%TERMINATE I/O ERROR            02713500 T  0000:0
TIP             = (*DLOC) #,            % LOAD I/O DESC                   02713600 T  0000:0
TOTREC          = H[7]    #,            % TOTAL RECORDS ON FILE           02713700 T  0000:0
UNLABELED       = (FIB[4].[2:1])#,   % UNLABELED FILE                     02713800 T  0000:0
```

```
        UT              = (FIB[4],[8:4])#,    % HARDWARE TYPE
        WAITDC          = P(DLOC,IOMASK,      % THIS SLEEPS ON I/O COMPLE
                            SLEEP,COM,=)#,    % AND LEAVES A FALSE ON STK
        WAITIO          = P(DLOC,IOMASK,      % THIS SLEEPS ON I/O
                          SLEEP,COM,DEL,DEL)#,% WAITING FOR A COMPLETE
        WORDSLEFT       = FIB[17]#,           % WORDS LEFT IN BUFFER
        WRITEPARITY     = FIB[5].[3:1]#,    % INDICATES FORCED REELSWITCH
        WRITEAFTEREOF   = FIB[13].[44:2]#,  %
        WRITBACK        = FIB[13].[23:1]#;    % FLAG TO SAY WRITE BACK
LABEL   LPRETURN,IOUT,START,IODONE,RANDOMLBL,SEEKRTN,SETUP;
LABEL   IMPROPER,DCPRL,FIXSTATNWD,DIDDLE,DIDDLEWRT,SERIALIO,EOFSETCK;
LABEL   DATACOM,RANDOMIO,REREAD;
SUBROUTINE GOUSE;                       % THIS CALLS USE ROUTINES
        BEGIN     COBOLINDEX ← T.[26:10];
                  P(MKS,T.[38:10],[COBOLCONTROL]); %THIS EXECUTES THE
        END GOUSE;                                 % CODE SEGMENT
SUBROUTINE MAYBEPARITY;
        BEGIN
                  SETPRESENTSBIT;
                  IF (T ←RT ←PGUSE[(DISK AND OPENIO)×3+9].[1:23])≠0
                     THEN GOUSE ;
                  IF (T←FIB [15].[1:23]) ≠0 THEN GOUSE;
$ SET OMIT = NOT SHAREDISK
                  IF RTOG THEN
                     IF (T OR RT) = 0 THEN IOERR(19);
        END MAYBEPARITY;
SUBROUTINE MOVREC;                  %THIS MOVES RECORDS TO&FROM WORK AREA
        BEGIN     IF CODE ≠ 4 THEN
                    P(BUFTOP INX(BS←(NUMWDS × (RCOUNT MOD NUMREC))+1))
                  ELSE
                    P(XCH);  %PICK UP VALUE LEFT FROM SERIALIO
                  P(BUFTOP INX (BUFFSIZE + 2));  %FIND END OF BUFFER
                  DEST := IF CODE THEN P (XCH) ELSE P;
                  STREAM (FROM:=P:NUMWDS,E:=NUMWDS.[36:6], XX:=DEST);
                       BEGIN
                       SI←FROM; E(DS←32WDS;DS←32WDS); DS←NUMWDS WDS;
                       END;
                  IF CODE THEN DEST := P
                  ELSE BEGIN
                       P(DEL);
                       IF CODE=0 AND PARITY THEN
                       BEGIN
$ SET OMIT = NOT SHAREDISK
                            MAYBEPARITY;
                       END;
                       END;
                  DLOC[0] ← TIP& DEST[33:33:15]
        END MOVREC;
SUBROUTINE  DIDDLEREC;              % THIS ROUTINE GETS THE NEXT RECORD
            BEGIN                   %  FOR ALL SERIAL FILES (READ & WRITE)
                  WORDSLEFT ← T ;
                  DLOC[0] ← NUMWDS INX TIP;
                  RCOUNT ← *P(DUP) + 1;
                  IF BREAK THEN BREAKOUT;
                  IF PARITY THEN MAYBEPARITY;
            END DIDDLE;
$ SET OMIT = NOT SHAREDISK
```

```
02713900 T  0000:0
02714000 T  0000:0
02714100 T  0000:0
02714200 T  0000:0
02714300 T  0000:0
02714400 T  0000:0
02714410 T  0000:0
02714450 T  0000:0
02714500 T  0000:0
02714600 T  0000:0
02714700 T  0000:0
02714800 T  0000:0
02714900 T  0000:0
02715000 T  0001:0
02715100 T  0002:1
02715200 T  0003:2
02715300 T  0003:3
02715400 T  0004:0
02715450 T  0004:0
02715500 T  0005:2
02715600 T  0010:1
02715700 T  0012:0
02715709 T  0016:0
02715800 T  0016:0
02715900 T  0016:1
02716000 T  0020:2
02716100 T  0020:3
02716200 T  0021:0
02716300 T  0021:3
02716400 T  0025:2
02716500 T  0025:3
02716600 T  0026:2
02716700 T  0028:3
02716800 T  0030:3
02716900 T  0032:3
02717000 T  0032:3
02717100 T  0034:3
02717200 T  0035:0
02717300 T  0035:3
02717400 T  0036:3
02717500 T  0037:0
02717510 T  0039:0
02717519 T  0039:2
02717530 T  0039:2
02717540 T  0041:0
02717600 T  0041:0
02717700 T  0041:0
02717800 T  0041:3
02717900 T  0042:3
02718000 T  0043:0
02718100 T  0043:0
02718200 T  0044:1
02718300 T  0045:3
02718400 T  0047:3
02718500 T  0053:0
02718600 T  0056:0
02718604 T  0056:1
```

```
SUBROUTINE  PREL;                    % THIS DOES ACTUAL I/O                    02718700 T  0056:1
              BEGIN                                                           02718800 T  0057:0
                   IF NOT (RT LSS 0) THEN                                     02718900 T  0057:0
                   BEGIN                                                      02719000 T  0057:3
                   P( TIP,DLOC);                                              02719100 T  0058:1
                   IF WRITBACK THEN                 % DO SPECIAL WRITE-IO      02719200 T  0059:0
                   BEGIN     WRITBACK ← FALSE;      % TURN OFF READ BIT        02719300 T  0060:0
                             DLOC[0]← TIP&RESETREADBIT;% TO MAKE WRITE         02719400 T  0063:0
                   END;                                                       02719500 T  0065:0
                   P(PRL,DEL);                      % DO I-O                   02719600 T  0065:0
                   END;                                                       02719700 T  0065:3
                   BCOUNT ← *P(DUP) + (RTOG←1);   %COUNT BLOCK&SET IOTOG      02719900 T  0065:3
                   IF CODE = 2 THEN GO TO SEEKRTN;                            02719950 T  0068:1
                   RCOUNT ← *P(DUP) + 1;            % COUNT RECS              02720000 T  0069:2
                   IF NOT DONE THEN                                           02720100 T  0071:2
$ SET OMIT = NOT SHAREDISK                                                    02720109 T  0072:3
                   WAITIO;                                                    02720150 T  0072:3
                   IF BREAK THEN BREAKOUT;                                    02720200 T  0074:3
              END PREL;                                  % ON NEW DESC        02720300 T  0080:0
SUBROUTINE REFLECTCHECKER ;          % WRITE PARITY ROUTINE                   02720400 T  0080:1
              BEGIN                                                           02720500 T  0081:0
                   IF NOT EOF THEN %TAPE WRITE PARITY OR BLANK TAPE           02720550 T  0081:0
                        BEGIN                                                 02720570 T  0082:1
                        IF OPENIO AND DISK THEN IF(T←PGUSE[12].[1:23])≠0      02720600 T  0082:3
                             THEN GOUSE ELSE ELSE                             02720605 T  0086:3
                        IF (T←PGUSE[9].[24:24])≠0 THEN GOUSE;                 02720610 T  0089:2
                        IF (T←FIB[15].[1:23]) ≠ 0 THEN GOUSE;                 02720620 T  0094:0
                        TERM(20);                                             02720650 T  0098:0
                        END;                                                  02720670 T  0099:1
                   SETPRESENTSBIT;      % MAKE DESC PRESENT                    02720700 T  0099:1
                   IF NOT DISK THEN NXTREEL;% REEL SWITCH                     02720800 T  0100:3
              END RELECTCHECKER;                                              02720900 T  0103:2
SUBROUTINE SKIPPER;                  % THIS DOES SKIPPING ON LINE PRINTER      02721000 T  0103:3
         BEGIN     NUMBUF  ← 1;     % INHIBIT BUFFER ROTATION                 02721100 T  0104:0
                   IF CHNNL ≠ 0 THEN LINES := 1;                              02721200 T  0106:2
                   DLOC[0] ← TIP & 1 [18:47:1]                                02721300 T  0108:2
                                     &(16+CHNNL) [27:42:6];                   02721400 T  0109:1
                   FOR  T←2 STEP 2 UNTIL LINES DO                             02721500 T  0112:0
                        BEGIN                                                 02721600 T  0113:0
                        PREL;                                                 02721700 T  0113:0
                        IF NOT PRESENT THEN IF EOF THEN SETPRESENTSBIT        02721800 T  0114:0
                                  ELSE REFLECTCHECKER;                        02721810 T  0118:3
                        END;                                                  02721900 T  0120:0
                   IF LINES THEN                                              02722000 T  0122:1
                        BEGIN                                                 02722100 T  0122:2
                        DLOC[0]← TIP & (2-(2×(CHNNL≠0)))[27:46:2];            02722200 T  0123:0
                        PREL;                                                 02722300 T  0126:2
                        IF NOT PRESENT THEN IF EOF THEN SETPRESENTSBIT        02722400 T  0128:0
                                  ELSE REFLECTCHECKER;                        02722410 T  0132:3
                        END;                                                  02722500 T  0134:0
                   NUMBUF  ← BUFFNUM;  % RESTORE BUFFER FOR ROTATION          02722600 T  0134:0
         END  SKIPPER;                                                        02722700 T  0137:1
SUBROUTINE  REVREAD;                 % THIS DOES A READ REVERSE                02722800 T  0137:2
         BEGIN     DLOC[0] ← FLAG (FIB [16]);                                 02722900 T  0138:0
                   PREL;                                                      02723000 T  0139:1
                   FIB[16].[33:15]← TIP;                                      02723100 T  0140:0
                   WORDSLEFT ← MEM [1 INX TIP];                               02723200 T  0142:1
```

```
        END;
SUBROUTINE  READREV;                    % THIS HANDLES A READ REVERSE
        BEGIN     IF NOT TECHA THEN
                        BEGIN
                        REVREAD;
                        DLOC [0] ← NOT(WORDSLEFT-2) INX TIP;
                        END
                  ELSE
                        IF (WORDSLEFT := T) LEQ 0 THEN
                             BEGIN
                             REVREAD;
                             DLOC[0] ← (NOT(MAXREC - 2 )INX TIP)&MAXR;
                             END
                        ELSE BEGIN
                             DLOC[0]← NOT(NUMWDS-1)INX (TIP
                                       &(NOT TIP) [2:28:1]);
                             RCOUNT ← *P(DUP) + 1;
                        END ;
                  IF NOT PRESENT THEN
                        BEGIN
                        SETPRESENTSBIT;
                        IF EOF THEN
                             BEGIN
                             ENDFILE ← TRUE;
                             HASHTOT;
                             P (1,RTN);
                             END;
                        IF (T ←PGUSE[9]&FIB[15] [25:1:23])≠0 THEN GOUSE;
                        IF RTOG THEN IOERR (29);
                        END;
                END READREV;
SUBROUTINE ERROR;
        BEGIN     IF EOF THEN
                        BEGIN
                        BCOUNT ← *P(DUP) - 1;
                        RCOUNT ← *P(DUP) - 1;
                        ENDFILE ← TRUE;
                        SETPRESENTSBIT;
                        IF READER THEN P(1,RTN);
                        IF NOT UNLABELED THEN
                             BEGIN
                             ENDREEL := FALSE;
                             IF NOT DISK THEN
                                  BEGIN
                                  READLBL;
                                 STREAM(SENT←0,BC←0,RC←0,WP←0:L←LBLPTR);
                                       BEGIN          % THIS RETRIVES END
                                       DI ← LOC SENT;% OF REEL SENTINAL,
                                       DI ← DI +7;    % BLOCK & REC COUNT
                                       SI ← L ; SI ←SI+39;
                                       DS ← CHR;DS←5 OCT;DS←7 OCT;
                                       DI ← DI+7; DS ← CHR;
                                       END;
                                  IF P=1 THEN WRITEPARITY ← TRUE;
                                  CHECK(RCOUNT) ONERR(16);
                                  CHECK(BCOUNT) ONERR(17);
                                  ENDREEL ← P ; % THIS STORES SENTINAL
```

```
02723300 T  0145:0
02723400 T  0145:1
02723500 T  0146:0
02723600 T  0147:2
02723700 T  0148:0
02723800 T  0149:0
02723900 T  0151:2
02724000 T  0151:2
02724100 T  0151:2
02724200 T  0153:3
02724300 T  0154:1
02724400 T  0155:0
02724500 T  0159:1
02724600 T  0159:1
02724700 T  0159:3
02724800 T  0161:1
02724900 T  0163:2
02725000 T  0165:2
02725100 T  0165:2
02725200 T  0166:3
02725300 T  0167:1
02725400 T  0168:3
02725500 T  0169:3
02725600 T  0170:1
02725700 T  0172:3
02725800 T  0178:2
02725900 T  0179:0
02726000 T  0179:0
02726100 T  0183:0
02726200 T  0185:3
02726300 T  0185:3
02726400 T  0186:0
02726500 T  0186:0
02726600 T  0187:0
02726700 T  0187:2
02726800 T  0189:2
02726900 T  0191:2
02727000 T  0194:0
02727100 T  0195:2
02727200 T  0197:3
02727300 T  0199:0
02727400 T  0199:2
02727500 T  0200:1
02727600 T  0201:0
02727700 T  0201:2
02727800 T  0203:2
02727900 T  0206:0
02728000 T  0206:1
02728100 T  0206:2
02728200 T  0206:3
02728300 T  0207:1
02728310 T  0208:0
02728400 T  0208:2
02728410 T  0208:3
02728500 T  0212:1
02728600 T  0216:3
02728700 T  0221:1
```

```
      IF NOT WRITEPARITY THEN                                          02728710 T  0221:3
      BEGIN                                                            02728720 T  0223:0
      HASHTOT;                                                         02728800 T  0223:2
      IF (T←PGUSE[3].[ 1:23])≠ 0 THEN GOUSE;                           02728900 T  0229:1
      IF (T←PGUSE[3].[24:24])≠ 0 THEN GOUSE;                           02729000 T  0232:2
      END;                                                             02729010 T  0236:1
      END                                                              02729100 T  0237:0
ELSE    STREAM (RECTOT← TOTREC + 1,                                    02729200 T  0237:0
                LABL  ← LBLPTR);                                       02729300 T  0238:3
        BEGIN                                                          02729400 T  0240:0
        SI ← LOC RECTOT ;                                              02729500 T  0240:0
        DI ← DI + 45;                                                  02729600 T  0240:1
        DS ← 7 DEC;                                                    02729700 T  0240:2
        END;                                                           02729800 T  0240:3
IF NOT ENDREEL THEN                                                    02729900 T  0241:0
    IF PGUSE[BS←(DISK AND OPENIO)×9+2]≠0                               02730000 T  0241:2
    THEN BEGIN                                                         02730100 T  0245:3
    IF (T←PGUSE[BS].[ 1:23])≠0 THEN GOUSE;                             02730200 T  0246:3
    IF (T←PGUSE[BS].[24:24])≠0 THEN GOUSE;                             02730300 T  0250:0
    END;                                                               02730400 T  0253:1
IF NOT DISK THEN % END OF REEL USE ROUTINES                            02730500 T  0254:0
IF NOT WRITEPARITY THEN                                                02730510 T  0254:3
    BEGIN                                                              02730600 T  0256:2
    IF (T←FIB [3].[ 1:23])≠0 THEN GOUSE;                               02730700 T  0257:0
    IF (T←FIB [3].[24:24])≠0 THEN GOUSE;                               02730800 T  0261:0
    END     ;                                                          02730900 T  0265:0
IF NOT ENDREEL THEN                                                    02731000 T  0265:0
    BEGIN                                                              02731100 T  0265:2
    IF (T← FIB [2].[ 1:23])≠0 THEN GOUSE;                              02731200 T  0266:0
    IF (T← FIB [2].[24:24])≠0 THEN GOUSE;                              02731300 T  0270:0
    P(1,RTN);                                                          02731400 T  0274:0
    END;                                                               02731500 T  0274:2
    END;                                                               02731600 T  0274:2
IF NONSTD THEN                                                         02731700 T  0274:2
    BEGIN                                                              02731800 T  0275:2
    ENDFILE := FALSE;                                                  02731900 T  0276:0
    CLOSEANDOPEN;                                                      02731950 T  0278:2
    P(1,RTN);                                                          02732000 T  0281:0
    END;                                                               02732100 T  0281:2
NXTREEL;                                                               02732200 T  0281:2
P(DEL,DEL);     %DELETE BRANCH RETURNS                                 02732300 T  0283:0
WRITEPARITY ← FALSE;                                                   02732310 T  0283:2
GO TO START;                                                           02732400 T  0286:0
END;                                                                   02732500 T  0286:2
MAYBEPARITY;                                                           02732600 T  0286:2
END ERROR;                                                             02732700 T  0288:0
SUBROUTINE DISKADDRESS;       %THIS COMPUTES THE DISK ADDRESS READ & WRIT 02732705 T  0288:1
  BEGIN                                                                02732710 T  0289:0
        IF CODE THEN RT ← SEGPERBLK × BCOUNT;                          02732715 T  0289:0
        IF P(RT DIV ROWLGTH,DUP) GEQ NUMBSPC THEN                      02732716 T  0292:0
        BEGIN                                                          02732717 T  0294:2
$ SET OMIT = NOT SHAREDISK                                             02732718 T  0295:0
            P(1,RTN);                                                  02732721 T  0295:0
        END;                                                           02732722 T  0295:2
        IF (T← P + 10) LSS 10 THEN T←10;                               02732723 T  0295:2
        IF (BS ← H[T]) = 0 THEN                                        02732725 T  0298:1
            BEGIN                                                      02732730 T  0299:3
```

```
                    GETSEG;                                                                 02732740 T  0300:1
                  IF INVALIDUSER THEN                                                       02732742 C  0307:0
                  BEGIN                                                                     02732744 C  0308:0
                  MAYBEPARITY;                                                              02732746 C  0308:2
                  END;                                                                      02732748 C  0310:0
                    IF HOWOPEN#0 THEN IF NOT OPENIO THEN IOERR(22);                         02732750 T  0310:0
                    BS ← H[T];                                                              02732760 T  0315:3
                    END;                                                                    02732765 T  0316:3
                STREAM( A ← BS ← BS + RT MOD ROWLGTH,                                       02732770 T  0316:3
                      B←T←BUFTOP,[CF]←(IF CODE THEN 0 ELSE WRITBACK));                      02732775 T  0319:0
                    BEGIN SI←LOC A;  DS←8 DEC; END;                                         02732780 T  0323:2
$ SET OMIT = NOT SHAREDISK                                                                  02732784 T  0324:1
    END DISKADDRESS;                                                                        02732795 T  0324:1
SUBROUTINE  WRIT;                    %     THIS WRITES A RECORD                             02732800 T  0324:2
        BEGIN     IF TECHC THEN                                                             02732900 T  0325:0
                      BEGIN                                                                 02733000 T  0326:2
                  ;   STREAM (A ← TIP, B ← [X1] );                                          02733100 T  0327:0
                          BEGIN                                                             02733200 T  0328:1
                          SI ← A; DI ← DI +4; DS ← 4 CHR;                                   02733300 T  0328:1
                                  DI ← DI +4; DS ← 4 CHR;                                   02733400 T  0329:0
                          END;                                                              02733500 T  0329:2
                      TECHCOFLO ←1&TIP[18:33:15]&NUMWDS[3:33:15];                           02733600 T  0329:3
                      NUMWDS ← -WORDSLEFT +(WORDSLEFT ← BUFFSIZE);                          02733700 T  0332:1
                      DLOC[0]← FLAG(FIB[16] & NUMWDS[8:38:10]);%TR840                       02733750 T  0335:3
                      END                                                                   02733800 T  0338:0
                  ELSE BEGIN                                                                02733900 T  0338:0
                      COUNT ← NUMREC;                                                       02734000 T  0338:2
                      NUMWDS← (WORDSLEFT ← BUFFSIZE) - T ;                                  02734100 T  0340:0
                      IF PUNCH THEN FIB[16],[32:1] ← CHNNL;  %TR 830 I                      02734150 T  0343:0
                      IF DISK  THEN                                                         02734200 T  0346:3
                          BEGIN                                                             02734300 T  0347:2
                          LASTIO ← 0;                                                       02734350 T  0348:0
                          %THIS COMPUTES THE AMT OF DISK USED IN ROWS                       02734400 T  0350:2
                          IF (RCOUNT+1) DIV RECPERBLK×SEGPERBLK DIV                         02734500 T  0350:2
                              ROWLGTH GEQ NUMBSPC THEN                                      02734600 T  0354:0
                          IF (RCOUNT + OPENIO) DIV                                          02734700 T  0356:0
                             RECPERBLK × SEGPERBLK DIV ROWLGTH                              02734800 T  0358:1
                             GEQ NUMBSPC THEN                                               02734810 T  0361:0
                          BEGIN                                                             02734820 T  0362:3
                          IF OPENIO THEN RCOUNT ← *P(DUP) + (SERIAL);                       02734830 T  0363:1
                    COUNT ← 0;                                                              02734835 T  0367:1
                          P(1,RTN)                                                          02734840 T  0369:1
                          END ELSE                                                          02734900 T  0369:3
                          IF SERIAL THEN COUNT←0 ELSE BADKEY←TRUE;                          02735000 T  0369:3
                          DLOC[0] ← FLAG(BUFTOP & RESETREADBIT);                            02735100 T  0376:2
                          P(CODE);                                                          02735200 T  0378:3
                          CODE ← 1;                                                         02735300 T  0379:0
                          DISKADDRESS;                                                      02735400 T  0379:3
                          CODE ← P;                                                         02735450 T  0381:0
$ SET OMIT = NOT SHAREDISK                                                                  02735599 T  0381:2
                          END                                                               02736100 T  0381:2
                      ELSE IF LINEPRINT THEN                                                02736200 T  0381:2
                          BEGIN                                                             02736300 T  0384:3
                          IF NOT SKIPAFT THEN                                               02736400 T  0385:1
                             BEGIN                                                          02736500 T  0385:3
                             SKIPPER;                                                       02736600 T  0386:1
                             LINES ← CHNNL← 0;                                              02736700 T  0387:0
```

```
                        GO TO SETUP;                                                    02736800 T  0388:1
                        END;                                                            02736900 T  0388:3
                    IF (CHNNL ≠0) OR (LINES ≤ 2) THEN                                   02737000 T  0388:3
SETUP:                  DLOC[0]← FLAG(FIB[16]&LINES [27:47:1]                           02737100 T  0390:2
                                      &LINES [28:46:1]                                  02737200 T  0391:3
                                      &CHNNL [29:44:4])                                 02737300 T  0392:3
                    ELSE BEGIN                                                          02737400 T  0393:3
                         DLOC[0] ← FLAG(FIB[16]&@20 [27:42:6]);                         02737500 T  0395:3
                         PREL;                                                          02737600 T  0397:1
                         IF NOT PRESENT THEN REFLECTCHECKER;                            02737650 T  0399:0
                         LINES ← LINES - 2;                                             02737700 T  0402:0
                         SKIPPER;                                                       02737800 T  0403:1
                         GO TO LPRETURN;                                                02737900 T  0404:0
                         END;                                                           02738000 T  0404:2
                    END LINEPRINTER                                                     02738100 T  0404:2
                ELSE DLOC[0] ← FLAG(FIB[16]&NUMWDS[8:38:10]);                           02738200 T  0404:2
                END;                                                                    02738300 T  0407:1
            IF TAPEE THEN IF NOT BINARY THEN ARROW;                                     02738400 T  0407:1
            IF DISK AND BS < 100 THEN TERM(69);                                         02738500 T  0412:1
            PREL;                                                                       02738600 T  0415:3
LPRETURN:   FIB[16].[33:15] ← TIP;                                                      02738700 T  0417:0
            DLOC[0] ← (DISK) INX TIP & MAXR;                                            02738800 T  0419:1
            IF TECHCOFLO THEN                                                           02738900 T  0422:2
                BEGIN                                                                   02739000 T  0422:3
               ;STREAM (I ← [X1],A←NUMWDS←TECHCOFLO.[3:15],                             02739100 T  0423:1
                        B ← TECHCOFLO.[18:15], K ←NUMWDS.[36:6],                        02739200 T  0425:0
                        X ← TIP OR MEM );                                               02739300 T  0425:3
                    BEGIN                                                               02739400 T  0427:3
                        SI ← B;                                                         02739500 T  0427:3
                        K(DS ← 32 WDS;DS← 32 WDS);                                      02739600 T  0428:0
                        DS ← A WDS;                                                     02739700 T  0429:1
                        SI ← I; DI ← X; SI ←SI+4;                                       02739800 T  0429:3
                        DS←4 CHR; SI←SI+4; DS ← 4 CHR;                                  02739900 T  0430:2
                    END;                                                                02740000 T  0431:1
                TECHCOFLO ← 0;                                                          02740100 T  0431:2
                DLOC[0]   ← NUMWDS INX TIP;                                             02740200 T  0432:1
                WORDSLEFT ← WORDSLEFT - NUMWDS;                                         02740300 T  0433:3
                END;                                                                    02740400 T  0435:3
            IF NOT PRESENT THEN                                  %CJC 021               02740500 T  0435:3
            BEGIN                                                                       02740510 T  0437:0
$ SET OMIT = NOT SHAREDISK                                                              02740519 T  0437:2
                REFLECTCHECKER;                                                         02740530 T  0437:2
            END ELSE                                                                    02740540 T  0439:0
            RESETPARITY;                                                                02740550 T  0439:0
        END WRIT;                                                                       02740600 T  0441:2
SUBROUTINE  WRITEADJUST;          % THIS ADJUSTS BLOCK+REC PTRS                          02740700 T  0441:3
        BEGIN                                                                           02740800 T  0442:0
            T := 0;                                                                     02740900 T  0442:0
            P(NUMWDS);                %SAVE OFF NUMWDS                                  02741000 T  0442:3
            BCOUNT := *P(DUP) - 1; %BACK UP BECAUSE WE                                  02741100 T  0443:0
            RCOUNT := *P(DUP) - 1; %WERE READING                                        02741200 T  0445:0
            WRIT;                                                                       02741300 T  0447:0
            BCOUNT := *P(DUP) + 1; %UP GRADE SO IT CAN STILL                            02741400 T  0448:0
            RCOUNT := *P(DUP) + 1; %THINK THAT WERE READING                             02741500 T  0450:0
            NUMWDS:= P;                                                                 02741600 T  0452:0
            WORDSLEFT := *P(DUP)-NUMWDS;%DONT LOSE LAST REC                             02741700 T  0452:2
        END OF WRITEADJUST;                                                             02741800 T  0454:2
```

```
SUBROUTINE  REED;                    % THIS READS A RECORD                                    02741900 T  0454:3
        BEGIN     IF DISK THEN                                                               02742000 T  0455:0
                       BEGIN LASTIO ← 1;                                                     02742100 T  0455:3
                       IF RCOUNT > TOTREC OR BADKEY THEN                                     02742200 T  0458:3
                            BEGIN                                                            02742300 T  0461:1
                            DLOC[0]← TIP &1[27:47:1];                                        02742400 T  0461:3
                            WRITEAFTEREOF ← 3;                                               02742420 T  0463:3
                            IF OPENIO AND SERIAL THEN                                        02742450 T  0466:1
                                 RCOUNT ← *P(DUP) +1;                                        02742460 T  0469:0
                            ERROR;                                                           02742500 T  0471:2
                            END;                                                             02742600 T  0473:0
                       DLOC[0]← FLAG(FIB[16]);                                               02742700 T  0473:0
                       RT ←(BCOUNT+(T←(NUMBUF-1)))×SEGPERBLK;                                02742800 T  0474:1
                            %RT = SEGMENTS READ ,T=BUFFERS                                   02742900 T  0478:3
                       IF (T←(T×RECPERBLK)+RCOUNT) GTR TOTREC OR                             02743000 T  0478:3
                          (T >LSUBU AND BOUNDED) THEN                                        02743100 T  0482:1
                            BEGIN                                                            02743200 T  0484:3
                            IF WRITBACK THEN                                                 02743300 T  0485:1
                                 BEGIN                                                       02743400 T  0486:1
                                 WRITEADJUST;                                                02743500 T  0486:3
                                 GO TO IOUT;                                                 02743600 T  0488:0
                                 END;                                                        02743700 T  0488:2
                            DLOC[0]←TIP&1[27:47:1]&0[2:47:1];                                02743800 T  0488:2
                            ROTATEBUF;  %THIS FLAGS ERROR DESC                               02743900 T  0491:2
                            RT ← -1;  % THIS INHIBITS PRL                                    02744000 T  0494:1
                            END                                                              02744100 T  0495:1
                       ELSE BEGIN ;                                                          02744200 T  0495:1
                       P(CODE);  CODE ← 0;                                                   02744300 T  0495:3
                                 DISKADDRESS;                                                02744400 T  0496:3
                                 CODE ← P;                                                   02744500 T  0498:0
 $ SET OMIT = NOT SHAREDISK                                                                  02744599 T  0498:2
                                 END  END                                                    02744800 T  0498:2
                            ELSE BEGIN                                                       02744900 T  0498:2
                                 IF NUMWDS<1 AND RCOUNT >0 THEN TERM(26);                    02745000 T  0499:0
                                 DLOC [0] ← FLAG (FIB[16]);                                  02745100 T  0502:3
                                 END;                                                        02745200 T  0504:0
                            IF CODE=2 THEN NUMBUF ←2;                                        02745300 T  0504:0
                            PREL;                                                            02745400 T  0507:3
IODONE:                     WORDSLEFT ← IF DISK THEN % DISK HAS NO SHORT BLOCKS               02745500 T  0509:0
                                        IF (BS←TOTREC-RCOUNT+2)≥RECPERBLK THEN               02745600 T  0510:1
                                           BUFFSIZE ELSE (BS×MAXREC) ELSE                    02745700 T  0514:1
                                                MEM [(NOT 0 ) INX TIP];                      02745800 T  0518:1
                            IF NOT PRESENT THEN                                              02745900 T  0520:3
                            BEGIN                                                            02745905 T  0522:0
 $ SET OMIT = NOT SHAREDISK                                                                  02745909 T  0522:2
                                 ERROR;                                                      02745975 T  0522:2
                            END ELSE                                                         02745980 T  0524:0
                            BEGIN RESETPARITY;                                               02745985 T  0524:0
 $ SET OMIT = NOT SHAREDISK                                                                  02745989 T  0526:2
                            END;                                                             02745995 T  0526:2
                            IF RANDOM THEN GO TO RANDOMLBL;                                  02746000 T  0526:2
                            FIB[16],[33:15] ← TIP;                                           02746100 T  0527:2
                            DLOC[0]← (DISK) INX TIP & MAXR;                                  02746200 T  0529:3
 $ SET OMIT = NOT SHAREDISK                                                                  02746249 T  0533:0
IOUT:       END  REED ;                                                                      02746300 T  0533:0
SUBROUTINE SEEK;    % THIS CHECKS FOR PRESENTS OF BLOCKS IF NOT IT READS                     02746400 T  0533:1
          BEGIN                                                                              02746500 T  0534:0
```

```
                    IF ((KEY ← KEY-1)<LSUBL ) OR (KEY >LSUBU AND BOUNDED)          02746600 T  0534:0
                    THEN BADKEY ← TRUE                                             02746700 T  0537:1
                    ELSE BEGIN                                                     02746800 T  0540:1
                         BCOUNT ← (RCOUNT←KEY) DIV NUMREC;                         02746900 T  0542:0
                         IF BCOUNT ≠ COUNT THEN                                    02747100 T  0545:0
                              BEGIN                                                02747200 T  0546:1
REREAD:                                                                           02747250 T  0546:3
                              IF NUMBUF=2 THEN                                    02747300 T  0546:3
                                   BEGIN                                          02747400 T  0548:1
                                   ROTATEBUF;                                     02747500 T  0548:3
                                   IF NOT DONE THEN                               02747600 T  0551:2
 $ SET OMIT = NOT SHAREDISK                                                       02747609 T  0552:3
                                   WAITIO;                                        02747650 T  0552:3
                                   NUMBUF ←1;                                     02747700 T  0554:3
                                   DLOC[0] ← DLOC[1];            %CJC 018         02747800 T  0557:1
                                   END;                                           02747900 T  0558:3
                              IF CODE=2 THEN DLOC[1]← TIP;                        02748000 T  0558:3
                              IF RCOUNT ≤ TOTREC THEN READ                        02748100 T  0561:3
                              ELSE IF BCOUNT =(TOTREC DIV NUMREC) THEN            02748200 T  0564:1
                                   BEGIN     %ABOVE CHECKS FOR LAST BLOCK         02748300 T  0567:2
                                   RCOUNT ← TOTREC;                               02748400 T  0568:0
                                   READ;                                          02748500 T  0569:2
                                   END;                                           02748600 T  0571:0
SEEKRTN:                           RCOUNT ← KEY;                                  02748700 T  0571:0
                                   IF RTOG THEN                                   02748750 T  0572:1
                                        COUNT ← BCOUNT ← *P(DUP) - 1;             02748800 T  0572:2
 $ SET OMIT = SHAREDISK                                                           02748899 T  0576:0
                              END;                                                02748900 T  0576:0
 $ POP OMIT                                                                       02748901 T  0576:0
 $ SET OMIT = NOT SHAREDISK                                                       02748909 T  0576:0
                                   BADKEY ← FALSE;                                02749000 T  0576:0
                                   END;                                           02749100 T  0578:2
                         SEEKEY ← (CODE=2);                                       02749200 T  0578:2
                         IF CODE = 2 THEN P(XIT);                                 02749300 T  0581:2
               END SEEK;                                                          02749400 T  0583:0
%%%%%%%%%%%%START HERE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%                           02749500 T  0583:1
               START :                                                            02749600 T  0583:1
                         FIB ← * (FLOC ← (NOT 2) INX DLOC); %SET UP IDS           02749700 T  0586:3
                           IF FPB[FNAM+3].[42:6]=43 THEN    % DUMMY               02749710 C  0589:1
                              IF CODE=0 THEN GO EOFSETCK ELSE                      02749720 C  0592:0
                                IF P(MKSCW,TOP,XCH,DEL) THEN P(XIT)                02749730 C  0593:1
                                                        ELSE P(0,RTN);             02749740 C  0595:3
                         IOMASK := @2000000000;                                   02749800 T  0596:3
                         IF (UNITYPE←UT)=4 THEN % DISK                            02749900 T  0597:2
                              BEGIN                                               02750000 T  0599:2
                              H ← * [FIB[14]];  % LOAD HEADER                     02750100 T  0600:0
                              IF RCOUNT > LSUBU AND BOUNDED THEN                  02750200 T  0601:1
                              IF CODE THEN                                        02750300 T  0603:3
                              IF (RCOUNT-(OPENIO AND (SERIAL))) > LSUBU THEN      02750305 T  0604:2
                              P(1,RTN) ELSE ELSE BADKEY ← TRUE;                   02750310 T  0609:1
 $ SET OMIT = NOT SHAREDISK                                                       02750314 T  0615:0
                              IF CODE.[1:1] THEN                                  02750325 T  0615:0
                              BEGIN CODE←ABS(CODE);                               02750330 T  0615:3
 $ SET OMIT = NOT SHAREDISK                                                       02750334 T  0617:1
                              END;                                                02750400 T  0617:1
                              END;                                                02750450 T  0617:1
                         IF NOT(ENDPROCESS=0 OR CODE) THEN GO TO EOFSETCK;        02750500 T  0617:1
```

```
                    IF CODE>1 THEN           % OTHER THAN READ OR WRIT
                         BEGIN
                         IF CODE=2 THEN       %  SEEK
                              BEGIN
                              IF HOWOPEN>1 THEN GO TO IMPROPER;
                              IF DISK THEN SEEK;
$ SET OMIT = TIMESHARING
                              TERM(28);   %IMPROPER SEEK
                              END;
                         IF CODE=6 THEN       % WRITE BLOCK
                              IF DISK THEN                        %CUBE XIX I
$ SET OMIT = NOT SHAREDISK
                              IF OPENIO AND                       %CUBE XIX I
                                 FIB[4].[27:3] = 1 THEN           %CUBE XIX I
                                 GO TO RANDOMIO                   %CUBE XIX I
                              ELSE TERM(35)                       %CUBE XIX I
                              ELSE   % THEN IT IS TAPE            %CUBE XIX I
                              IF FIB[5].[41:4] ≠ 0 THEN           %CUBE XIX I
               %     TERM(34) WHEN WRITE BLOCK ON INPUT OR ON     %CUBE XIX I
               %              REVERSED OR ON UNOPENED FILE.       %CUBE XIX I
                              TERM(34) ELSE                       %CUBE XIX I
                                   BEGIN T := WORDSLEFT; %WRITE BLOCK
                                        IF T=BUFFSIZE THEN     %TR 857
                                             P(XIT);%NULL BLOCK%TR 857
                                        WRIT;
                                        RCOUNT ← *P(DUP) - 1;
                                        P(XIT);
                                   END WRITE BLOCK;
                         TERM(25);            % UN RECOGNISED CODE
                         END;
                    IF  (1=CODE) ≠HOWOPEN THEN   % CHECK USE VS HOW OPEN
IMPROPER:                IF HOWOPEN>1 THEN  TERM (31+CODE) ELSE %CLOSED
                         IF NOT OPENIO THEN TERM(PROPER);      %USAGE
                    IF UNITYPE =10 OR UNITYPE =13 THEN GO TO DATACOM;
                    IF SERIAL THEN
                         BEGIN
                         T ← WORDSLEFT -(NUMWDS←NUMWDS);  %COUNT WORDLEFT
                         IF OPENIO THEN GO TO SERIALIO;
                         IF CODE THEN       % CODE=1 ON WRITE
                              BEGIN
                              IF NUMWDS<1 THEN TERM(36);
                              HASH;
                              IF TECHC THEN
                                   BEGIN
                                   IF NUMWDS > MAXREC THEN
                                             T ← WORDSLEFT-(NUMWDS←MAXREC);
                                   IF T> MAXREC THEN DIDDLEREC ELSE WRIT;
                                   END
                              ELSE IF(COUNT ← *P(DUP) - 1)>0 THEN
                                             DIDDLEREC ELSE  WRIT;
                              IF NOT DISK THEN
                                     P(XIT);
                              IF (T←RCOUNT-1) > TOTREC THEN TOTREC ← T;
                              P(0,RTN);
                              END;
                                        %CODE=0 ON READ
                         IF REVERSE THEN READREV ELSE
```

```
02750600 T  0619:3
02750700 T  0620:2
02750800 T  0621:0
02750900 T  0621:3
02751000 T  0622:1
02751100 T  0624:1
02751200 T  0627:0
02751400 T  0627:0
02751500 T  0628:1
02751600 T  0628:1
02751610 T  0629:0
02751614 T  0630:1
02751620 T  0630:1
02751630 T  0631:3
02751640 T  0633:2
02751650 T  0633:2
02751660 T  0635:2
02751670 T  0635:2
02751680 T  0637:2
02751690 T  0637:2
02751700 T  0637:2
02751800 T  0639:1
02751830 T  0640:3
02751850 T  0642:1
02751900 T  0643:0
02752000 T  0644:0
02752100 T  0646:0
02752200 T  0646:1
02752300 T  0646:1
02752400 T  0647:2
02752500 T  0647:2
02752600 T  0649:2
02752700 T  0653:3
02752800 T  0659:2
02752900 T  0662:0
02753000 T  0663:2
02753100 T  0664:0
02753200 T  0666:0
02753300 T  0667:3
02753400 T  0668:0
02753450 T  0668:2
02753500 T  0671:0
02753600 T  0675:2
02753700 T  0677:0
02753800 T  0677:2
02753900 T  0679:0
02754000 T  0681:2
02754100 T  0686:1
02754200 T  0687:0
02754300 T  0690:0
02754400 T  0694:0
02754500 T  0694:3
02754600 T  0695:2
02754700 T  0699:2
02754800 T  0700:0
02754900 T  0700:0
02755000 T  0700:0
```

```
                        IF TECH = 0 THEN REED ELSE              %TR 899
                        IF T < 1 OR BADKEY THEN REED ELSE
                        IF NUMWDS<1 THEN GO TO EOFSETCK ELSE
                        DIDDLEREC;
                        HASH;
                        P (0,RTN)
                        END;
%RANDOM AND RANDOM I=0 HERE ON
RANDOMIO:               FIB[13],[44:1] ← 0;                          %CUBE XIX I
                        IF SEEKEY THEN KEY ←RCOUNT ELSE SEEK;
                        IF BADKEY OR (KEY > TOTREC AND CODE=0) THEN
                             BEGIN
                             SANDBKEY ← 0;        %RESET SEEKEY&BADKEY
                             IF WRITBACK THEN  %RESTORE THE ADDRESS
                             RCOUNT ← ((BCOUNT ← COUNT) × NUMREC);
 $ SET OMIT = NOT SHAREDISK
                             P(1,RTN);
                             END;
                        IF SEEKEY THEN
                        BEGIN  IF NUMBUF = 2 THEN
                        BEGIN       ROTATEBUF;
                                    IF NOT DONE THEN
 $ SET OMIT = NOT SHAREDISK
                                    WAITIO;
                                    NUMBUF ←1;
                        END;
                                SEEKEY←FALSE;
 $ SET OMIT = NOT SHAREDISK
                                RTOG←RANDOM←TRUE;
                                GO TO IODONE;
                        END;
RANDOMLBL:              IF INVALIDUSER THEN
                        BEGIN
                             MAYBEPARITY;
                        END;
                        T ←RANDOM ← FALSE;
                        IF CODE = 6 THEN IF WRITBACK THEN            %CUBE XIX I
                   BEGIN P(BCOUNT);                                  %CUBE XIX I
                        WRIT;                                        %CUBE XIX I
                        P([BCOUNT],←);                               %CUBE XIX I
                        RCOUNT ← KEY;                                %CUBE XIX I
                        P(XIT);                                      %CUBE XIX I
                   END  ELSE  P(XIT);                                %CUBE XIX I
                        IF WRITBACK AND CODE THEN
                             IF COUNT ≠ BCOUNT THEN
                                  BEGIN
                                  P(NUMWDS,BCOUNT); %LEAVE BLOCK COUNT ON STK
                                  RCOUNT← (BCOUNT←COUNT) × NUMREC;
                                  WRIT;
                                  P([BCOUNT],←);%PICK UP BLOCK COUNT
                                  NUMWDS ← P;
                                  RCOUNT ← KEY;
                                  END;
                        MOVREC;
                        IF CODE THEN   % IF RANDOM WRITE THEN WRITE
                             BEGIN
                             IF LASTIO THEN FIB[13],[44:1] ← 1;
```

```
02755050 T  0703:0
02755100 T  0707:0
02755200 T  0711:0
02755300 T  0712:1
02755400 T  0714:0
02755500 T  0718:2
02755600 T  0719:0
02755700 T  0719:0
02755710 T  0719:0
02755800 T  0721:2
02755900 T  0726:0
02756000 T  0729:1
02756100 T  0729:3
02756110 T  0732:1
02756120 T  0733:1
02756129 T  0737:0
02756200 T  0737:0
02756300 T  0737:2
02756400 T  0737:2
02756500 T  0738:2
02756600 T  0740:2
02756700 T  0743:3
02756709 T  0745:0
02756795 T  0745:0
02756800 T  0747:0
02756900 T  0749:2
02756910 T  0749:2
02756919 T  0752:0
02757000 T  0752:0
02757200 T  0753:1
02757300 T  0753:3
02757400 T  0753:3
02757410 T  0754:3
02757430 T  0755:1
02757440 T  0756:0
02757500 T  0756:0
02757510 T  0757:1
02757520 T  0759:2
02757530 T  0760:2
02757540 T  0762:0
02757550 T  0762:3
02757560 T  0764:0
02757570 T  0764:1
02757600 T  0765:0
02757700 T  0766:2
02757800 T  0768:1
02757900 T  0768:3
02758000 T  0769:2
02758100 T  0772:3
02758200 T  0774:0
02758300 T  0774:3
02758400 T  0775:1
02758500 T  0776:2
02758600 T  0776:2
02758700 T  0778:0
02758800 T  0778:1
02758850 T  0778:3
```

```
                        IF NOT (COUNT = BCOUNT AND WRITBACK) THEN                02758900 T  0782:3
                              BEGIN                                              02758960 T  0785:2
                              DISKADDRESS;%COMPUTE  NEW ADDRESS                  02759000 T  0786:0
                              COUNT ← BCOUNT;                                    02759010 T  0787:0
                              END;                                               02759050 T  0788:2
                        IF KEY > TOTREC THEN TOTREC ← KEY;                       02759100 T  0788:2
$ SET OMIT = NOT SHAREDISK                                                       02759104 T  0791:1
                        WRITBACK ← TRUE;                                         02759165 T  0791:1
                        END;                                                     02759200 T  0793:3
                     P(0,RTN);                                                   02759300 T  0793:3
%END OF MAIN LOGIC NEXT IS SPECIAL ROUTINES                                      02759400 T  0794:1
SERIALIO:     %THIS HANDLES SERIAL I-O                                           02759500 T  0794:1
          IF BADKEY THEN REED;                                    %CJC 022       02759550 T  0794:1
          IF NOT (LASTDONE AND CODE) THEN                                        02759600 T  0797:0
          IF (COUNT ← *P(DUP) -1) > 0 THEN                                       02759620 T  0798:3
          BEGIN WORDSLEFT ← T;                                                   02759640 T  0801:3
               GO TO DIDDLE;                                                     02759660 T  0803:2
          END  ELSE                                                              02759680 T  0804:0
          BEGIN IF BOUNDED THEN                                                  02759700 T  0804:0
               IF RCOUNT = LSUBL THEN                                            02759720 T  0805:2
               BEGIN COUNT ← NUMREC - (BS ← RCOUNT MOD NUMREC);                  02759740 T  0807:1
                    BCOUNT ← *P(DUP) - (BS ≠ 0);                                 02759760 T  0811:1
               END  ELSE                                                         02759780 T  0813:3
                    COUNT ← NUMREC   ELSE                                        02759800 T  0813:3
          COUNT ← NUMREC;                                                        02759900 T  0815:3
          IF CODE THEN   % TWO WRITES IN A ROW                                   02759920 T  0817:3
          BEGIN P(TIP INX 0); % LEAVES POINTER FOR MOVEREC                       02759940 T  0818:0
               IF RCOUNT GEQ TOTREC THEN                                         02759960 T  0819:2
               BEGIN IF TECH = 0 AND WRITEAFTEREOF = 2 THEN                      02759980 T  0820:3
                    BEGIN WRITEAFTEREOF ← 1;                                     02760000 T  0824:2
                         RCOUNT ← *P(DUP) +2;                                    02760020 T  0827:2
                         BCOUNT ← *P(DUP) +2;                                    02760040 T  0829:2
                    END;                                                         02760100 T  0831:2
                    IF RCOUNT = 0 THEN                                           02760200 T  0831:2
                    BEGIN P(DEL);                                                02760300 T  0832:2
                         RCOUNT ← *P(DUP) +1;                                    02760400 T  0833:1
                         BCOUNT ← *P(DUP) +1;                                    02760500 T  0835:1
                         WORDSLEFT ← BUFFSIZE - NUMWDS;                          02760600 T  0837:1
                         GO TO DIDDLEWRT;                                        02760700 T  0839:3
                    END  ELSE                                                    02760800 T  0840:1
                    WRITEADJUST;                                                 02760900 T  0840:1
               END  ELSE REED;                                                   02761000 T  0842:0
               CODE ← 4;  MOVREC;   CODE ← 1;                                    02761100 T  0844:0
               END  ELSE REED;                                                   02761200 T  0846:3
DIDDLEWRT:     IF CODE THEN WRITBACK ← TRUE;                                     02761300 T  0848:0
               IF WRITEAFTEREOF = 3 THEN WRITEAFTEREOF ← 2;                      02761400 T  0851:1
               RCOUNT ← *P(DUP) -1;                                              02761500 T  0855:3
DIDDLE:        MOVREC;                                                           02761600 T  0857:3
               RCOUNT ← *P(DUP) +1;                                              02761700 T  0859:0
          END  ELSE GO TO DIDDLEWRT;                                             02761800 T  0861:0
          LASTDONE ← NOT CODE;                                                   02761900 T  0861:0
          IF (T ← RCOUNT -1) > TOTREC THEN                                       02762000 T  0863:3
          IF CODE THEN TOTREC ← T  % UPDATE EOF POINTER                          02762100 T  0866:0
          ELSE GO TO EOFSETCK;     % PASSED EOF ON READ                         02762200 T  0867:3
          P(0,RTN);                                                              02762300 T  0868:2
%END SERIAL I-O                                                                  02762400 T  0869:0
 DATACOM:     % ALL DATA COMM GOES THRU HERE                                     02762500 T  0869:0
```

```
 $ SET OMIT = NOT(TIMESHARING)                                          02762600 T  0869:0
                IF NOT CODE THEN REMOTEREAD; REMOTEWRIT;                02762700 T  0869:0
 $ POP OMIT                                                             02762750 T  0876:3
 $ SET OMIT = TIMESHARING                                               02762800 T  0876:3
EOFSETCK:       IF ENDFILE THEN TERM (15);                              02765800 T  0876:3
                ENDFILE ← TRUE ;                                        02765900 T  0879:3
                P(1,RTN);                                               02766000 T  0882:1
% END OF EOF SET CHECK                                                  02766100 T  0882:3
     END OF COBOL I O INTRINSICS;                                       02766200 T  0882:3
                                                                                SIZE= 0883 WORDS
```

```
PROCEDURE FBINBACKBLOCK(FILX,DKADDR,FI,FMT,LISX,EDITCODE,EOFL,PARL) ;   02767050 T  0000:0
                                                                        START OF REL SEGMENT; DISK ADDRESS = 00295
VALUE DKADDR,FI,LISX,EDITCODE,EOFL,PARL ;   % INT # @160.               02767100 T  0000:0
REAL DKADDR,FI,LISX,EDITCODE,EOFL,PARL;  ARRAY FMT[*];  NAME FILX ;     02767150 T  0000:0
    BEGIN                                                               02767200 T  0000:0
                                                                        02767250 T  0000:0
    INTEGER BSIZE, LSTRN=19 ;                                           02767300 T  0000:0
                                                                        02767350 T  0000:0
    REAL LISTYPE=20, ARRAYSTUFF=18, ALGOLWRITE=12, ALGOLREAD=13, T6, T7,  02767400 T  0000:0
         SELECT=14, FORTERR=24, ARY, TYPE, DBLPREC=20, INDX=EOFL, B6700,  02767450 T  0000:0
         OUT, FLG, IOINT, T1, T2, T3, T4, T5, TWDT, WH1=18, WH2=17,     02767500 T  0000:0
         SIZE=PARL, INTINT=5, PRNTR, CKPBI, FMTWRD=CKPBI, FLB=FILX ;    02767550 T  0000:0
                                                                        02767600 T  0000:0
    NAME LISTADDR, ADDR ;                                               02767650 T  0000:0
                                                                        02767700 T  0000:0
    ARRAY AR1=LISTADDR[*], FIB[*], IOBUFF[*], TPAR=23[*], FPB=3[*] ;    02767750 T  0000:0
                                                                        02767800 T  0000:0
    LABEL ALIST, BLKDTA, SEVENS, ENDALL, AWAY, SPCL1, SPCL2, DSZ,       02767850 T  0000:0
          CMPXL, DUBEL, LOGCL, STRNL, INTREL, BDERR, ENDIT, ERROR,      02767900 T  0000:0
          BO, BI, BO1, BI1, BDERR1, BDERR2, BI2, MAX, LOOP, STRNL1,     02767950 T  0000:0
          PRINTER, OUTL, BKSPC, BO2, BO3, BO4, BIO, BI3, BI4, BI5, ENDG,  02767975 T  0000:0
          B6 ;                                                          02767980 T  0000:0
                                                                        02768000 T  0000:0
    SWITCH TYPL←INTREL, STRNL, INTREL, LOGCL, DUBEL, CMPXL ;            02768050 T  0000:0
                                                                        02768100 T  0000:0
    DEFINE DONE = LSTRN=(-1) #,                                         02768150 T  0000:0
           NOTDONE = LSTRN≠(-1) #,                                      02768200 T  0000:0
           KIND = FIB[4].[8:4] #,                                       02768250 T  0000:0
           TAPEF = 2 #,                                                 02768300 T  0000:0
           REMOTEF = 13 #,                                              02768325 T  0000:0
           DATACOMF = 10 #,                                             02768330 T  0000:0
           INTEGR = 1 #,                                                02768350 T  0000:0
           STRING = 2 #,                                                02768400 T  0000:0
           REEL = 3 #,                                                  02768450 T  0000:0
           LOGICAL = 4 #,                                               02768500 T  0000:0
           DBLPRECSN = 5 #,                                             02768550 T  0000:0
           COMPLEX = 6 #,                                               02768600 T  0000:0
           TYPEF = [44:4] #,                                            02768650 T  0000:0
           INDXF = [18:15] #,                                           02768700 T  0000:0
           SIZEF = [33:15] # ;                                          02768750 T  0000:0
                                                                        02768800 T  0000:0
    SUBROUTINE BLANKIT ;                                                02768805 T  0000:0
        BEGIN                                                           02768810 T  0001:0
```

holds internal type code
see MCP Ref Manual page 122

```
                STREAM(C←P(XCH),A←BSIZE-1,B←P(DUP),[36:6]:D←IOBUFF) ;                    02768815 T  0001:0
                     BEGIN                                                               02768820 T  0004:0
                     SI←LOC A; 8(SI←SI-1; DS←CHR); SI←D; DS←A WDS ;                      02768825 T  0004:0
                     B(DS←32WDS; DS←32WDS) ;                                            02768830 T  0006:0
                     END ;                                                              02768835 T  0007:1
                P(DEL,DEL,DEL) ;                                                        02768840 T  0007:2
                END OF BLANKIT ;                                                        02768845 T  0008:1
                                                                                        02768846 T  0008:2
           SUBROUTINE CKPB ;                                                            02768850 T  0008:2
                BEGIN                                                                   02768900 T  0009:0
                P(MKS,FLG); IF OUT THEN P(DKADDR,0,(-1)) ELSE P(CKPBI) ;                02768950 T  0009:0
                IF (BSIZE←P(FILX,IOINT))<0 THEN GO ENDIT ;                              02769000 T  0012:0
                END OF CKPB ;                                                           02769050 T  0014:0
                                                                                        02769100 T  0014:1
           SUBROUTINE IO ;                                                              02769700 T  0014:1
                BEGIN P(0) ;                                                            02769750 T  0015:0
      ENDALL:   P(MKS,FLG,DKADDR); IF OUT THEN P(0,BSIZE); P(FILX,IOINT) ;              02769800 T  0015:1
                IF P THEN                                                               02769850 T  0017:3
      ENDIT:         BEGIN FILX[NOT 3]←FILX[NOT 4]←0; P(XIT) END ;                      02769900 T  0017:3
                CKPB ;                                                                  02769950 T  0021:3
                END OF IO ;                                                             02770000 T  0023:0
                                                                                        02770050 T  0023:1
           REAL SUBROUTINE NXTITM ;                                                     02770100 T  0023:1
                BEGIN                                                                   02770150 T  0024:0
                P(IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX AND 255,CDC)                  02770200 T  0024:0
                  ELSE [AR1[INDX]]) ;                                                   02770250 T  0027:0
                INDX←INDX+1;  NXTITM←P ;                                                02770300 T  0028:0
                END OF NXTITM ;                                                         02770350 T  0029:2
                                                                                        02770400 T  0029:3
           SUBROUTINE GETNEXTLISTITEM ;                                                 02770450 T  0029:3
                BEGIN                                                                   02770500 T  0030:0
                IF ARY THEN                                                             02770550 T  0030:0
                     BEGIN                                                              02770600 T  0030:1
      ALIST:         P(NXTITM) ;                                                       02770650 T  0030:3
                     IF DBLPREC THEN IF OUT THEN WH2←*NXTITM ELSE INDX←INDX+1 ;         02770700 T  0032:0
                     ARY←SIZE>INDX ;                                                    02770750 T  0036:3
                     END                                                                02770800 T  0038:3
                ELSE IF TYPE=COMPLEX THEN                                               02770850 T  0038:3
                          BEGIN TYPE←-COMPLEX; P([LISTADDR[1]]) END                     02770900 T  0040:0
                     ELSE BEGIN                                                         02770950 T  0042:1
                          P(ARRAYSTUFF←LISTYPE←0); LISTADDR←[LISX] ;                     02771000 T  0042:3
                          DBLPREC←(TYPE←LISTYPE.TYPEF)=DBLPRECSN ;                      02771050 T  0044:3
                          IF ARY←ARRAYSTUFF≠0 THEN                                      02771100 T  0047:0
                               BEGIN                                                    02771150 T  0048:1
                               IF TYPE=COMPLEX THEN TYPE←-COMPLEX ;                     02771200 T  0048:3
                               SIZE←(INDX←ARRAYSTUFF.INDXF)+ARRAYSTUFF.SIZEF ;          02771250 T  0051:0
                               P(LISTADDR←MEM[LISTADDR,[18:15]]) ;                      02771300 T  0053:3
                               TWDT←NOT P(LOD,TOP); P(DEL) ;                            02771350 T  0055:3
                               IF EDITCODE=2 THEN GO ENDG ELSE GO ALIST ;               02771400 T  0057:1
                               END ;                                                    02771500 T  0059:0
                          P(DEL,[LISTADDR[0]]) ;                                        02771550 T  0059:0
                          IF DBLPREC THEN IF OUT THEN WH2←LISTADDR[1] ;                 02771600 T  0059:2
                          END ;                                                         02771650 T  0062:2
                IF OUT THEN WH1←*P ELSE ADDR←P ;                                        02771700 T  0062:2
      ENDG:     END OF GETNEXTLISTITEM ;                                              02771750 T  0065:0
                                                                                        02771800 T  0065:1
           SUBROUTINE GETANDCHECK ;                                                     02772050 T  0065:1
```

I/O

probably block size

name param to this procedure

local variable

```
            BEGIN                                                                  02772100 T  0066:0
            GETNEXTLISTITEM;  T1←T1-1 ;                                            02772150 T  0066:0
            IF DONE THEN                                                           02772200 T  0068:1
BDERR1:          BEGIN P(1) ;                                                      02772250 T  0069:1
BDERR:           T1←P;  P(MKS,T1,TYPE,T2,FLG,BSIZE,(-2),FORTERR) ;                  02772425 T  0070:0
                 END ;                                                             02772950 T  0072:3
            FLG←FLG+1 ;                                                            02772975 T  0072:3
            END OF GETANDCHECK ;                                                   02773000 T  0074:0
                                                                                   02773050 T  0074:1
%************************:: CODE STARTS HERE ::*************************%         02773100 T  0074:1
                                                                                   02773150 T  0074:1
    LSTRN←CKPBI←1 ;                                                                02773200 T  0074:1
    IF EDITCODE=6 THEN                                                             02773250 T  0081:2
        BEGIN % BLOCKDATA.                                                         02773300 T  0082:1
BLKDTA:  GETNEXTLISTITEM;  P((FMTWRD←FMT[FI←P+FI])=0) ;                             02773400 T  0082:3
         T2←FMTWRD.[18:15];  BSIZE←(FMTWRD≠0)+BSIZE ;                              02773425 T  0086:1
         IF DONE THEN BEGIN IF NOT P THEN GO BDERR1; P(XIT) END ;                  02773450 T  0089:1
         FLG←FLG+1 ;                                                               02773475 T  0091:2
         IF P THEN BEGIN P(2); GO BDERR END;  T1←FMTWRD.[33:15]-1 ;                 02773500 T  0092:3
         T3←FMT[FI←FI+1];  T4←FMT[FI+1] ;                                          02773550 T  0095:3
         GO TYPL[T2-1] ;                                                           02773600 T  0099:1
CMPXL:   IF ABS(TYPE)≠COMPLEX THEN GO BDERR2;  ADDR[0]←T3 ;                        02773650 T  0103:3
         GETNEXTLISTITEM;  ADDR[0]←T4;  IF T1≤0 THEN GO SPCL1 ;                    02773700 T  0106:0
         GETANDCHECK ;                                                             02773725 T  0109:0
         GO CMPXL ;                                                                02773750 T  0110:0
DUBEL:   IF NOT DBLPREC THEN GO BDERR2;  ADDR[0]←T3;  ADDR[1]←T4 ;                 02773800 T  0110:2
         IF T1 LEQ 0 THEN                                                          02773850 T  0113:2
SPCL1:        BEGIN P(2); GO BLKDTA END ;                                          02773900 T  0114:1
         GETANDCHECK ;                                                             02773950 T  0115:2
         GO DUBEL ;                                                                02774000 T  0117:0
LOGCL:   IF TYPE≠LOGICAL THEN                                                      02774050 T  0117:2
BDERR2:       BEGIN P(3); GO BDERR END ;                                           02774100 T  0118:1
         ADDR[0]←T3;  IF T1 LEQ 0 THEN GO SPCL2;  GETANDCHECK ;                    02774150 T  0119:2
         GO LOGCL ;                                                                02774200 T  0123:0
STRNL:   T4←FI;  T3←T1+1;  FMTWRD←FMTWRD.[3:15] ;                                  02774220 T  0123:2
STRNL1:  IF ABS(TYPE)=COMPLEX OR DBLPREC THEN GO BDERR2 ;                          02774225 T  0126:3
         ADDR[0]←FMT[FI] ;                                                         02774230 T  0129:0
         IF T1>0 THEN FI←FI+1                                                      02774250 T  0130:0
         ELSE BEGIN                                                                02774275 T  0131:2
              IF (FMTWRD←FMTWRD-1)≤0 THEN GO SPCL2;  FI←T4;  T1←T3 ;                02774300 T  0133:0
              END ;                                                                02774325 T  0136:3
         GETANDCHECK ;                                                             02774350 T  0136:3
         GO STRNL1 ;                                                               02774375 T  0138:0
INTREL:  IF ABS(TYPE)=COMPLEX THEN GO BDERR2;  P(T3,[ADDR[0]]) ;                   02774400 T  0138:2
         IF TYPE=INTEGR OR TYPE=LOGICAL THEN                                       02774410 T  0140:2
              BEGIN                                                                02774420 T  0142:1
              IF T3>P(MAX) THEN BEGIN P(4); GO BDERR END ;                         02774430 T  0142:3
              P(ISD) ;                                                             02774440 T  0144:3
              END                                                                  02774450 T  0145:0
         ELSE P(←) ;                                                               02774460 T  0145:0
         IF DBLPREC THEN ADDR[1]←0 ;                                               02774470 T  0145:3
         IF T1 LEQ 0 THEN                                                          02774480 T  0148:0
SPCL2:        BEGIN P(1); GO BLKDTA END ;                                          02774500 T  0148:3
         GETANDCHECK ;                                                             02774550 T  0150:0
         GO INTREL ;                                                               02774600 T  0151:0
        END OF BLOCKDATA ;                                                         02774650 T  0151:2
    FIB←FILX[NOT 2];  FILX[NOT 3]←PARL;  FILX[NOT 4]←EOFL ;                        02774700 T  0151:2
```

```
        P(FIB[5]) ;                                                                                  02774750 T  0156:3
        IF FI<0 THEN GO OUTL;  P(P,[43:2]≠T1←(EDITCODE=5)+2,*P(,ALGOLREAD));                         02774800 T  0157:1
        FLG←DKADDR;  GO DSZ ;                                                                        02774850 T  0161:2
MAX:::  @7777777777777 ;                                                                             02774900 T  0162:3
OUTL:                                                                                                02774950 T  0164:0
        OUT←1;  P(P,[43:1],*P(,ALGOLWRITE)) ;                                                        02775000 T  0164:0
        IF FLG←DKADDR<0 THEN                                                                         02775050 T  0165:3
DSZ:        DKADDR←0 ;                                                                               02775100 T  0167:0
        IOINT←P;  IF P THEN P(MKS,0,T1,FILX,1,SELECT) ;                                              02775250 T  0168:1
        IF EDITCODE=5 THEN                                                                           02775300 T  0170:3
            BEGIN % BACKSPACE.                                                                       02775350 T  0171:2
            IF FIB[5],[41:2]≠0 THEN GO ENDIT;  CKPBI←3;  CKPB;  IO ;                                 02775400 T  0172:0
            IF NOT (FIB[FLG←0]≠1 AND KIND=TAPEF) THEN GO ENDIT ;                                     02775450 T  0177:0
BKSPC:      IF (*(*[FILX])),[3:15]≠P(SEVENS) THEN BEGIN IO; GO BKSPC END ;                           02775500 T  0180:3
            IF (*(*[FILX])),[18:15]≠P(SEVENS) THEN GO AWAY;  GO ENDIT ;                              02775550 T  0184:2
            END ;                                                                                    02775650 T  0187:2
        T2←(FIB[5] AND 96)≠0;  CKPB;  T4←(T1←KIND)=TAPEF;  CKPBI←3 ;                                 02775675 T  0187:2
        IF PRNTR←(T1=1 OR T1=7 OR T1=12) AND FPB[FIB[4],[13:11]+3],[43:5]<20                          02775700 T  0194:1
        THEN BEGIN                                                                                   02775725 T  0199:1
            IF BSIZE>17 THEN BSIZE←17 ;                                                              02775727 T  0201:0
            IF T2 THEN BEGIN IOBUFF←TPAR; P(" "); BLANKIT END ;                                      02775730 T  0203:0
            END                                                                                      02775735 T  0206:0
        ELSE IF T2 AND T4 THEN FIB[8],[3:15]←0 ;                                                     02775737 T  0206:0
        IF FIB[0]=0 THEN FIB[0]←2;  T5←T1=REMOTEF OR T1=DATACOMF ;                                   02775740 T  0210:1
        IF FIB[0]≠2 AND T4 THEN                                                                      02775750 T  0215:1
            BEGIN T1←4 ;                                                                             02775800 T  0216:3
ERROR:      P(MKS,FIB[6],FILX,[33:15],T1,FORTERR) ;                                                  02775850 T  0218:0
            END ;                                                                                    02775900 T  0220:0
        IF T4 AND NOT FIB[13],[24:1] THEN P(MKS,(-1),FORTERR) ;                                      02775925 T  0220:0
        T3←P(SEVENS) ;                                                                               02775950 T  0223:1
        IF EDITCODE=0 THEN                                                                           02776000 T  0224:0
            BEGIN % NO FORMAT, NO LIST.                                                              02776050 T  0224:3
            IOBUFF←*FILX ;                                                                           02776100 T  0225:1
            IF OUT THEN                                                                              02776150 T  0226:1
                BEGIN                                                                                02776200 T  0226:2
                IF PRNTR THEN                                                                        02776205 T  0227:0
                    BEGIN                                                                            02776210 T  0227:1
PRINTER:            IF NOT T2 THEN FIB[17]←*P(DUP)+BSIZE ;                                           02776220 T  0227:3
                    P(MKS,1,0,T2,BSIZE,FILX,ALGOLWRITE);  CKPB ;                                     02776225 T  0230:3
                    FIB[17]←*P(DUP)-BSIZE ;                                                          02776230 T  0234:0
                    STREAM(TPAR,BSIZE,S←*FILX) ;                                                     02776235 T  0236:0
                        BEGIN                                                                        02776240 T  0237:3
                        SI←TPAR; DS←BSIZE WDS; DI←TPAR; 18(DS←8LIT" ") ;                             02776245 T  0237:3
                        END ;                                                                        02776250 T  0240:2
                    GO ENDIT ;                                                                       02776255 T  0240:3
                    END ;                                                                            02776280 T  0241:1
                IF T5 THEN P(" ") ELSE P("0");  BLANKIT ;                                            02776320 T  0241:1
                IF T4 THEN IOBUFF[0]←(NOT 0)&(BSIZE-1)[33:33:15] ;                                   02776550 T  0244:0
                END                                                                                  02776600 T  0247:1
            ELSE IF T4 THEN GO BIO ;                                                                 02776650 T  0247:1
AWAY:       P(1);  GO ENDALL ;                                                                       02776700 T  0248:3
            END ;                                                                                    02776750 T  0249:2
        IF T4 THEN IF (*FILX),[8:10]<3 THEN P(MKS,(-4),FORTERR) ;                                    02776775 T  0249:2
        GETNEXTLISTITEM;  IF NOT OUT THEN GO BIO ;                                                   02776800 T  0253:1
BO: T1←T4;  IOBUFF←IF PRNTR THEN TPAR ELSE *FILX ;                                                   02776850 T  0254:3
        IF T5 THEN BEGIN P(" "); BLANKIT END ;                                                       02776875 T  0258:1
BO1:IF ARY THEN                                                                                      02776880 T  0260:0
```

this is FIB[5].[43:1] = 1 for COBOL input

≠0 if file is closed COBOL only

FIB[0] used for COBOL only

FIB[13].[24:1] = 0 means alpha = 1 means binary

obscure!

BOn output
BIn input
BIOn in/out

```
B02:      BEGIN WH1←1 ;                                                                    02776885 T  0260:1
          IF P((BSIZE-T1) AND NOT DBLPREC,DUP)>SIZE-INDX THEN                              02776890 T  0261:2
             P(DEL,SIZE-INDX) ;                                                            02776895 T  0264:1
          IF TWDT THEN                                                                     02776900 T  0265:3
               BEGIN                                                                       02776905 T  0266:0
               IF P(DUP)>(WH2←256-INDX.[40:8]) THEN                                        02776910 T  0266:2
                    BEGIN P(DEL,WH2); WH1←0 END ;                                          02776915 T  0268:3
               P(*[AR1[INDX.[33:7]]],INDX.[40:8],CDC) ;                                    02776920 T  0270:2
               END                                                                         02776925 T  0272:3
          ELSE P([AR1[INDX]]) ;                                                            02776930 T  0272:3
          WH2←P(XCH) ;                                                                     02776935 T  0273:3
          STREAM(S←P:WH2,N←P(DUP).[38:4],D←[IOBUFF[T1]]) ;                                 02776940 T  0274:2
               BEGIN SI←S; DS←WH2 WDS; N(DS←32WDS; DS←32WDS) END ;                         02776945 T  0276:3
          P(DEL); T1←T1+WH2 ;                                                              02776950 T  0279:0
          IF ARY←(INDX←INDX+WH2)<SIZE THEN IF WH1 THEN GO B04 ELSE GO B02                  02776955 T  0280:2
          ELSE GO B03 ;                                                                    02776957 T  0283:2
          END ;                                                                            02776960 T  0284:2
    IOBUFF[T1]←WH1; IF DBLPREC THEN IOBUFF[T1←T1+1]←WH2; T1←T1+1 ;                         02776965 T  0284:2
B03:GETNEXTLISTITEM ;                                                                      02776970 T  0290:0
    IF NOT (DONE OR T1+DBLPREC≥BSIZE) THEN GO B01 ;                                        02777000 T  0291:0
B04:IF PRNTR THEN GO PRINTER; IF NOT T4 THEN GO AWAY ;                                     02777050 T  0294:0
    P((T1-1)&T3[3:33:15]) ;                                                                02777100 T  0295:3
    IF DONE THEN BEGIN P(SEVENS,CFX,[IOBUFF[0]],←); GO AWAY END ;                          02777150 T  0297:2
    P([IOBUFF[T3←0]],←) ;                                                                  02777200 T  0300:3
    IO; GO B0 ;                                                                            02777250 T  0302:0
SEVENS:::@77777                                                                            02777260 T  0303:2
BI0:IF T4 THEN                                                                             02777265 T  0305:0
          BEGIN T7←BSIZE ;                                                                 02777267 T  0305:1
          IF T2 THEN FIB[8].[3:1]←B6700←(*(*[FILX])).[1:15]=P(B6)                          02777270 T  0306:2
          ELSE IF B6700←FIB[8].[3:1] THEN T6←FIB[8].[4:14] ;                               02777290 T  0309:3
          IF EDITCODE=0 THEN                                                               02777292 T  0315:3
               BEGIN BSIZE←(IOBUFF[T6] AND T3)+1; GO BI5 END ;                             02777293 T  0316:2
          END ;                                                                            02777295 T  0319:2
BI: IOBUFF←*FILX ;                                                                         02777300 T  0319:2
    IF T4 THEN BEGIN BSIZE←(IOBUFF[T6] AND T3)+1; P(T6+1) END ELSE P(0);                   02777302 T  0320:2
    T1←P ;                                                                                 02777303 T  0324:3
BI1:IF ARY THEN                                                                            02777305 T  0325:1
BI2:      BEGIN WH1←1 ;                                                                    02777310 T  0325:2
          IF P((BSIZE+T6-T1) AND NOT DBLPREC,DUP)>SIZE-INDX THEN                           02777315 T  0326:3
             P(DEL,SIZE-INDX) ;                                                            02777320 T  0330:0
          IF TWDT THEN                                                                     02777325 T  0331:2
               BEGIN                                                                       02777330 T  0331:3
               IF P(DUP)>(WH2←256-INDX.[40:8]) THEN                                        02777335 T  0332:1
                    BEGIN P(DEL,WH2); WH1←0 END ;                                          02777340 T  0334:2
               P(*[AR1[INDX.[33:7]]],INDX.[40:8],CDC) ;                                    02777345 T  0336:1
               END                                                                         02777350 T  0338:2
          ELSE P([AR1[INDX]]) ;                                                            02777355 T  0338:2
          IF (WH2←P(XCH))+(TYPE←P(IOBUFF INX T1))>HUNT(TYPE) THEN P(FLB);                  02777360 T  0339:2
          STREAM(S←P:WH2,N←P(DUP).[38:4],TYPE) ;                                           02777365 T  0343:2
               BEGIN SI←TYPE; DI←S; DS←WH2 WDS; N(DS←32WDS; DS←32WDS)END;                  02777370 T  0345:2
          P(DEL); T1←T1+WH2 ;                                                              02777375 T  0347:3
          IF ARY←(INDX←INDX+WH2)<SIZE THEN IF WH1 THEN GO BI4 ELSE GO BI2                  02777380 T  0349:2
          ELSE GO BI3 ;                                                                    02777385 T  0352:2
          END ;                                                                            02777390 T  0353:2
    ADDR[0]←IOBUFF[T1]; IF DBLPREC THEN ADDR[1]←IOBUFF[T1←T1+1] ;                          02777395 T  0353:2
    T1←T1+1 ;                                                                              02777400 T  0358:0
BI3:GETNEXTLISTITEM ;                                                                      02777425 T  0359:1
```

```
    IF NOT (DONE OR T1+DBLPREC≥BSIZE) THEN GO BI1 ;                                  02777450 T  0360:0
BI4:IF NOT T4 THEN GO AWAY ;                                                         02777500 T  0363:0
    IF DONE THEN                                                                     02777550 T  0363:3
        BEGIN                                                                        02777600 T  0364:3
BI5:    IF B6700 THEN                                                                02777650 T  0365:1
            BEGIN                                                                    02777655 T  0365:2
            IF BSIZE+T6<T7 THEN T6←T6+BSIZE                                          02777660 T  0366:0
            ELSE BEGIN                                                               02777665 T  0368:0
                WHILE NOT (*(T6 INX *FILX)).[32:1]                                   02777670 T  0369:2
                DO BEGIN IO; T6←0 END ;                                              02777672 T  0371:1
                IF (T6←((*(*[FILX])) AND T3)+1)≥T7 THEN                              02777675 T  0374:1
                    BEGIN IO; T6←0 END ;                                             02777680 T  0377:0
                END ;                                                                02777685 T  0379:3
            FIB[8].[4:14]←T6;  GO ENDIT ;                                            02777690 T  0379:3
            END ;                                                                    02777695 T  0384:0
        WHILE (*(*[FILX])).[18:15]≠T3 DO IO;  GO AWAY ;                              02777700 T  0384:0
B6:::   @10000 ;                                                                     02777725 T  0389:0
        END ;                                                                        02777750 T  0390:0
    IF B6700 THEN                                                                    02777755 T  0390:0
        BEGIN                                                                        02777760 T  0390:1
        IF IOBUFF[T6].[32:1] THEN BEGIN T1←5; GO ERROR END ;                         02777765 T  0390:3
        IF BSIZE+T6≥T7 THEN BEGIN IO; BSIZE←0 END ;                                  02777770 T  0393:2
        T6←BSIZE; GO BI ;                                                            02777775 T  0396:3
        END ;                                                                        02777780 T  0398:0
    IF IOBUFF[0].[18:15]=T3 THEN BEGIN T1←5; GO ERROR END ;                          02777800 T  0398:0
    IO;  GO BI ;                                                                     02777850 T  0401:2
    END OF FBINBACKBLOCK ;                                                           02777950 T  0403:2
                                                                                     SIZE= 0404 WORDS
```

```
PROCEDURE FTINTFIX(FILX,DKADDR,FI,FMT,LISX,EDITCODE,EOFL,PARL); %INT@156          02780000 T  0000:0
                                                                                     START OF REL SEGMENT; DISK ADDRESS = 00309
VALUE DKADDR,FI,LISX,EDITCODE,EOFL,PARL;  ARRAY FMT[*];  NAME FILX ;              02780050 T  0000:0
REAL DKADDR,FI,LISX,EDITCODE,EOFL,PARL ;                                         02780100 T  0000:0
    BEGIN                                                                        02780150 T  0000:0
                                                                                 02780200 T  0000:0
    INTEGER LSTRN=19, IT3 ;                                                      02780250 T  0000:0
                                                                                 02780300 T  0000:0
    REAL LISTYPE=20, HOLTOG=21, ARRAYSTUFF=18, ALGOLREAD=13, SELECT=14,          02780350 T  0000:0
        FORTERR=24, CHR, MAXCHR, FMTW, BUFF, TYPE, INDX, SIZE, TWDT,             02780400 T  0000:0
        INTINT=5, SGN, NEEDNEWLISTADDRESS, SCALE, R, W, D, T3=IT3, T2,           02780450 T  0000:0
        XTRA, DBLPREC, ARY, T1, EXP=PARL, DECPT=EOFL, E=18, WH1=17,              02780500 T  0000:0
     SAVD,                                                                       02780540 C  0000:0
        WH2=9, C=20, CODE, T4, COMMAS, DLRSGN, VL, DC10 ;                        02780550 T  0000:0
                                                                                 02780600 T  0000:0
    NAME LISTADDR, ADDR ;                                                        02780650 T  0000:0
                                                                                 02780700 T  0000:0
    ARRAY TEN=22[*], AR1=LISTADDR[*], TPAR=23[*], FIB[*] ;                       02780750 T  0000:0
                                                                                 02780800 T  0000:0
    LABEL ALIST, GETNEXTPHRASE, REPEAT, TT, XX, SS, PP, AA, OO, HH,              02780850 T  0000:0
          CC, GG, LL, FF, EE, II, DD, ERR3, TEST1, AWAY, JJ,                     02780900 T  0000:0
          ERROR, BACK, GOTNUMBER, MAX, ENDALL, EDIT, BLANKS, ADJT, CHKC,         02780950 T  0000:0
          FERROR, EX, ASK, ERR1, EX1, O1, O2, E1, E2, STNRD, OUTSUB,             02781000 T  0000:0
          CD, NLEL, F094, F095, VERROR, HV, CD1 ;                                02781005 T  0000:0
                                                                                 02781050 T  0000:0
```

```
SWITCH PHRASE←SS,HH,PP,XX,TT,AA,OO,LL,JJ,II,GG,FF,EE,DD,CC ;                    02781100 T  0000:0
                                                                                02781150 T  0000:0
DEFINE DONE = LSTRN=(-1) #,                                                     02781200 T  0000:0
       REEL = 3 #,                                                              02781250 T  0000:0
       LOGICAL = 4 #,                                                           02781300 T  0000:0
       INTEGR = 1 #,                                                            02781350 T  0000:0
       DBLPRECSN = 5 #,                                                         02781400 T  0000:0
       COMPLEX = 6 #,                                                           02781450 T  0000:0
       MAXCODE = 15 #,                                                          02781455 T  0000:0
       VERR(VERR1) = BEGIN P(VERR1); GO VERROR END #,                           02781460 T  0000:0
       ERR(ERR1) = BEGIN P(ERR1); GO ERROR END #,                               02781500 T  0000:0
             H(H1,H2,H3,H4) = IF SC≥"≥" THEN H1: DS←CHR                         02781550 T  0000:0
                              ELSE BEGIN                                        02781600 T  0000:0
                                   IF SC="#" THEN H2:DS←LIT"=" ELSE             02781650 T  0000:0
                                   IF SC="&" THEN H3:DS←LIT"+" ELSE             02781700 T  0000:0
                                   IF SC="%" THEN DS←LIT"(" ELSE                02781750 T  0000:0
                                   IF SC="[" THEN DS←LIT")" ELSE                02781800 T  0000:0
                                   IF SC="@" THEN H4:DS←LIT"'" ELSE             02781850 T  0000:0
                                   IF SC≠":" THEN IF SC≠"<" THEN                02781900 T  0000:0
                                   IF SC≠">" THEN GO H1 ELSE GO H2              02781950 T  0000:0
                                   ELSE GO H3 ELSE GO H4 ;                      02782000 T  0000:0
                                   SI←SI+1 ;                                    02782050 T  0000:0
                                   END #,                                       02782100 T  0000:0
       TWOD = LISTYPE.[38:1] #,                                                 02782150 T  0000:0
       INDXF = [18:15] #,                                                       02782200 T  0000:0
       TYPEF = [44:4] #,                                                        02782250 T  0000:0
       SIZEF = [33:15] # ;                                                      02782300 T  0000:0
                                                                                02782350 T  0000:0
REAL SUBROUTINE NEXTCHR ;                                                       02782400 T  0000:0
     BEGIN                                                                      02782450 T  0001:0
     STREAM(C←0,BUFF:T←T1←T1-1) ;                                               02782500 T  0001:0
          BEGIN DI←LOC BUFF; DI←DI-1; SI←BUFF; DS←CHR; BUFF←SI END ;             02782550 T  0003:2
     BUFF←P;  NEXTCHR←C←P ;                                                     02782600 T  0004:3
     END OF NEXTCHR ;                                                           02782650 T  0006:1
                                                                                02782700 T  0006:2
SUBROUTINE RNDADJ ;                                                             02782705 T  0006:2
     BEGIN                                                                      02782710 T  0007:0
     E←(T4←P(XCH)+T2)+E;  T2←T2-T4;  T3←T3-T4;  VL←1 ;                           02782715 T  0007:0
     P(XCH,TEN[T4],/,,T4,ISN,XCH) ;                                             02782720 T  0012:2
     END OF RNDADJ ;                                                            02782725 T  0014:1
                                                                                02782730 T  0014:2
SUBROUTINE CONVERT ;                                                            02782750 T  0014:2
     BEGIN                                                                      02782800 T  0015:0
     STREAM(V←0,Q←0,R←0,C←0,DECPT,N←0,W←IF T1≥8 THEN 8 ELSE T1:BUFF,             02782850 T  0015:0
            T←0,J←CODE≠9,Z←T1<9) ;                                              02782900 T  0019:1
          BEGIN                                                                 02782950 T  0021:2
          SI←BUFF; DI←LOC T ;                                                   02783000 T  0021:2
          W(L3: IF SC≥"0" THEN BEGIN TALLY←TALLY+1; DS←CHR END                  02783050 T  0022:0
            ELSE IF SC="," THEN                                                 02783100 T  0023:2
                 BEGIN                                                          02783150 T  0024:1
                 CI←CI+DECPT; GO L1; GO L2; L1: Q←TALLY; TALLY←1;               02783200 T  0024:1
                 R←TALLY; DECPT←TALLY; TALLY←Q; SI←SI+1 ;                       02783250 T  0025:3
                 CI←CI+Z; GO L3 ;                                               02783275 T  0027:1
                 END                                                            02783300 T  0028:0
            ELSE IF SC=" " THEN                                                 02783350 T  0028:0
                 BEGIN                                                          02783400 T  0028:3
                 CI←CI+J; GO L2; TALLY←TALLY+1; DS←CHR ;                        02783450 T  0028:3
```

```
                                    END                                                                  02783500 T  0030:0
                                ELSE IF SC="0" THEN                                                      02783550 T  0030:0
                                    BEGIN                                                                02783600 T  0030:3
                                    CI←CI+J; GO L2; DI←DI+1; SI←SI+1 ;                                   02783650 T  0030:3
                                    TALLY←TALLY+1 ;                                                      02783700 T  0032:0
                                    END                                                                  02783750 T  0032:1
                                ELSE BEGIN                                                               02783800 T  0032:1
L2:                                 DI←LOC DECPT; DI←DI-1; DS←CHR ;                                      02783850 T  0032:2
                                    JUMP OUT TO CV ;                                                     02783900 T  0033:1
                                    END) ;                                                               02783950 T  0033:3
CV:         W←SI; SI←LOC T; DI←LOC V; N←TALLY; DS←N OCT ;                                                02784000 T  0034:0
            END ;                                                                                        02784050 T  0035:2
        BUFF←P;  T3←P;  DC10←(DECPT←P) AND CODE>10 ;                                                     02784100 T  0035:3
        T1←T1-(((C←P(XCH))≠0)-P(XCH)-T3;  T4←P;  P(XCH) ;                                                02784125 T  0038:3
        IF COMMAS THEN IF C="," AND NOT DC10 THEN C←0 ;                                                  02784150 T  0042:3
        END OF CONVERT ;                                                                                 02784200 T  0046:1
                                                                                                         02784250 T  0046:2
    REAL SUBROUTINE DEBLANKDEZEROGETSGN ;                                                                02784300 T  0046:2
        BEGIN                                                                                            02784350 T  0047:0
        STREAM(C←0,BUFF,VL←1,SGN←0,T2←0,T1:T←T1.[36:6],HADSGN←SGN) ;                                     02784400 T  0047:0
            BEGIN SI←BUFF; DI←T2 ;                                                                       02784450 T  0050:1
            T1(IF SC≠" " THEN                                                                            02784500 T  0050:3
                    IF SC>"0" THEN GO ENDS                                                               02784550 T  0051:3
                    ELSE IF SC="-" THEN                                                                  02784600 T  0052:2
                        BEGIN                                                                            02784650 T  0053:1
                        BUFF←TALLY; TALLY←1; SGN←TALLY; TALLY←BUFF;                                      02784700 T  0053:1
L1:                     CI←CI+HADSGN; GO L2; HADSGN←DI; VL←DI ;                                          02784750 T  0054:3
                        END                                                                              02784850 T  0056:0
                    ELSE IF SC≠"0" THEN                                                                  02784900 T  0056:0
                      BEGIN                                                                              02784950 T  0056:3
                      IF SC≠"+" THEN IF SC≠"&" THEN                                                      02785000 T  0056:3
                            BEGIN                                                                        02785050 T  0057:3
L2:                         TALLY←TALLY+1; DI←LOC BUFF; DI←DI-1 ;                                        02785100 T  0057:3
                            DS←CHR; ENDS: T2←TALLY ;                                                     02785150 T  0058:2
                            JUMP OUT TO ENDS1 ;                                                          02785200 T  0059:0
                            END ;                                                                        02785250 T  0059:2
                      GO L1 ;                                                                            02785300 T  0059:2
                      END ELSE VL←DI ;                                                                   02785350 T  0059:3
          SI←SI+1; TALLY←TALLY+1) ;                                                                      02785400 T  0060:1
      DI←T1; T2←TALLY ;                                                                                  02785450 T  0061:0
      T(2(32(IF SC≠" " THEN                                                                              02785500 T  0061:2
           IF SC>"0" THEN BEGIN T1←DI; JUMP OUT 3 TO ENDS1 END                                           02785550 T  0063:0
           ELSE IF SC="-" THEN                                                                           02785600 T  0064:3
                    BEGIN TALLY←1; SGN←TALLY ; ;                                                         02785650 T  0065:2
L3:                 CI←CI+HADSGN; GO L4; TALLY←0; HADSGN←TALLY;                                          02785700 T  0066:0
                    VL←TALLY ;                                                                           02785725 T  0067:1
                    END                                                                                  02785750 T  0067:2
                ELSE IF SC≠"0" THEN                                                                      02785800 T  0067:2
                    BEGIN                                                                                02785850 T  0068:1
                    IF SC≠"+" THEN IF SC≠"&" THEN                                                        02785900 T  0068:1
                        BEGIN                                                                            02785950 T  0069:1
L4:                     DI←DI-8; T1←DI; DI←LOC BUFF; DI←DI-1 ;                                           02786000 T  0069:1
                        DS←CHR; JUMP OUT 3 TO ENDS1 ;                                                    02786050 T  0070:1
                        END ;                                                                            02786100 T  0071:2
                    GO L3 ;                                                                              02786150 T  0071:2
                    END ELSE BEGIN TALLY←0; VL←TALLY END ;                                               02786200 T  0071:3
              SI←SI+1; DI←DI-8))) ;                                                                      02786250 T  0072:2
```

```
ENDS1:          BUFF←SI ;                                                    02786300 T  0073:3
                END ;                                                        02786350 T  0074:0
          T1←P(SUB,SSP); SGN←P; VL←P; BUFF←P ;                               02786400 T  0074:1
          DEBLANKDEZEROGETSGN←(C←P)>9 ;                                      02786425 T  0076:3
          END OF DEBLANKDEZEROGETSGN ;                                       02786450 T  0078:0
                                                                             02786500 T  0078:1
     SUBROUTINE CKPB ;                                                       02786550 T  0078:1
          BEGIN                                                              02786600 T  0079:0
          MAXCHR←P(MKS,DKADDR,1,FILX,ALGOLREAD)×8 ;                          02786650 T  0079:0
          BUFF←(P(*FILX)).[33:15] ;                                          02786700 T  0081:1
          END OF CKPB ;                                                      02786750 T  0082:3
                                                                             02786800 T  0083:0
     SUBROUTINE INPUT ;                                                      02786850 T  0083:0
          BEGIN P(0) ;                                                       02786900 T  0083:0
ENDALL:   P(MKS,DKADDR,CHR←0,FILX,ALGOLREAD) ;                               02786950 T  0083:1
          IF P THEN BEGIN FILX[NOT 3]←FILX[NOT 4]←0; P(XIT) END ;            02787000 T  0085:0
          CKPB ;                                                             02787300 T  0089:0
          END OF INPUT ;                                                     02787350 T  0090:0
                                                                             02787400 T  0090:1
     SUBROUTINE GETNEXTLISTADDRESS ;                                         02787450 T  0090:1
          BEGIN                                                              02787500 T  0091:0
          IF NEEDNEWLISTADDRESS THEN                                         02787550 T  0091:0
               BEGIN                                                         02787600 T  0091:1
               IF ARY THEN                                                   02787650 T  0091:3
                    BEGIN                                                    02787700 T  0092:0
ALIST:              IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX AND 255,CDC)     02787750 T  0092:2
                    ELSE P([AR1[INDX]]) ;                                    02787800 T  0095:2
                    ARY←(INDX←INDX+1+DBLPREC) LSS SIZE ;                     02787850 T  0096:2
                    END                                                      02787900 T  0099:1
               ELSE IF TYPE=COMPLEX THEN                                     02787950 T  0099:1
                         BEGIN TYPE←-COMPLEX; P([LISTADDR[1]]) END           02788000 T  0100:2
                    ELSE BEGIN                                               02788050 T  0102:3
                         P(ARRAYSTUFF←0); LISTADDR←[LISX] ;                  02788100 T  0103:1
                         DBLPREC←(TYPE←LISTYPE.TYPEF)=DBLPRECSN ;            02788150 T  0104:3
                         IF ARY←ARRAYSTUFF≠0 THEN                            02788200 T  0107:0
                              BEGIN                                          02788250 T  0108:1
                              IF TYPE=COMPLEX THEN TYPE←-COMPLEX ;           02788300 T  0108:3
                              SIZE←(INDX←ARRAYSTUFF.INDXF)+                  02788350 T  0111:0
                                   ARRAYSTUFF.SIZEF ;                        02788400 T  0112:1
                              P(LISTADDR←MEM[LISTADDR,[18:15]]) ;            02788450 T  0113:3
                              TWDT←NOT P(LOD,TOP); P(DEL) ;                  02788500 T  0115:3
                              GO ALIST ;                                     02788550 T  0117:1
                              END ;                                          02788650 T  0117:3
                         P(DEL,[LISTADDR[0]]) ;                              02788700 T  0117:3
                         END ;                                               02788750 T  0118:1
               NEEDNEWLISTADDRESS←0; ADDR←P ;                                02788800 T  0118:1
               END ;                                                         02788850 T  0119:2
          IF DONE OR EDITCODE=1 THEN                                         02788900 T  0119:2
AWAY:          BEGIN P(1); GO ENDALL END ;                                   02788950 T  0121:2
          END OF GETNEXTLISTADDRESS ;                                        02788955 T  0122:3
                                                                             02788960 T  0123:0
     SUBROUTINE NLE ;                                                        02788965 T  0123:0
          BEGIN P(XCH); GETNEXTLISTADDRESS; NEEDNEWLISTADDRESS←1 ;           02788970 T  0123:0
          IF WH2←DBLPREC THEN WH2←ADDR[1] ;                                  02788975 T  0124:3
          IF (WH1←ADDR[0])+4>P(F094) THEN                                    02788980 T  0127:2
               BEGIN IF T1 THEN VERR(P+10); P(DEL,F094) END                  02788985 T  0129:1
          ELSE IF P(DEL,(-P(F094)),DUP)<WH1 THEN P(DEL,WH1) ;                02788990 T  0132:0
```

```
                P(XCH) ;
                END OF NLE ;

          SUBROUTINE HANDLEVARIABLES ;
                BEGIN T1←1 ;
                IF R=P(F095) THEN
                     BEGIN P(0); NLE; T1←P(,R,ISN)>0; DLRSGN.[18:15]←R ;
                     IF CODE=29 THEN
                          BEGIN P(FI+W) ;
                          IF R≥0 THEN P([FMT[P]],DUP,LOD,P&R[6:36:12],XCH)
                          ELSE P(,FI) ;
                          P(STN) ;
OUTSUB:                   P(DEL,DEL); GO GETNEXTPHRASE ;
                          END ;
                     END ;
                IF T4←CODE=30 THEN
                     BEGIN P(2); NLE; P(,T2,ISN) ;
                     STREAM(P1←P:P2←P(CD),P3←P(CD1)) ;
                          BEGIN SI←LOC P1; SI←SI+7; DI←LOC P2; DI←DI+1 ;
                          32(IF SC=DC THEN JUMP OUT; TALLY←TALLY+1; SI←SI-1) ;
                          P1←TALLY ;
                          END ;
                     IF (T2 AND 63)≠T2 THEN P(DEL,32) ;
                     IF (CODE←P+3)>MAXCODE AND T1 THEN VERR(2) ;
                     T1←CODE>4 AND T1 ;
                     END ;
                T2←T1 ;
                IF P(CODE≥11 AND CODE≤14,F095)=W THEN
                     BEGIN P(,W,4) ;
NLEL:                NLE; P(XCH,ISD); T1←P(DUP) AND T2←T2 AND W>0 ;
                     END ;
                IF D=P(F095) THEN BEGIN P(,D,6); GO NLEL END ;
                IF CODE≤4 THEN
                     BEGIN IF T4 THEN W←R;
                     FMTW←FMTW&(P(DUP),[41:1]+(W<0))[41:47:1]; GO HV ;
                     END ;
                IF NOT T2 THEN GO OUTSUB; IF CODE=5 THEN HV: R←1 ;
                IF P(DUP) AND D<0 THEN VERR(16) ;
                IF T4 THEN IF W=P(F094) THEN
                                BEGIN IF CODE≠9 THEN VERR(6); W←0 END
                           ELSE IF P(DUP) AND D=P(F094) THEN
                                BEGIN P(7) ;
VERROR:                         T4←P; P(MKS,CODE,R,W,D,T4,WH1,WH2,FMTW,
                                     (-5),FORTERR) ;
F094:::  4094 ;
F095:::  4095 ;
CD:::    @0047676321464341 ;   % OPXTAOLJ
CD1:::   @3127262524230000 ;   % IGFEDCOO
                           END ;
                IF NOT P THEN SAVD:=D:=0 ELSE SAVD:=D;
                END OF HANDLEVARIABLES ;

          SUBROUTINE ADJUSTBUFF ;
                BUFF←(P(*FILX) INX T2,[33:12])&T2[30:45:3] ;

          SUBROUTINE SKIP ;
                IF (T1←P(XCH)) GEQ W THEN T1←W
```

```
02788995 T  0135:0
02789000 T  0135:1
02789005 T  0135:2
02789010 T  0135:2
02789015 T  0136:0
02789020 T  0136:3
02789025 T  0137:2
02789030 T  0141:3
02789035 T  0142:2
02789040 T  0143:3
02789045 T  0147:0
02789050 T  0147:3
02789055 T  0148:0
02789060 T  0149:0
02789065 T  0149:0
02789070 T  0149:0
02789075 T  0150:1
02789080 T  0152:2
02789085 T  0153:3
02789090 T  0154:3
02789095 T  0156:3
02789100 T  0157:0
02789105 T  0157:1
02789110 T  0159:2
02789115 T  0162:3
02789120 T  0164:2
02789125 T  0164:2
02789130 T  0165:1
02789135 T  0167:3
02789140 T  0168:3
02789145 T  0173:1
02789150 T  0173:1
02789155 T  0175:2
02789160 T  0176:1
02789165 T  0178:1
02789170 T  0182:0
02789175 T  0182:0
02789180 T  0184:3
02789185 T  0187:1
02789187 T  0188:3
02789190 T  0192:0
02789195 T  0193:3
02789200 T  0194:2
02789205 T  0197:1
02789210 T  0198:0
02789215 T  0199:0
02789220 T  0200:0
02789225 T  0201:0
02789230 T  0202:0
02789235 P  0202:0
02789240 T  0205:1
02789250 T  0205:2
02789300 T  0205:2
02789350 T  0206:0
02789400 T  0209:1
02789450 T  0209:1
02789500 T  0210:0
```

```
          ELSE BEGIN T2←CHR-T1; ADJUSTBUFF END ;                                                    02789550 T  0211:3
                                                                                                    02789600 T  0215:1
%************************:: CODE STARTS HERE ::************************%                           02789650 T  0215:1
                                                                                                    02789700 T  0215:1
    FIB←FILX[NOT 2];  FILX[NOT 3]←PARL;  FILX[NOT 4]←EOFL ;                                         02789750 T  0215:1
    IF FIB[5].[43:2]≠2 THEN P(MKS,0,2,FILX,1,SELECT);  CKPB ;                                       02789800 T  0228:3
    IF NOT(NOT(NEEDNEWLISTADDRESS←EDITCODE=3) OR FMT[FI])THEN GO FERROR;                            02789850 T  0233:0
    IF FIB[0]=0 THEN FIB[0]←1 ;                                                                     02789900 T  0235:2
    IF (LSTRN←1)≠FIB[0] AND FIB[4].[8:4]=2 THEN                                                     02789950 T  0238:2
         BEGIN T3←4 ;                                                                               02790000 T  0241:3
FERROR:  P(MKS,FIB[7],FILX.[33:15],T3,FORTERR) ;                                                    02790050 T  0243:0
         END ;                                                                                      02790100 T  0245:0
    P(0) ;                                                                                          02790150 T  0245:0
GETNEXTPHRASE:                                                                                      02790200 T  0245:1
    R←P(FMT[FI←FI+1],DUP).[6:12];  IF (CODE←P(DUP).[1:5])=2 THEN GO HH ;                             02790250 T  0245:1
    W:=P(DUP).[18:12];  SAVD:=D:=(FMTW:=P(DUP)).[30:12];                                             02790255 P  0249:3
    IF (XTRA←P(DUP) AND 63).[44:2]=0 THEN P(0,0)                                                    02790260 T  0253:3
    ELSE P(P((T4←P(DUP) AND 15)=12,DUP) OR T4=8,P(XCH) OR T4=4) ;                                    02790265 T  0257:0
    DLRSGN←P;  COMMAS←P ;                                                                           02790270 T  0261:3
    IF P.[42:1] THEN IF (FMTW AND 3)=0 THEN HANDLEVARIABLES ;                                       02790275 T  0262:3
    IF CODE=0 THEN                                                                                  02790300 T  0267:0
         BEGIN                                                                                      02790350 T  0267:3
         IF D≠0 THEN BEGIN GETNEXTLISTADDRESS; INPUT END ;                                          02790400 T  0268:1
         IF P(DUP).[18:15]≠FI THEN P(R&FI[18:33:15]) ;                                              02790450 T  0272:0
         IF P((NOT 0),XCH,INX,DUP).[33:15]=0 THEN P(DEL) ELSE FI←FI-W ;                             02790500 T  0274:3
         GO GETNEXTPHRASE ;                                                                         02790550 T  0279:2
         END ;                                                                                      02790600 T  0280:0
    IF CODE=5 THEN CHR←0 ;                                                                          02790650 T  0280:0
REPEAT:                                                                                             02790750 T  0282:0
    IF CODE>5 THEN BEGIN GETNEXTLISTADDRESS; NEEDNEWLISTADDRESS←1 END ;                             02790800 T  0282:0
    IF CODE≠3 AND CODE≠9 THEN                                                                       02790850 T  0284:3
         IF (CHR←CHR+W)>MAXCHR THEN GO AWAY ;                                                       02790900 T  0286:2
    T1←W;  SGN←1;  E←EXP←DECPT←T2←WH1←WH2←0 ;                                                        02790950 T  0289:1
    GO PHRASE[CODE-1] ;                                                                             02791000 T  0294:0
TT:  CHR←W-1 ;                                                                                      02791050 T  0303:0
XX:  T2←CHR;  ADJUSTBUFF;  GO TEST1 ;                                                               02791100 T  0304:1
SS:  INPUT;  GO TEST1 ;                                                                             02791150 T  0306:2
LL:  IF NOT (NEXTCHR≠" " OR T1=0) THEN GO LL ;                                                      02791200 T  0308:2
    IF NOT ((ADDR[0]←C≠"F" AND C≠" ") AND C≠"T") THEN GO XX ;                                       02791250 T  0312:0
    P("G");  IF CODE≠11 THEN P(12,+);  ERR(1) ;                                                     02791300 T  0315:3
PP:  SCALE←W&FMTW[1:41:1];  GO TEST1 ;                                                              02791350 T  0318:2
00:  P(16);  SKIP;  P(DEBLANKDEZEROGETSGN);  IF T1=0 THEN SGN←SGN OR VL ;                           02791400 T  0320:3
01:  IF (T2←C<8 OR C=" ") AND T1>0 THEN                                                             02791450 T  0324:2
         BEGIN                                                                                      02791500 T  0328:3
         P(NEXTCHR&P(XCH)[1:4:44]) ;                                                                02791550 T  0329:1
         IF T2←T1=15 AND C>3 THEN BEGIN P(DEL); GO 02 END ELSE GO 01 ;                              02791600 T  0331:0
         END ;                                                                                      02791650 T  0334:2
    IF SGN THEN P(CHS);  ADDR[0]←P;  IF T2 THEN GO TEST1 ;                                          02791700 T  0334:2
02:  P("0");  ERR(1+T2+T2) ;                                                                        02791750 T  0337:0
GG:  IF TYPE=LOGICAL THEN GO LL;  IF TYPE=INTEGR THEN D:=0;                                         02791800 P  0339:0
    P("G"); GO EDIT ;                                                                               02791825 C  0342:1
EE:  P("E");  GO EDIT ;                                                                             02791850 T  0343:0
FF:  P("F");  GO EDIT ;                                                                             02791900 T  0343:3
DD:  P("D");  GO EDIT ;                                                                             02791950 T  0344:2
II:  P("I");  GO EDIT ;                                                                             02792000 T  0345:1
JJ:  P("J") ;                                                                                       02792050 T  0346:0
EDIT:                                                                                               02792100 T  0346:1
```

```
IF DEBLANKDEZEROGETSGN THEN                                                    02792150 T  0346:1
     BEGIN                                                                     02792200 T  0347:0
     IF DLRSGN THEN                                                            02792250 T  0347:2
          IF C="$" THEN IF NOT DEBLANKDEZEROGETSGN THEN GO E2 ;                02792300 T  0347:3
     IF CODE=9 THEN GO GOTNUMBER ;                                             02792350 T  0351:2
     IF DECPT←C="." THEN                                                       02792355 T  0352:3
          BEGIN IF CODE=10 THEN ERR(2); P((-T1)) ;                             02792360 T  0354:0
          STREAM(BUFF,C←0,T2←0,T1:T←T1,[36:6]) ;                               02792365 T  0357:0
               BEGIN SI←BUFF ;                                                 02792370 T  0359:2
               T1(IF SC≠"0" THEN IF SC≠" " THEN IF SC≠"0" THEN                 02792375 T  0359:3
                    BEGIN                                                      02792380 T  0361:3
                    IF SC<"0" THEN                                             02792385 T  0361:3
                         BEGIN                                                 02792390 T  0362:1
                         DI←LOC T2; DI←DI-1; DS←CHR; TALLY←TALLY+1 ;           02792395 T  0362:1
                         END ;                                                 02792400 T  0363:1
                    JUMP OUT TO L ;                                            02792405 T  0363:1
                    END ;                                                      02792410 T  0363:3
                  TALLY←TALLY+1; SI←SI+1) ;                                    02792415 T  0363:3
               DI←T1 ;                                                         02792420 T  0364:2
               T(2(32(IF SC≠"0" THEN IF SC≠" " THEN IF SC≠"0" THEN             02792425 T  0364:3
                         BEGIN T1←DI ;                                         02792430 T  0367:1
                         IF SC<"0" THEN                                        02792435 T  0367:2
                              BEGIN DI←DI-8 ;                                  02792440 T  0368:0
                              T1←DI; DI←LOC T2; DI←DI-1; DS←CHR ;              02792445 T  0368:1
                              END ;                                            02792450 T  0369:1
                         JUMP OUT 3 TO L ;                                     02792455 T  0369:1
                         END ;                                                 02792460 T  0370:1
                       DI←DI-8; SI←SI+1))) ;                                   02792465 T  0370:1
               T1←DI ;                                                         02792470 T  0371:2
          L:   BUFF←SI; T2←TALLY ;                                             02792475 T  0371:3
               END ;                                                           02792480 T  0372:1
          T1←P(SUB,SSP); C←P; BUFF←P; E←P+T1 ;                                 02792485 T  0372:2
          IF C<10 THEN IF T1=0 THEN GO GOTNUMBER ELSE GO E2 ;                  02792490 T  0375:2
          END ;                                                                02792495 T  0378:2
     IF C="*" THEN GO ASK; WH1←1-DECPT; GO EX1 ;                               02792500 T  0378:2
     END                                                                       02792505 T  0381:2
ELSE IF T1=0 THEN BEGIN SGN←VL OR SGN; GO GOTNUMBER END ;                      02792510 T  0381:2
E2: IF CODE≤10 THEN DECPT←1; VL←0 ;                                            02792550 T  0385:0
BACK:                                                                          02792600 T  0387:3
     CONVERT ;                                                                 02792650 T  0387:3
     IF T3=0 THEN IF COMMAS THEN GO CHKC ELSE P(DEL)                           02792700 T  0389:0
     ELSE BEGIN                                                                02792750 T  0391:2
          IF VL THEN                                                           02792900 T  0392:0
               BEGIN                                                           02792905 T  0392:1
               IF DC10 THEN P(T4)                                              02792910 T  0392:3
               ELSE                                                            02792915 T  0393:2
ADJT:                P(T3) ;                                                   02792920 T  0393:3
               E←E+P; P(DEL) ;                                                 02792925 T  0394:2
               END                                                             02792930 T  0395:3
          ELSE BEGIN                                                           02792950 T  0395:3
               IF DC10 THEN E←E+T4-T3 ;                                        02793000 T  0396:1
               IF (T2←T2+T3)>T3 THEN                                           02793050 T  0398:3
                    BEGIN                                                      02793075 T  0400:2
                    IF T2<12 THEN GO STNRD ;                                   02793100 T  0401:0
                    IF DBLPREC THEN                                            02793125 T  0402:1
                         BEGIN                                                 02793130 T  0402:2
                         IF P(DUP)=0 THEN IF T2>23 OR T1=0 OR C≠0 THEN         02793135 T  0403:0
```

```
                              BEGIN VL←1; T2←T2-T3; GO ADJT END ;                                  02793140 T  0407:0
                         IF T2>23 THEN                                                             02793160 T  0410:0
                              BEGIN                                                                02793165 T  0410:3
                              P((TEN[T2-T3-12]×P(MAX)<WH1)-24);  RNDADJ ;                          02793170 T  0411:1
                              END ;                                                                02793175 T  0415:0
                         WH2←P(0,XCH,WH2,WH1,0,TEN[T3],DLM,DLA,XCH) ;                              02793180 T  0415:0
                         END                                                                       02793185 T  0418:0
                    ELSE BEGIN                                                                     02793190 T  0418:0
                         P((TEN[T3-T2+12]×WH1>P(MAX))-12);  RNDADJ ;                               02793195 T  0418:2
STNRD:                   P(TEN[T3]×WH1,+) ;                                                        02793200 T  0423:0
                         END ;                                                                     02793205 T  0424:1
                    END ;                                                                          02793245 T  0424:1
               WH1←P ;                                                                             02793250 T  0424:1
               END ;                                                                               02793300 T  0424:3
CHKC:     IF NOT (C≠0 OR T1≤0) THEN GO BACK ;                                                      02793350 T  0424:3
          END ;                                                                                    02793400 T  0427:0
     IF C>9 AND CODE≠9 THEN ELSE GO GOTNUMBER ;                                                    02793450 T  0427:0
EX1:IF NOT (C≠"E" AND C≠"D") THEN GO EX ;                                                          02793500 T  0429:1
     IF C="+" OR C="-" THEN BEGIN P(T2,T1+1); SKIP; T2←P END                                       02793510 T  0431:2
     ELSE IF C≠"@" OR HOLTOG THEN GO ERR1 ;                                                        02793515 T  0436:2
EX:  IF CODE=10 THEN ERR(2);  P(SGN);  SGN←EXP←1 ;                                                 02793550 T  0439:0
     IF DEBLANKDEZEROGETSGN THEN                                                                   02793600 T  0442:2
          BEGIN P(DEL) ;                                                                           02793650 T  0444:0
ERR1:     IF C="*" THEN GO GOTNUMBER;  ERR(1+(CODE=10 AND C=",")) ;                                02793700 T  0444:3
          END ;                                                                                    02793750 T  0448:3
     P(DECPT);  DECPT←1;  CONVERT;  IF SGN THEN P(SSN);  E←(T4←P)+E ;                               02793800 T  0448:3
     DECPT←P;  SGN←P;  IF C>9 THEN GO ERR1 ;                                                       02793850 T  0453:2
E1:  IF T1>0 THEN IF NEXTCHR>9 AND C≠" " THEN GO ERR1 ELSE GO E1 ;                                 02793900 T  0455:3
GOTNUMBER:                                                                                         02793950 T  0460:2
     IF CODE>10 THEN                                                                               02794000 T  0460:2
          BEGIN IF NOT DECPT THEN E←E-D; IF NOT EXP THEN E←E-SCALE END                             02794050 T  0461:1
     ELSE IF CODE=9 THEN                                                                           02794100 T  0466:1
               IF (CHR←CHR+W-T1-(C>9)+(C=C←"*"))>MAXCHR THEN GO AWAY ;                             02794150 T  0467:2
     IF ABS(E)>44 THEN                                                                             02794200 T  0473:1
          BEGIN IF E+T2>68 THEN GO ERR3;  IF E<(-68) THEN E←-68 END ;                              02794250 T  0474:1
     IF DBLPREC THEN                                                                               02794300 T  0479:0
          BEGIN                                                                                    02794350 T  0479:1
          P(WH2,WH1) ;                                                                             02794400 T  0479:3
          IF E≠0 THEN P(TEN[ABS(E)+69],TEN[ABS(E)],IF E>0 THEN P(DLM)                              02794450 T  0480:1
                                                           ELSE P(DLD)) ;                          02794500 T  0485:0
          END                                                                                      02794550 T  0485:3
     ELSE BEGIN                                                                                    02794600 T  0485:3
          P(WH1);  IF E≠0 THEN P(TEN[ABS(E)],IF E>0 THEN P(×) ELSE P(/));                          02794650 T  0486:1
          END ;                                                                                    02794700 T  0490:3
     IF SGN THEN P(SSN) ;                                                                          02794750 T  0490:3
     IF TYPE=INTEGR OR TYPE=LOGICAL THEN                                                           02794800 T  0491:3
          BEGIN                                                                                    02794850 T  0493:2
          IF P(DUP)>P(MAX) THEN BEGIN P(DEL); ERR3: ERR(3) END ;                                   02794900 T  0494:0
          P([ADDR[0]],ISD);  GO ASK ;                                                              02794950 T  0496:1
          END ;                                                                                    02795000 T  0497:1
     P([ADDR[0]],←);  IF DBLPREC THEN P([ADDR[1]],←) ;                                             02795350 T  0497:1
ASK:P(DEL);  IF C≠"*" THEN GO TEST1;  GO XX ;                                                      02795400 T  0499:2
MAX:::   @0007777777777777 ;                                                                       02795410 T  0501:2
HH:  P(DEL);  IF (CHR←CHR+R)>MAXCHR THEN GO AWAY ;                                                 02795450 T  0503:0
     STREAM(BUFF:R,S←R,[36:6],HOLTOG,Q←[FMT[FI]]) ;                                                02795460 T  0505:2
          BEGIN                                                                                    02795500 T  0508:1
          DI←DI+3; SI←BUFF; CI←CI+HOLTOG; GO L1; R(H(A,B,C,D)) ;                                   02795550 T  0508:1
```

```
          GO L2; L1: GO L3; L2: S(2(32(H(W,X,Y,Z)))); GO L4; L3:                02795600 T  0521:0
          DS←R CHR; S(DS←32 CHR; DS←32 CHR); L4: BUFF←SI ;                      02795650 T  0534:1
          END ;                                                                 02795700 T  0536:1
     FI←FI+(R+2).[36:9]; BUFF←P; GO GETNEXTPHRASE ;                             02795750 T  0536:2
CC:                                                                             02795800 T  0541:0
AA: P(6); SKIP; ADDR[0]←IF P(CODE=6,DUP) THEN P(BLANKS) ELSE 0 ;                02795850 T  0541:0
     STREAM(T←IF P THEN 2 ELSE 8=T1:BUFF,T1,HOLTOG,ADDR) ;                      02795900 T  0545:0
          BEGIN                                                                 02795950 T  0548:3
          DI←DI+T; SI←BUFF; CI←CI+HOLTOG; GO L1; T1(H(A,B,C,D)) ;               02796000 T  0548:3
          GO L2; L1: DS←T1 CHR; L2: T←SI ;                                      02796050 T  0561:3
          END ;                                                                 02796100 T  0562:3
     BUFF←P ;                                                                   02796150 T  0563:0
TEST1:                                                                          02796200 T  0563:2
     IF (R←R-1)>0 THEN GO REPEAT ;                                              02796250 T  0563:2
     IF (XTRA AND 3)=0 THEN GO GETNEXTPHRASE ;                                  02796300 T  0566:1
     P(XTRA); XTRA←W←0 ;                                                        02796350 T  0568:0
     IF P(DUP) THEN BEGIN W←P.[42:5]; CODE←4; GO REPEAT END ;                   02796400 T  0569:2
     CODE←1; R←P.[42:4]; GO SS ;                                                02796450 T  0574:0
BLANKS:::"      " ;                                                             02796500 T  0578:0
ERROR:                                                                          02796700 T  0579:0
     P(FILX[NOT 3]); FILX[NOT 3]←FILX[NOT 4]←0; P(MKS,9,INTINT) ;               02796750 T  0579:0
     T3←P(DEL); T2←P ;                                                          02796800 T  0584:1
P(MKS,T2,T3,W,SAVD,CODE,TYPE,CHR=T1,FIB[7],BUFF,FMT[FI],DLRSGN,(-3),            02796850 P  0585:2
       FORTERR) ;                                                               02796900 T  0590:0
     END OF FTINTFIX ;                                                          02797450 T  0590:1
                                                                                      SIZE= 0591 WORDS
```

```
PROCEDURE FTINT ;          % 050                                                02800000 T  0000:0
                                                                        START OF REL SEGMENT; DISK ADDRESS = 00329
          BEGIN                                                                 02800100 T  0000:0
COMMENT FILX        FILE TOP IO DESCRIPTOR                                      02800200 T  0000:0
        FMTA        FORMAT OR NAMELIST OR 0                                     02800300 T  0000:0
        LISX        ACCIDENTAL ENTRY DESC. OR 0                                 02800400 T  0000:0
        EDITCODE     0    NO FORMAT, NO LIST                                    02800500 T  0000:0
                     1    FORMAT, NO LIST                                       02800600 T  0000:0
                     2    NO FORMAT, LIST                                       02800700 T  0000:0
                     3    FORMAT, LIST                                          02800800 T  0000:0
                     4    NAMELIST                                              02800900 T  0000:0
                     5    BACKSPACE                                             02801000 T  0000:0
                     6    BLOCKDATA;                                            02801100 T  0000:0
REAL     PARL       =  -1,                                                      02801200 T  0000:0
         EOFL       =  -2,                                                      02801300 T  0000:0
        FORTERR     = 24,                                                       02801301 T  0000:0
         EDITCODE   =  -3,                                                      02801400 T  0000:0
         LISX       =  -4,                                                      02801500 T  0000:0
         FI         =  -6,                                                      02801600 T  0000:0
         DKADR      =  -7,                                                      02801700 T  0000:0
         READINT    =  13,                                                      02801900 T  0000:0
         SELECT     =  14;                                                      02802000 T  0000:0
ARRAY    FMTA       =  -5[*];                                                   02802100 T  0000:0
NAME     FILX       =  -8,                                                      02802160 T  0000:0
         MEM        =   2;                                                      02802190 T  0000:0
INTEGER     LSTRN        = 19;                                                  02802300 T  0000:0
REAL        LISTYPE      = 20,                                                  02802400 T  0000:0
```

```
                ARRAYSTUFF  = 18,                                                                  02802500 T  0000:0
                HOLTOG      = 21;                                                                  02802600 T  0000:0
ARRAY           TEN         = 22[*],                                                               02802900 T  0000:0
                FIB[*];                                                                            02803000 T  0000:0
NAME            LISTADR;                                                                           02803100 T  0000:0
REAL            BUFF                    , %  FIRST BUFFER POSITION                                 02803200 T  0000:0
                BSIZE                   , %  ARGUMENTS                                             02803300 T  0000:0
                NBC,                                                                               02803400 T  0000:0
                NFCI,                                                                              02803600 T  0000:0
                DH1,                                                                               02803700 T  0000:0
                WH1                     , %                                                        02803800 T  0000:0
                WH2;                                                                               02803900 T  0000:0
NAME            W1;                                                                                02804100 T  0000:0
ARRAY           IOBUFF      = BUFF[*];                                                             02804200 T  0000:0
INTEGER         RPT,                                                                               02804300 T  0000:0
                W                       , %  FIELD                                                 02804400 T  0000:0
                BDTYP       = W,                                                                   02804500 T  0000:0
                WT                      , %       WIDTH                                            02804600 T  0000:0
                T1                      , %                                                        02804700 T  0000:0
                D                       , %  DEC-                                                 02804800 T  0000:0
                DT                      , %      IMAL P-                                           02804900 T  0000:0
                D1                      , %          LA-                                           02805000 T  0000:0
                D2                      , %              CE-                                       02805100 T  0000:0
                CNT,                                                                               02805200 T  0000:0
                EXP                     , %  EXPONENT                                              02805300 T  0000:0
                EXPSGN,                                                                            02805400 T  0000:0
                CODE                    , %  EDITING FUNCTION                                      02805500 T  0000:0
                SKP                     , %  REDUNDANT POSITIONS                                    02805600 T  0000:0
                NCR                     , %  CURRENT BUFFER POSITION                               02805700 T  0000:0
                LCR                     , %  BUFFER SIZE IN CHARACTERS                             02805800 T  0000:0
                CHR                     , %  CURRENT CHAR FROM FORMAT                              02805900 T  0000:0
                PRCW        ,     % PAREN CONTROL WORD                                            02805910 T  0000:0
                PCT,          % PAREN COUNTER                                                     02805920 T  0000:0
                PS                      ; %  SCALE FACTOR                                         02806000 T  0000:0
BOOLEAN         DONETOG                 , %  RETURN AFTER WRITE                                   02806100 T  0000:0
                SGN                     , %  SIGN                                                  02806200 T  0000:0
                FRTOG,        % TRUE IF NUM HAS FRACTION PART                                     02806210 T  0000:0
                LGTG,                                                                              02806300 T  0000:0
                DTAERR,                                                                            02806400 T  0000:0
                FMERRTOG                , %  FORMAT ERROR                                         02806500 T  0000:0
                GTOG,                                                                              02806700 T  0000:0
                CTOG;                                                                              02806800 T  0000:0
DEFINE LOGV         = 4#,                                                                          02806900 T  0000:0
       INTEGV        =1#,                                                                          02807000 T  0000:0
       STRGV      =2#,                                                                           02807100 T  0000:0
       DBLV         = 5#,                                                                         02807300 T  0000:0
       CMPLXV       = 6#,                                                                         02807400 T  0000:0
         NUM = 2#,                                                                                 02807600 T  0000:0
       GTYPE        = 1#,                                                                         02807800 T  0000:0
       FTYPE        = 2#,                                                                         02807900 T  0000:0
       ETYPE        = 3#,                                                                         02808000 T  0000:0
       DTYPE        = 4#,                                                                         02808100 T  0000:0
       ITYPE        = 5#,                                                                         02808200 T  0000:0
       LTYPE        = 6#,                                                                         02808300 T  0000:0
       ATYPE        = 7#,                                                                         02808400 T  0000:0
       OTYPE        = 8#,                                                                         02808500 T  0000:0
       KIND          = (FIB[4].[8:4])#,                                                           02808600 T  0000:0
```

```
        TAPEF       =2#,                                                    02808800 T  0000:0
        MAX        = @7777777777777#,                                       02809000 T  0000:0
        ELMTYP      = LISTYPE . [44:4]#,                                    02809100 T  0000:0
        DLN         = (LISTYPE.[44:4] = DBLV)#,                             02809200 T  0000:0
        CMPLX       = (LISTYPE.[44:4] = CMPLXV)#,                           02809300 T  0000:0
        TWOD        = LISTYPE.[38:1]#,                                      02809400 T  0000:0
        LPPS       = 15:30:18#,                                             02809500 T  0000:0
        LPPR       = [15:18]#,                                              02809600 T  0000:0
        RPTF       = [33:15]#,                                              02809700 T  0000:0
        NORF        = (P(XCH,DUP) < 0)#,                                    02809800 T  0000:0
       PCF        = [9:6]#,                                                02809810 T  0000:0
        ENDLIST     = (LSTRN  = (-1))#,                                     02809900 T  0000:0
        SIZEF       = [33:15]#,                                             02810000 T  0000:0
        BASEF       = [18:15]#;                                             02810100 T  0000:0
LABEL    TYPERR, FMCYC, FMERR, MON, FNOL, FMTLST, FRMTCD, NFPH,             02810300 T  0000:0
         STRT,REPEAT,LPAR,RTPAR,SLASH,STRING,TFMT,FMTERR,                   02810400 T  0000:0
         CL1,CL2,CL3,CL4,SCAL,HOL,SKIP,CL3A,STRA,TFMA,TIX,                  02810500 T  0000:0
         G,F,E,DC,I,L,A,O,COMM,                                             02810600 T  0000:0
         LL,LL1;                                                            02810900 T  0000:0
COMMENT  * * * * * START OF SUBROUTINE DECLARATIONS * * * * * * * * * ;    02811000 T  0000:0
SUBROUTINE CKPB;                                                            02811100 T  0000:0
BEGIN COMMENT   INITIALIZE FILE AND ACQUIRE RECORD SIZE;                    02811200 T  0001:0
LCR ← 8×(BSIZE ← P(MKS,DKADR,1,FILX,READINT));                              02811300 T  0001:0
BUFF ← (*FILX).[33:15]; NCR ← 0;                                            02811400 T  0003:3
END CKPB;                                                                   02811500 T  0006:0
SUBROUTINE READS;                                                           02811600 T  0006:1
BEGIN                                                                       02811700 T  0007:0
P(MKS,DKADR,0,FILX,READINT);                                                02811800 T  0007:0
IF DONETOG THEN P(XIT);                                                     02811900 T  0008:1
 $ SET OMIT = NOT(TIMESHARING)                                              02811909 T  0009:1
IF KIND≠10 THEN %%% THEN NOT FROM DATA COM.                                 02811910 T  0009:1
    IF NOT ((*FILX).[19:1]) THEN P(FILX,@2000000000,36,COM,DEL,DEL);        02811920 T  0010:3
 $ SET OMIT = TIMESHARING                                                   02811950 T  0014:2
CKPB;                                                                       02812200 T  0014:2
END READS;                                                                  02812300 T  0016:0
LABEL NFCL;                                                                 02812400 T  0018:0
REAL SUBROUTINE NFC;                                                        02812500 T  0018:0
BEGIN                                                                       02812600 T  0018:0
NFCL:                                                                       02812700 T  0018:0
WHILE NFCI.[45:3] < 2 DO NFCI ← NFCI + 1;                                   02812800 T  0018:0
STREAM(P1 ← 0:P2 ←FMTA[NFCI.[30:15]],P3 ← NFCI.[45:3]);                     02812900 T  0021:2
   BEGIN  DI ← LOC P1;   DS ← 7 LIT "0";                                    02813000 T  0024:1
          SI ← LOC P2; SI ←SI + P3;DS ← CHR;                                02813100 T  0025:3
        SI ← SI - 1; DI ← DI - 1;                                           02813200 T  0026:3
        IF SC < "A" THEN                                                    02813300 T  0027:1
    BEGIN                                                                   02813400 T  0027:3
        IF SC = "@" THEN DS ← LIT """ ELSE                                  02813500 T  0027:3
        IF SC = "[" THEN DS ← LIT ")" ELSE                                  02813600 T  0029:0
        IF SC = "%" THEN DS ←LIT "(";                                       02813700 T  0030:1
    END;                                                                    02813800 T  0031:1
   END;                                                                     02813900 T  0031:1
NFCI ← NFCI + 1; IF (CHR ← P) = " " THEN IF NOT LGTG THEN GO NFCL;          02814000 T  0031:2
NFC ← CHR;                                                                  02814100 T  0035:0
END NFC;                                                                    02814200 T  0035:2
SUBROUTINE PUT;                                                             02814300 T  0035:3
BEGIN ;                                                                     02814400 T  0036:0
            WHILE NFCI.[45:3] <2 DO NFCI ← NFCI + 1;                        02814500 T  0036:0
```

```
                STREAM(P2←[FMTA[NFCI,[30:15]]],P3←NFCI.[45:3],P4←NBC);          02814600 T  0039:2
            BEGIN                                                               02814700 T  0042:0
                SI ← LOC P4; SI←SI+1;                                           02814800 T  0042:0
                DI ← P2; DI←DI+P3; DS←CHR;                                      02814900 T  0042:2
        END;                                                                    02815000 T  0043:2
            NFCI ← NFCI +1;                                                     02815100 T  0043:3
    END PUT;                                                                    02815200 T  0045:0
    SUBROUTINE GET;                                                             02815300 T  0045:1
        BEGIN;                                                                  02815400 T  0046:0
            STREAM(P1 ←[NBC]:P2 ← BUFF);                                        02815500 T  0046:0
            BEGIN                                                               02815600 T  0047:1
                SI ← P2; DI ← P1;                                               02815700 T  0047:1
                DI ← DI+1; DS ← CHR;                                            02815800 T  0047:3
                P1 ← SI;                                                        02815900 T  0048:1
            END;                                                                02816000 T  0048:2
            BUFF ← P;                                                           02816100 T  0048:3
                IF HOLTOG THEN                                                  02816200 T  0049:1
            BEGIN                                                               02816300 T  0049:2
                IF (NBC ← NBC.[6:6]) = "#" THEN NBC ← "=" ELSE                  02816400 T  0050:0
                IF NBC = "&" THEN NBC ← "+" ELSE                                02816500 T  0053:0
                IF NBC = "%" THEN NBC ← "(" ELSE                                02816600 T  0055:2
                IF NBC = "[" THEN NBC ← ")" ELSE                                02816700 T  0058:0
                IF NBC = "@" THEN NBC ← "'";                                    02816800 T  0060:2
                NBC ← 0&NBC[6:42:6];                                            02816900 T  0063:0
            END;                                                                02817000 T  0064:3
        END GET;                                                                02817100 T  0064:3
            % PARAMETERS FOR LIST CONTROL                                       02817200 T  0065:0
    BOOLEAN ATOG,TWDT;                                                          02817300 T  0065:0
    ARRAY AR1 = LISTADR[*];                                                     02817400 T  0065:0
        INTEGER INDX, SIZE;                                                     02817600 T  0065:0
    LABEL RTNLST,SRT;                                                           02817700 T  0065:0
    DEFINE NXTELM = IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX,[40:8],CDC)         02817800 T  0065:0
                            ELSE [AR1[INDX]]#;                                  02817900 T  0065:0
    SUBROUTINE GETLIST;                                                         02818000 T  0065:0
        BEGIN                                                                   02818100 T  0065:0
        SRT: IF ATOG THEN                                                       02818200 T  0065:0
        BEGIN                                                                   02818300 T  0065:1
            W1 ← NXTELM;                                                        02818400 T  0065:3
            INDX ← INDX + DLN;                                                  02818500 T  0070:1
            IF (INDX ←INDX + 1) ≥ SIZE THEN                                     02818600 T  0072:2
        BEGIN                                                                   02818700 T  0074:1
            ARRAYSTUFF ← 0;                                                     02818800 T  0074:3
            ATOG ← FALSE;                                                       02818900 T  0075:2
        END;                                                                    02819000 T  0076:1
            GO TO RTNLST;                                                       02819100 T  0076:1
        END;                                                                    02819200 T  0076:3
            IF CTOG THEN                                                        02819300 T  0076:3
        BEGIN     % IMAGINARY PART OF COMPLEX                                   02819400 T  0077:0
            W1 ← [LISTADR[1]];                                                  02819500 T  0077:2
            CTOG ← FALSE;                                                       02819600 T  0078:3
            GO TO RTNLST;                                                       02819700 T  0079:2
        END;                                                                    02819800 T  0080:0
            P(0);                                                               02819900 T  0080:0
            LISTADR ← [LISX];                                                   02820000 T  0080:1
            IF ARRAYSTUFF ≠ 0 THEN                                              02820100 T  0081:0
        BEGIN                                                                   02820200 T  0081:3
            ATOG ← TRUE;                                                        02820300 T  0082:1
```

```
            TWDT←NOT P(*(LISTADR←MEM[LISTADR,[18:15]]),TOP);  P(DEL) ;          02820400 T  0083:0
            SIZE←(INDX←ARRAYSTUFF.BASEF)+ARRAYSTUFF.SIZEF ;                    02820700 T  0086:2
            GO TO SRT;                                                         02820800 T  0089:1
        END;                                                                   02820900 T  0089:3
            W1 ← [LISTADR[0]];                                                 02821000 T  0089:3
            P(DEL);                                                            02821100 T  0090:2
            CTOG ← CMPLX;                                                      02821200 T  0090:3
        RTNLST:                                                                02821300 T  0092:2
        END GETLIST;                                                           02821400 T  0092:2
    SUBROUTINE FORMATCONTROL;                                                  02821500 T  0092:3
    BEGIN                                                                      02821600 T  0093:0
        STRT:                                                                  02821700 T  0093:0
                W←D←CODE←SKP←RPT←0;                                            02821800 T  0093:0
                SGN←DONETOG←FMERRTOG←FALSE;                                    02821900 T  0095:3
        CL1: COMMENT CHECK FOR SINGLE CHARACTER EDITING TYPES;                 02822000 T  0097:2
                IF NFC≤9 THEN GO TO REPEAT; % MUST BE REPEAT FIELD             02822100 T  0097:2
                IF CHR="(" THEN GO LPAR;                                       02822200 T  0100:0
                IF CHR=")" THEN GO RTPAR;                                      02822300 T  0101:1
                IF CHR="/" THEN GO SLASH;                                      02822400 T  0102:2
                IF CHR="'" THEN GO STRING;                                     02822500 T  0103:3
                IF CHR="T" THEN GO TO TFMT;                                    02822600 T  0105:0
                SGN←(CHR="-") & (CHR="+")[2:47:1];                             02822690 T  0106:1
                IF SGN THEN                                                    02822700 T  0109:0
            BEGIN                                                              02822800 T  0109:1
                IF NFC≤9 THEN GO TO REPEAT                                     02822900 T  0109:3
                         ELSE GO TO FMTERR;                                    02823000 T  0111:2
            END;                                                               02823100 T  0112:2
                IF CHR="," THEN GO TO STRT;                                    02823200 T  0112:2
                RPT←1;                                                         02823300 T  0113:3
        CL2: COMMENT  TYPES WHICH MAY HAVE REPEAT FIELDS;                      02823400 T  0114:2
                IF SGN THEN RPT←-RPT;                                          02823500 T  0114:2
                IF CHR="P" THEN GO TO SCAL;                                    02823600 T  0116:1
                IF RPT<0 OR SGN.[2:1] THEN GO TO FMTERR;                       02823700 T  0117:2
                IF CHR="(" THEN GO TO LPAR;                                    02823800 T  0120:0
                IF CHR="H" THEN GO TO HOL;                                     02823900 T  0121:1
                IF RPT=0 THEN RPT←1;                                           02824000 T  0122:2
                IF CHR = "X" THEN GO TO SKIP;                                  02824100 T  0124:2
        CL3: COMMENT TYPES WHICH HAVE W FIELDS;                                02824200 T  0125:3
                IF CHR="I" THEN CODE ← ITYPE ELSE                              02824300 T  0125:3
                 IF CHR="A" THEN CODE ← ATYPE ELSE                             02824400 T  0127:3
                  IF CHR="L" THEN CODE ← LTYPE ELSE                            02824500 T  0130:1
                   IF CHR="O" THEN CODE ← OTYPE;                               02824600 T  0132:3
                IF CODE ≥ ITYPE THEN GO TO CL3A;                               02824700 T  0135:1
        CL4: COMMENT  TYPES WITH W AND D FIELDS;                               02824800 T  0136:2
                IF CHR="D" THEN CODE ← DTYPE ELSE                              02824900 T  0136:2
                 IF CHR="E" THEN CODE ← ETYPE ELSE                             02825000 T  0138:2
                  IF CHR="F" THEN CODE ← FTYPE ELSE                            02825100 T  0141:0
                   IF CHR="G" THEN CODE ← GTYPE ELSE                           02825200 T  0143:2
                    GO TO FMTERR;                                              02825300 T  0146:0
        CL3A: COMMENT DEVELOP VALUE OF W FIELD;                                02825400 T  0146:0
                IF NFC>9 THEN GO TO FMTERR;                                    02825500 T  0146:0
                W←CHR;                                                         02825600 T  0148:0
                WHILE NFC≤9 DO W←10×W+CHR; % CONVERT TO OCTAL                  02825700 T  0148:3
                NFCI←NFCI-1;                                                   02825800 T  0153:1
                IF W>63 THEN GO TO FMTERR;                                     02825900 T  0154:2
                IF CODE≥ITYPE THEN GO TIX;                                     02826000 T  0155:3
            COMMENT DEVELOP D FIELD;                                           02826100 T  0157:0
```

```
          IF NFC≠"." THEN GO TO FMTERR;                                    02826200 T  0157:0
          IF NFC>9 THEN GO TO FMTERR;                                      02826300 T  0159:0
          D←CHR;                                                           02826400 T  0161:0
          WHILE NFC≤9 DO D←10×D+CHR; % CONVERT TO OCTAL                    02826500 T  0161:3
          NFCI←NFCI-1;                                                     02826600 T  0166:1
          GO TO TIX;                                                       02826700 T  0167:2
LPAR: COMMENT GENERITE PAREN CONTROL WORD;                                 02826800 T  0168:0
          IF PCT≠0 AND RPT=0 THEN RPT←1 ;                                  02826810 T  0168:0
          T1 ← RPT&NFCI[LPPS]&(RPT≤0)[1:47:1];                             02826900 T  0171:0
          IF PCT ≤ 1 THEN PRCW ← T1 & PCT[9:42:6];                         02826910 T  0174:1
          P(T1,XCH); PCT ← PCT + 1;                                        02826920 T  0177:1
          GO TO STRT;                                                      02827000 T  0179:0
RTPAR: COMMENT POINT AT LEFT PAR IF REPEAT NOT EXAUSTED;                   02827100 T  0179:2
          IF NORF THEN                                                     02827200 T  0179:2
     BEGIN     % NO REPEAT FIELD                                           02827300 T  0180:2
          DONETOG ← ENDLIST ;                                              02827400 T  0181:0
          READS;                                                           02827500 T  0182:2
          IF (PCT ← PCT - 1) ≤ 0 THEN IF PRCW.PCF ≠0                       02827510 T  0184:0
           THEN BEGIN P(XCH,PRCW); PCT ← 2 END ELSE PCT ← 1;               02827520 T  0187:0
     END ELSE                                                              02827600 T  0190:2
     BEGIN                                                                 02827700 T  0190:2
          IF (RPT←P(DUP).RPTF) ≤ 1                                         02827800 T  0191:0
           THEN BEGIN P(DEL);PCT ← PCT - 1; GO TO STRT END                 02827900 T  0192:1
           ELSE P(RPT - 1,CCX);                                           02828000 T  0195:1
     END;                                                                  02828100 T  0196:3
          NFCI←P(DUP).LPPR; % RESET TO LEFT PAREN                          02828200 T  0196:3
          P(XCH);                                                          02828300 T  0198:0
          GO TO STRT;                                                      02828400 T  0198:1
REPEAT: COMMENT CONVERT REPEAT FIELD TO OCTAL IN RPT;                      02828500 T  0198:3
          RPT←CHR;                                                         02828600 T  0198:3
          WHILE NFC≤9 DO RPT← 10×RPT+CHR;                                  02828700 T  0199:2
          GO TO CL2;                                                       02828800 T  0204:1
SLASH:    % READ NEXT RECORD                                               02828900 T  0204:3
          READS;                                                           02829000 T  0204:3
          GO TO STRT;                                                      02829100 T  0206:0
STRING:   % MOVE STRING FROM BUFFER TO FORMAT                              02829200 T  0206:2
          LGTG ← TRUE;                                                     02829300 T  0206:2
          GET; PUT; NCR ← NCR + 1;                                         02829400 T  0207:1
STRA:     IF NFC = """ THEN BEGIN LGTG ← FALSE; GO TO STRT END;            02829500 T  0210:1
          IF (NCR ← NCR + 1) > LCR THEN GO TO FMTERR;                      02829600 T  0213:1
          GET; NFCI ← NFCI-1;                                              02829700 T  0215:2
          PUT;                                                             02829800 T  0218:1
          GO TO STRA;                                                      02829900 T  0219:0
TFMT:  COMMENT SET BUFFER TO CHARACTER POSITION INDICATED BY FIELD         02830000 T  0219:2
          FOLLOWING "T";                                                   02830100 T  0219:2
          IF (RPT←NFC)>9 THEN GO TO FMTERR;                                02830200 T  0219:2
          WHILE NFC≤9 DO RPT←10×RPT+CHR;                                   02830300 T  0222:2
          IF RPT>LCR THEN GO TO FMTERR;                                    02830400 T  0227:1
          NCR←RPT-1;                                                       02830500 T  0228:2
TFMA:     BUFF ← ((*FILX) INX NCR,[33:12])&NCR[30:45:3];                   02830600 T  0229:3
          GO TO STRT;                                                      02830700 T  0232:3
SCAL: COMMENT SCALE FACTOR OF P PHRASE;                                    02830800 T  0233:1
          PS←RPT;                                                          02830900 T  0233:1
          GO TO STRT;                                                      02831000 T  0234:0
HOL: COMMENT HOLLERITH STRING;                                             02831100 T  0234:2
          WHILE RPT > 0 DO                                                 02831200 T  0234:2
     BEGIN                                                                 02831300 T  0235:3
```

```
                IF (NCR ← NCR + 1) > LCR THEN GO TO FMTERR;                          02831400 T  0235:3
                GET; PUT;                                                            02831500 T  0238:0
                RPT←RPT-1;                                                           02831600 T  0240:0
            END;                                                                     02831700 T  0241:1
                GO TO STRT;                                                          02831800 T  0241:3
        SKIP: COMMENT X PHRASE;                                                      02831900 T  0242:1
                IF (NCR ← NCR+RPT) > LCR THEN GO TO FMTERR;                          02832000 T  0242:1
                GO TO TFMA;                                                          02832100 T  0244:2
        FMTERR:   FMERRTOG←TRUE;                                                     02832200 T  0245:0
TIX:                                                                                 02832300 T  0245:3
END FORMATCONTROL;                                                                   02832400 T  0245:3
SUBROUTINE SKPC;    % SKIPS CURRENT CHARACTERS, PUTS NEXT CHARACTERS                  02832500 T  0246:0
    BEGIN;          % IN NBC                                                        02832600 T  0246:0
        STREAM(P1←BUFF,P2←0:P3←0);                                                   02832700 T  0246:0
        BEGIN                                                                        02832800 T  0247:2
            SI ← P1; SI ← SI +1; P1 ← SI;                                            02832900 T  0247:2
            DI ← LOC P2; DI ← DI+7; DS ← CHR;                                        02833000 T  0248:1
        END;                                                                         02833100 T  0249:0
        NBC ← P; BUFF ← P;                                                           02833200 T  0249:1
        WT ← WT -1;                                                                  02833300 T  0250:1
    END SKPC;                                                                        02833400 T  0251:2
SUBROUTINE SCALE;                                                                    02833500 T  0251:3
    BEGIN                                                                            02833600 T  0252:0
        IF (D1 ← D1 + CNT) > 11                                                      02833700 T  0252:0
         THEN DOUBLE(WH1,WH2,TEN[CNT],TEN[69+CNT],X,                                 02833800 T  0253:1
                     DH1,0,+,←,WH1,WH2)                                              02833900 T  0256:3
         ELSE WH1← WH1×TEN[CNT]+DH1;                                                 02834000 T  0258:1
        DH1 ← 0;                                                                     02834100 T  0261:0
    END SCALE;                                                                       02834200 T  0261:3
SUBROUTINE GETNUM;                                                                   02834300 T  0262:0
    BEGIN;                                                                           02834400 T  0262:0
        STREAM(P1←BUFF,P2←IF WT ≤ 8 THEN WT ELSE 8,P3←0,P4←0:P5←0);                   02834500 T  0262:0
        BEGIN                                                                        02834600 T  0266:0
            SI←P1; DI←LOC P5 ;                                                       02834700 T  0266:0
            P2(IF SC<"0" THEN                                                        02834800 T  0266:2
                 IF SC≠" " THEN                                                      02834900 T  0267:2
                     IF SC="0" THEN BEGIN DS←LIT"0"; SI←SI+1;GO L END                02835000 T  0268:0
                     ELSE JUMP OUT ;                                                 02835100 T  0269:2
               DS←CHR ;                                                              02835200 T  0270:1
L:             TALLY←TALLY+1) ;                                                      02835300 T  0270:2
            P2←TALLY; P1←SI ;                                                        02835400 T  0271:0
            SI←LOC P5; DI←LOC P3; DS←P2 OCT ;                                        02835500 T  0271:2
            SI←P1 ;                                                                  02835600 T  0272:2
            DI ← LOC P4; DI ← DI + 7; DS ← CHR;                                      02835700 T  0272:3
        END;                                                                         02835800 T  0273:2
        NBC ← P; DH1 ← P; CNT ← P; BUFF ← P;                                         02835900 T  0273:3
    END GETNUM;                                                                      02836000 T  0275:3
SUBROUTINE GETSIGN;                                                                  02836100 T  0276:0
    BEGIN;                                                                           02836200 T  0276:0
        STREAM(P1←BUFF,P2←(IF WT > 63 THEN 63 ELSE WT),P3←0,P4←(-1):                 02836300 T  0276:0
          P5←0);                                                                     02836310 T  0279:2
        BEGIN                                                                        02836400 T  0280:1
           SI←P1;  DI←P2 ;                                                           02836500 T  0280:1
           P2(DI←DI-8;  IF SC≠" " THEN JUMP OUT TO L1;                               02836600 T  0280:3
               SI ← SI + 1; TALLY ← TALLY + 1);                                      02836700 T  0282:2
            P1 ← SI;                                                                 02836800 T  0283:1
            GO TO RTNSGN;                                                            02836900 T  0283:2
```

```
        L1:  IF SC ≥ "0" THEN                                                    02837000 T  0283:3
        BEGIN                                                                    02837100 T  0284:1
        L3:  P2 ← TALLY;                                                         02837200 T  0284:1
        L2:  P5(P1←DI; TALLY←P2; P1(IF SC≠" " THEN                               02837300 T  0284:2
             JUMP OUT;  TALLY←TALLY+1;  SI←SI+1);  P2←TALLY);  P1←SI ;            02837310 T  0287:0
             DI ← LOC P4; DS ← 7 LIT "0"; DS ← CHR;                              02837400 T  0289:0
             GO TO RTNSGN;                                                       02837500 T  0290:3
        END;                                                                     02837600 T  0291:0
             IF SC "," THEN GO TO L3;                                            02837700 T  0291:0
             TALLY ← TALLY+1;                                                    02837800 T  0291:3
             P2 ← TALLY;                                                         02837900 T  0292:0
             TALLY←1;  P5←TALLY ;                                                02838000 T  0292:1
             IF SC="-" THEN TALLY←1 ELSE IF SC="+" THEN TALLY←0 ELSE             02838010 T  0292:3
             IF SC="&" THEN TALLY←0 ELSE                                         02838020 T  0294:3
                BEGIN TALLY←0;  P1←TALLY;  GO TO RTNSGN END;                     02838025 T  0295:3
             P3 ← TALLY;                                                         02838100 T  0296:2
             SI ← SI + 1;                                                        02838200 T  0296:3
             GO TO L2;                                                           02838300 T  0297:0
        RTNSGN:                                                                  02838400 T  0297:1
        END;                                                                     02838500 T  0297:1
        NBC←P;  SGN←P;  CNT←P;  DTAERR←((BUFF←P)=0) ;                            02838600 T  0297:2
    END GETSIGN;                                                                 02838700 T  0300:2
    LABEL     NCRTN,BLSGN;                                                       02838800 T  0300:3
SUBROUTINE NUMCONVERT;                                                           02838900 T  0300:3
    BEGIN                                                                        02839000 T  0301:0
        DH1 ← D1 ← D2 ← EXP ← EXPSGN ← FRTOG ←0;                                 02839100 T  0301:0
        WH1 ← WH2 ← =0;                                                          02839200 T  0304:1
        WT ← W;                                                                  02839300 T  0305:3
BLSGN:                                                                           02839310 T  0306:2
     GETSIGN;                                                                    02839400 T  0306:2
     IF DTAERR THEN GO TO NCRTN ;                                                02839405 T  0308:0
     WT ← WT - CNT; IF NBC <0       % BLANK FIELD                                02839500 T  0309:0
       THEN IF WT ≤ 0 THEN GO TO NCRTN ELSE GO TO BLSGN;                         02839510 T  0310:3
     IF NBC ≤ 9 THEN                                                             02839600 T  0313:1
BEGIN                                                                            02839700 T  0314:0
     GETNUM;  WH1 ← DH1;                                                         02839800 T  0314:2
     IF (WT ← WT - (D1 ← CNT)) ≤ 0 THEN GO TO NCRTN;                             02839900 T  0316:3
     WHILE NBC≤9 OR NBC=" " OR NBC="0" DO                                        02840000 T  0319:2
BEGIN                                                                            02840100 T  0322:3
     GETNUM; SCALE;                                                              02840200 T  0322:3
     IF (WT ← WT-CNT) ≤ 0 THEN GO TO NCRTN;                                      02840300 T  0325:0
END;                                                                             02840400 T  0327:1
END;                                                                             02840500 T  0327:3
     IF NBC = "." THEN                                                           02840600 T  0327:3
BEGIN                                                                            02840700 T  0328:2
     FRTOG ← TRUE;                                                               02840800 T  0329:0
     SKPC;                                                                       02840900 T  0329:3
     IF WT≤0 THEN GO TO NCRTN ;                                                  02840910 T  0331:0
     WHILE NBC≤9 OR NBC=" " OR NBC="0" DO                                        02841000 T  0332:1
BEGIN                                                                            02841100 T  0335:2
     GETNUM; SCALE;                                                              02841200 T  0335:2
     D2 ← D2 + CNT;                                                              02841300 T  0338:0
     IF ( WT ← WT - CNT) ≤ 0 THEN GO TO NCRTN;                                   02841400 T  0339:1
END;                                                                             02841500 T  0341:2
END;                                                                             02841600 T  0342:0
     IF NBC = "D" OR NBC = "E" THEN SKPC;                                        02841700 T  0342:0
     IF WT≤0 THEN BEGIN DTAERR←TRUE;  GO TO NCRTN END ;                          02841710 T  0345:0
```

```
          IF (NBC="+") OR (NBC="&") OR (NBC=" ") OR (EXPSGN←(NBC="-"))          02841800 T  0347:2
            THEN SKPC;                                                          02841900 T  0351:0
          IF WT≤0 THEN BEGIN DTAERR←TRUE;  GO TO NCRTN END ;                    02841910 T  0353:0
          IF NBC > "9" THEN DTAERR ← TRUE                                       02842000 T  0355:2
          ELSE                                                                  02842100 T  0357:0
     BEGIN                                                                      02842200 T  0357:2
          GETNUM;                                                               02842300 T  0358:0
          EXP ← IF EXPSGN THEN (-DH1) ELSE DH1;                                 02842400 T  0359:0
          IF (WT←WT-CNT) ≤ 0 THEN GO TO NCRTN;                                  02842500 T  0361:2
          WHILE WT > 0 DO SKPC;                                                 02842600 T  0363:3
     END;                                                                       02842700 T  0366:2
     NCRTN:                                                                     02842800 T  0366:2
          IF WH1 = 0 THEN IF SGN THEN WH1 ← -0;                                 02842900 T  0366:2
     END NUMCONVERT;                                                            02843000 T  0369:2
SUBROUTINE CONVERT;                                                             02843100 T  0369:3
     BEGIN                                                                      02843200 T  0370:0
          WH1 ← WH2 ← 0; WT ←W;                                                 02843300 T  0370:0
          GO TO P(CODE,DUP,ADD);                                                02843400 T  0372:0
          GO TO FMERR;                                                          02843500 T  0373:0
          GO TO G;  GO TO F; GO TO E; GO TO DC; GO TO I;                        02843600 T  0373:2
          GO TO L;  GO TO A; GO TO 0;                                           02843700 T  0376:0
     0:        % OCTAL CONVERSION                                              02843800 T  0377:2
          IF W>16 THEN SKP←W-WT←16 ELSE SKP←0 ;                                 02843900 T  0377:2
          STREAM(P1←BUFF,P2←0;P3←[WH1],P4←SKP,P5←WT,P6←16-WT,P7←0,               02843950 T  0381:3
          P8←(WT=16)) ;                                                         02844000 T  0384:2
               BEGIN                                                            02844050 T  0385:2
               SI←P1;  P1←TALLY;  TALLY←1 ;                                     02844100 T  0385:2
               P4(IF SC=" " THEN SI←SI+1 ELSE IF SC="0" THEN SI←SI+1            02844150 T  0386:1
                  ELSE BEGIN P7(JUMP OUT 2 TO MAST);  IF SC="-" THEN            02844200 T  0388:1
                  P2←TALLY ELSE IF SC≠"+" THEN IF SC≠"&" THEN JUMP OUT          02844250 T  0390:3
                  TO MAST;  P7←TALLY;  SI←SI+1 END) ;                           02844300 T  0392:1
               P8(IF SC>"3" THEN JUMP OUT TO MAST) ;                            02844350 T  0393:2
               DI←P3;  P6(SKIP 3 DB) ;                                          02844400 T  0395:1
               GO TO FAST;  MAST:  GO TO LAST;  FAST:                           02844410 T  0396:2
               P5(IF SC>"0" THEN IF SC<"8" THEN BEGIN SKIP 3 SB;  P7←           02844450 T  0397:0
                  TALLY;  3(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB) END       02844500 T  0398:3
                  ELSE JUMP OUT TO LAST ELSE BEGIN IF SC≠" " THEN               02844550 T  0401:0
                  IF SC≠"0" THEN BEGIN P7(JUMP OUT 2 TO LAST);                  02844600 T  0402:2
                  IF SC="-" THEN P2←TALLY ELSE IF SC≠"+" THEN                   02844650 T  0404:2
                  IF SC≠"&" THEN JUMP OUT TO LAST;  P7←TALLY END ;              02844700 T  0406:0
                  SI←SI+1;  SKIP 3 DB END) ;                                    02844750 T  0407:1
               P1←SI ;                                                          02844800 T  0408:0
               LAST:                                                            02844850 T  0408:1
               END ;                                                            02844900 T  0408:1
          SGN←P;  IF (DTAERR←((BUFF←P)=0)) THEN GO TO COMM ;                    02844950 T  0408:2
          W1[0]←IF SGN THEN -WH1 ELSE WH1 ;                                     02844975 T  0411:1
          GO TO COMM;                                                           02845000 T  0413:3
     A:   % ALPHA CONVERSION                                                    02845100 T  0414:1
          IF W > 6 THEN SKP ← W - (WT ← 6);                                     02845200 T  0414:1
          WH1 ← "      ";                                                       02845300 T  0417:1
          STREAM(P1← BUFF:P2←[WH1],P3←SKP,P4←WT,P5←HOLTOG);                     02845400 T  0418:0
          BEGIN                                                                 02845500 T  0420:0
               SI ← P1; SI ← SI + P3;                                           02845600 T  0420:0
               DI ← P2; DI ← DI + 2;                                            02845700 T  0420:3
               P5(P4(                                                           02845800 T  0421:1
               IF SC ≥ "A" THEN DS ← CHR ELSE                                   02845900 T  0422:1
               IF SC = "#" THEN BEGIN SI ← SI +1; DS ← LIT "=" END ELSE          02846000 T  0423:1
```

```
            IF SC = "&" THEN BEGIN SI ← SI +1; DS ← LIT "+" END ELSE          02846100 T  0424:3
            IF SC = "%" THEN BEGIN SI ← SI +1; DS ← LIT "(" END ELSE          02846200 T  0426:1
            IF SC = "[" THEN BEGIN SI ← SI +1; DS ← LIT ")" END ELSE          02846300 T  0427:3
            IF SC = "@" THEN BEGIN SI ← SI +1; DS ← LIT """ END ELSE          02846400 T  0429:1
            DS ← CHR;);                                                       02846500 T  0430:3
            JUMP OUT TO X;);                                                  02846600 T  0431:1
            DS ← P4 CHR;                                                      02846700 T  0432:0
        X:                                                                    02846800 T  0432:2
            P1 ← SI;                                                          02846900 T  0432:2
        END;                                                                  02847000 T  0432:3
        BUFF ← P;                                                             02847100 T  0433:0
        W1[0] ← WH1;                                                          02847200 T  0433:2
        GO TO COMM;                                                           02847300 T  0434:1
    L:  % LOGICAL CONVERSION                                                  02847400 T  0436:0
        STREAM(P1←BUFF,P2←0:P3←0,P4←W);                                       02847500 T  0436:0
        BEGIN                                                                 02847600 T  0437:3
            SI ← P1; P3←SI;                                                   02847700 T  0437:3
            P4(IF SC≠" " THEN JUMP OUT 1 TO LL ELSE SI ← SI + 1);             02847800 T  0438:1
            TALLY ← 0; GO TO LL1;                                             02847900 T  0440:2
            LL: IF SC = "T" THEN TALLY ← 1;                                   02848000 T  0441:0
       LL1: P2 ← TALLY; SI ← P3; SI ← SI +P4; P1 ← SI;                        02848100 T  0441:3
        END;                                                                  02848200 T  0443:0
        W1[0] ← P;                                                            02848300 T  0443:1
        BUFF  ← P;                                                            02848400 T  0443:3
        DTAERR ← ELMTYP ≠ LOGV;                                               02848600 T  0444:1
        GO TO COMM;                                                           02848700 T  0446:0
    I:  % INTEGER CONVERSION                                                  02848800 T  0446:2
        NUMCONVERT;                                                           02848900 T  0446:2
        IF (DTAERR ← DTAERR OR D2≠0 OR EXP ≠ 0                                02849000 T  0448:0
                    OR ELMTYP = DBLV OR WH1 > MAX)                            02849100 T  0449:2
         THEN GO TO COMM;                                                     02849200 T  0452:1
        W1[0] ← IF SGN THEN -WH1 ELSE WH1;                                    02849300 T  0454:0
        GO TO COMM;                                                           02849400 T  0456:2
        % SINGLE PRECISION                                                    02849500 T  0458:0
    E:  % E FORMAT                                                            02849600 T  0458:0
    F:  % F FORMAT                                                            02849700 T  0458:0
    G:  % G FORMAT                                                            02849800 T  0458:0
        NUMCONVERT;                                                           02849900 T  0458:0
        IF (W1[0] ← WH1) = 0 THEN GO TO COMM;                                 02850000 T  0459:0
        IF (DTAERR ← DTAERR OR ELMTYP = LOGV                                  02850100 T  0460:3
                    OR ELMTYP = INTEGV OR ELMTYP= DBLV)                       02850200 T  0462:0
         THEN GO TO COMM;                                                     02850300 T  0465:0
        T1 ← (IF EXP ≠ 0 THEN EXP ELSE -PS)                                   02850400 T  0466:3
            -(IF FRTOG THEN D2 ELSE D);                                       02850500 T  0468:3
        IF T1<(-68) THEN T1←-68 ELSE IF DTAERR←T1>68 THEN GO TO COMM ;        02850510 T  0471:3
        IF D1 GTR 11 THEN IF T1 GTR 0 THEN                                    02850530 T  0476:3
          DOUBLE(WH1,WH2,TEN[ T1],TEN[69+T1],TIMES,:=,WH1,WH2)                02850535 T  0478:3
         ELSE                                                                 02850540 T  0482:2
          BEGIN                                                               02850545 T  0482:3
           DOUBLE(WH1,WH2,TEN[-T1],TEN[69-T1],/,:=,WH1,WH2);                  02850550 T  0483:1
           IF WH2 > @0007777777777700 THEN                                    02850553 T  0487:0
           IF WH1.[3:6] LSS 14 THEN WH1 := WH1 + 1 & WH1[2:2:7];              02850555 T  0487:3
          END                                                                 02850560 T  0492:1
        ELSE                                                                  02850565 T  0492:1
        WH1 ← IF T1≥ 0 THEN WH1×TEN[T1]                                       02850600 T  0492:1
                       ELSE WH1/TEN[-T1];                                     02850700 T  0495:3
        W1[0] ← IF SGN THEN -WH1 ELSE WH1;                                    02850800 T  0498:2
```

```
            GO TO COMM;                                                                                  02850900 T  0501:0
        DC: % DOUBLE PRECISION CONVERSION                                                                02851000 T  0501:2
            NUMCONVERT;                                                                                  02851100 T  0501:2
            IF WH1 = 0 THEN BEGIN W1[0] ← W1[1] ←WH1; GO TO COMM END;                                    02851200 T  0503:0
            IF (DTAERR ← DTAERR OR ELMTYP ≠ DBLV )                                                       02851300 T  0506:3
             THEN GO TO COMM;                                                                            02851400 T  0508:0
            T1 ← (IF EXP ≠ 0 THEN EXP ELSE -PS)                                                          02851500 T  0509:3
                -(IF FRTOG THEN D2 ELSE D);                                                              02851600 T  0511:3
            IF T1<(-68) THEN T1←-68 ELSE IF DTAERR←T1>68 THEN GO TO COMM ;                               02851610 T  0514:3
            IF SGN THEN WH1 ← - WH1;                                                                     02851700 T  0519:3
            IF T1 > 0 THEN                                                                               02851800 T  0521:2
             DOUBLE(WH1,WH2,TEN[ T1],TEN[69+T1],×,←,W1[0],W1[1])                                         02851900 T  0522:1
                   ELSE                                                                                  02852000 T  0526:0
             DOUBLE(WH1,WH2,TEN[-T1],TEN[69-T1],/,←,W1[0],W1[1]);                                        02852100 T  0526:3
        COMM:                                                                                            02852200 T  0531:2
END CONVERT;                                                                                             02852300 T  0531:2
COMMENT * * * * * * * * * * * END OF DECLARATIONS * * * * * * * * * * * ;                                02862200 T  0531:3
IF EDITCODE≠1 AND EDITCODE≠3 THEN                                                                        02862210 T  0531:3
    BEGIN P(MKS) ;                                                                                       02862220 T  0544:0
    IF EDITCODE≠6 THEN P(FILX,DKADR);  P(FI,FMTA,*P(,LISX)) ;                                            02862230 T  0544:3
    IF EDITCODE=4 THEN P(EOFL,INTCALL(PARL,@154))                                                        02862250 T  0547:3
    ELSE P(EDITCODE,EOFL,INTCALL(PARL,@160)) ;                                                           02862260 T  0551:0
    P(XIT) ;                                                                                             02862270 T  0553:3
    END ;                                                                                                02862300 T  0554:0
        FILX[NOT 4] ← EOFL;   FILX[NOT 3] ← PARL;                                                        02862310 T  0554:0
FIB ← FILX[NOT 2];      % OPEN FILE IF NOT OPEN                                                          02862400 T  0557:2
IF FIB[5],[43:2] ≠ (T1 ← 2 + (EDITCODE=5)) THEN                                                          02862500 T  0559:1
    P(MKS,0,T1,FILX,1,SELECT);                                                                           02862600 T  0562:1
        CKPB; ARRAYSTUFF←0;                                                                              02862705 T  0564:1
        IF FIB[0] = 0 THEN                                                                               02862706 T  0565:3
         FIB[0] ← 1 + (EDITCODE =0 OR EDITCODE =2)                                                       02862708 T  0566:3
        ELSE                                                                                             02862710 T  0569:1
         IF FIB[0] ≠1 + (EDITCODE =0 OR EDITCODE = 2)                                                    02862712 T  0570:2
         THEN P(MKS,FIB[6],FILX,[33:15],4,FORTERR);                                                      02862714 T  0573:0
IF EDITCODE=1 THEN GO FNOL;  GO FMTLST ;                                                                 02862800 T  0576:2
    FNOL:                                                                                                02863900 T  0578:1
            LSTRN←-1;                                                                                    02864000 T  0578:1
            GO TO FRMTCD;                                                                                02864100 T  0579:1
    FMTLST:                                                                                              02882900 T  0579:3
        LSTRN ← 1;                                                                                       02883000 T  0579:3
            CTOG ← DONETOG ← FALSE;                                                                      02883100 T  0580:2
            GETLIST;                                                                                     02883200 T  0581:3
            GO TO FRMTCD;                                                                                02883300 T  0583:0
    MON:                                                                                                 02883400 T  0583:2
    FRMTCD:                                                                                              02883500 T  0583:2
            PS ← 0;                                                                                      02883600 T  0583:2
            NFCI ← (FI×8) + 2;          % FIRST FORMAT CHARACTER                                         02883700 T  0584:1
            IF NFC ≠ "(" THEN GO TO FMERR;                                                               02883800 T  0586:0
            NFCI ← (FI×8) + 2;          % FIRST FORMAT CHARACTER                                         02883900 T  0588:0
    NFPH:   FORMATCONTROL; % ANAYLSIS OF FORMAT STATEMENT                                                02884000 T  0589:3
            IF FMERRTOG THEN GO TO FMERR;                                                                02884100 T  0591:0
    FMCYC:  IF (DONETOG ← ENDLIST) THEN READS;                                                           02884200 T  0592:0
            IF W + NCR > LCR THEN GO TO FMERR;                                                           02884300 T  0595:0
            NCR ← W + NCR;                                                                               02884500 T  0596:3
            CONVERT;                                                                                     02884600 T  0598:0
            IF DTAERR THEN GO TO TYPERR;                                                                 02884700 T  0599:0
            GETLIST;                                                                                     02884800 T  0600:0
```

```
            IF (RPT←RPT-1) > 0 THEN GO TO FMCYC;                                  02884900 T  0601:0
            GO TO NFPH;                                                           02885000 T  0603:1
FMERR:                                                                            02885100 T  0603:3
     P(MKS,FIB[6],FILX,[33:15],0,FORTERR);                                        02885110 T  0603:3
TYPERR:                                                                           02885400 T  0605:3
     P(MKS,FIB[6],FILX,[33:15],2,FORTERR);                                        02885500 T  0605:3
END FTINT;                                                                        02885800 T  0607:3
                                                                                          SIZE= 0608 WORDS
```

```
PROCEDURE FTOUTFIX(FILX,DKADDR,FI,FMT,LISX,EDITCODE,EOFL,PARL); %INT@157          02886000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00350
VALUE DKADDR,FI,LISX,EDITCODE,EOFL,PARL ;                                         02886040 T  0000:0
ARRAY FMT[*];  NAME FILX;  REAL DKADDR,FI,LISX,EDITCODE,PARL,EOFL ;               02886080 T  0000:0
    BEGIN                                                                         02886120 T  0000:0
                                                                                  02886160 T  0000:0
    INTEGER LSTRN=19, E=17 ;                                                      02886200 T  0000:0
                                                                                  02886240 T  0000:0
    REAL LISTYPE=20, ARRAYSTUFF=18, ALGOLWRITE=12, SELECT=14,                     02886280 T  0000:0
         FORTERR=24, CHR, MAXCHR, BSIZE, PRNTR, TYPE, INDX, SIZE, TWDT,           02886320 T  0000:0
         SGN, BUFF, T1, T2, T3, WH1, WH2, WH3, ARY, W, D, R, SAVW=EOFL,           02886360 T  0000:0
         E1=9, XTRA, C, FMTW, SAVBUFF, DBLPREC, CODE, T4, T5, SKP=PARL,           02886400 T  0000:0
         NEEDNEWLISTELEMENT, FLG, SCALE, T8, T6=18, SAVD, DECPT=20, ND,           02886440 T  0000:0
         COMMAS, DLRSGN, T21 ;                                                    02886480 T  0000:0
                                                                                  02886520 T  0000:0
    NAME LISTADDR ;                                                               02886560 T  0000:0
                                                                                  02886600 T  0000:0
    ARRAY TEN=22[*], AR1=LISTADDR[*], TPAR=23[*], FPB=3[*], FIB[*] ;              02886640 T  0000:0
                                                                                  02886680 T  0000:0
    LABEL ALIST, GETNEXTPHRASE, REPEAT, TT, XX, SS, PP, AA, A1, OO, HH,           02886720 T  0000:0
          CC, ERROR, GG, LL, FF, EE, II, DD, TEST, TEST1, AWAY, OVRFLW,           02886760 T  0000:0
          BUMPWH3, MAXI, LOG8, THREH, THREL, HLF, CONVERT, D1, OVRFLW1,           02886800 T  0000:0
          FIVPT, JJ, RAPUP, X1, OVRFLW2, ONE, OUTSUB, CD, NLEL, F094,             02886840 T  0000:0
          F095, VERROR, HV, CD1, CMSK, REPEAT1, IEDIT, TEN11, ONDG, CKH,          02886845 T  0000:0
          STNRD, SE, TWHLF, DREST, DREST1, HLF1, FIVPT1, SQN, OVRFLW3,            02886850 T  0000:0
          TEST2, REPEAT2, STNRD1, XPIV, GOTE, NK ;                                02886855 T  0000:0
                                                                                  02886880 T  0000:0
    SWITCH PHRASE←SS,HH,PP,XX,TT,AA,OO,LL,JJ,II,GG,FF,EE,DD,CC ;                  02886920 T  0000:0
                                                                                  02886960 T  0000:0
    DEFINE DONE = LSTRN=(-1) #,                                                   02887000 T  0000:0
           REEL = 3 #,                                                            02887040 T  0000:0
           LOGICAL = 4 #,                                                         02887080 T  0000:0
           INTEGR = 1 #,                                                          02887120 T  0000:0
           DBLPRECSN = 5 #,                                                       02887160 T  0000:0
           COMPLEX = 6 #,                                                         02887200 T  0000:0
           MAXCODE = 15 #,                                                        02887210 T  0000:0
           VERR(VERR1) = BEGIN P(VERR1); GO VERROR END #,                         02887215 T  0000:0
           MAYBE(MAYBE1,MAYBE2,MAYBE3) = CI←CI+MAYBE1; GO TO MAYBE2 ;             02887240 T  0000:0
                                         DS←LIT MAYBE3; MAYBE2: #,                02887280 T  0000:0
           TWOD = LISTYPE,[38:1] #,                                               02887320 T  0000:0
           INDXF = [18:15] #,                                                     02887360 T  0000:0
           TYPEF = [44:4] #,                                                      02887400 T  0000:0
           SIZEF = [33:15] # ;                                                    02887440 T  0000:0
                                                                                  02887480 T  0000:0
    SUBROUTINE BLANKIT ;                                                          02887482 T  0000:0
```

```
          BEGIN                                                                                  02887484 T  0001:0
          STREAM(A←BSIZE-1,B←P(DUP).[36:6],T21,BUFF) ;                                           02887486 T  0001:0
               BEGIN                                                                             02887488 T  0003:2
               SI←T21; DS←WDS; SI←BUFF; DS←A WDS; B(DS←32WDS; DS←32WDS) ;                        02887490 T  0003:2
               END ;                                                                             02887492 T  0006:0
          END OF BLANKIT ;                                                                       02887496 T  0006:1
                                                                                                 02887498 T  0006:2
     SUBROUTINE OUTPUT ;                                                                         02887520 T  0006:2
          BEGIN                                                                                  02887560 T  0007:0
          IF PRNTR THEN                                                                          02887600 T  0007:0
               BEGIN                                                                             02887640 T  0007:1
               STREAM(Q←0:SAVBUFF) ;                                                             02887680 T  0007:3
                    BEGIN DI←LOC Q; SI←SAVBUFF; DI←DI+7; DS←CHR END ;                            02887720 T  0009:0
               T1←IF (T1←P)="+" THEN 0 ELSE IF T1>9 THEN 16                                      02887760 T  0010:1
                                       ELSE IF T1=0 THEN 32 ELSE T1;                             02887800 T  0013:1
               IF NOT C THEN FIB[17]←*P(DUP)+BSIZE ;                                             02887840 T  0016:2
               P(MKS,T1.[42:2],T1 AND 15,C,BSIZE,FILX,ALGOLWRITE) ;                              02887880 T  0020:1
               FIB[6]←*P(DUP)-((C←0)=T1) ;                                                       02887920 T  0023:0
               P(MKS,FLG,DKADDR,0,(-1),FILX,ALGOLWRITE,DEL) ;                                    02888160 T  0026:0
               STREAM(Q←BUFF←SAVBUFF,BSIZE,BSZ←P(DUP)-1,T21,S←*FILX) ;                            02888200 T  0028:1
                    BEGIN                                                                        02888240 T  0031:1
                    SI←Q; SI←SI+1; DS←BSIZE WDS; DI←Q; SI←T21; DS←9CHR ;                          02888280 T  0031:1
                    SI←Q; SI←SI+1; DS←BSZ WDS ;                                                  02888320 T  0033:0
                    END ;                                                                        02888360 T  0034:0
               FIB[17]←*P(DUP)-BSIZE ;                                                           02888400 T  0034:1
               END                                                                               02888440 T  0036:1
          ELSE BEGIN                                                                             02888480 T  0036:1
               P(MKS,FLG,DKADDR,0,BSIZE,FILX,ALGOLWRITE) ;                                       02888520 T  0036:3
               IF LSTRN≠(-1) THEN                                                                02888560 T  0038:2
                    BEGIN                                                                        02888600 T  0039:2
                    P(MKS,FLG,DKADDR,0,(-1),FILX,ALGOLWRITE,DEL) ;                               02888640 T  0040:0
                    BUFF←SAVBUFF←(*FILX).[33:15];  BLANKIT ;                                     02888680 T  0042:1
                    END ;                                                                        02888720 T  0045:0
               END ;                                                                             02888760 T  0045:0
          CHR←0 ;                                                                                02888800 T  0045:0
          END OF OUTPUT ;                                                                        02888840 T  0045:3
                                                                                                 02888880 T  0046:0
     SUBROUTINE SKIP ;                                                                           02888920 T  0046:0
          IF (T1←P(XCH)) GEQ W THEN T1←W                                                         02888960 T  0046:0
          ELSE BEGIN                                                                             02889000 T  0047:3
               STREAM(T21:Q←W-T1,T←P(DUP),[36:6],BUFF) ;                                         02889040 T  0049:0
                    BEGIN                                                                        02889080 T  0051:3
                    SI←T21; DS←Q CHR; T(SI←T21; DS←32CHR; DS←32CHR) ;                            02889120 T  0051:3
                    T21←DI ;                                                                     02889140 T  0054:0
                    END ;                                                                        02889160 T  0054:1
               BUFF←P ;                                                                          02889200 T  0054:2
               END OF SKIP ;                                                                     02889240 T  0055:0
                                                                                                 02889280 T  0055:1
     REAL SUBROUTINE NXTELM ;                                                                    02889320 T  0055:1
          BEGIN                                                                                  02889360 T  0056:0
          P(IF TWDT THEN P(*[AR1[INDX.[33:7]]],INDX AND 255,COC)                                 02889400 T  0056:0
            ELSE AR1[INDX]) ;                                                                    02889440 T  0059:0
          INDX←INDX+1;  NXTELM←P ;                                                               02889480 T  0060:0
          END OF NXTELM ;                                                                        02889520 T  0061:2
                                                                                                 02889560 T  0061:3
     SUBROUTINE GETNEXTLISTELEMENT ;                                                             02889600 T  0061:3
          BEGIN                                                                                  02889640 T  0062:0
```

```
          IF NEEDNEWLISTELEMENT THEN                                                    02889680 T  0062:0
               BEGIN                                                                    02889720 T  0062:1
               IF ARY THEN                                                              02889760 T  0062:3
                    BEGIN                                                               02889800 T  0063:0
ALIST:              P(NXTELM); IF DBLPREC THEN WH2←NXTELM; ARY←INDX<SIZE;               02889840 T  0063:2
                    END                                                                 02889880 T  0067:3
               ELSE IF TYPE=COMPLEX THEN                                                02889920 T  0068:3
                         BEGIN TYPE←REEL; P(LISTADDR[1]) END                            02889960 T  0070:0
                    ELSE BEGIN                                                          02890000 T  0072:1
                         P(ARRAYSTUFF←0);  LISTADDR←[LISX] ;                            02890040 T  0072:3
                         DBLPREC←(TYPE←LISTYPE,TYPEF)=DBLPRECSN ;                       02890080 T  0074:1
                         IF ARY←ARRAYSTUFF≠0 THEN                                       02890120 T  0076:2
                              BEGIN                                                     02890160 T  0077:3
                              IF TYPE=COMPLEX THEN TYPE←REEL ;                          02890200 T  0078:1
                              SIZE←(INDX←ARRAYSTUFF,INDXF)+                             02890240 T  0080:1
                                   ARRAYSTUFF,SIZEF ;                                   02890280 T  0081:2
                              P(LISTADDR←MEM[LISTADDR,[18:15]]) ;                       02890320 T  0083:0
                              TWDT←NOT P(LOD,TOP);  P(DEL) ;                            02890360 T  0085:0
                              GO ALIST ;                                                02890400 T  0086:2
                              END ;                                                     02890480 T  0087:0
                         P(DEL,LISTADDR[0]) ;                                           02890520 T  0087:0
                         IF DBLPREC THEN WH2←LISTADDR[1] ;                              02890560 T  0087:2
                         END ;                                                          02890600 T  0089:3
               T5←WH1←P ;                                                               02890640 T  0089:3
               END ;                                                                    02890680 T  0090:3
          IF (NEEDNEWLISTELEMENT←1)=EDITCODE OR DONE THEN                               02890720 T  0090:3
AWAY:          BEGIN OUTPUT; P(XIT) END ;                                               02890725 T  0093:1
          END OF GETNEXTLISTELEMENT ;                                                   02890730 T  0095:1
                                                                                        02890735 T  0095:2
NLE     SUBROUTINE NLE ;                                                                02890740 T  0095:2
next list element    BEGIN P(XCH); WH2←0; GETNEXTLISTELEMENT ;                          02890745 T  0096:0
               IF WH1+4>P(F094) THEN                                                    02890750 T  0098:0
                    BEGIN IF T1 THEN VERR(P+10); P(DEL,F094) END                        02890755 T  0099:1
               ELSE IF P(DEL,(-P(F094)),DUP)<WH1 THEN P(DEL,WH1) ;                      02890760 T  0102:0
               P(XCH) ;                                                                 02890765 T  0105:0
               END OF NLE ;                                                             02890770 T  0105:1
                                                                                        02890775 T  0105:2
          SUBROUTINE HANDLEVARIABLES ;                                                  02890780 T  0105:2
               BEGIN T1←1 ;                                                             02890785 T  0106:0
               IF R=P(F095) THEN                                                        02890790 T  0106:3
                    BEGIN P(0); NLE; T1←P(,R,ISN)>0 ;                                   02890795 T  0107:2
                    IF CODE=29 THEN                                                     02890800 T  0110:2
                         BEGIN P(FI+SAVW) ;                                             02890805 T  0111:1
                         IF R≥0 THEN P([FMT[P]],DUP,LOD,P&R[6:36:12],XCH)               02890810 T  0112:2
                         ELSE P(,FI) ;                                                  02890815 T  0115:3
                         P(STN) ;                                                       02890820 T  0116:2
OUTSUB:                  P(DEL,DEL);  GO GETNEXTPHRASE ;                            02890825 T  0116:3
                         END ;                                                          02890830 T  0117:3
                    END ;                                                               02890835 T  0117:3
               IF T4←CODE=30 THEN                                                       02890840 T  0117:3
                    BEGIN P(2); NLE; P(,ND,ISN) ;                                       02890845 T  0119:0
                    STREAM(P1←P;P2←P(CD),P3←P(CD1)) ;                                   02890850 T  0121:2
                         BEGIN SI←LOC P1; SI←SI+7; DI←LOC P2; DI←DI+1 ;                 02890855 T  0122:3
                         32(IF SC=DC THEN JUMP OUT; TALLY←TALLY+1; SI←SI-1) ;           02890860 T  0123:3
                         P1←TALLY ;                                                     02890865 T  0125:3
                         END ;                                                          02890870 T  0126:0
                    IF (ND AND 63)≠ND THEN P(DEL,32) ;                                  02890875 T  0126:1
```

```
                    IF (CODE←P+3)>MAXCODE AND T1 THEN VERR(2) ;                                  02890880 T  0128:2
                    T1←CODE>4 AND T1 ;                                                           02890885 T  0131:3
                    END ;                                                                        02890890 T  0133:2
               T2←T1 ;                                                                           02890895 T  0133:2
               IF P(CODE≥11 AND CODE≤14,F095)=SAVW THEN                                          02890900 T  0134:1
                    BEGIN P(,SAVW,4) ;                                                           02890905 T  0136:3
NLEL:               NLE; P(XCH,ISD); T1←P(DUP) AND T2←T2 AND SAVW>0 ;                             02890910 T  0137:3
                    END ;                                                                        02890915 T  0142:1
               IF SAVD=P(F095) THEN BEGIN P(,SAVD,6); GO NLEL END ;                              02890920 T  0142:1
               IF CODE≤4 THEN                                                                    02890925 T  0144:2
                    BEGIN IF T4 THEN SAVW←R;                                                     02890930 T  0145:1
                    FMTW←FMTW&(P(DUP),[41:1]+(SAVW<0))[41:47:1];  GO HV ;                        02890935 T  0147:1
                    END ;                                                                        02890940 T  0151:0
               IF NOT T2 THEN GO OUTSUB;  IF CODE=5 THEN HV: R←1 ;                               02890945 T  0151:0
               IF P(DUP) AND SAVD<0 THEN VERR(16) ;                                              02890950 T  0153:3
               IF T4 THEN IF SAVW=P(F094) THEN BEGIN IF CODE≠9 THEN VERR(6)END                   02890955 T  0156:1
                         ELSE IF P(DUP) AND SAVD=P(F094) THEN                                    02890960 T  0160:1
                              BEGIN P(7) ;                                                       02890965 T  0162:0
VERROR:                       T4←P;  P(MKS,CODE,R,SAVW,SAVD,T4,WH1,WH2,FMTW,                    02890970 T  0162:3
                                   (-5),FORTERR) ;                                           02890975 T  0165:2
F094:::  4094 ;                                                                                  02890980 T  0166:1
F095:::  4095 ;                                                                                  02890985 T  0168:0
CD:::    @0047676321464341 ;   % OPXTAOLJ                                                       02890990 T  0169:0
CD1:::   @3127262524230000 ;   % IGFEDCOO                                                       02890995 T  0170:0
                         END ;                                                                   02891000 T  0171:0
               IF NOT P THEN SAVD←0 ;                                                            02891005 T  0171:0
               END OF HANDLEVARIABLES ;                                                          02891010 T  0172:2
                                                                                                 02891015 T  0172:3
          REAL SUBROUTINE SETUP ;                                                                02891020 T  0172:3
               BEGIN                                                                             02891025 T  0173:0
               P(XCH,DUP) ;                                                                      02891030 T  0173:0
               IF DBLPREC THEN                                                                   02891035 T  0173:2
                    BEGIN                                                                        02891040 T  0173:3
                    IF P>ND THEN BEGIN T6←P-ND; P(ND) END ;                                      02891080 T  0174:1
                    IF (T5←(T4←P)-T3←ND-16)<0 THEN                                               02891120 T  0176:2
                         BEGIN P(WH3/TEN[-T5],,WH3,ISD); T3←T4 END                               02891160 T  0179:2
                    ELSE IF T5 LSS 8 THEN                                                        02891200 T  0182:2
                         BEGIN                                                                   02891240 T  0183:3
                         IF P(WH2/TEN[8-T2←T5],,WH2,ISN)=TEN[T2] THEN                            02891280 T  0184:1
BUMPWH3:                 WH3←WH3+1 ;                                                             02891320 T  0187:2
                         END                                                                     02891360 T  0189:1
                    ELSE IF P(WH1/TEN[16-T5],,WH1,ISN)=TEN[T1←T5-T2←8]                           02891400 T  0189:1
                         THEN IF (WH2←WH2+1)=T8 THEN GO BUMPWH3 ;                                02891440 T  0193:2
                    END                                                                          02891480 T  0196:3
               ELSE BEGIN                                                                        02891520 T  0196:3
                    IF (T3←P)>11 THEN T6←T3-T3←(P(WH1,TEN[ABS(E)],IF E>0                         02891560 T  0197:1
                                   THEN P(/) ELSE P(×))≤P(FIVPT))+11 ;                           02891600 T  0200:1
                    IF CODE=12 THEN P(SCALE,+) ;                                                 02891640 T  0204:2
                    P(P-E1-T6,WH3,XCH,TEN[ABS(P(DUP))],IF P(XCH)<0 THEN P(/)                     02891680 T  0206:1
                                                       ELSE P(×),,WH3,ISD) ;                     02891720 T  0210:0
                    END ;                                                                        02891760 T  0211:1
               E1←P(TEN[T3]=WH3,DUP)+E1;  SETUP←P ;                                              02891800 T  0211:1
               END OF SETUP ;                                                                    02891840 T  0213:3
                                                                                                 02891880 T  0214:0
%************************:: CODE STARTS HERE ::************************%                        02891920 T  0214:0
                                                                                                 02891960 T  0214:0
     FIB←FILX[NOT 2];  P(TEN[8],,T8,ISD) ;                                                       02892000 T  0214:0
```

```
          IF FLG←DKADDR<0 THEN DKADDR←0 ;                                                    02892040 T  0226:2
          IF P(FIB[5],DUP).[43:1] THEN P(MKS,0,0,FILX,1,SELECT) ;                            02892080 T  0229:0
          C←(P AND 96)≠0 ;                                                                   02892120 T  0232:1
          MAXCHR←(BSIZE←P(MKS,FLG,DKADDR,0,(-1),FILX,ALGOLWRITE))×8+                         02892160 T  0233:3
                PRNTR←((T1←FIB[4].[8:4])=1 OR T1=12 OR T1=7) AND                             02892200 T  0236:3
                      FPB[FIB[4].[13:11]+3].[43:5]<20 ;                                      02892205 T  0240:3
          IF NOT(NOT(NEEDNEWLISTELEMENT←EDITCODE=3) OR FMT[FI]) THEN GO ERROR;               02892240 T  0245:0
          IF NOT TPAR.[14:1] THEN                                                            02892250 T  0247:2
               BEGIN                                                                         02892255 T  0249:0
               E←P(1,[E],CFX,SFB)&29[8:38:10] ;                                              02892260 T  0249:2
               STREAM(A←P(21,[E])); BEGIN DS←8LIT" "; SI←A; DS←7WDS END ;                     02892265 T  0252:0
               P(TPAR,1,25,COM,DEL,DEL);  E←0 ;                                              02892270 T  0255:0
               END ;                                                                         02892275 T  0257:2
          T21←P([TPAR[21]]) INX 0 ;                                                          02892276 T  0257:2
          IF PRNTR THEN                                                                      02892280 T  0259:0
               BEGIN                                                                         02892320 T  0259:1
               IF BSIZE>16 THEN BEGIN BSIZE←17; MAXCHR←133 END ;                             02892360 T  0259:3
               IF C THEN BEGIN BUFF←TPAR INX 1; TPAR[0]←" "; BLANKIT END ;                   02892440 T  0262:2
               P(P(CMSK) OR TPAR) ;                                                          02892445 T  0267:0
               END                                                                           02892480 T  0268:0
          ELSE P(P(*FILX),[33:15]) ;                                                         02892520 T  0268:0
          BUFF←SAVBUFF←P ;                                                                   02892560 T  0269:2
          IF NOT PRNTR THEN BLANKIT ;                                                        02892565 T  0270:2
          IF FIB[0]=0 THEN FIB[0]←1 ;                                                        02892600 T  0273:0
          IF (LSTRN←1)≠FIB[0] AND T1=2 THEN                                                  02892640 T  0275:3
               BEGIN  T3←4 ;                                                                 02892680 T  0278:1
ERROR:    P(MKS,FIB[7],FILX,[33:15],T3,FORTERR) ;                                            02892720 T  0279:2
               END ;                                                                         02892760 T  0281:2
          P(0) ;                                                                             02892800 T  0281:2
GETNEXTPHRASE:                                                                               02892840 T  0281:3
          R←P(FMT[FI←FI+1],DUP).[6:12];  IF (CODE←P(DUP).[1:5])=2 THEN GO HH ;               02892880 T  0281:3
          SAVW←P(DUP).[18:12];  SAVD←(FMTW←P(DUP)).[30:12] ;                                  02892885 T  0286:1
          IF (XTRA←P(DUP) AND 63).[44:2]=0 THEN P(0,0)                                       02892890 T  0289:3
          ELSE P(P((D←P(DUP) AND 15)=12,DUP) OR D=8,P(XCH) OR D=4) ;                          02892895 T  0293:0
          DLRSGN←P;  COMMAS←P ;                                                              02892900 T  0297:3
          IF P.[42:1] THEN IF (FMTW AND 3)=0 THEN HANDLEVARIABLES ;                          02892905 T  0298:3
          IF CODE=0 THEN                                                                     02892920 T  0303:0
               BEGIN                                                                         02892960 T  0303:3
               IF SAVD≠0 THEN                                                                02893000 T  0304:1
                    BEGIN GETNEXTLISTELEMENT; OUTPUT; NEEDNEWLISTELEMENT←0END;               02893005 T  0305:0
               IF P(DUP).[18:15]≠FI THEN P(R&FI[18:33:15]) ;                                 02893040 T  0308:3
               IF P((NOT 0),XCH,INX,DUP).[33:15]=0 THEN P(DEL)ELSE FI←FI-SAVW;               02893080 T  0311:2
               GO GETNEXTPHRASE ;                                                            02893120 T  0315:1
FIVPT::: 5.49755813885 ;                                                                     02893160 T  0316:3
               END ;                                                                         02893200 T  0318:0
          IF CODE=5 THEN CHR←R←0 ;                                                           02893240 T  0318:0
REPEAT:                                                                                      02893320 T  0320:2
          IF CODE>5 THEN                                                                     02893360 T  0320:2
REPEAT1: GETNEXTLISTELEMENT ;                                                                02893365 T  0321:1
REPEAT2:                                                                                     02893370 T  0323:0
          IF (CHR←(W←SAVW)+CHR)>MAXCHR THEN IF CODE≠3 AND CODE≠9 THEN GO AWAY;               02893375 T  0323:0
          IF CODE≥9 THEN IF CODE≤14 THEN                                                     02893380 T  0327:2
               BEGIN                                                                         02893385 T  0330:1
               SGN←WH1.[1:1];  DECPT←CODE>10 ;                                               02893390 T  0330:3
               IF CODE<13 THEN                                                               02893395 T  0333:1
                    IF ABS(WH1)<P(TEN11) AND NOT CODE THEN                                   02893400 T  0334:0
                         IF W<64 AND NOT(COMMAS OR DLRSGN OR DBLPREC) THEN                   02893405 T  0336:1
```

```
BEGIN                                                                  02893410 T  0339:1
IF NOT DECPT THEN                                                      02893415 T  0339:3
     BEGIN                                                             02893420 T  0340:1
     IF P(E1←W,ABS(WH1),,WH2,ISN)>9                                    02893425 T  0340:3
     THEN GO IEDIT ;                                                   02893430 T  0342:2
     IF P(SGN+1,"-",DUP)≥0 THEN GO ONDG ;                              02893435 T  0343:2
     GO OVRFLW1 ;                                                      02893440 T  0345:3
     END ;                                                             02893445 T  0346:1
IF (T1←11-D←SAVD)<0 THEN GO STNRD1 ;                                   02893450 T  0346:1
IF (E1←W"D-1)<0 THEN GO OVRFLW ;                                       02893455 T  0349:0
P(T6←0); IF WH1=0 THEN BEGIN WH2←0; GO CKH END;                        02893460 T  0351:3
P(TEN[D],DUP,ABS(WH1)) ;                                               02893465 T  0355:0
IF SCALE≠0 THEN                                                        02893470 T  0356:1
     BEGIN                                                             02893475 T  0357:0
     IF P(TEN[ABS(SCALE)],IF SCALE>0 THEN P(×)                         02893480 T  0357:2
     ELSE P(/),DUP)≥P(TEN11) THEN GO STNRD ;                           02893485 T  0359:3
     P(,WH1,STN) ;                                                     02893490 T  0361:3
     END ;                                                             02893495 T  0362:1
IF P(DUP,HLF1,"-",,WH2,ISN,"-",×,,T5,ISN)=P THEN                       02893500 T  0362:1
     BEGIN T5←0; WH2←WH2+1 END ;                                       02893505 T  0364:3
IF T5≠0 THEN                                                           02893510 T  0367:1
     BEGIN P(DEL,10) ;                                                 02893515 T  0368:0
     IF D>8 THEN                                                       02893520 T  0369:0
          BEGIN P(DEL,5) ;                                             02893525 T  0369:3
          T2←T5 DIV T8; D←D-8 ;                                        02893530 T  0370:3
          END ;                                                        02893535 T  0373:1
     END ;                                                             02893540 T  0373:1
IF WH2<10 THEN                                                         02893545 T  0373:1
     BEGIN                                                             02893550 T  0374:0
CKH:                                                                   02893555 T  0374:2
          IF P(E1-SGN-1,DUP)<0 THEN                                    02893560 T  0376:2
               BEGIN                                                   02893565 T  0377:0
               IF WH2≠0 THEN GO OVRFLW2 ;                              02893570 T  0378:1
               P(DEL) ;                                                02893575 T  0378:2
               IF E1=0 THEN                                            02893580 T  0379:1
                    BEGIN                                              02893585 T  0379:3
                    IF SGN THEN GO OVRFLW1                             02893590 T  0380:0
                    ELSE GO DREST1 ;
CMSK:::  @700000 ;                                                     02893595 T  0381:0
HLF1:::  0.5 ;                                                         02893600 T  0382:0
TEN11::: 99999999999.0 ;                                               02893605 T  0383:0
                    END ;                                              02893610 T  0384:0
               P(SGN←0); WH2←"-" ;                                     02893615 T  0384:0
               END ;                                                   02893620 T  0385:2
ONDG:          STREAM(S←P:T21,WH2,SGN,BUFF) ;                          02893625 T  0385:2
               BEGIN                                                   02893630 T  0387:1
               SI←T21; DS←S CHR; MAYBE(SGN,L1,"-") ;                   02893635 T  0387:1
               SI←LOC SGN; SI←SI-1; DS←CHR; S←DI ;                     02893640 T  0389:1
               END ;                                                   02893645 T  0390:1
          IF NOT DECPT THEN GO TEST ;                                  02893650 T  0390:2
          BUFF←P; P(WH2≠0) ;                                           02893655 T  0391:1
DREST:    IF (E←P)>T1 THEN                                             02893660 T  0392:2
             T6←E-T1-(ABS(WH1)≤TEN[E-1]×P(FIVPT1)) ;                   02893665 T  0393:2
DREST1:   STREAM(Q←P:D,T2,T5,T6,BUFF) ;                                02893670 T  0397:3
               BEGIN                                                   02893675 T  0399:3
               DS←LIT"."; CI←CI+Q; SI←LOC T5;SI←SI+2;                   02893680 T  0399:3
               DS←D CHR; GO L2; SI←LOC T2; DS←D DEC ;                  02893685 T  0401:1
               DS←8DEC; GO L2; SI←LOC T5; DS←D DEC ;                   02893690 T  0402:3
```

```
                    L2: Q←DI ;                                                  02893695 T  0404:0
                    T6(DI←DI-T6;T6(DS←LIT"0"); JUMP OUT);                       02893700 P  0404:1
                    END ;                                                       02893705 T  0407:1
               GO TEST ;                                                        02893710 T  0407:2
               END ;                                                            02893715 T  0408:0
IEDIT:         IF P(E1-SGN,DUP)<9 THEN                                          02893720 T  0408:0
               BEGIN                                                            02893725 T  0409:2
               STREAM(E1←P,WH2,SGN:BUFF) ;                                      02893730 T  0410:0
                    BEGIN SI←LOC WH2; CI←CI+SGN; GO L1 ;                        02893735 T  0411:2
                    DS←LIT"0"; DS←E1 DEC; E1←DI; DI←BUFF ;                      02893740 T  0412:2
                    IF TOGGLE THEN TALLY←1; DS←8 FILL ;                         02893745 T  0414:0
                    DI←DI-1; DS←LIT"-"; GO L2; L1:                              02893750 T  0414:3
                    DS←E1 DEC; IF TOGGLE THEN TALLY←1 ;                         02893755 T  0415:3
                    E1←DI; DI←BUFF; DS←8FILL; L2: WH2←DI ;                      02893760 T  0416:3
                    SGN←TALLY ;                                                 02893765 T  0417:3
                    END ;                                                       02893770 T  0418:0
               IF P THEN GO SQN ELSE GO OVRFLW3 ;                               02893775 T  0418:1
               END ;                                                            02893780 T  0419:1
               IF WH2<T8 THEN                                                   02893785 T  0419:1
               BEGIN                                                            02893790 T  0420:0
               P(DEL) ;                                                         02893795 T  0420:2
               STREAM(WH2,S←E1-8:T21,SGN,BUFF) ;                                02893800 T  0420:3
                    BEGIN                                                       02893805 T  0423:1
                    SI←T21; DS←S CHR; S←DI; SI←LOC WH2 ;                        02893810 T  0423:1
                    DS←8DEC; WH2←DI; DI←S; DS←8FILL; S←DI;                      02893815 T  0424:2
                    CI←CI+SGN; GO L1; DI←DI-1; DS←LIT"-" ;                      02893820 T  0425:3
                    L1:                                                         02893825 T  0427:1
                    END ;                                                       02893830 T  0427:1
                    GO SQN ;                                                    02893835 T  0427:2
                    END ;                                                       02893840 T  0428:0
               E1←P ;                                                           02893845 T  0428:0
               STREAM(WH1←WH2 DIV T8,WH2,T21:S←IF E1>16 THEN P(                 02893850 T  0428:2
                      E1-16,8) ELSE P(0,E1-8),E1←P,SGN,BUFF) ;                  02893855 T  0431:2
                    BEGIN                                                       02893860 T  0434:3
                    SI←T21; DS←S CHR; DS←SGN CHR; S←DI ;                        02893865 T  0434:3
                    SI←LOC WH1; DS←E1 DEC; IF TOGGLE THEN                       02893870 T  0436:1
                    TALLY←1; DS←8DEC; WH1←DI; T21←TALLY; DI←S ;                 02893875 T  0437:1
                    DS←8FILL; WH2←DI; CI←CI+SGN; GO L1 ;                        02893880 T  0438:2
                    DI←DI-1; DS←LIT"-"; L1:                                     02893885 T  0439:3
                    END ;                                                       02893890 T  0440:2
               IF NOT P THEN GO OVRFLW3 ;                                       02893895 T  0440:3
SQN:           IF NOT DECPT THEN                                                02893900 T  0441:1
                    BEGIN P(DEL,XCH,DEL); GO TEST END ;                         02893905 T  0441:3
               E1←P;  BUFF←P ;                                                  02893910 T  0443:2
               IF W<13 THEN GO DREST1 ;                                         02893915 T  0444:2
               P(0&P((1+BUFF) INX (NOT E1))[43:46:2]                            02893920 T  0445:3
                 +BUFF,[30:3]-E1,[30:3]) ;                                      02893925 T  0447:2
               GO DREST ;                                                       02893930 T  0450:1
FIVPT1:::5,49755813885 ;                                                        02893935 T  0450:3
STNRD:         P(DEL,DEL,DEL,DEL) ;                                             02893940 T  0452:0
               END ;                                                            02893945 T  0453:0
          D←SAVD ;                                                              02893950 T  0453:0
STNRD1:   P(XPIV,WH1=0) ;                                                       02893955 T  0453:3
          IF NOT DBLPREC THEN                                                   02893960 T  0454:3
               BEGIN                                                            02893965 T  0455:1
               IF P THEN GO GOTE ;                                              02893970 T  0455:3
               P(TEN[ABS(E←(P&(WH1←ABS(WH1) MOD P(MAXI))[9:3:6]&                02893975 T  0456:2
```

```
                    WH1[1:2:1]+P(TWHLF))×P(LOG8))],WH1) ;                                02893980 T  0458:3
               IF E<0 THEN P(x,ONE,XCH) ;                                                02893985 T  0462:0
               IF P(>) THEN                                                              02893990 T  0464:0
                    BEGIN P(E-1) ;                                                       02893995 T  0464:1
GOTE:               E←P ;                                                              02894000 T  0465:2
                    END ;                                                                02894005 T  0466:0
               IF CODE=13 THEN                                                           02894010 T  0466:0
                    IF NOT (WH1≠0 AND (DLRSGN OR D>16 OR D+SCALE>11))                    02894015 T  0466:3
                    THEN GO SE ;                                                         02894020 T  0470:3
               ND←12;  WH3←WH1 ;                                                         02894025 T  0471:2
               END                                                                       02894030 T  0473:0
          ELSE IF P AND WH2=0 THEN BEGIN WH3←E←P; ND←24 END ELSE                         02894035 T  0473:0
               BEGIN                                                                     02894040 T  0476:3
               P(WH1←ABS(WH1)) ;                                                         02894045 T  0477:1
               IF (P AND P(NK))=0 THEN WH2←P(0,ONE,WH2,WH1,DLM,,WH1,←) ;                 02894050 T  0478:1
               IF (E←(P&(WH3←WH1)[9:3:6]&WH1[1:2:1]+P(TWHLF))×P(LOG8))<0                  02894055 T  0482:0
                    THEN P(0,ONE,TEN[69-E],TEN[-E],DLD)                                  02894080 T  0486:0
               ELSE P(TEN[E+69],TEN[E]) ;                                                02894120 T  0489:2
               T1←P;  IF (P>WH2 AND T1=WH1) OR T1>WH1 THEN E←E-1 ;                        02894160 T  0491:2
               P(WH2,WH1,TEN[69+ABS(E)],TEN[ABS(E)],IF 0>E THEN                          02894200 T  0496:1
                 P(DLM) ELSE P(DLD)) ;                                                   02894240 T  0499:2
               T1←P;  T3←P;  P(24) ;                                                     02894280 T  0501:0
               IF T1≥P(THREH) THEN P(T1>P(THREH) OR T3>P(THREL),-) ;                      02894320 T  0502:1
               ND←P ;                                                                    02894360 T  0505:2
               IF ND≥70+E THEN P(WH2,WH1,TEN[ND+69],TEN[ND],DLM,                         02894400 T  0506:0
                                 TEN[68-E],TEN[-E-1],DLM)                                02894410 T  0510:0
               ELSE P(WH2,WH1,TEN[(T1←ABS(ND-E-1))+69],TEN[T1],IF ND≤E                   02894425 T  0512:2
                    THEN P(DLD) ELSE P(DLM)) ;                                           02894440 T  0517:0
               WH1←P ;                                                                   02894480 T  0519:0
               P(T3←P,WH1,T6←TEN[85],T4←TEN[16],DLD,HLF,-,,WH3,ISD,DEL,                   02894520 T  0519:2
                 T3,WH1,0,WH3,T6,T4,DLM,DLS) ;                                           02894560 T  0523:3
               WH1←P ;                                                                   02894600 T  0525:3
               P(T3←P,WH1,0,T8,DLD,HLF,-,,WH2,ISD,DEL,T3,WH1,                            02894640 T  0526:1
                 0,WH2,0,T8,DLM,DLS,,WH1,ISD,DEL) ;                                      02894680 T  0529:2
               END ;                                                                     02894720 T  0531:3
          IF T4←(T1←T2←T3←T6←0)=WH3 THEN P((-ND))                                        02894760 T  0531:3
          ELSE BEGIN P(E+1); IF CODE=12 THEN P(SCALE,+) END ;                            02894800 T  0536:0
          E1←P ;                                                                         02894840 T  0539:0
          END ;                                                                          02894880 T  0539:2
     GO PHRASE[CODE-1] ;                                                                 02894960 T  0539:2
HLF:::   0.5 ;                                                                           02894965 T  0548:2
MAXI:::  @0777777777777777 ;                                                             02894967 T  0550:0
ONE:::   1.0 ;                                                                           02895000 T  0551:0
XPIV:::  @1130000000000000 ;                                                             02895040 T  0552:0
TWHLF::: 12.5 ;                                                                          02895045 T  0553:0
NK:::    @0007000000000000 ;                                                             02895050 T  0554:0
THREH::: @1153013331500045 ;                                                             02895055 T  0555:0
THREL::: @0003112121167260 ;                                                             02895060 T  0556:0
LOG8:::  0.90308998709 ;                                                                 02895065 T  0557:0
SE: IF P(W-D-5-SGN,DUP)<0 THEN GO OVRFLW1 ;                                              02895070 T  0558:0
     IF P(DUP)>63 THEN BEGIN P(W,XCH,SUB,63,+); SKIP; P(63) END ;                        02895075 T  0561:0
     IF WH1=0 THEN                                                                       02895080 T  0565:1
          BEGIN                                                                          02895085 T  0566:0
          STREAM(SKP←P:T21,SGN,D←D+3,D1←P(DUP),[36:6],BUFF) ;                            02895090 T  0566:2
               BEGIN                                                                     02895095 T  0569:2
               SI←T21; DS←SKP CHR; MAYBE(SGN,L1,"-"); DS←2LIT",0" ;                      02895100 T  0569:2
               SI←T21; DS←D CHR; D1(SI←T21; DS←32CHR; DS←32CHR); SKP←DI ;                02895105 T  0572:0
```

```
          END ;                                                                                    02895110 T  0574:2
     GO TEST ;                                                                                     02895115 T  0574:3
     END ;                                                                                         02895120 T  0575:1
ND←E ;                                                                                             02895125 T  0575:1
IF SCALE≠0 THEN                                                                                    02895130 T  0576:0
     IF SCALE<0 THEN                                                                               02895135 T  0576:3
          BEGIN IF P(1-SCALE,DUP)>D THEN GO OVRFLW2 END                                            02895140 T  0578:0
     ELSE BEGIN                                                                                    02895145 T  0580:2
          IF P(SCALE,-,DUP)<0 THEN GO OVRFLW1 ;                                                    02895150 T  0581:0
          P(SKP←P); D←D+SCALE; E←E-SCALE; P(1) ;                                                   02895155 T  0582:3
          END                                                                                      02895160 T  0586:0
ELSE BEGIN IF D=0 THEN GO OVRFLW1; P(1) END ;                                                      02895165 T  0586:0
T3←P ;                                                                                             02895170 T  0588:0
IF P(WH1,TEN[ABS(P((E1←D-T3)-ND,DUP))],IF P(XCH)>0 THEN                                             02895175 T  0588:2
P(×) ELSE P(/),,T4,ISN)=TEN[E1+1] THEN                                                             02895180 T  0592:0
     BEGIN P(TEN[E1],,T4,ISD); E←E+1 END ;                                                         02895185 T  0595:1
P(5);  IF D>8 THEN BEGIN E1←T4 DIV T8; D←D-8; P(DEL,0) END ;                                        02895190 T  0598:0
STREAM(SKP←P,Q←P:D,E1,T4,E←ABS(E+T3),ES←E<(-T3),SGN,T21,BUFF) ;                                     02895195 T  0602:2
     BEGIN                                                                                         02895200 T  0606:3
     SI←T21; DS←SKP CHR; MAYBE(SGN,L1,"-"); DS←LIT","; CI←CI+Q;                                    02895205 T  0606:3
     SI←LOC E1; DS←D DEC; DS←8DEC; GO L2; SI←LOC T4; DS←D DEC ;                                    02895210 T  0609:3
     L2: DS←2LIT"E "; CI←CI+ES; GO L3; DI←DI-1; DS←LIT"-" ;                                        02895215 T  0611:3
     L3: DS←2DEC; SKP←DI ;                                                                         02895220 T  0613:3
     END ;                                                                                         02895225 T  0614:1
P(DEL) ;                                                                                           02895230 T  0614:2
IF SCALE>0 THEN STREAM(SCALE,SKP←SKP+SGN,BUFF) ;                                                   02895235 T  0614:3
     BEGIN                                                                                         02895240 T  0617:3
     DI←DI+SKP; SKP←DI; SI←SKP; SI←SI+1; DS←SCALE CHR ;                                            02895245 T  0617:3
     DS←LIT"." ;                                                                                   02895250 T  0619:2
     END ;                                                                                         02895255 T  0620:0
     GO TEST ;                                                                                     02895260 T  0620:1
TT:  P(CHR←W-1);  IF PRNTR THEN P(DEL,W+6) ;                                                       02895265 T  0620:3
     P((P(DUP),[33:12] INX SAVBUFF)&P(XCH)[30:45:3]);  GO TEST ;                                   02895280 T  0623:3
XX:  P(0) ;                                                                                        02895320 T  0626:2
X1:  SKIP;  GO TEST1 ;                                                                             02895360 T  0626:3
SS:  OUTPUT; IF (R←R-1)>0 THEN GO SS ELSE GO TEST2 ;                                               02895400 T  0628:2
CC:                                                                                                02895440 T  0632:3
AA:  P(WH1,6) ;                                                                                    02895480 T  0632:3
A1:  SKIP ;                                                                                        02895520 T  0633:1
     STREAM(Q←P:T←IF CODE=6 THEN 2 ELSE 8-T1,T1,BUFF) ;                                            02895560 T  0634:0
          BEGIN SI←LOC Q; SI←SI+T; DS←T1 CHR; Q←DI END ;                                           02895600 T  0638:0
     GO TEST ;                                                                                     02895640 T  0639:3
LL:  P("F");  IF T5 THEN P(29,+);  P(1);  GO A1 ;                                                  02895680 T  0640:1
PP:  SCALE←W&FMTW[1:41:1]; CHR←CHR-W; GO TEST1 ;                                                   02895720 T  0642:2
OO:  P(16);  SKIP ;                                                                                02895760 T  0646:0
     STREAM(Q←3×(16-T1):T1,WH1,BUFF) ;                                                             02895800 T  0647:0
          BEGIN                                                                                    02895840 T  0649:3
          SI←LOC WH1;  SKIP Q SB ;                                                                 02895880 T  0649:3
          T1(DS←3RESET; 3(IF SB THEN DS←SET ELSE DS←RESET; SKIP SB)) ;                             02895920 T  0650:2
          Q←DI ;                                                                                   02895960 T  0653:2
          END ;                                                                                    02896000 T  0653:3
     GO TEST ;                                                                                     02896040 T  0654:0
HH:  P(DEL);  IF (CHR←CHR+R)>MAXCHR THEN GO AWAY ;                                                 02896080 T  0654:2
     STREAM(Q←[FMT[FI]]:R,S←R,[36:6],BUFF) ;                                                       02896100 T  0657:2
         BEGIN SI←Q; SI←SI+3; DS←R CHR; S(DS←32CHR; DS←32CHR); Q←DI END ;                           02896120 T  0660:0
     BUFF←P;  FI←(R+2),[36:9]+FI;  GO GETNEXTPHRASE ;                                              02896160 T  0662:2
GG:  IF TYPE=INTEGR THEN BEGIN DECPT←D←0; GO II END ;                                              02896200 T  0668:0
```

```
    IF TYPE=LOGICAL THEN GO LL ;                                                                    02896240 T  0671:0
    IF E1≥0 AND E1≤D THEN                                                                           02896280 T  0672:1
         BEGIN                                                                                      02896320 T  0674:0
         W←W-4;  D←D-E1 ;                                                                           02896360 T  0674:2
II:                                                                                                 02896400 T  0677:0
JJ:                                                                                                 02896440 T  0677:0
FF:      IF P(0,E1+D,DUP)<0 THEN P(DEL,WH3←0) ;                                                     02896480 T  0677:0
         IF T4 AND DECPT THEN T6←P ELSE T3←SETUP+T3 ;                                               02896520 T  0680:1
         P(CODE=9) ;                                                                                02896600 T  0685:0
         E←(T4←IF (ND←0)<E1 THEN E1 ELSE P(DUP))+(T5←IF COMMAS THEN                                 02896640 T  0685:3
           (T4-1) DIV 3 ELSE 0)+DLRSGN+SGN ;                                                        02896680 T  0689:1
         IF P THEN                                                                                  02896720 T  0694:0
              BEGIN IF (CHR←CHR+E-W)>MAXCHR THEN GO AWAY; P(WH3=0) END                               02896725 T  0694:0
         ELSE IF P(W-D-DECPT-E,DUP)<0 THEN GO OVRFLW2 ;                                             02896760 T  0698:0
         SKP←P+T5 ;                                                                                 02896800 T  0703:0
         IF E1 LSS 1 THEN                                                                           02896840 T  0704:0
              BEGIN IF (ND←(T4←SKP≠0)-E1)>D THEN ND←D+T4;SKP←SKP-T4 END;                             02896880 T  0704:3
         GO CONVERT ;                                                                               02896920 T  0711:0
         END ;                                                                                      02896960 T  0711:2
EE: P("E");  GO D1 ;                                                                                02897000 T  0711:2
DD: P("D") ;                                                                                        02897040 T  0712:1
D1: IF T4 THEN BEGIN P(DEL); GO SE END; IF P(SCALE,DUP)<0 THEN P(DEL,0);                            02897080 T  0712:2
    IF (SKP←-P(D+P,DUP)+W-5-SGN-DLRSGN)<0 THEN GO OVRFLW2 ;                                         02897100 T  0716:0
    IF D←SCALE≤0 THEN IF -SCALE≥SAVD THEN GO OVRFLW2 ELSE P(SCALE,+) ;                              02897120 T  0720:2
    IF SETUP THEN P(TEN[T3-1],,WH3,ISD) ;                                                           02897360 T  0724:1
    IF (T4←ND←0)≠SCALE THEN                                                                         02897400 T  0727:0
         BEGIN                                                                                      02897440 T  0728:3
         IF ABS(E1←E1-SCALE)>99 THEN GO OVRFLW1 ;                                                   02897480 T  0729:1
         IF D THEN ND←-SCALE ELSE T4←SCALE ;                                                        02897520 T  0731:3
         END ;                                                                                      02897560 T  0734:3
CONVERT:                                                                                            02897600 T  0734:3
    IF NOT DBLPREC AND T3>8 THEN                                                                    02897640 T  0734:3
         BEGIN WH3←(WH2←WH3) DIV T8; T3←T3-T2←8 END ;                                               02897680 T  0736:1
    STREAM(ND,SKP:T6,SGN,E←DECPT,S←SKP,[36:6],T←T6,[36:6],DLRSGN,T4,                                02897720 T  0740:1
           T21,WH3,T3,WH2,T2,WH1,T1,BUFF) ;                                                         02897760 T  0744:0
         BEGIN SI←T21; DS←SKP CHR ;                                                                 02897800 T  0746:1
         S(SI←T21; DS←32CHR; DS←32CHR); SKP←DI; MAYBE(DLRSGN,L3,"$") ;                               02897840 T  0747:0
         MAYBE(SGN,L1,"-"); SGN←DI; DI←DI+E; ND(DS←LIT"0") ;                                         02897880 T  0750:0
         SI←LOC WH3; DS←T3 DEC; SI←LOC WH2; DS←T2 DEC; SI←LOC WH1 ;                                 02897920 T  0753:1
         DS←T1 DEC;T6(DS←LIT"0");T(32(DS←2LIT"0"));ND←DI;CI←CI+E;                                    02897960 P  0755:0
         GO L2; SI←SGN; DI←SGN; SI←SI+1; DS←T4 CHR; DS←LIT","; L2:                                  02898000 T  0759:1
         END ;                                                                                      02898040 T  0761:1
    T6←P ;                                                                                          02898080 T  0761:2
    IF (T4←P(XCH))≠0 THEN                                                                           02898120 T  0762:0
         STREAM(BUFF←P:T4,SGN←IF E1 LSS 0 THEN "-" ELSE " ",S←ABS(E1)) ;                            02898160 T  0763:1
              BEGIN                                                                                 02898200 T  0767:1
              DI←BUFF; SI←LOC T4; SI←SI+7; DS←CHR; SI←SI+7; DS←CHR ;                                02898240 T  0767:2
              DS←2 DEC; BUFF←DI ;                                                                   02898280 T  0769:0
              END                                                                                   02898320 T  0769:2
    ELSE IF T5>0 THEN                                                                               02898360 T  0769:2
         STREAM(T←T5-1,Q←E-T5×4,T5,T6) ;                                                            02898400 T  0771:0
              BEGIN                                                                                 02898440 T  0774:2
              SI←T6; DI←DI-T5; DS←Q CHR; DS←LIT"," ;                                                02898480 T  0774:2
              T(DS←3CHR; DS←LIT",") ;                                                               02898520 T  0776:1
              END ;                                                                                 02898560 T  0777:3
    IF W≠SAVW THEN BEGIN BUFF←P; P(W+W-SAVW); GO X1 END ;                                           02898600 T  0778:0
    GO TEST ;                                                                                       02898640 T  0781:2
```

```
OVRFLW3:
    P(DEL) ;
OVRFLW2:
    P(DEL) ;
OVRFLW1:
    P(DEL) ;
OVRFLW:
    STREAM(W←SAVW:W1←SAVW.[36:6]›BUFF) ;
        BEGIN W(DS←LIT"*"); W1(32(DS←2LIT"*")); W←DI END ;
TEST:
    BUFF←P ;
TEST1:
    IF (R←R-1)>0 THEN GO REPEAT1 ;
TEST2:
    IF (XTRA AND 3)=0 THEN GO GETNEXTPHRASE ;
    P(XTRA);  XTRA←SAVW←0 ;
    IF P(DUP) THEN BEGIN SAVW←P.[42:5]; CODE←4; GO REPEAT2 END ;
    CODE←1;  R←P.[42:4];  GO SS ;
    END OF FTOUTFIX ;
```

```
02898660 T  0782:0
02898665 T  0782:0
02898680 T  0782:1
02898720 T  0782:1
02898760 T  0783:1
02898800 T  0783:1
02898840 T  0784:1
02898880 T  0784:1
02898920 T  0787:0
02898960 T  0790:2
02899000 T  0790:2
02899040 T  0791:2
02899080 T  0791:2
02899085 T  0793:3
02899120 T  0793:3
02899160 T  0795:2
02899200 T  0797:0
02899240 T  0802:0
02899280 T  0804:1
                SIZE= 0805 WORDS
```

```
        % FORTRAN OUTPUT INTRINSIC
PROCEDURE FTOUT;            % 051

          BEGIN
COMMENT FILX        FILE TOP IO DESCRIPTOR
        FMTA        FORMAT OR NAMELIST OR 0
        LISX        ACCIDENTAL ENTRY DESC. OR 0
        EDITCODE     0    NO FORMAT, NO LIST
                     1    FORMAT, NO LIST
                     2    NO FORMAT, LIST
                     3    FORMAT, LIST
                     4    NAMELIST
                                        ;
REAL     EDITCODE  =  -1,
        FORTERR    =  24,
         LISX      =  -2,
         FI        =  -4,
         DKADR     =  -5;
ARRAY    FMTA      =  -3[*], FPB = 3[*] ;
NAME     FILX      =  -6,
         MEM       =   2;
REAL        ALGOLWRITE  = 12,
            SELECT      = 14;
INTEGER     LSTRN       = 19;
REAL        LISTYPE     = 20,
            ARRAYSTUFF  = 18;
ARRAY       TEN         = 22[*],
            TPAR        = 23[*],
            FIB[*];
NAME        LISTADR;
REAL        BUFF                    , %  FIRST BUFFER POSITION
            BSIZE                   , %  ARGUMENTS
            FLG             , % TRUE FOR SERIAL I/O
```

```
02900000 T  0000:0
02900100 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00377
02900200 T  0000:0
02900300 T  0000:0
02900400 T  0000:0
02900500 T  0000:0
02900600 T  0000:0
02900700 T  0000:0
02900800 T  0000:0
02900900 T  0000:0
02901000 T  0000:0
02901100 T  0000:0
02901200 T  0000:0
02901210 T  0000:0
02901300 T  0000:0
02901400 T  0000:0
02901500 T  0000:0
02901600 T  0000:0
02901700 T  0000:0
02901800 T  0000:0
02901900 T  0000:0
02902000 T  0000:0
02902300 T  0000:0
02902400 T  0000:0
02902500 T  0000:0
02902900 T  0000:0
02903000 T  0000:0
02903100 T  0000:0
02903200 T  0000:0
02903300 T  0000:0
02903400 T  0000:0
02903410 T  0000:0
```

```
                WH1,
                WH2                         , %                                                 02903500 T  0000!0
                W1                          , %                                                 02903600 T  0000!0
                W2                          , %                                                 02903700 T  0000!0
                NFCI                        ; %  NEXT FORMAT CHAR LOCATION                      02903800 T  0000!0
ARRAY           IOBUFF      = BUFF[*];                                                          02903900 T  0000!0
INTEGER         DH1                         , %   CONV-                                        02904000 T  0000!0
                DH2                         , %      ERTED NU-                                  02904100 T  0000!0
                DH3                         , %           MBER                                  02904200 T  0000!0
                RPT                         , %  REPEAT INDICATOR                             02904300 T  0000!0
                W                           , %  FIELD                                         02904400 T  0000!0
                WT                          , %       WIDTH                                    02904500 T  0000!0
                T1                          , %                                                02904600 T  0000!0
                D                           , %   DEC-                                         02904700 T  0000!0
                DT                          , %       IMAL P-                                  02904800 T  0000!0
                D1                          , %           LA-                                  02904900 T  0000!0
                D2                          , %             CE-                                02905000 T  0000!0
                D3                          , %               S                                02905100 T  0000!0
                ZEROS                       , %  TRAILING ZEROES                              02905200 T  0000!0
                EXP                         , %  EXPONENT                                      02905300 T  0000!0
                SHFT                        , %  INTEGER PART OF SHIFT                         02905400 T  0000!0
                CODE                        , %  EDITING FUNCTION                              02905500 T  0000!0
                SKP                         , %  REDUNDANT POSITIONS                           02905600 T  0000!0
                NCR                         , %  CURRENT BUFFER POSITION                       02905700 T  0000!0
                LCR                         , %  BUFFER SIZE IN CHARACTERS                     02905800 T  0000!0
                QUOTE                       , %  STRING DELIMITER (" OR @)                     02905900 T  0000!0
                CHR                         , %  CURRENT CHAR FROM FORMAT                      02905910 T  0000!0
                PRCW        ,     % PAREN CONTROL WORD                                        02906000 T  0000!0
                PCT,              % PAREN COUNTER                                             02906010 T  0000!0
                PS                          ; %   SCALE FACTOR                                 02906020 T  0000!0
BOOLEAN         DONETOG                     , %   RETURN AFTER WRITE                          02906100 T  0000!0
                SGN                         , %   SIGN                                        02906200 T  0000!0
                PRNTR                ,% TRUE IF PRINTER OUT PUT                             02906300 T  0000!0
                FMERRTOG                    , %   FORMAT ERROR                                02906400 T  0000!0
                LGTG        ,                                                                02906500 T  0000!0
                DTOG                        , %   DOUBLE PRECISION TOG                        02906600 T  0000!0
                CTOG                        , %   COMPLEX NUMBER TOG                          02906700 T  0000!0
                GTOGA                       , %  G EDITING TOG W-D - SGN > 4                  02906800 T  0000!0
                GTOG                        ; %   G EDITING TOG                               02906810 T  0000!0
DEFINE DBLV         = 5#,                                                                      02906900 T  0000!0
       CMPLXV       = 6#,                                                                      02907300 T  0000!0
       GTYPE        = 1#,                                                                      02907400 T  0000!0
       FTYPE        = 2#,                                                                      02907500 T  0000!0
       ETYPE        = 3#,                                                                      02907600 T  0000!0
       DTYPE        = 4#,                                                                      02907700 T  0000!0
       ITYPE        = 5#,                                                                      02907800 T  0000!0
       LTYPE        = 6#,                                                                      02907900 T  0000!0
       ATYPE        = 7#,                                                                      02908000 T  0000!0
       OTYPE        = 8#,                                                                      02908100 T  0000!0
       KIND          = (FIB[4],[8:4])#,                                                       02908200 T  0000!0
       TAPEF         =2#,                                                                     02908300 T  0000!0
       MAX          = @7777777777777#,                                                        02908500 T  0000!0
       DLN           = (LISTYPE,[44:4] =DBLV)#,                                               02908700 T  0000!0
       CMPLX         = (LISTYPE,[44:4] = CMPLXV)#,                                            02908900 T  0000!0
       TWOD          = LISTYPE,[38:1]#,                                                       02909000 T  0000!0
       LPPS         = 15:30:18#,                                                              02909100 T  0000!0
       LPPR         = [15:18]#,                                                               02909200 T  0000!0
                                                                                              02909300 T  0000!0
```

```
        RPTF        = [33:15]#,                                          02909400 T  0000:0
        NORF         = (P(XCH,DUP) < 0)#,                                02909500 T  0000:0
       PCF         = [9:6]#,                                             02909510 T  0000:0
        ENDLIST      = (LSTRN = (-1))#,                                  02909600 T  0000:0
        SIZEF        = [33:15]#,                                         02909700 T  0000:0
        BASEF        = [18:15]#;                                         02909800 T  0000:0
LABEL   TYPERR,NMLST,                                                    02910000 T  0000:0
        STRT,REPEAT,LPAR,RTPAR,SLASH,STRING,TFMT,FMTERR,                 02910100 T  0000:0
        CL1,CL2,CL3,CL4,SCAL,HOL,SKIP,CL3A,STRA,TFMA,TIX,                02910200 T  0000:0
        ERTN,G,F,E,DC,I,L,A,AA,O,FA,GA,AST,COMM,                         02910300 T  0000:0
        NOFL,FNOL,BINARY,FMTLST,                                         02910400 T  0000:0
        FRMTCD,NFPH,FMCYC,FMERR,ZAP,ZIPIT;                               02910500 T  0000:0
COMMENT  * * * * * START OF SUBROUTINE DECLARATIONS * * * * * * * * ;   02910600 T  0000:0
SUBROUTINE CKPB;                                                         02910700 T  0000:0
BEGIN COMMENT   INITIALIZE FILE AND ACQUIRE RECORD SIZE;                 02910800 T  0001:0
LCR ← 8×(BSIZE ← P(MKS,FLG,DKADR,0,(-1),FILX,ALGOLWRITE));               02910900 T  0001:0
IF PRNTR←PRNTR&(((T1←FIB[4].[8:4])=1 OR T1=7 OR T1=12) AND FPB[FIB[4]    02911000 T  0004:2
        .[13:11]+3].[43:5]<20)[47:47:1] THEN                             02911005 T  0009:0
    IF BSIZE ≥ 17 THEN BEGIN LCR ← 132; BSIZE ← 17 END;                  02911010 T  0013:0
BUFF←(IF T1←PRNTR AND (EDITCODE=1 OR EDITCODE>2) THEN TPAR ELSE *FILX)   02911100 T  0016:1
.[33:15] ;                                                               02911200 T  0020:2
IF ((NOT T1) OR PRNTR.[46:1]) AND EDITCODE≠2 THEN                        02911400 T  0022:0
STREAM(P2 ← (BSIZE-1).[36:6],                                            02911500 T  0024:2
          P3←BSIZE+T1-1,P4←BUFF) ;                                       02911600 T  0026:2
   BEGIN  DI ← P4;   DS ← 8 LIT " ";                                     02911700 T  0028:1
          SI ← P4;                                                       02911800 T  0029:3
          P2(DS ← 32 WDS;  DS ← 32 WDS);                                 02911900 T  0030:0
          DS ← P3 WDS;                                                   02912000 T  0031:1
   END;                                                                  02912100 T  0031:3
NCR ← 0;                                                                 02912300 T  0032:0
END CKPB;                                                                02912400 T  0032:3
SUBROUTINE PRNT;                                                         02912500 T  0033:0
BEGIN COMMENT    GENERATE A CALL FOR CAR. CONT. AND FOR OUTPUT;          02912600 T  0033:0
IF PRNTR AND (EDITCODE = 1 OR EDITCODE ≥ 3) THEN                         02912700 T  0033:0
BEGIN;                                                                   02912800 T  0035:1
    NCR ← 0;                                                             02912900 T  0035:3
    STREAM(P1←0;P2←TPAR);                                                02913000 T  0036:2
    BEGIN SI ←P2; DI ← LOC P1; DI ←DI + 7; DS ←CHR;                      02913100 T  0038:0
          DI ← P2; DS ← LIT " ";END;                                     02913200 T  0039:0
         NCR ← P;                                                        02913300 T  0040:0
         IF NCR = " " THEN D2 ← 16 ELSE                                  02913400 T  0040:2
         IF NCR = "0" THEN D2 ← 32 ELSE                                  02913500 T  0042:2
         IF NCR = "+" THEN D2 ← 0 ELSE                                   02913600 T  0045:0
              IF (D2 ← NCR) > 9 THEN D2 ← 16;                            02913700 T  0047:2
         IF NOT PRNTR.[46:1] THEN FIB[17]←FIB[17]+BSIZE ;                02913900 T  0050:2
         P(MKS,D2.[42:2],D2.[44:4],PRNTR.[46:1],BSIZE,FILX,ALGOLWRITE) ; 02914000 T  0054:0
         FIB[6]←FIB[6]-(D2=0) ;                                          02914010 T  0057:1
         IF NOT (*FILX).[19:1] THEN P(FILX,@2000000000,2,COM,DEL,DEL);   02914100 T  0059:3
         PRNTR←1;  CKPB ;                                                02914200 T  0063:0
         STREAM(P1←TPAR,P2←*FILX,P3←BSIZE.[36:6],P4←BSIZE);              02914300 T  0065:0
         BEGIN                                                           02914400 T  0067:2
              SI ← P1; DI ← P2; DS ← P4 WDS;                             02914500 T  0067:2
              P3(DS ←32 WDS; DS ← 32 WDS);                               02914600 T  0068:2
              DI←P1; P4(DS←8LIT" ") ;                                    02914610 T  0069:3
         END;                                                            02914700 T  0072:0
    FIB[17]←FIB[17]-BSIZE;  IF DONETOG THEN P(XIT) ;                     02914800 T  0072:1
    END ELSE BEGIN P(MKS,FLG,DKADR,0,BSIZE,FILX,ALGOLWRITE);             02914900 T  0075:1
```

```
    IF DONETOG THEN P(XIT);                                                                  02915000 T  0078:3
    CKPB END ;                                                                               02915100 T  0079:3
END PRNT;                                                                                    02915200 T  0081:0
LABEL NFCL;                                                                                  02915300 T  0081:1
REAL SUBROUTINE NFC;                                                                         02915400 T  0081:1
BEGIN                                                                                        02915500 T  0082:0
NFCL:                                                                                        02915600 T  0082:0
WHILE NFCI.[45:3] < 2 DO NFCI ← NFCI + 1;                                                    02915700 T  0082:0
STREAM(P1 ← 0:P2 ←FMTA[NFCI.[30:15]],P3 ← NFCI.[45:3]);                                      02915800 T  0085:2
   BEGIN  DI ← LOC P1;   DS ← 7 LIT "0";                                                     02915900 T  0088:1
          SI ← LOC P2; SI ←SI + P3;DS ← CHR;                                                 02916000 T  0089:3
         SI ← SI - 1; DI ← DI - 1;                                                           02916100 T  0090:3
   END;                                                                                      02916800 T  0091:1
NFCI ← NFCI + 1; IF (CHR ← P) = " " THEN IF NOT LGTG THEN GO NFCL;                           02916900 T  0091:2
NFC ← CHR;                                                                                   02917000 T  0095:0
END NFC;                                                                                     02917100 T  0095:2
SUBROUTINE IST;                                                                              02917200 T  0095:3
BEGIN ;                                                                                      02917300 T  0096:0
STREAM(P1 ← 0:P2 ← BUFF,P3 ← CHR);                                                           02917400 T  0096:0
   BEGIN  SI ← LOC P3;   SI ← SI + 7;                                                        02917500 T  0097:2
          DI ← P2;   DS ← CHR;  P1 ← DI;                                                     02917600 T  0098:0
   END;                                                                                      02917700 T  0098:3
BUFF ← P;                                                                                    02917800 T  0099:0
    END IST;                                                                                 02917900 T  0099:2
        % PARAMETERS FOR LIST CONTROL                                                        02918000 T  0099:3
BOOLEAN ATOG,TWDT;                                                                           02918100 T  0099:3
ARRAY AR1 = LISTADR[*];                                                                      02918200 T  0099:3
REAL  INDX,SIZE,NLI,NLE;                                                                     02918400 T  0099:3
LABEL RTNLST,SRT;                                                                            02918500 T  0099:3
DEFINE NXTELM = IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX,[40:8],COC)                          02918600 T  0099:3
                        ELSE AR1[INDX]#;                                                     02918700 T  0099:3
SUBROUTINE GETLIST;                                                                          02920200 T  0099:3
    BEGIN                                                                                    02920300 T  0100:0
    SRT: IF ATOG THEN                                                                        02920400 T  0100:0
    BEGIN                                                                                    02920500 T  0100:1
         IF DLN THEN                                                                         02920600 T  0100:3
    BEGIN                                                                                    02920700 T  0102:0
         WH1 ← NXTELM;                                                                       02920800 T  0102:2
         INDX ← INDX + 1;                                                                    02920900 T  0107:0
         WH2 ← NXTELM;                                                                       02921000 T  0108:1
    END ELSE                                                                                 02921100 T  0112:3
    BEGIN                                                                                    02921200 T  0112:3
         WH1 ← NXTELM;                                                                       02921300 T  0113:1
         WH2 ← 0;                                                                            02921400 T  0117:3
    END;                                                                                     02921500 T  0118:2
         IF (INDX ←INDX + 1) ≥ SIZE THEN                                                     02921600 T  0118:2
    BEGIN                                                                                    02921700 T  0120:1
         ARRAYSTUFF ← 0;                                                                     02921800 T  0120:3
         ATOG ← FALSE;                                                                       02921900 T  0121:2
    END;                                                                                     02922000 T  0122:1
         GO TO RTNLST;                                                                       02922100 T  0122:1
    END;                                                                                     02922200 T  0122:3
         IF CTOG THEN                                                                        02922300 T  0122:3
    BEGIN     % IMAGINARY PART OF COMPLEX                                                    02922400 T  0123:0
         WH1 ← LISTADR[1];                                                                   02922500 T  0123:2
         WH2 ← 0;                                                                            02922600 T  0125:0
         CTOG ← FALSE;                                                                       02922700 T  0125:3
```

```
            GO TO RTNLST;                                                                    02922800 T  0126:2
        END;                                                                                 02922900 T  0127:0
            P(0); LISTADR ← [LISX];                                                          02923600 T  0127:0
            IF ARRAYSTUFF ≠ 0 THEN                                                           02923800 T  0128:0
        BEGIN                                                                                02923900 T  0128:3
            ATOG ← TRUE;                                                                     02924000 T  0129:1
            SIZE←(INDX←ARRAYSTUFF.BASEF)+ARRAYSTUFF.SIZEF ;                                  02924400 T  0130:0
            TWDT←NOT P(*(LISTADR←MEM[LISTADR.[18:15]]).TOP);  P(DEL) ;                       02924500 T  0132:3
            GO TO SRT;                                                                       02924600 T  0136:1
        END;                                                                                 02924700 T  0136:3
            P(DEL);                                                                          02924800 T  0136:3
            WH1 ← LISTADR[0];                                                                02924900 T  0137:0
            WH2 ← IF DLN THEN LISTADR[1] ELSE 0;                                             02925000 T  0137:3
            CTOG ← CMPLX;                                                                    02925100 T  0141:3
        RTNLST:                                                                              02925200 T  0143:2
        END GETLIST;                                                                         02925300 T  0143:2
    SUBROUTINE FORMATCONTROL;                                                                02927400 T  0143:3
    BEGIN                                                                                    02927500 T  0144:0
        STRT:                                                                                02927600 T  0144:0
                W←D←CODE←SKP←RPT←0;                                                          02927700 T  0144:0
                SGN←DONETOG←FMERRTOG←FALSE;                                                  02927800 T  0146:3
        CL1: COMMENT CHECK FOR SINGLE CHARACTER EDITING TYPES;                               02927900 T  0148:2
                IF NFC≤9 THEN GO TO REPEAT; % MUST BE REPEAT FIELD                           02928000 T  0148:2
                IF CHR = "(" OR CHR = "%" THEN GO TO LPAR;                                   02928100 T  0151:0
                IF CHR = ")" OR CHR = "[" THEN GO TO RTPAR;                                  02928200 T  0153:2
                IF CHR="/" THEN GO SLASH;                                                    02928300 T  0156:0
                IF CHR = """ OR CHR = "@" THEN GO TO STRING;                                 02928400 T  0157:1
                IF CHR="T" THEN GO TO TFMT;                                                  02928500 T  0159:3
                SGN←(CHR="-") & (CHR="+")[2:47:1];                                           02928590 T  0161:0
                IF SGN THEN                                                                  02928600 T  0163:3
            BEGIN                                                                            02928700 T  0164:0
                IF NFC≤9 THEN GO TO REPEAT                                                   02928800 T  0164:2
                        ELSE GO TO FMTERR;                                                   02928900 T  0166:2
            END;                                                                             02929000 T  0167:2
                IF CHR="," THEN GO TO STRT;                                                  02929100 T  0167:2
                RPT←1;                                                                       02929200 T  0168:3
        CL2: COMMENT  TYPES WHICH MAY HAVE REPEAT FIELDS;                                    02929300 T  0169:2
                IF SGN THEN RPT←-RPT;                                                        02929400 T  0169:2
                IF CHR="P" THEN GO TO SCAL;                                                  02929500 T  0171:1
                IF RPT<0 OR SGN.[2:1] THEN GO TO FMTERR;                                     02929600 T  0172:2
                IF CHR = "(" OR CHR = "%" THEN GO TO LPAR;                                   02929700 T  0175:0
                IF CHR="H" THEN GO TO HOL;                                                   02929800 T  0177:2
                IF RPT=0 THEN RPT←1;                                                         02929900 T  0178:3
                IF CHR = "X" THEN GO TO SKIP;                                                02930000 T  0180:3
        CL3: COMMENT TYPES WHICH HAVE W FIELDS;                                              02930100 T  0182:0
                IF CHR="I" THEN CODE ← ITYPE ELSE                                            02930200 T  0182:0
                 IF CHR="A" THEN CODE ← ATYPE ELSE                                           02930300 T  0184:0
                  IF CHR="L" THEN CODE ← LTYPE ELSE                                          02930400 T  0186:2
                    IF CHR="O" THEN CODE ← OTYPE;                                            02930500 T  0189:0
                IF CODE ≥ ITYPE THEN GO TO CL3A;                                             02930600 T  0191:2
        CL4: COMMENT  TYPES WITH W AND D FIELDS;                                             02930700 T  0192:3
                IF CHR="D" THEN CODE ← DTYPE ELSE                                            02930800 T  0192:3
                 IF CHR="E" THEN CODE ← ETYPE ELSE                                           02930900 T  0194:3
                  IF CHR="F" THEN CODE ← FTYPE ELSE                                          02931000 T  0197:1
                   IF CHR="G" THEN CODE ← GTYPE ELSE                                         02931100 T  0199:3
                    GO TO FMTERR;                                                            02931200 T  0202:1
        CL3A: COMMENT DEVELOP VALUE OF W FIELD;                                              02931300 T  0202:1
```

```
            IF NFC>9 THEN GO TO FMTERR;                                    02931400 T  0202:1
            W←CHR;                                                         02931500 T  0204:0
            WHILE NFC≤9 DO W←10×W+CHR; % CONVERT TO OCTAL                  02931600 T  0204:3
            NFCI←NFCI-1;                                                   02931700 T  0209:1
            IF W>63 THEN GO TO FMTERR;                                     02931800 T  0210:2
            IF CODE≥ITYPE THEN GO TIX;                                     02931900 T  0211:3
         COMMENT DEVELOP D FIELD;                                          02932000 T  0213:0
            IF NFC≠"." THEN GO TO FMTERR;                                  02932100 T  0213:0
         IF NFC >9 THEN GO TO FMTERR;                                      02932200 T  0215:0
            D←CHR;                                                         02932300 T  0217:0
            WHILE NFC≤9 DO D←10×D+CHR; % CONVERT TO OCTAL                  02932400 T  0217:3
            NFCI←NFCI-1;                                                   02932500 T  0222:1
            GO TO TIX;                                                     02932600 T  0223:2
LPAR:    COMMENT GENERITE PAREN CONTROL WORD;                              02932700 T  0224:0
            IF PCT≠0 AND RPT=0 THEN RPT←1 ;                                02932710 T  0224:0
            T1 ← RPT&NFCI[LPPS]&(RPT≤0)[1:47:1];                           02932800 T  0227:0
            IF PCT ≤ 1 THEN PRCW ← T1 & PCT[9:42:6];                       02932810 T  0230:1
            P (T1, XCH); PCT←PCT+1;                                        02932820 T  0233:1
            GO TO STRT;                                                    02932900 T  0235:0
RTPAR:   COMMENT POINT AT LEFT PAR IF REPEAT NOT EXAUSTED;                 02933000 T  0235:2
            IF NORF THEN                                                   02933100 T  0235:2
         BEGIN      % NO REPEAT FIELD                                      02933200 T  0236:2
            DONETOG ← ENDLIST;                                             02933300 T  0237:0
            PRNT;% WRITE OUT RECORD                                        02933400 T  0238:2
            IF (PCT ← PCT - 1) ≤ 0 THEN IF PRCW.PCF ≠0                     02933410 T  0240:0
             THEN BEGIN P(XCH,PRCW); PCT ← 2 END ELSE PCT ← 1;             02933420 T  0243:0
         END ELSE                                                          02933500 T  0246:2
         BEGIN                                                             02933600 T  0246:2
            IF (RPT←P(DUP).RPTF) ≤ 1                                       02933700 T  0247:0
             THEN BEGIN P(DEL);PCT ← PCT - 1; GO TO STRT END               02933800 T  0248:1
             ELSE P(RPT - 1,CCX);                                          02933900 T  0251:1
         END;                                                              02934000 T  0252:3
            NFCI←P(DUP).LPPR; % RESET TO LEFT PAREN                        02934100 T  0252:3
            P(XCH);                                                        02934200 T  0254:0
            GO TO STRT;                                                    02934300 T  0254:1
REPEAT:  COMMENT CONVERT REPEAT FIELD TO OCTAL IN RPT;                     02934400 T  0254:3
            RPT←CHR;                                                       02934500 T  0254:3
            WHILE NFC≤9 DO RPT← 10×RPT+CHR;                                02934600 T  0255:2
            GO TO CL2;                                                     02934700 T  0260:1
SLASH:   COMMENT WRITE OUT BUFFER;                                         02934800 T  0260:3
            PRNT;                                                          02934900 T  0260:3
            GO TO STRT;                                                    02935000 T  0262:0
STRING:  COMMENT MOVE STRING FROM FORMAT ARRAY TO BUFFER;                  02935100 T  0262:2
            QUOTE ← CHR;   % SAVE STRING DELIMITER                         02935110 T  0262:2
            LGTG ← TRUE; CHR ← NFC;                                        02935200 T  0263:1
STRA:       IF (NCR ← NCR + 1) > LCR THEN GO TO FMTERR;                    02935300 T  0265:2
            IST;                                                           02935400 T  0267:3
            IF NFC ≠ QUOTE THEN GO TO STRA;    % " OR @                   02935500 T  0269:0
            LGTG ← FALSE ;  GO TO STRT;                                    02935600 T  0271:0
TFMT:    COMMENT SET BUFFER TO CHARACTER POSITION INDICATED BY FIELD       02935700 T  0272:1
            FOLLOWING "T";                                                 02935800 T  0272:1
            IF (RPT←NFC)>9 THEN GO TO FMTERR;                              02935900 T  0272:1
            WHILE NFC≤9 DO RPT←10×RPT+CHR;                                 02936000 T  0274:2
            IF RPT>LCR THEN GO TO FMTERR;                                  02936100 T  0279:1
            NCR←RPT-1;                                                     02936200 T  0280:2
TFMA:       BUFF ←((IF PRNTR THEN TPAR ELSE (*FILX)) INX                   02936300 T  0281:3
                 NCR.[33:12])&NCR[30:45:3];                                02936400 T  0284:0
```

```
                GO TO STRT;
          SCAL:  COMMENT SCALE FACTOR OF P PHRASE;
                PS←RPT;
                GO TO STRT;
          HOL: COMMENT HOLLERITH STRING;
                LGTG ← TRUE;
                WHILE RPT > 0 DO
               BEGIN
                IF (NCR ← NCR + 1) > LCR THEN GO TO FMTERR;
                CHR ← NFC;  IST;
                RPT←RPT-1;
               END;
                LGTG ← FALSE; GO TO STRT;
          SKIP: COMMENT X PHRASE;
               IF (NCR ← NCR+RPT) > LCR THEN GO TO FMTERR;
                GO TO TFMA;
          FMTERR:   FMERRTOG←TRUE;
     TIX:
     END FORMATCONTROL;
     SUBROUTINE FUNNYZERO;
          BEGIN
               SKP ← W - (D+6+SGN);
               STREAM(P1← BUFF:P2←SKP,P3←SGN,P4←(D+4));
               BEGIN
                    DI ←P1; DI ← DI + P2;
                    P3(DS ← LIT "-"; JUMP OUT TO L);
               L:   DS ← 2 LIT "0.";
                    P4(DS ← LIT " ");
                    P1 ← DI;
               END;
               BUFF ← P;
          END FUNNYZERO;
     SUBROUTINE FINDE;
     BEGIN          IF DTOG THEN
                    DOUBLE(TEN[0],0,WH1,WH2,×,←,WH1,WH2)
                         ELSE WH1 ← TEN[0] × WH1;
                    EXP←(0&WH1[42:3:6]&WH1[1:2:1]+12,5)×,90308998709 ;
                    W2 ← 0;
                    IF DTOG THEN
                    IF EXP ≥ 0 THEN DOUBLE(TEN[EXP],TEN[69+EXP],←,W1,W2)
                     ELSE DOUBLE(1,0,TEN[-EXP],TEN[69-EXP],/,←,W1,W2)
                    ELSE W1 ← IF EXP ≥ 0 THEN TEN[EXP] ELSE 1/TEN[-EXP];
                    IF WH1 > W1 THEN GO TO ERTN;
                    IF WH1 = W1 THEN
                     IF WH2 ≥ W2 THEN GO TO ERTN;
                    EXP ← EXP-1;
          ERTN:
     END FINDE;
     SUBROUTINE NUMCONVERT;
     BEGIN
                    IF D1 > 0 THEN
               BEGIN
                    DOUBLE(WH1,WH2,TEN[16],TEN[85],/,←,W1,W2);
                    DH1 ← W1 DIV 1.0;
               END;
                    IF D2 > 0 THEN
               BEGIN     IF DTOG  THEN
```

```
02936500 T  0286:2
02936600 T  0287:0
02936700 T  0287:0
02936800 T  0287:3
02936900 T  0288:1
02937000 T  0288:1
02937100 T  0289:0
02937200 T  0290:1
02937300 T  0290:1
02937400 T  0292:2
02937500 T  0296:0
02937600 T  0297:1
02937700 T  0297:3
02937800 T  0299:0
02937900 T  0299:0
02938000 T  0301:1
02938100 T  0301:3
02938200 T  0302:2
02938300 T  0302:2
02938400 T  0302:3
02938500 T  0303:0
02938600 T  0303:0
02938700 T  0305:1
02938800 T  0307:2
02938900 T  0307:2
02939000 T  0308:1
02939100 T  0310:0
02939200 T  0310:2
02939300 T  0311:3
02939400 T  0312:0
02939500 T  0312:1
02939600 T  0312:3
02939700 T  0313:0
02939800 T  0313:0
02939900 T  0313:1
02940000 T  0316:1
02940100 T  0318:2
02940150 T  0322:1
02940200 T  0323:0
02940300 T  0323:1
02940400 T  0327:2
02940500 T  0334:3
02940600 T  0339:2
02940700 T  0340:3
02940800 T  0341:2
02940900 T  0343:1
02941000 T  0344:2
02941100 T  0344:2
02941200 T  0344:3
02941300 T  0345:0
02941400 T  0345:0
02941500 T  0345:3
02941600 T  0346:1
02941700 T  0349:1
02941800 T  0350:2
02941900 T  0350:2
02942000 T  0351:1
```

```
                BEGIN                                                                    02942100 T  0352:0
                   DOUBLE(WH1,WH2,DH1,0,TEN[16],TEN[85],x,=,                             02942200 T  0352:2
                          TEN[ 8],TEN[77],/,←,W1,W2);                                    02942300 T  0355:1
                   DH2 ← W1 DIV 1;                                                       02942400 T  0357:3
                END                                                                      02942500 T  0359:0
                   ELSE DH2 ← WH1 DIV TEN[8];                                            02942600 T  0359:0
             END;                                                                        02942700 T  0362:2
                   IF DTOG THEN                                                          02942800 T  0362:2
             BEGIN                                                                       02942900 T  0362:3
                   DOUBLE(WH1,WH2,DH1,0,TEN[16],TEN[85],x,                               02943000 T  0363:1
                                DH2,0,TEN[ 8],TEN[77],x,+,=,←,W1,W2);                    02943100 T  0365:3
                   DH3 ← W1 DIV 1;                                                       02943200 T  0369:1
             END                                                                         02943300 T  0370:2
                   ELSE DH3 ← WH1 DIV 1;                                                 02943400 T  0370:2
                   EXP ← EXP+1;                                                          02943500 T  0372:1
END NUMCONVERT;                                                                          02943600 T  0373:2
SUBROUTINE SETD;                                                                         02943700 T  0373:3
BEGIN                                                                                    02943800 T  0374:0
                   IF DLN AND DT > 23 THEN                                               02943900 T  0374:0
             BEGIN                                                                       02944000 T  0376:1
                   ZEROS←DT-23; DT ← 23; D1 ← 7; D2 ← D3 ← 8;                            02944100 T  0376:3
             END ELSE IF DT>12 AND NOT DLN THEN                                          02944200 T  0380:3
             BEGIN                                                                       02944300 T  0383:2
                   ZEROS←DT-12; DT ← 12; D1←0; D2 ← 4; D3 ← 8;                           02944400 T  0384:0
             END ELSE IF DT>16 THEN                                                      02944500 T  0388:1
             BEGIN                                                                       02944600 T  0389:2
                   D1←DT-16; D2←D3←8;                                                    02944700 T  0390:0
             END ELSE IF DT > 8 THEN                                                     02944800 T  0392:2
             BEGIN                                                                       02944900 T  0393:3
                   D1←0;D2←DT-8; D3←8;                                                   02945000 T  0394:1
             END ELSE                                                                    02945100 T  0397:0
             BEGIN                                                                       02945200 T  0397:0
                   D1←D2←0;D3←DT;                                                        02945300 T  0397:2
             END;                                                                        02945400 T  0399:2
END SETD;                                                                                02945500 T  0399:2
SUBROUTINE RNDOFF;                                                                       02945600 T  0399:3
BEGIN              IF DTOG THEN                                                          02945700 T  0400:0
                   IF T1 ≥ 0 THEN                                                        02945800 T  0400:1
                    DOUBLE(WH1,WH2,,5,TEN[T1],TEN[T1+69],x,+,←,WH1,WH2) ELSE             02945900 T  0401:2
                    DOUBLE(WH1,WH2,,5,TEN[-T1],TEN[69-T1],/,+,←,WH1,WH2)                 02946000 T  0406:1
                   ELSE WH1 ← WH1 + (IF T1≥0 THEN 5×TEN[T1] ELSE 5/TEN[-T1]);            02946100 T  0411:0
END RNDOFF;                                                                              02946200 T  0416:0
SUBROUTINE SCALE;                                                                        02946300 T  0417:0
BEGIN              IF DTOG THEN                                                          02946400 T  0417:0
    BEGIN          IF T1 ≥ 0                                                             02946500 T  0417:1
                    THEN DOUBLE(WH1,WH2,TEN[T1],TEN[T1+69],x,←,WH1,WH2)                  02946600 T  0418:0
                    ELSE DOUBLE(WH1,WH2,TEN[-T1],TEN[69-T1],/,←,WH1,WH2);                02946700 T  0422:1
                   IF WH1 ≥ TEN[DT] THEN                                                 02946800 T  0426:3
             BEGIN                                                                       02946900 T  0427:3
                   EXP ← EXP + 1;                                                        02947000 T  0428:1
                   DOUBLE(WH1,WH2,TEN[1],0,/,←,WH1,WH2);                                 02947100 T  0429:2
             END                                                                         02947200 T  0432:1
    END ELSE WH1 ← IF T1 ≥ 0 THEN WH1×TEN[T1] ELSE WH1/TEN[-T1];                         02947300 T  0432:1
END SCALE;                                                                               02947400 T  0437:1
              %*************** S T A R T  O F  EDIT-CONTROL*****************%            02947500 T  0437:2
SUBROUTINE CONVERT;                                                                      02947600 T  0437:2
BEGIN                                                                                    02947700 T  0438:0
```

```
DTOG ← GTOG ← FALSE;                                                                02947800 T  0438:0
SGN ←WH1.[1:1]; IF CODE < LTYPE THEN WH1 ← ABS(WH1); WT ← W; DT ← D;                 02947900 T  0439:1
DH1 ← DH2 ← DH3 ← ZEROS ←EXP ← SKP ← SHFT ← D1 ← D2 ← D3 ← 0;                        02948000 T  0444:1
GO TO P(CODE,DUP,ADD);                                                              02948100 T  0449:2
GO TO FMERR;                                                                        02948200 T  0450:2
GO TO G;                                                                            02948300 T  0451:0
GO TO F;                                                                            02948400 T  0451:2
GO TO E;                                                                            02948500 T  0452:0
GO TO DC;                                                                           02948600 T  0452:2
GO TO I;                                                                            02948700 T  0453:0
GO TO L;                                                                            02948800 T  0453:2
GO TO A;                                                                            02948900 T  0454:0
GO TO O;                                                                            02949000 T  0454:2
O:   COMMENT OCTAL CONVERSION * * * * * * * * * * ;                                 02949100 T  0455:0
        IF W > 16 THEN SKP ← W - (WT ← 16);                                         02949200 T  0455:0
STREAM(P1 ← BUFF:P2 ← WH1,P3 ← SKP,P4 ← WT,P5 ← 16-WT);                             02949300 T  0458:0
   BEGIN SI ← LOC P2; DI ← P1;                                                      02949400 T  0460:2
         DI ← DI + P3;   P5(SKIP 3 SB);                                             02949500 T  0461:0
         P4(DS ← 3 RESET; 3(IF SB THEN DS ← SET                                     02949600 T  0462:2
                                  ELSE DS ← RESET;  SKIP SB));                      02949700 T  0464:1
         P1 ← DI;                                                                   02949800 T  0465:2
   END;                                                                             02949900 T  0465:3
BUFF ← P;                                                                           02950000 T  0466:0
GO TO COMM;                                                                         02950100 T  0466:2
A:  COMMENT ALPHA CONVERSION * * * * * * * * * ;                                    02950200 T  0467:0
IF W > 6 THEN SKP ← W - (WT ← 6);                                                   02950300 T  0467:0
AA:   STREAM(P1 ← BUFF:P2 ← WH1,P3 ← SKP, P4 ← WT);                                 02950400 T  0470:0
   BEGIN DI ← P1; DI ← DI + P3;                                                     02950500 T  0471:3
         SI ← LOC P2;   SI ← SI + 2;                                                02950600 T  0472:2
         DS ← P4 CHR;   P1 ← DI;                                                    02950700 T  0473:0
   END;                                                                             02950800 T  0473:3
BUFF ← P;                                                                           02950900 T  0474:0
GO TO COMM;                                                                         02951000 T  0474:2
    L:   COMMENT LOGICIAL CONVERSION;                                               02951100 T  0475:0
         IF W >1 THEN SKP←W-(WT←1);                                                 02951200 T  0475:0
         WH1← 0&(IF WH1 THEN "T" ELSE "F")[12:42:6];                                02951300 T  0478:0
         GO TO AA;                                                                  02951400 T  0481:1
    I:   COMMENT INTEGER CONVERSION;                                                02951500 T  0481:3
         IF WH1=0 AND WH2=0 THEN DT ← D3 ← 1 ELSE                                   02951600 T  0481:3
        BEGIN     IF DTOG THEN                                                      02951700 T  0485:1
         DOUBLE(WH1,WH2,,,5,+,←,WH1,WH2)  % ROUND OFF                               02951800 T  0486:0
         ELSE WH1 ← T1 ← WH1;                                                       02951900 T  0488:3
         IF WH1=0 AND WH2=0 THEN EXP←-1 ELSE FINDE ;                                02952000 T  0492:1
         IF EXP < 0 THEN DT ← D3 ← 1 ELSE                                           02952100 T  0497:0
        BEGIN                                                                       02952200 T  0499:2
         IF (DLN AND EXP≥24) OR (NOT DLN AND EXP≥12) THEN GO AST;                   02952300 T  0500:0
         DT ← EXP+1; SETD; NUMCONVERT;                                              02952400 T  0505:2
        END;                                                                        02952500 T  0509:0
        END;                                                                        02952600 T  0509:0
         IF DT + SGN > W THEN GO TO AST;                                            02952700 T  0509:0
         IF W > DT + SGN THEN SKP ← W - DT - SGN;                                   02952800 T  0510:3
STREAM(P1←0:P2 ← D1,P3 ← DH1,P4 ← D2,P5 ← DH2,                                      02952900 T  0514:1
      P6 ← D3,P7 ← DH3,P8 ← SGN,P9 ← SKP,P10 ← BUFF);                               02953000 T  0516:0
   BEGIN  DI ← P10;    P9(DI ← DI + 1);                                             02953100 T  0517:2
         P8(DS ← LIT "-");                                                          02953200 T  0518:3
         SI ←LOC P3;  DS ← P2 DEC;                                                  02953210 T  0520:0
         SI ← LOC P5;   DS ← P4 DEC;                                                02953300 T  0520:3
```

```
            SI ← LOC P7;    DS ← P6 DEC;                                    02953400 T  0521:2
            P1 ← DI;                                                        02953500 T  0522:1
   END;                                                                     02953600 T  0522:2
BUFF ← P;                                                                   02953700 T  0522:3
GO TO COMM;                                                                 02953800 T  0523:1
    DC:  COMMENT DOUBLE PRECISION CONVERT,SAME AS E CONVERT;                02953900 T  0523:3
    E:   COMMENT E CONVERSION;                                              02954000 T  0523:3
            DTOG ← TRUE;                                                    02954100 T  0523:3
            SETD;                                                           02954200 T  0524:2
            IF WH1=0 AND WH2 = 0 THEN                                       02954300 T  0526:0
         BEGIN                                                              02954400 T  0527:3
            IF W < (D+6+ SGN) THEN GO TO AST;                               02954500 T  0528:1
            FUNNYZERO; GO TO COMM;                                          02954600 T  0530:2
         END  ELSE                                                          02954700 T  0532:2
         BEGIN                                                              02954800 T  0532:2
            FINDE;                                                          02954900 T  0533:0
            IF PS ≤ 0 THEN                                                  02955000 T  0534:0
         BEGIN                                                              02955100 T  0534:3
            IF (SKP ← W - D - 5 - SGN) < 0 THEN GO TO AST; SETD;            02955200 T  0535:1
            IF (DT ← DT + PS) < 0 THEN DT ← 0;                              02955300 T  0540:0
            T1 ← EXP - DT; RNDOFF;                                          02955400 T  0543:0
         END ELSE                                                           02955500 T  0545:0
         BEGIN                                                              02955600 T  0545:0
            DT ← DT + (SHFT ← PS);  SETD;                                   02955700 T  0545:2
            T1 ← EXP - DT; RNDOFF;                                          02955800 T  0548:0
            IF W<(T1←DT+5+ SGN + ZEROS)    THEN GO TO AST;                  02955900 T  0550:0
            SKP←W-T1;                                                       02956000 T  0553:1
         END;                                                               02956100 T  0554:2
            T1←DT-1-EXP; SCALE;                                             02956200 T  0554:2
            NUMCONVERT;                                                     02956300 T  0557:0
            EXP←EXP-PS;                                                     02956400 T  0558:0
      END;                                                                  02956500 T  0559:1
STREAM(P1 ← 0;P2 ← SKP,P3 ← SGN,P4 ← D1,P5 ← DH1,                           02956600 T  0559:1
       P6 ← D2,P7 ← DH2,P8 ← D3,P9 ← DH3,P10 ← (DLN),                       02956700 T  0561:0
       P11 ← (EXP < 0),P12 ← ABS(EXP),P13 ← SHFT,P14 ← ZEROS,P15 ←BUFF);    02956800 T  0563:1
   BEGIN  DI ← P15;   DI ← DI + P2;   P3(DS ← LIT "-");                     02956900 T  0565:1
          P2 ← DI;   DS ← LIT ".";                                          02957000 T  0567:2
          SI ← LOC P5;   DS ← P4 DEC;                                       02957100 T  0568:1
          SI ← LOC P7;   DS ← P6 DEC;                                       02957200 T  0569:0
          SI ← LOC P9;   DS ← P8 DEC;                                       02957300 T  0569:3
          P14(DS ← LIT " ");  DS ← LIT "E";                                 02957400 T  0570:2
          P10(DI ← DI - 1;  DS ← LIT "D");                                  02957500 T  0572:1
         DS ← LIT " " ;                                                     02957600 T  0573:3
          P11(DI ← DI - 1;  DS ← LIT "-");                                  02957700 T  0574:1
          SI ← LOC P12;   DS ← 2 DEC;                                       02957800 T  0575:3
          P1 ← DI;                                                          02957900 T  0576:1
          P13(DI ← P2;  SI ← P2;  SI ← SI + 1;                              02958000 T  0576:2
            DS ← P13 CHR;  DS ← LIT ".";  JUMP OUT TO X);  X:               02958100 T  0577:3
   END;                                                                     02958200 T  0579:2
BUFF ← P;                                                                   02958300 T  0579:3
            GO TO COMM;                                                     02958400 T  0580:1
    F:   COMMENT F CONVERSION;                                              02958500 T  0580:3
            IF DTOG THEN                                                    02958600 T  0580:3
            IF PS>0                                                         02958700 T  0581:0
             THEN DOUBLE(WH1,WH2,TEN[PS],TEN[69+PS],×,←,WH1,WH2)            02958800 T  0581:3
             ELSE DOUBLE(WH1,WH2,TEN[-PS],TEN[69-PS],/,←,WH1,WH2)           02958900 T  0586:0
            ELSE WH1 ← IF PS > 0 THEN WH1×TEN[PS] ELSE WH1/TEN[-PS];        02959000 T  0590:1
```

```
 FA:       IF WH1=0 AND WH2=0 THEN EXP←0 ELSE                                  02959100 T  0595:2
      BEGIN                                                                    02959200 T  0598:2
           T1 ← -(DT+1); RNDOFF;                                               02959300 T  0599:0
           FINDE;                                                              02959400 T  0602:0
           IF EXP<0 THEN EXP←0;                                                02959500 T  0603:0
           IF (T1←DT+EXP+1)> 12 THEN DT←DT-(ZEROS←T1-12);                      02959600 T  0605:0
           T1 ← DT; SCALE;                                                     02959700 T  0610:0
           IF ABS(WH1) > MAX THEN                                              02959800 T  0612:0
      BEGIN                                                                    02959900 T  0613:0
           DT ← DT - 1; ZEROS ← ZEROS + 1;                                     02960000 T  0613:2
           T1 ← -1; SCALE;                                                     02960100 T  0616:0
           END;                                                                02960200 T  0618:0
      END;                                                                     02960300 T  0618:0
           DT←DT + EXP +1; SETD;                                               02960400 T  0618:0
           IF W<(T1←D+2+ SGN + EXP)    THEN GO TO AST;                         02960500 T  0621:0
           SKP←W-T1;                                                           02960600 T  0624:1
           NUMCONVERT;                                                         02960700 T  0625:2
STREAM(P1 ← 0;P2 ← SKP,P3 ← SGN,P4 ← D1,P5 ← DH1,                              02960800 T  0627:0
     P6 ← D2,P7 ← DH2,P8 ← D3,P9 ← DH3,P10 ← ZEROS,                           02960900 T  0628:3
     P11 ← EXP,P12 ← BUFF);                                                    02961000 T  0630:0
  BEGIN  DI ← P12;   DI ← DI + P2;   P3(DS ← LIT "-");                         02961100 T  0630:3
         P2 ← DI;   DS ← LIT ".";                                              02961200 T  0632:3
         SI ← LOC P5;   DS ← P4 DEC;                                           02961300 T  0633:2
         SI ← LOC P7;   DS ← P6 DEC;                                           02961400 T  0634:1
         SI ← LOC P9;   DS ← P8 DEC;                                           02961500 T  0635:0
         P10(DS ← LIT "0");                                                    02961600 T  0635:3
         P1 ← DI;                                                              02961700 T  0637:0
         P11(DI ← P2;  SI ← P2;  SI ← SI + 1;   DS ← P11 CHR;                  02961800 T  0637:1
             DS ← LIT ".";  JUMP OUT TO X);   X:                               02961900 T  0639:0
  END;                                                                         02962000 T  0640:1
BUFF ← P;                                                                      02962100 T  0640:2
           IF GTOG THEN GO TO GA;                                              02962200 T  0641:0
           GO TO COMM;                                                         02962300 T  0642:0
   G:   COMMENT G CONVERSION;                                                  02962400 T  0644:0
           GTOG ← TRUE;                                                        02962500 T  0644:0
           IF WH1=0 AND WH2=0 THEN EXP←0 ELSE FINDE;                           02962600 T  0644:3
           IF (GTOGA←W-D-SGN>4) THEN                                           02962610 T  0649:0
           IF EXP< (-1) THEN GO TO E;                                          02962700 T  0651:1
           IF (T1←D-EXP-1)<0 AND GTOGA THEN GO TO E;                           02962800 T  0653:1
           WT ← D; W ← W-4;                                                    02962900 T  0656:3
           D ← DT ← T1;                                                        02963000 T  0658:3
           GO TO FA;                                                           02963100 T  0660:0
GA:                                                                            02963200 T  0660:2
STREAM(P1 ← 0;P2 ← BUFF);                                                      02963300 T  0660:2
  BEGIN  DI ← P2;   DI ← DI + 4;   P1 ← DI;   END;                             02963400 T  0661:3
BUFF ← P;                                                                      02963500 T  0662:3
W ← W + 4;   D ← WT;                                                           02963600 T  0663:1
GO TO COMM;                                                                    02963700 T  0665:1
AST:                                                                           02963800 T  0665:3
STREAM(P1 ← 0;P2 ← BUFF,P3 ← W);                                               02963900 T  0665:3
  BEGIN  DI ← P2;     P3(DS ← LIT "*");   P1 ← DI;   END;                      02964000 T  0667:1
BUFF ← P;                                                                      02964100 T  0669:1
IF GTOG THEN GO TO GA;                                                         02964150 T  0669:3
COMM:                                                                          02964200 T  0670:3
END CONVERT;                                                                   02964300 T  0670:3
COMMENT * * * * * * * * * * END OF DECLARATIONS * * * * * * * * * * * ;       02964400 T  0671:0
IF EDITCODE=0 OR EDITCODE=2 OR EDITCODE=4 THEN                                 02964405 T  0671:0
```

```
    BEGIN                                                                                  02964410 T  0686:0
    P(MKS,FILX,DKADR) ;                                                                    02964415 T  0686:2
    IF EDITCODE=4 THEN P(FI,FMTA,INTCALL(*P(,LISX),@155))                                  02964420 T  0687:1
    ELSE P((-1),FMTA,*P(,LISX),EDITCODE,0,INTCALL(0,@160)) ;                               02964425 T  0691:1
    P(XIT) ;                                                                               02964430 T  0695:2
    END ;                                                                                  02964435 T  0695:3
IF EDITCODE=6 THEN GO ZAP;                                                                 02964450 T  0695:3
FIB ← FILX[NOT 2];      % OPEN FILE IF NOT OPEN                                           02964500 T  0697:0
IF DKADR < 0 THEN BEGIN FLG ← 1; DKADR ←0 END;                                             02964510 T  0698:3
IF FIB[5].[43:1] THEN P(MKS,0,0,FILX,1,SELECT);                                            02964600 T  0701:2
PRNTR←2×(FIB[5].[41:2]≠0) ; %%% IFF FILE IS CLOSED, SETS PRNTR.[46:1]=1.                   02964610 T  0704:2
CKPB; ARRAYSTUFF ← 0;                                                                      02964700 T  0707:0
        IF FIB[0] = 0 THEN                                                                 02964710 T  0708:3
         FIB[0] ← 1 + (EDITCODE =0 OR EDITCODE =2)                                         02964720 T  0709:3
        ELSE                                                                               02964730 T  0712:1
         IF FIB [0] ≠1 + (EDITCODE =0 OR EDITCODE = 2)                                     02964740 T  0713:2
         THEN P(MKS,FIB[6],FILX.[33:15],4,FORTERR);                                        02964750 T  0716:0
IF PRNTR THEN STREAM(TPAR); DS←8LIT" " ;                                                   02964765 T  0719:2
GO TO P(EDITCODE,DUP,ADD);                                                                 02964800 T  0722:3
GO TO NOFL;         %  NO FORMAT, NO LIST                                                 02964900 T  0723:3
GO TO FNOL;         %  FORMAT, NO LIST                                                    02965000 T  0724:1
GO TO BINARY;       %  NO FORMAT, LIST                                                    02965100 T  0724:3
GO TO FMTLST;       %  FORMAT, LIST                                                       02965200 T  0725:1
    NOFL:                                                                                  02965400 T  0725:3
P(XIT) ;                                                                                   02965500 T  0725:3
    FNOL:                                                                                  02965700 T  0726:0
            LSTRN←-1;                                                                      02965800 T  0726:0
            GO TO FRMTCD;                                                                  02965900 T  0727:0
    NMLST:                                                                                 02966000 T  0727:2
        P(XIT);                                                                            02966100 T  0727:2
    BINARY:                                                                                02971000 T  0727:3
P(XIT) ;                                                                                   02971100 T  0727:3
    FMTLST:                                                                                02973100 T  0728:0
        LSTRN ← 1;                                                                         02973200 T  0728:0
            CTOG ← DONETOG ← FALSE;                                                        02973300 T  0728:3
            GETLIST;                                                                       02973400 T  0730:0
    FRMTCD:                                                                                02973500 T  0731:0
            PS ← 0;                                                                        02973600 T  0731:0
            NFCI ← (FI×8) + 2;        % FIRST FORMAT CHARACTER                          02973700 T  0731:3
            IF NOT(NFC="(" OR CHR="%") THEN GO TO FMERR ;                                  02973800 T  0733:2
            NFCI ← (FI×8) + 2;                                                             02973900 T  0737:0
    NFPH:   FORMATCONTROL; % ANAYLSIS OF FORMAT STATEMENT                                  02974000 T  0738:3
            IF FMERRTOG THEN GO TO FMERR;                                                  02974100 T  0740:0
      FMCYC: IF(DONETOG ← ENDLIST) THEN                                                    02974200 T  0741:0
               IF EDITCODE=6 THEN GO ZIPIT ELSE PRNT;                                      02974250 T  0742:2
            IF W + NCR > LCR THEN GO TO FMERR;                                             02974300 T  0745:0
            NCR ← W + NCR;                                                                 02974500 T  0746:3
            CONVERT;                                                                       02974600 T  0748:0
            GETLIST;                                                                       02974700 T  0749:0
            IF (RPT←RPT-1) > 0 THEN GO TO FMCYC;                                           02974800 T  0750:0
            IF EDITCODE=6 THEN GO ZIPIT ELSE                                               02974850 T  0752:1
            GO TO NFPH;                                                                    02974900 T  0753:0
        ZAP:RPT←27;CODE←ATYPE;W←WT←6;D←0;BUFF←TPAR.[33:15];GETLIST;                         02974950 T  0754:0
            LCR←168; GO FMCYC;                                                             02974953 T  0760:0
        ZIPIT: STREAM(P1←BUFF); BEGIN DI←P1; DS← 5 LIT ";END.."; END;                      02974954 T  0761:1
            P(.TPAR,LOD,4,COM,DEL);                                                        02974956 T  0763:2
            BUFF ← TPAR.[33:15];                                                           02974958 T  0764:3
```

```
                STREAM(P1←BUFF);BEGIN DI←P1; 17(DS ← 8 LIT " ");END;                02974960 T  0766:1
                P(XIT);                                                            02974962 T  0769:1
FMERR:                                                                             02975000 T  0769:2
        P(MKS,FIB[6],FILX,[33:15],0,FORTERR);                                      02975100 T  0769:2
    TYPERR:                                                                        02975200 T  0772:0
        P(MKS,FIB[6],FILX,[33:15],2,FORTERR);                                      02975210 T  0772:0
    END FTOUT;                                                                     02975300 T  0774:0
                                                                                          SIZE= 0775 WORDS
```

```
PROCEDURE FORTRANFREEWRITE(FILX,DKADDR,R,W,LISX,NI,NAMS,SUBS) ;%INT @153        02976020 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00403
VALUE DKADDR,R,LISX,W,NI;  INTEGER R,W;  REAL DKADDR,LISX,NI ;                  02976035 T  0000:0
ARRAY SUBS[*], NAMS[*];  NAME FILX ;                                           02976050 T  0000:0
    BEGIN                                                                      02976065 T  0000:0
                                                                               02976080 T  0000:0
    INTEGER LSTRN=19, E, CHR, MAXCHR, PRNTR, TYPE, INDX, SIZE,                 02976095 T  0000:0
            WDTH, MAXWDTH, SGN, CC ;                                           02976110 T  0000:0
                                                                               02976125 T  0000:0
    REAL LISTYPE=20, ARRAYSTUFF=18, ALGOLWRITE=12, SELECT=14, T1, NID,         02976140 T  0000:0
         T2, FORTERR=24, BUFF, BSIZE, FLG, WH1=R, D, WH2, ARY, T3, WH1S,       02976155 T  0000:0
         RNDUP=9, MNTSSA=17, FNR, FNL, ENR, MS3, DECPT, LSS1, SVMAXWDTH,       02976170 T  0000:0
         GTRMI, WH3, HALF, TRZ, NN1, TWOT, VH=-1, VL=-2, TIPE=-4 ;             02976185 T  0000:0
                                                                               02976200 T  0000:0
    NAME LISTADDR ;                                                            02976215 T  0000:0
                                                                               02976230 T  0000:0
ARRAY TEN=22[*], AR1=LISTADDR[*], TPAR=23[*], FPB=3[*], FIB[*] ;               02976245 T  0000:0
                                                                               02976260 T  0000:0
LABEL BACK, START, ITYPE, ALIST, TRU, FALS, TWOPT5, MAXI, SETUP,               02976275 T  0000:0
      LOG8, FTYPE, ETYPE, FUNNYE, CHOOSEI, OVRFLW, FIVEPT, GET3,               02976290 T  0000:0
      HUNT, ZERO, HLF, NOFIT, GOTOQ, ZEROES, FITYPE, ETYPE2, Q,                02976305 T  0000:0
      ETYPE1, DFTYPE, DTYPE, BUMPWH3, THREL, THREH, MAXWDTHOF1, Q1,            02976320 T  0000:0
      MAXI1 ;                                                                  02976321 T  0000:0
                                                                               02976335 T  0000:0
DEFINE DONE = (LSTRN=(-1)) #,                                                  02976350 T  0000:0
       REEL = 3 #,                                                             02976365 T  0000:0
       LOGICAL = 4 #,                                                          02976380 T  0000:0
       INTEGR = 1 #,                                                           02976395 T  0000:0
       DBLPREC = 5 #,                                                          02976410 T  0000:0
       COMPLEXR = 6 #,                                                         02976425 T  0000:0
       COMPLEXI = 7 #,                                                         02976440 T  0000:0
       MAYBE(MAYBE1,MAYBE2,MAYBE3) = CI←CI+MAYBE1; GO TO MAYBE2 ;              02976455 T  0000:0
                                     DS←LIT MAYBE3; MAYBE2: #,                 02976470 T  0000:0
       TWOD = LISTYPE,[38:1] #,                                                02976485 T  0000:0
       INDXF = [18:15] #,                                                      02976500 T  0000:0
       TYPEF = [44:4] #,                                                       02976515 T  0000:0
       SIZEF = [33:15] # ;                                                     02976530 T  0000:0
                                                                               02976545 T  0000:0
SUBROUTINE GETWDTH ;                                                           02976560 T  0000:0
    WDTH←((FNR←E+RNDUP)<FNL←0)+1+GTRMI←1+(ABS(FNR)>9) ;                         02976575 T  0001:0
                                                                               02976590 T  0006:3
REAL SUBROUTINE FTEST ;                                                        02976605 T  0006:3
    FTEST←P(XCH)+(FNR←T2-FNL←FNL-LSS1)<(-E) OR FNR<0 ;                         02976620 T  0007:0
                                                                               02976635 T  0012:0
REAL SUBROUTINE NXTELM ;                                                       02976650 T  0012:0
```

```
            BEGIN                                                                    02976665 T  0012:0
            P(IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX,[40:8],COC)                    02976680 T  0012:0
              ELSE AR1[INDX]) ;                                                      02976695 T  0015:0
            INDX←INDX+1;  NXTELM←P ;                                                 02976710 T  0016:0
            END OF NXTELM ;                                                          02976725 T  0017:2
                                                                                     02976740 T  0017:3
        SUBROUTINE COUNTZ ;                                                          02976755 T  0017:3
            BEGIN                                                                    02976770 T  0018:0
            T3←0 ;                                                                   02976785 T  0018:0
HUNT:       IF LISTYPE MOD TEN[T3←T3+3]=0 THEN GO HUNT ;                             02976800 T  0018:3
            IF LISTYPE MOD TEN[T1←T3-1]≠0                                            02976815 T  0021:3
            THEN T1←T1-1-(LISTYPE MOD TEN[T1-1]≠0) ;                                 02976830 T  0023:3
            END OF COUNTZ ;                                                          02976845 T  0028:1
                                                                                     02976860 T  0028:2
        REAL SUBROUTINE USEXPNOTATION ;                                              02976875 T  0028:2
            BEGIN                                                                    02976890 T  0029:0
            FNL←IF LSS1 THEN P((-E),1) ELSE P(0,E+1) ;                               02976905 T  0029:0
            IF (FNR←P+ENR-FNL-T1)≤0 THEN FNR←1 ;                                     02976920 T  0032:2
            USEXPNOTATION←(FNR+FNL>1+(ENR←ENR-T1)+((DECPT←1)+T3←(-1>E)+              02976935 T  0036:1
                          (ABS(E+1)>9))+(TRZ OR LSS1) AND (ABS(E)≥4 OR               02976950 T  0040:2
                          FNL+FNR>MS3+2)) OR (ENR+T3≤MS3 AND ABS(E)≥5) ;             02976965 T  0045:2
            END OF USEXPNOTATION ;                                                   02976980 T  0050:3
                                                                                     02976995 T  0051:0
        SUBROUTINE ROUNDANDSPLIT ;                                                   02977010 T  0051:0
            BEGIN                                                                    02977025 T  0051:0
            T1←T2←LISTYPE←0;  T3←RNDUP+7 ;                                           02977040 T  0051:0
            IF (MNTSSA←GTRMI-7-RNDUP) LSS 0 THEN                                     02977055 T  0054:0
                BEGIN                                                                02977070 T  0056:1
                P(WH3/TEN[-MNTSSA]-1,,WH3,ISD);  T3←GTRMI ;                          02977085 T  0056:3
BUMPWH3:        IF (WH3←WH3+1)=TEN[T3] THEN E←(LISTYPE←1)+E ;                        02977100 T  0059:3
                END                                                                  02977115 T  0064:0
            ELSE IF MNTSSA LSS 8 THEN                                                02977130 T  0064:0
                BEGIN                                                                02977145 T  0065:1
                IF P(WH2/TEN[8-T2←MNTSSA],,WH2,ISN)=TEN[MNTSSA]                      02977160 T  0065:3
                THEN GO BUMPWH3 ;                                                    02977175 T  0068:2
                END                                                                  02977190 T  0069:2
            ELSE IF P(WH1/TEN[16-MNTSSA],,WH1,ISN)=TEN[T1←MNTSSA-T2←8]               02977205 T  0069:2
                THEN IF (WH2←WH2+1)=TEN[8] THEN GO BUMPWH3 ;                         02977220 T  0073:3
            END OF ROUNDANDSPLIT ;                                                   02977235 T  0077:1
                                                                                     02977250 T  0077:2
        SUBROUTINE OUTPUT ;                                                          02977265 T  0077:2
            BEGIN                                                                    02977280 T  0078:0
            IF PRNTR THEN                                                            02977295 T  0078:0
                BEGIN                                                                02977310 T  0078:1
                FIB[17]←FIB[17]+BSIZE ;                                              02977325 T  0078:3
                P(MKS,2,0,CC,BSIZE,FILX,ALGOLWRITE) ;                                02977340 T  0080:3
                IF NOT(*FILX).[1:19]                                                 02977355 T  0082:2
$ SET OMIT = TIMESHARING                                                             02977370 T  0083:2
$ SET OMIT = NOT(TIMESHARING )                                                       02977400 T  0083:2
                THEN IF FIB[4].[8:4]≠10                                              02977415 T  0083:2
                    THEN P(FILX,@2000000000,36,COM,DEL,DEL) ;                        02977430 T  0085:1
$ POP OMIT                                                                           02977431 T  0087:3
                END                                                                  02977445 T  0087:3
            ELSE IF NOT CC THEN P(MKS,FLG,DKADDR,0,BSIZE,FILX,ALGOLWRITE) ;          02977460 T  0087:3
            P(MKS,FLG,DKADDR,CHR←0,(-1),FILX,ALGOLWRITE,DEL) ;                       02977475 T  0092:3
            IF PRNTR THEN FIB[17]←FIB[17]-BSIZE ;                                    02977490 T  0095:2
            STREAM(BS←BSIZE-1,B←P(DUP).[36:6],BUF←BUFF←(*FILX).[33:15]) ;             02977505 T  0098:1
```

```
          BEGIN                                                                    02977520 T  0101:3
          DS←8LIT" "; SI←BUF; DS←BS WDS; B(DS←32WDS; DS←32WDS) ;                   02977535 T  0101:3
          END ;                                                                    02977550 T  0105:0
     END OF OUTPUT ;                                                               02977565 T  0105:1
                                                                                   02977580 T  0105:2
SUBROUTINE CHECKBUMPANDSKIP ;                                                      02977595 T  0105:2
     BEGIN                                                                         02977610 T  0106:0
     IF P(W←MAXWDTH-WDTH,DUP)≤0 THEN P(DEL,0) ;                                    02977625 T  0106:0
     IF P(P(DUP)+WDTH+2+SVMAXWDTH-MAXWDTH,DUP)+CHR>MAXCHR                          02977640 T  0109:0
     THEN OUTPUT ;                                                                 02977655 T  0112:0
     CHR←P+CHR ;                                                                   02977670 T  0114:0
     STREAM(SKP←LSS1←P:L←LSS1,[36:6],BUFF) ;                                       02977685 T  0115:0
          BEGIN DI←DI+SKP; L(DI←DI+32; DI←DI+32); SKP←DI END ;                     02977690 T  0117:1
     BUFF←P ;                                                                      02977695 T  0119:2
     IF NID≠NI THEN                                                                02977700 T  0120:0
          BEGIN                                                                    02977715 T  0120:3
          STREAM(N←(LSS1←NAMS[NID]),[9:3]:LSS1,T←(NN1≥0),BUFF) ;                   02977730 T  0121:1
               BEGIN                                                               02977745 T  0124:3
               SI←LOC LSS1; SI←SI+2; DS←N CHR; MAYBE(T,L,"("); N←DI;               02977760 T  0124:3
               END ;                                                               02977775 T  0127:1
          BUFF←P ;                                                                 02977790 T  0127:2
          FOR MS3←0 STEP 1 UNTIL NN1 DO                                            02977805 T  0128:0
               BEGIN                                                               02977820 T  0129:0
               STREAM(N←SUBS[MS3],[15:3]:Q←SUBS[MS3],[33:15],BUFF) ;               02977835 T  0129:0
                    BEGIN SI←LOC Q; DS←N DEC; DS←LIT","; N←DI END ;                02977850 T  0131:3
               BUFF←P ;                                                            02977865 T  0133:3
               END ;                                                               02977880 T  0134:1
          STREAM(T←(NN1≥0):R←TYPE=COMPLEXR,I←TYPE=COMPLEXI,BUFF) ;                 02977895 T  0136:2
               BEGIN                                                               02977910 T  0139:3
               CI←CI+T; GO TO L; DI←DI-1; DS←LIT")"; L:                            02977925 T  0139:3
               CI←CI+R; GO TO L1; DS←2LIT"-R"; L1:                                 02977940 T  0141:1
               CI←CI+I; GO TO L2; DS←2LIT"-I"; L2: DS←LIT"="; T←DI ;               02977955 T  0142:2
               END ;                                                               02977970 T  0144:2
          BUFF←P ;                                                                 02977985 T  0144:3
          END ;                                                                    02978000 T  0145:1
     END OF CHECKBUMPANDSKIP ;                                                     02978030 T  0145:1
                                                                                   02978045 T  0145:2
SUBROUTINE BASICONVERT ;                                                           02978060 T  0145:2
     BEGIN                                                                         02978075 T  0146:0
     IF TYPE≠DBLPREC THEN                                                          02978090 T  0146:0
          BEGIN T3←0 ;                                                             02978105 T  0146:3
          IF E>8 THEN BEGIN WH2←WH1 DIV TEN[8]; T2←E-T1←8 END                      02978120 T  0148:0
          ELSE BEGIN T2←0; T1←E END ;                                              02978135 T  0152:2
          END ;                                                                    02978150 T  0154:2
     WDTH←WDTH+T1+T2+T3+SGN+DECPT ;                                                02978165 T  0154:2
     CHECKBUMPANDSKIP ;                                                            02978180 T  0157:3
     STREAM(WH3:WH2,WH1,T3,T2,T1,FNL,DECPT,ENR,SGN,TRZ,BUFF) ;                     02978195 T  0159:0
          BEGIN                                                                    02978210 T  0162:3
          MAYBE(SGN,L1,"-");  SGN←DI ;                                             02978215 T  0162:3
          DI←DI+DECPT; ENR(DS←LIT"0"); SI←LOC WH3; DS←T3 DEC ;                     02978225 T  0164:1
          SI←LOC WH2; DS←T2 DEC; SI←LOC WH1; DS←T1 DEC ;                           02978240 T  0166:3
          TRZ(DS←LIT"0"); WH3←DI; CI←CI+DECPT; GO TO L ;                           02978255 T  0168:1
          SI←SGN; SI←SI+1; DI←SGN; DS←FNL CHR; DS←LIT"."; L:                       02978270 T  0170:2
          END ;                                                                    02978285 T  0172:1
     P(XCH) ;                                                                      02978300 T  0172:2
     END OF BASICONVERT ;                                                          02978315 T  0172:3
                                                                                   02978330 T  0173:0
```

```
     SUBROUTINE FINDE ;                                                                  02978345 T  0173:0
          E←(0&T1[42:3:6]&T1[1:2:1]+12+P(HLF))×P(LOG8) ;                                 02978360 T  0173:0
                                                                                         02978375 T  0177:2
     SUBROUTINE GETW ;                                                                   02978390 T  0177:2
          IF (T1←P(XCH))≥W←MAXCHR/R-P(TWOPT5) THEN W←T1                                  02978405 T  0178:0
          ELSE IF (T1←IF NAMS=0 THEN 30 ELSE P(NAMS[NI],DUP)×6+(P(XCH)>0)                02978420 T  0181:1
               +37)<W THEN MAXCHR←((W←T1)+2)×R ;                                         02978435 T  0186:2
                                                                                         02978450 T  0191:2
%************************:: CODE STARTS HERE ::**************************%             02978465 T  0191:2
                                                                                         02978480 T  0191:2
     HALF←P(HLF) ;                                                                       02978485 T  0191:2
     IF NI<0 THEN                                                                        02978487 T  0202:0
          BEGIN % SPECIAL SINGLE EDIT.                                                   02978489 T  0202:3
          BUFF←R;  NID←NI;  WH1←VH;  WH2←VL ;                                            02978491 T  0203:1
          IF (TYPE←IF TIPE=2 THEN IF VL=0 THEN IF ABS(VH)≤P(MAXI1) THEN                  02978493 T  0206:1
          IF P(VH,,TYPE,ISN)=VH THEN 1 ELSE 3 ELSE 3 ELSE 5 ELSE TIPE)<0                 02978495 T  0209:3
          OR TYPE>5 THEN DO UNTIL FALSE ;                                                02978497 T  0215:3
          W←(W←ABS(W))-SVMAXWDTH←MAXWDTH←IF W≠0 THEN W ELSE 62 ;                         02978499 T  0218:2
          MAXCHR←64;  GO Q1 ;                                                            02978501 T  0223:2
          END ;                                                                          02978503 T  0224:3
     FIB←FILX[NOT 2];  IF DKADDR<0 THEN BEGIN FLG←1; DKADDR←0 END ;                      02978510 T  0224:3
     D←IF R.[1:1] THEN "," ELSE " ";  R←ABS(R) ;                                         02978525 T  0229:1
     IF FIB[5].[43:1] THEN P(MKS,0,0,FILX,1,SELECT) ;                                    02978540 T  0233:0
     CC←(FIB[5] AND 96)≠0 ;                                                              02978555 T  0236:0
     P(MKS,FLG,DKADDR,0,(-1),FILX,ALGOLWRITE) ;                                          02978560 T  0238:0
     IF P(*[FIB[14]],TOP) THEN P(DEL)                                                    02978565 T  0240:0
     ELSE P((T1←(*(4 INX P(XCH))).[36:6])≠0 AND T1≠8 AND T1≠9,SUB) ;                     02978570 T  0241:3
     MAXCHR←P(BSIZE←P)×8 ;                                                               02978575 T  0247:0
     IF PRNTR←((T1←FIB[4].[8:4])=1 OR T1=7 OR T1=12) AND FPB[FIB[4]                      02978585 T  0248:2
               .[13:11]+3].[43:5]<20 THEN                                                02978590 T  0252:3
          BEGIN IF BSIZE>16 THEN BEGIN MAXCHR←132; BSIZE←17 END; END                     02978600 T  0256:0
     ELSE CC←1 ;                                                                         02978615 T  0259:1
     OUTPUT ;                                                                            02978630 T  0260:2
     IF (CC←0)=FIB[0] THEN FIB[0]←1 ;                                                    02978645 T  0262:0
     IF (LSTRN←1)≠FIB[0] AND T1=2                                                        02978660 T  0265:1
     THEN P(MKS,FIB[7],FILX.[33:15],4,FORTERR) ;                                         02978661 T  0267:0
     MAXWDTH←MAXCHR-2 ;                                                                  02978675 T  0270:1
     IF (W←ABS(W))≠0 THEN                                                                02978690 T  0271:2
          BEGIN                                                                          02978705 T  0273:0
          IF W>MAXWDTH THEN W←MAXWDTH ELSE MAXWDTH←W ;                                   02978720 T  0273:2
          IF R≠0 THEN BEGIN P(MAXWDTH); GETW END ;                                       02978735 T  0276:3
          END                                                                            02978750 T  0279:0
     ELSE IF R≠0 THEN BEGIN P(1); GETW; MAXWDTH←W END ;                                  02978765 T  0279:0
     W←W-SVMAXWDTH←MAXWDTH ;                                                             02978780 T  0282:3
     GO START ;                                                                          02978795 T  0284:2
HLF:::   0.5 ;                                                                           02978810 T  0285:0
LOG8:::  0.90308998709 ;                                                                 02978825 T  0286:0
TWOPT5:::2.4999999999 ;                                                                  02978840 T  0287:0
MAXI1::: @0007777777777777 ;                                                            02978842 T  0288:0
BACK::                                                                                   02978855 T  0289:0
     BUFF←P;  IF NI<0 THEN P(BUFF,RTN);  TYPE←ABS(TYPE) ;                                02978870 T  0289:0
     STREAM(D:BUFF); BEGIN SI←LOC D; SI←SI+7; DS←CHR; DI←DI+1; D←DI END ;                 02978885 T  0292:1
     BUFF←P ;                                                                            02978900 T  0294:3
START:                                                                                   02978915 T  0295:2
     IF ARY THEN                                                                         02978930 T  0295:2
          BEGIN                                                                          02978945 T  0295:3
ALIST:    WH1←NXTELM ;                                                                   02978960 T  0296:1
```

```
          IF NID≠NI THEN                                                                    02978975 T  0297:2
               BEGIN                                                                        02978990 T  0298:1
               P(INDX←ARRAYSTUFF.INDXF) ;                                                   02979005 T  0298:3
               IF TYPE≥DBLPREC THEN P((P+1) DIV 2);  T2←P ;                                 02979020 T  0300:0
               IF TYPE≥COMPLEXR THEN TYPE←COMPLEXR+(TYPE=COMPLEXR) ;                        02979035 T  0302:3
               FOR T1←NN1 STEP -1 UNTIL 0 DO                                                02979050 T  0305:3
                    BEGIN                                                                   02979065 T  0307:0
                    SUBS[T1].[33:15]←1+(T2-1) DIV (MS3←SUBS[T1].[18:15]);                   02979080 T  0307:0
                    IF (T2←T2 MOD MS3)=0 THEN T2←MS3 ;                                      02979095 T  0310:3
                    END ;                                                                   02979110 T  0315:0
               END ;                                                                        02979125 T  0317:1
          IF TYPE=DBLPREC THEN WH2←NXTELM;  ARY←INDX<SIZE ;                                 02979140 T  0317:1
          END                                                                               02979155 T  0321:3
     ELSE IF TYPE=COMPLEXR THEN BEGIN WH1←LISTADDR[1]; TYPE←COMPLEXI END                    02979170 T  0321:3
          ELSE BEGIN                                                                        02979185 T  0325:3
               P(ARRAYSTUFF←0);  LISTADDR←[LISX];  TYPE←LISTYPE.TYPEF ;                     02979200 T  0326:1
               IF (NID←ARRAYSTUFF.[3:15]+NI)≠NI                                             02979215 T  0329:0
               THEN NN1←NAMS[NID].[1:8]-1 ;                                                 02979230 T  0330:3
               IF ARY←ARRAYSTUFF.[18:30]≠0 THEN                                             02979245 T  0333:3
                    BEGIN                                                                   02979260 T  0335:2
                    IF NID≠NI THEN                                                          02979275 T  0336:0
                         BEGIN                                                              02979290 T  0336:3
                         SUBS[0].[18:15]←T1←1 ;                                             02979305 T  0337:1
                         FOR T2←1 STEP 1 UNTIL NN1 DO SUBS[T2].[18:15]←T1                   02979320 T  0339:3
                                                      ←T1×SUBS[T2-1].[33:15];               02979335 T  0342:0
                         END ;                                                              02979350 T  0343:0
                    IF TYPE=COMPLEXR THEN TYPE←COMPLEXI ;                                   02979365 T  0347:2
                    SIZE←(INDX←ARRAYSTUFF.INDXF)+ARRAYSTUFF.SIZEF ;                         02979380 T  0349:2
                    P(LISTADDR←MEM[LISTADDR.[18:15]]) ;                                     02979395 T  0352:1
                    TWDT←NOT P(LOD,TOP);  P(DEL) ;                                          02979410 T  0354:1
                    GO ALIST ;                                                              02979425 T  0355:3
                    END ;                                                                   02979440 T  0356:1
               WH1←LISTADDR[0];  P(DEL) ;                                                   02979455 T  0356:1
               IF TYPE=DBLPREC THEN WH2←LISTADDR[1] ;                                       02979470 T  0357:1
               END ;                                                                        02979485 T  0360:0
     IF DONE THEN                                                                           02979500 T  0360:0
          BEGIN                                                                             02979515 T  0361:0
          STREAM(TPAR); 18(DS←8LIT" ") ;                                                    02979530 T  0361:2
          IF NOT PRNTR THEN P(MKS,FLG,DKADDR,0,BSIZE,FILX,ALGOLWRITE) ;                     02979545 T  0364:2
          P(XIT) ;                                                                          02979560 T  0367:1
          END ;                                                                             02979575 T  0367:2
     MAXWDTH←SVMAXWDTH ;                                                                    02979590 T  0367:2
     IF NID≠NI THEN                                                                         02979605 T  0368:1
          BEGIN                                                                             02979620 T  0369:0
          T3←(NN1≥0)+(NAMS[NID].[9:3])+2+NN1+P(ABS(TYPE)≥COMPLEXR,DUP,+);                   02979635 T  0369:2
          FOR T1←NN1 STEP -1 UNTIL 0 DO                                                     02979650 T  0374:0
               BEGIN  T2←0;  MS3←SUBS[T1].[33:15] ;                                         02979665 T  0376:0
               WHILE TEN[T2]≤MS3 DO T2←T2+1 ;                                               02979680 T  0378:1
               T3←T3+T2;  SUBS[T1].[15:3]←T2 ;                                              02979695 T  0381:2
               END ;                                                                        02979710 T  0385:1
          IF (MAXWDTH←MAXWDTH-T3)≤0 THEN GO OVRFLW ;                                        02979725 T  0387:2
          END ;                                                                             02979740 T  0389:3
Q1:  IF TYPE=LOGICAL THEN                                                                   02979755 T  0389:3
          BEGIN                                                                             02979770 T  0390:2
          IF (WDTH←7-WH1←WH1 AND 1)>MAXWDTH THEN WDTH←MAXWDTH ;                             02979785 T  0391:0
          CHECKBUMPANDSKIP ;                                                                02979800 T  0395:0
          STREAM(S←IF WH1 THEN P(TRU) ELSE P(FALS):Z←T1←WDTH>1,                             02979815 T  0396:0
```

```
                    Q←T2←WDTH>2,WDTH←WDTH-T1-T2,F←3+WH1,BUFF) ;                    02979830 T  0399:2
                  BEGIN                                                            02979845 T  0403:1
                  SI←LOC S;  SI←SI+F;  MAYBE(Z,L1,",") ;                           02979860 T  0403:1
                  DS←WDTH CHR;  MAYBE(Q,L2,",");  S←DI ;                          02979875 T  0405:1
                  END ;                                                            02979890 T  0407:1
             GO BACK ;                                                             02979905 T  0407:2
             END OF LOGICAL ;                                                      02979920 T  0408:0
        SGN←WH1,[1:1] ;                                                            02979935 T  0408:0
Q:      IF T3←NOT (GTRMI←ABS(WH1)>P(MAXI)) AND TYPE=INTEGR                         02979950 T  0409:1
        THEN P(WH1,,WH1,ISD) ;                                                     02979965 T  0411:2
        T2←(MS3←MAXWDTH-3-SGN)+2;  WH1←ABS(WH1) ;                                  02979980 T  0413:3
        IF TYPE=BBLPREC THEN                                                       02979995 T  0417:2
             BEGIN                                                                 02980010 T  0418:1
             WH1←T1←P(WH2,WH1,0,TEN[0],DLM);  WH2←P ;                              02980025 T  0418:3
             IF WH1=0 THEN BEGIN TYPE←-DBLPREC; GO ZERO END;  FINDE ;              02980040 T  0421:3
             IF MAXWDTH<7                                                          02980055 T  0426:0
             THEN P(WH2,WH1,HALF,(NOT P(MAXI)) AND WH1,DLA,,WH1S,←,DEL) ;          02980070 T  0426:1
             IF LSS1←E LSS 0 THEN P(0,TEN[0],TEN[69-E],TEN[-E],DLD)                02980085 T  0430:0
             ELSE P(TEN[69+E],TEN[E]) ;                                            02980100 T  0434:2
             T1←P;  T3←P ;                                                         02980115 T  0436:2
             IF T1 GEQ WH1 THEN IF T1>WH1 OR T3>WH2 THEN E←E-1 ;                   02980130 T  0437:2
             ENR←24 ;                                                              02980145 T  0442:1
             P(WH2,WH1,TEN[69+ABS(E)],TEN[ABS(E)],IF LSS1←E LSS 0 THEN             02980160 T  0443:0
               P(DLM) ELSE P(DLD)) ;                                               02980175 T  0446:3
             T1←P;  T3←P ;                                                         02980190 T  0448:1
             IF T1≥P(THREH) THEN IF T1>P(THREH) OR T3>P(THREL) THEN ENR←23 ;       02980205 T  0449:1
             P(WH2,WH1,TEN[(T1←ABS(ENR-E-1))+69],TEN[T1],IF ENR≤E THEN             02980220 T  0453:0
               P(DLD) ELSE P(DLM)) ;                                               02980235 T  0458:0
             WH1←P;  RNDUP←ENR=24 ;                                                02980250 T  0459:2
             P(T3←P,WH1,TEN[85],TEN[16],DLD,HALF,-,,WH3,ISD,DEL,T3,WH1,0,WH3       02980265 T  0461:1
               ,TEN[85],TEN[16],DLM,DLS) ;                                         02980280 T  0465:1
             WH1←P ;                                                               02980295 T  0467:0
             P(T3←P,WH1,0,TEN[8],DLD,HALF,-,,WH2,ISD,DEL,T3,WH1,0,WH2,0,TEN[       02980310 T  0467:2
               8],DLM,DLS,,WH1,ISD,DEL) ;                                          02980325 T  0471:3
             IF P(0,0)=LISTYPE←WH1 THEN                                            02980340 T  0473:2
                  BEGIN P(DEL,8) ;                                                 02980355 T  0475:0
                  IF (LISTYPE←WH2)=0 THEN BEGIN P(DEL,16); LISTYPE←WH3 END ;       02980370 T  0476:0
                  END ;                                                            02980385 T  0479:0
             COUNTZ;  T1←P+T1 ;                                                    02980400 T  0479:0
             IF USEXPNOTATION THEN                                                 02980415 T  0481:0
                  BEGIN                                                            02980430 T  0482:0
DTYPE:            IF ENR+T3>MS3 THEN                                               02980445 T  0482:2
                       IF E=T2 THEN BEGIN DECPT←FNR←0; GO DFTYPE END               02980460 T  0483:3
                       ELSE IF (ENR←MS3-T3)≤0 THEN                                 02980475 T  0487:1
GOTOQ:                      BEGIN WH1←WH1S; TYPE←-DBLPREC; GO Q END ;             02980490 T  0489:2
                  GTRMI←ENR;  ROUNDANDSPLIT ;                                      02980505 T  0492:1
                  IF NOT (9≠E OR T3≠1) THEN GO GOTOQ ;                             02980520 T  0494:0
                  IF LISTYPE THEN P(TEN[(T3←T3+(IF LSS1 THEN -10=E ELSE -(9=       02980535 T  0496:1
                            E)))-1],,WH3,ISD) ;                                    02980550 T  0499:3
                  RNDUP←1;  GO ETYPE1 ;                                            02980565 T  0502:2
MAXI:::   @0007777777777777 ;                                                      02980580 T  0503:3
THREH:::  @1143013331500045 ;                                                      02980595 T  0505:0
THREL:::  @0003112121167260 ;                                                      02980610 T  0506:0
TRU:::    "TRUE" ;                                                                 02980625 T  0507:0
FALS:::   "FALSE" ;                                                                02980640 T  0508:0
                  END ;                                                            02980655 T  0509:0
             IF FNL←FNR≤T2 THEN                                                    02980670 T  0509:0
```

```
                    BEGIN                                                                  02980685 T  0510:1
DFTYPE:             ENR←TRZ←0;  T1←FNL ;                                                   02980700 T  0510:3
                    IF LSS1 THEN ENR←FNL-E-1                                               02980715 T  0512:3
                    ELSE IF 23+RNDUP≤FNL THEN                                              02980730 T  0514:1
                         BEGIN TRZ←FNL+FNR-T1←23+RNDUP; FNR←0 END ;                        02980745 T  0517:0
                    GTRMI←T1+FNR-ENR;  ROUNDANDSPLIT ;                                     02980760 T  0521:0
                    IF LISTYPE THEN                                                        02980775 T  0524:0
                         BEGIN                                                             02980790 T  0524:1
                         IF DECPT THEN DECPT←FNR+TRZ≥1                                     02980805 T  0524:3
                         ELSE BEGIN                                                        02980820 T  0526:1
                              T1←T2←0;  DECPT←WH3←T3←RNDUP←1;  GETWDTH ;                   02980835 T  0527:3
                              IF (TRZ←-WDTH+WDTH←MS3+1)≤0 THEN GO GOTOQ ;                  02980850 T  0532:0
                              GO ETYPE2 ;                                                  02980865 T  0535:2
                              END ;                                                        02980880 T  0536:0
                         FNL←FNL+(E≥0) ;                                                   02980895 T  0536:0
                         IF LSS1 THEN IF NOT (TRZ←(ENR←ENR-1)≥0) THEN ENR←0 ;              02980910 T  0537:3
                         IF T3<DECPT THEN GO GOTOQ ;                                       02980925 T  0542:1
                         P(TEN[(T3←T3+1-DECPT)-1],,WH3,ISD) ;                              02980940 T  0543:2
                         END ;                                                             02980955 T  0546:2
FITYPE:             WDTH←ENR+TRZ;  BASICONVERT;  GO BACK ;                                 02980970 T  0546:2
                    END ;                                                                  02980985 T  0549:2
               P(WH3/TEN[RNDUP+6]>5);  IF NOT FTEST THEN GO DFTYPE ;                       02981000 T  0549:2
               GO DTYPE ;                                                                  02981015 T  0553:2
               END OF DBLPREC ;                                                            02981030 T  0554:0
          IF MAXWDTH=1 THEN GO TO MAXWDTHOF1 ;                                             02981045 T  0554:0
          T1←TEN[0]×WH1 ;                                                                  02981060 T  0555:1
          IF E←WH1≠0 THEN                                                                  02981075 T  0556:3
               BEGIN FINDE; E←E-((IF E>0 THEN TEN[E] ELSE 1/TEN[-E])>T1) END ;             02981090 T  0558:0
          IF T3 AND E≤T2 THEN                                                              02981105 T  0565:0
               BEGIN E←E+1 ;                                                               02981120 T  0566:1
ITYPE:         DECPT←0;  GO SETUP ;                                                       02981135 T  0568:0
               END ;                                                                       02981150 T  0569:1
          IF WH1=0 THEN                                                                    02981165 T  0569:1
ZERO:          BEGIN FNR←0; FNL←1; GO FTYPE END ;                                         02981180 T  0570:0
          RNDUP←(MNTSSA←P(WH1←T1,TEN[ABS(E)],IF LSS1←0>E THEN                              02981195 T  0572:2
                P(×) ELSE P(/))) GEQ 5 ;                                                 02981210 T  0575:1
          LISTYPE←P(WH1,TEN[ABS((ENR←12-(MNTSSA>P(FIVEPT)))-E-1)],                       02981225 T  0578:1
                    IF GTRMI THEN P(/) ELSE P(×)) ;                                      02981240 T  0581:3
          COUNTZ ;                                                                       02981255 T  0584:0
          IF GTRMI OR USEXPNOTATION THEN                                                 02981270 T  0585:0
               BEGIN                                                                     02981285 T  0586:1
               IF ENR+T3 LEQ MS3 THEN                                                    02981300 T  0586:3
ETYPE:              BEGIN                                                              02981315 T  0588:0
                    P(WH1) ;                                                             02981330 T  0588:2
                    IF NOT RNDUP←TYPE≠INTEGR OR DECPT=0 THEN                             02981345 T  0588:3
                         BEGIN                                                           02981360 T  0591:1
                         ENR←P(E≥10 AND (E←E-ENR)≤10,DUP)+1+ENR ;                        02981375 T  0591:3
                         DECPT←0;  P(TEN[E←-P+E],/) ;                                    02981390 T  0596:1
                         END                                                             02981405 T  0598:3
                    ELSE P(TEN[ABS(E+1-ENR)],IF E<ENR THEN P(×) ELSE P(/)) ;             02981420 T  0598:3
                    P(,WH1,ISD) ;                                                        02981435 T  0603:1
                    IF WH1=TEN[ENR] THEN                                                 02981450 T  0603:3
                         BEGIN                                                           02981465 T  0604:3
                         IF (ENR←ENR+P(E+RNDUP,IF LSS1 THEN P=(-10) ELSE-(P=9)           02981480 T  0605:1
                         ))<0 THEN ENR←DECPT←0 ;                                         02981495 T  0608:2
                         E←E+1;  P(TEN[ABS(ENR-1)],,WH1,ISD) ;                           02981510 T  0612:0
                         END ;                                                           02981525 T  0615:0
```

```
ETYPE1:         IF P(RNDUP AND DECPT=0,DUP) THEN RNDUP←0 ;                          02981540 T  0615:0
                GETWDTH ;                                                           02981555 T  0617:3
                IF P THEN IF E=(-10) AND MNTSSA<1+HALF THEN ENR←0                   02981570 T  0619:0
                          ELSE DECPT←FNL←E=9 ;                                      02981585 T  0622:2
                IF MAXWDTH<SGN+DECPT+(E←ENR)+WDTH THEN                              02981600 T  0625:2
NOFIT:               BEGIN IF NOT LSS1 THEN GO OVRFLW; GO ZEROES END ;              02981615 T  0628:1
                ENR←TRZ←0 ;                                                         02981630 T  0630:0
ETYPE2:         BASICONVERT;  BUFF←P ;                                              02981645 T  0631:1
                STREAM(GTRMI:FNR←ABS(FNR),S←FNR<0,C←IF ABS(TYPE)=DBLPREC            02981660 T  0632:2
                     THEN "D" ELSE "E",BUFF) ;                                      02981675 T  0635:1
                     BEGIN                                                          02981690 T  0637:2
                     SI←LOC C; SI←SI+7; DS←CHR; MAYBE(S,L,"-") ;                    02981705 T  0637:2
                     SI←LOC FNR; DS←GTRMI DEC; GTRMI←DI ;                           02981720 T  0639:2
                     END ;                                                          02981735 T  0640:2
                GO BACK ;                                                           02981750 T  0640:3
FIVEPT:::       5,49755813885 ;                                                     02981765 T  0641:1
GET3:           P(USEXPNOTATION,DEL) ;                                              02981780 T  0643:0
                END ;                                                               02981795 T  0644:1
FUNNYE:   IF NOT(E≠T2 OR GTRMI) THEN GO CHOOSEI ;                                   02981810 T  0644:1
          IF (ENR←MS3-T3)≥1 THEN GO ETYPE;  DECPT←0 ;                               02981825 T  0646:0
          IF NOT ((ENR←MS3≥T3) OR RNDUP OR MNTSSA<1+HALF) THEN GO NOFIT ;           02981840 T  0649:0
          GO ETYPE ;                                                                02981855 T  0652:2
ZEROES:   WH1←FNL←0; IF SGN AND MAXWDTH=2 THEN GO MAXWDTHOF1; FNR←T2 ;              02981870 T  0653:1
FTYPE:    P(WH1×TEN[FNR],,WH1,ISD);  DECPT←1 ;                                      02981885 T  0657:1
          IF -2<E AND WH1=TEN[E←FNL+FNR] THEN                                       02981900 T  0659:2
             BEGIN FNL←FNL+1; P(TEN[(E←E+1-DECPT←FNR>0)-1],,WH1,ISD) END;            02981915 T  0662:3
SETUP:       ENR←TRZ←0;  GO FITYPE ;                                                02981930 T  0668:2
             END ;                                                                  02981945 T  0670:1
     IF FNR+FNL LEQ T2 THEN GO FTYPE ;                                              02981960 T  0670:1
     IF LSS1 AND SGN THEN IF MAXWDTH=2 THEN GO MAXWDTHOF1 ;                         02981975 T  0672:0
     P(RNDUP);  IF NOT FTEST THEN GO FTYPE ;                                        02981990 T  0674:2
     GO FUNNYE ;                                                                    02982005 T  0676:2
MAXWDTHOF1:                                                                         02982020 T  0677:0
     E←0 ;                                                                          02982035 T  0677:0
CHOOSEI:                                                                            02982050 T  0677:3
     IF NOT GTRMI THEN P(WH1,,WH1,ISD) ;                                            02982065 T  0677:3
     IF TEN[E←E+1]=WH1 THEN GO GET3 ;                                               02982080 T  0679:2
     IF P(MAXWDTH=1,DUP) THEN SGN←SGN AND WH1≠0 ;                                   02982095 T  0682:0
     IF NOT (P AND (SGN OR WH1>9)) THEN GO ITYPE ;                                  02982110 T  0685:1
OVRFLW:                                                                             02982125 T  0687:1
     WDTH←MAXWDTH;  P(NID);  NID←NI;  CHECKBUMPANDSKIP;  NID←P ;                    02982140 T  0687:1
     STREAM(SVMAXWDTH:S←SVMAXWDTH,[36:6],BUFF) ;                                    02982155 T  0691:2
          BEGIN SVMAXWDTH(DS←LIT"x"); S(32(DS←2LIT"x"));SVMAXWDTH←DI END;           02982160 T  0693:2
     GO BACK ;                                                                      02982170 T  0696:3
     END OF FORTRANFREEWRITE ;                                                      02982185 T  0697:2
                                                                                              SIZE= 0698 WORDS
```

```
PROCEDURE FINNAME;          %154                                                    02982500 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00427
          BEGIN                                                                     02982520 T  0000:0
COMMENT FILX       FILE TOP IO DESCRIPTOR                                         02982540 T  0000:0
        FMTA       FORMAT OR NAMELIST OR 0                                        02982560 T  0000:0
        LISX       ACCIDENTAL ENTRY DESC, OR 0                                    02982580 T  0000:0
                   ;                                                              02982600 T  0000:0
```

```
REAL      PARL      =  -1,                                                  02982740 T  0000:0
          EOFL      =  -2,                                                  02982760 T  0000:0
         FORTERR    = 24,                                                   02982780 T  0000:0
          LISX      =  -3,                                                  02982820 T  0000:0
          FI        =  -5,                                                  02982840 T  0000:0
          DKADR     =  -6,                                                  02982860 T  0000:0
          READINT   =  13,                                                  02982880 T  0000:0
          SELECT    =  14;                                                  02982900 T  0000:0
ARRAY     FMTA      =  -4[*];                                               02982920 T  0000:0
NAME      FILX      =  -7,                                                  02982940 T  0000:0
          MEM       =   2;                                                  02982960 T  0000:0
INTEGER     JUNK1       = 17;                                               02982980 T  0000:0
REAL        LISTYPE     = 20;                                               02983000 T  0000:0
ARRAY       PRTBASE     = 10[*],                                            02983020 T  0000:0
            TEN         = 22[*],                                            02983040 T  0000:0
            FIB[*];                                                         02983060 T  0000:0
NAME        LISTADR;                                                        02983080 T  0000:0
REAL        BUFF                        , %  FIRST BUFFER POSITION          02983100 T  0000:0
            BSIZE                       , %  ARGUMENTS                      02983120 T  0000:0
            NBC,                                                            02983140 T  0000:0
            NLI,NLE,SBS,                                                    02983160 T  0000:0
            NFCI,                                                           02983180 T  0000:0
            DH1,                                                            02983200 T  0000:0
            WH1                         , %                                 02983220 T  0000:0
            WH2;                                                            02983240 T  0000:0
REAL        NAMEV;                                                          02983260 T  0000:0
NAME        W1;                                                             02983280 T  0000:0
INTEGER     RPT,                                                            02983300 T  0000:0
            W                           , %  FIELD                         02983320 T  0000:0
            WT                          , %       WIDTH                     02983340 T  0000:0
            T1                          , %                                 02983360 T  0000:0
            D1                          , %            LA-                  02983380 T  0000:0
            D2                          , %              CE-                02983400 T  0000:0
            CNT,                                                            02983420 T  0000:0
            EXP                         , %  EXPONENT                       02983440 T  0000:0
            EXPSGN,                                                         02983460 T  0000:0
            NCR                         , %  CURRENT BUFFER POSITION        02983480 T  0000:0
            LCR                         , %  BUFFER SIZE IN CHARACTERS       02983500 T  0000:0
                  CHR                     ;                                 02983520 T  0000:0
BOOLEAN     DONETOG                     , %  RETURN AFTER WRITE             02983540 T  0000:0
            SGN                         , %  SIGN                           02983560 T  0000:0
            LGTG,                                                           02983580 T  0000:0
            DTAERR,                                                         02983600 T  0000:0
            NLT,                                                            02983620 T  0000:0
            NFL,                                                            02983625 T  0000:0
            CTOG;                                                           02983640 T  0000:0
DEFINE LOGV       = 4#,                                                     02983660 T  0000:0
       INTEGV      =1#,                                                     02983680 T  0000:0
       DBLV       = 5#,                                                     02983700 T  0000:0
       CMPLXV     = 6#,                                                     02983720 T  0000:0
         SPC = 3#,                                                          02983740 T  0000:0
         NUM = 2#,                                                          02983760 T  0000:0
         ID = 1#,                                                           02983780 T  0000:0
       KIND        = (FIB[4].[8:4])#,                                       02983800 T  0000:0
       MAX        = @7777777777777#,                                       02983820 T  0000:0
       ELMTYP      = LISTYPE . [44:4]#,                                     02983840 T  0000:0
       DLN         = (LISTYPE.[44:4] = DBLV)#,                              02983860 T  0000:0
```

```
        CMPLX       = (LISTYPE.[44:4] = CMPLXV)#,                                  02983880 T  0000:0
        TWOD        = LISTYPE.[38:1]#,                                             02983900 T  0000:0
        SIZEF       = [33:15]#,                                                    02983920 T  0000:0
        BASEF       = [18:15]#;                                                    02983940 T  0000:0
LABEL   NMLST,NLERR,NLP,NLPA,NRP,NLISRT,NLL,NLPB,NRPL;                             02983960 T  0000:0
COMMENT  * * * * * START OF SUBROUTINE DECLARATIONS * * * * * * * * * ;           02983980 T  0000:0
SUBROUTINE CKPB;                                                                   02984000 T  0000:0
BEGIN COMMENT   INITIALIZE FILE AND ACQUIRE RECORD SIZE;                           02984020 T  0001:0
LCR ← 8×(BSIZE ← P(MKS,DKADR,1,FILX,READINT));                                     02984040 T  0001:0
BUFF ← (*FILX).[33:15]; NCR ← 0;                                                   02984060 T  0003:3
END CKPB;                                                                          02984080 T  0006:0
SUBROUTINE READS;                                                                  02984100 T  0006:1
BEGIN                                                                              02984120 T  0007:0
P(MKS,DKADR,0,FILX,READINT);                                                       02984140 T  0007:0
IF DONETOG THEN P(XIT);                                                            02984160 T  0008:1
IF ((*FILX).[27:1]) THEN P(XIT);                                                   02984280 T  0009:1
CKPB;                                                                              02984300 T  0011:0
END READS;                                                                         02984320 T  0012:0
        % PARAMETERS FOR LIST CONTROL                                             02984340 T  0012:1
BOOLEAN ATOG,TWDT;                                                                 02984360 T  0012:1
ARRAY AR1 = LISTADR[*];                                                            02984380 T  0012:1
    INTEGER INDX, SIZE;                                                            02984400 T  0012:1
DEFINE NXTELM = IF TWDT THEN P(*[AR1[INDX,[33:7]]],INDX,[40:8],CDC)                02984420 T  0012:1
                        ELSE [AR1[INDX]]#;                                         02984440 T  0012:1
SUBROUTINE SKPC;    % SKIPS CURRENT CHARACTERS. PUTS NEXT CHARACTERS                02984460 T  0012:1
    BEGIN;          % IN NBC                                                       02984480 T  0013:0
        STREAM(P1←BUFF,P2←0:P3←0);                                                 02984500 T  0013:0
        BEGIN                                                                      02984520 T  0014:2
            SI ← P1; SI ← SI +1; P1 ← SI;                                          02984540 T  0014:2
            DI ← LOC P2; DI ← DI+7; DS ← CHR;                                      02984560 T  0015:1
        END;                                                                       02984580 T  0016:0
        NBC ← P; BUFF ← P;                                                         02984600 T  0016:1
        WT ← WT -1;                                                                02984620 T  0017:1
    END SKPC;                                                                      02984640 T  0018:2
SUBROUTINE SCALE;                                                                  02984660 T  0018:3
    BEGIN                                                                          02984680 T  0019:0
        IF (D1 ← D1 + CNT) > 11                                                    02984700 T  0019:0
         THEN DOUBLE(WH1,WH2,TEN[CNT],TEN[69+CNT],×,                               02984720 T  0020:1
                    DH1,0,+,←,WH1,WH2)                                             02984740 T  0023:3
         ELSE WH1← WH1×TEN[CNT]+DH1;                                               02984760 T  0025:1
        DH1 ← 0;                                                                   02984780 T  0028:0
    END SCALE;                                                                     02984800 T  0028:3
SUBROUTINE GETNUM;                                                                 02984820 T  0029:0
    BEGIN;                                                                         02984840 T  0029:0
        STREAM(P1←BUFF,P2←IF WT ≤ 8 THEN WT ELSE 8,P3←0,P4←0:P5←0);                 02984860 T  0029:0
        BEGIN                                                                      02984880 T  0033:0
            SI ← P1;                                                               02984900 T  0033:0
            P2(IF SC = " " THEN TALLY ← TALLY +1                                   02984920 T  0033:1
                           ELSE                                                    02984940 T  0034:1
               BEGIN  IF SC ≥ "0" THEN TALLY ← TALLY + 1                           02984960 T  0034:3
                 ELSE JUMP OUT;                                                    02984980 T  0035:1
                 END;                                                              02985000 T  0036:1
                 SI ← SI + 1);                                                     02985020 T  0036:1
            P2 ← TALLY;                                                            02985040 T  0036:3
            SI ← P1; DI ← LOC P3; DS ← P2 OCT;                                     02985060 T  0037:0
            P1 ← SI;                                                               02985080 T  0038:0
            DI ← LOC P4; DI ← DI + 7; DS ← CHR;                                    02985100 T  0038:1
```

```
        END;
        NBC ← P; DH1 ← P; CNT ← P; BUFF ← P;
    END GETNUM;
SUBROUTINE GETSIGN;
    BEGIN;
        STREAM(P1←BUFF,P2←(IF WT > 63 THEN 63 ELSE WT),P3←0,P4←(-1);
            P5←0);
        BEGIN
           SI←P1;  DI←P2 ;
           P2(DI←DI"8;  IF SC≠" " THEN JUMP OUT TO L1;
                SI ← SI + 1; TALLY ← TALLY + 1);
            P1 ← SI;
            GO TO RTNSGN;
        L1:  IF SC ≥ "0" THEN
        BEGIN
        L3:  P2 ← TALLY;
        L2:  P5(P1←DI; TALLY←P2; P1(IF SC≠" " THEN
             JUMP OUT;  TALLY←TALLY+1;  SI←SI+1);  P2←TALLY);  P1←SI ;
             DI ← LOC P4; DS ← 7 LIT "0"; DS ← CHR;
             GO TO RTNSGN;
        END;
             IF SC ="," THEN GO TO L3;
             TALLY ← TALLY+1;
             P2 ← TALLY;
             TALLY←1;  P5←TALLY ;
             IF SC="-" THEN TALLY←1 ELSE IF SC="+" THEN TALLY←0 ELSE
             IF SC="&" THEN TALLY←0 ELSE
                BEGIN TALLY←0;  P1←TALLY;  GO TO RTNSGN END;
             P3 ← TALLY;
             SI ← SI + 1;
             GO TO L2;
        RTNSGN:
        END;
        NBC←P;  SGN←P;  CNT←P;  DTAERR←((BUFF←P)=0) ;
    END GETSIGN;
    LABEL     NCRTN,BLSGN;
SUBROUTINE NUMCONVERT;
    BEGIN
        DH1 := D1 := D2 := EXP := EXPSGN := 0;
        WH1 ← WH2 ← -0;
        WT ← W;
    BLSGN:
        GETSIGN;
        IF DTAERR THEN GO TO NCRTN ;
        WT ← WT - CNT; IF NBC <0       % BLANK FIELD
          THEN IF WT ≤ 0 THEN GO TO NCRTN ELSE GO TO BLSGN;
        IF NBC ≤ 9 THEN
    BEGIN
        GETNUM;  WH1 ← DH1;
        IF (WT ← WT - (D1 ← CNT)) ≤ 0 THEN GO TO NCRTN;
        WHILE NBC ≤ "9" OR NBC = " " DO
    BEGIN
        GETNUM; SCALE;
        IF (WT ← WT"CNT) ≤ 0 THEN GO TO NCRTN;
    END;
    END;
        IF NBC = "." THEN
```

```
02985120 T  0039:0
02985140 T  0039:1
02985160 T  0041:1
02985180 T  0041:2
02985200 T  0042:0
02985220 T  0042:0
02985240 T  0045:2
02985260 T  0046:1
02985280 T  0046:1
02985300 T  0046:3
02985320 T  0048:2
02985340 T  0049:1
02985360 T  0049:2
02985380 T  0049:3
02985400 T  0050:1
02985420 T  0050:1
02985440 T  0050:2
02985460 T  0053:0
02985480 T  0055:0
02985500 T  0056:3
02985520 T  0057:0
02985540 T  0057:0
02985560 T  0057:3
02985580 T  0058:0
02985600 T  0058:1
02985620 T  0058:3
02985640 T  0060:3
02985660 T  0061:3
02985680 T  0062:2
02985700 T  0062:3
02985720 T  0063:0
02985740 T  0063:1
02985760 T  0063:1
02985780 T  0063:2
02985800 T  0066:2
02985820 T  0066:3
02985840 T  0066:3
02985860 T  0067:0
02985880 T  0067:0
02985900 T  0069:3
02985920 T  0071:1
02985940 T  0072:0
02985960 T  0072:0
02985980 T  0073:0
02986000 T  0074:0
02986020 T  0075:3
02986040 T  0078:1
02986060 T  0079:0
02986080 T  0079:2
02986100 T  0081:3
02986120 T  0084:2
02986140 T  0086:3
02986160 T  0086:3
02986180 T  0089:0
02986200 T  0091:1
02986220 T  0091:3
02986240 T  0091:3
```

```
        BEGIN                                                                    02986260 T  0092:2
             SKPC;                                                               02986280 T  0093:0
             IF WT≤0 THEN GO TO NCRTN ;                                          02986300 T  0094:0
             WHILE (NBC ≤ "9") OR (NBC = " ") DO                                 02986320 T  0095:1
        BEGIN                                                                    02986340 T  0097:2
             GETNUM; SCALE;                                                      02986360 T  0097:2
             D2 ← D2 + CNT;                                                      02986380 T  0100:0
             IF ( WT ← WT - CNT) ≤ 0 THEN GO TO NCRTN;                           02986400 T  0101:1
        END;                                                                     02986420 T  0103:2
        END;                                                                     02986440 T  0104:0
             IF NBC = "D" OR NBC = "E" THEN SKPC;                                02986460 T  0104:0
             IF WT≤0 THEN BEGIN DTAERR←TRUE;  GO TO NCRTN END ;                  02986480 T  0107:0
             IF (NBC="+") OR (NBC="&") OR (NBC=" ") OR (EXPSGN←(NBC="-"))        02986500 T  0109:2
               THEN SKPC;                                                        02986520 T  0113:0
             IF WT≤0 THEN BEGIN DTAERR←TRUE;  GO TO NCRTN END ;                  02986540 T  0115:0
             IF NBC > "9" THEN DTAERR ← TRUE                                     02986560 T  0117:2
             ELSE                                                                02986580 T  0119:0
        BEGIN                                                                    02986600 T  0119:2
             GETNUM;                                                             02986620 T  0120:0
             EXP ← IF EXPSGN THEN (-DH1) ELSE DH1;                               02986640 T  0121:0
             IF (WT←WT-CNT) ≤ 0 THEN GO TO NCRTN;                                02986660 T  0123:2
             IF NOT NLT THEN WHILE WT > 0 DO SKPC;                               02986680 T  0125:3
        END;                                                                     02986700 T  0129:2
        NCRTN:                                                                   02986720 T  0129:2
             IF WH1 = 0 THEN IF SGN THEN WH1 ← -0;                               02986740 T  0129:2
        END NUMCONVERT;                                                          02986760 T  0132:2
   LABEL NMBLNK;                                                                 02986780 T  0132:3
   REAL SUBROUTINE NMSCN;                                                        02986800 T  0132:3
   BEGIN;                                                                        02986820 T  0133:0
   NMBLNK:                                                                       02986840 T  0133:0
        IF NCR ≥ LCR THEN READS;                                                 02986860 T  0133:0
        STREAM(P1←BUFF,P2←(IF(T1←LCR-NCR)>63 THEN 63 ELSE T1),                   02986880 T  0135:0
             P3 ←0:P4←0);                                                        02986900 T  0138:3
        BEGIN                                                                    02986920 T  0139:3
            SI ←P1;                                                              02986940 T  0139:3
             P2(IF SC ≠" " THEN JUMP OUT TO L1;                                  02986960 T  0140:0
                SI ← SI + 1; TALLY ← TALLY + 1;);                                02986980 T  0141:2
             P2 ← TALLY; TALLY ← 1;                                              02987000 T  0142:1
             GO TO L2;                                                           02987020 T  0142:3
        L1:  P2 ← TALLY; TALLY ← 0;                                              02987040 T  0143:0
        L2:  P1 ← TALLY; P3 ← SI;                                                02987060 T  0143:2
        END;                                                                     02987080 T  0144:0
        BUFF ← P; NCR ← NCR+P; IF P THEN GO TO NMBLNK;                           02987100 T  0144:1
        STREAM(P1←BUFF,P2←0,P3←0,P4←"      ",P5←0:NFL) ;                         02987120 T  0146:2
        BEGIN                                                                    02987140 T  0148:3
             SI ← P1;                                                            02987160 T  0148:3
             NFL(IF SC≥"0" THEN JUMP OUT TO NU) ;                                02987180 T  0149:0
             IF SC = ALPHA THEN                                                  02987200 T  0150:3
        BEGIN                                                                    02987220 T  0151:1
             DI ← LOC P4; DI ← DI + 2;                                           02987240 T  0151:1
             6(IF SC < "A" THEN JUMP OUT;                                        02987260 T  0151:3
               DS ← CHR; TALLY ← TALLY + 1);                                     02987280 T  0153:0
             P1 ← SI; P3 ← TALLY;                                                02987300 T  0153:3
             TALLY ← 1; GO TO EXIT;                                              02987320 T  0154:1
        END;                                                                     02987340 T  0154:3
             NFL(IF SC≠"-" THEN IF SC≠"+" THEN IF SC≠"&" THEN JUMP OUT;          02987360 T  0154:3
                 JUMP OUT TO NU) ;                                               02987380 T  0156:3
```

```
                    TALLY ← 1; P3 ← TALLY;
                    DI ← LOC P2; DI ← DI + 7;
                    IF SC ="#" THEN BEGIN DS←LIT"="; SI←SI+1 END ELSE

                    IF SC ="[" THEN BEGIN DS←LIT ")"; SI ← SI+1 END ELSE
                    IF SC ="%" THEN BEGIN DS←LIT "("; SI ← SI+1 END ELSE
                     DS ←CHR;
                    P1 ← SI;
                    TALLY ← 3; GO TO EXIT;

               NU:
                    P1 ← SI; TALLY ← 2;
               EXIT:
                    P5 ← TALLY;
               END;
               T1 ← P; NAMEV ← P; NCR ← NCR + P; NBC ← P; BUFF ← P;
               IF T1 = ID THEN NBC ← NAMEV;
               NMSCN ← T1;
          END NMSCN;
     SUBROUTINE NLCONV;
          BEGIN
               IF NBC = "." THEN
          BEGIN;
               STREAM(P1←BUFF:P2←0);
               BEGIN DI ← P1; DI ← DI - 1; P1 ← DI; END;
               BUFF ← P; NCR ← NCR - 1;
          END;
               W ← LCR-NCR;
               NUMCONVERT;
               IF (NCR ← NCR + (W-WT)) ≥ LCR THEN READS;
               T1 ← EXP - D2;
               IF WH1 > MAX
                 THEN
               IF T1 ≥ 0 THEN DOUBLE(WH1,WH2,TEN[T1],TEN[69+T1],×,
                                    ←,WH1,WH2)
                         ELSE DOUBLE(WH1,WH2,TEN[-T1],TEN[69-T1],/,
                                    ←,WH1,WH2)
                 ELSE
               IF T1 ≥ 0 THEN WH1 ← WH1 × TEN[T1]
                         ELSE WH1 ← WH1 / TEN[-T1];
               IF SGN THEN WH1 ← - WH1;
          END NLCONV;
     LABEL NLR,SUBD;
     SUBROUTINE SUBEVUL;
          BEGIN
               IF NMSCN ≠NUM THEN GO TO NLERR;
               CHR ← FMTA[NLI←NLI+1],[1:6];  % # DIM
               NFCI ← 1; NLCONV;
               IF (D2≠0) OR (EXP ≠ 0) THEN GO TO NLERR;
               SBS ← WH1-1;
          NLR: IF NMSCN ≠ SPC THEN GO TO NLERR;
               IF NBC = ")" THEN GO TO SUBD;
               IF NFCI = CHR THEN GO TO NLERR;
               NLCONV;
```

```
02987400 T  0158:0
02987420 T  0158:0
02987440 T  0158:0
02987460 T  0158:0
02987480 T  0158:2
02987500 T  0159:0
02987520 T  0160:2
02987540 T  0160:2
02987560 T  0162:0
02987580 T  0163:2
02987600 T  0163:3
02987620 T  0164:0
02987640 T  0164:2
02987660 T  0164:2
02987680 T  0164:2
02987700 T  0165:0
02987720 T  0165:0
02987740 T  0165:1
02987760 T  0165:2
02987780 T  0168:2
02987800 T  0170:2
02987820 T  0171:0
02987840 T  0173:0
02987860 T  0173:0
02987880 T  0173:0
02987900 T  0173:3
02987920 T  0174:1
02987940 T  0175:2
02987960 T  0176:2
02987980 T  0178:1
02988000 T  0178:1
02988020 T  0179:2
02988040 T  0181:0
02988060 T  0185:0
02988080 T  0186:1
02988100 T  0186:3
02988120 T  0187:0
02988140 T  0191:1
02988160 T  0192:0
02988180 T  0196:3
02988200 T  0197:2
02988220 T  0197:3
02988240 T  0200:0
02988260 T  0203:1
02988280 T  0205:0
02988300 T  0205:1
02988320 T  0205:1
02988340 T  0206:0
02988360 T  0206:0
02988380 T  0208:0
02988400 T  0210:2
02988420 T  0212:0
02988440 T  0214:2
02988460 T  0215:3
02988480 T  0218:0
02988500 T  0219:1
02988520 T  0220:2
```

```
            IF (D2≠0) OR (EXP≠0) THEN GO TO NLERR;                                  02988540 T  0222:0
            WH1 ← WH1 - 1; D2 ← 0;                                                  02988560 T  0224:2
            WHILE D2<NFCI DO WH1 ← WH1×FMTA[NLI+(D2←D2+1)];                         02988580 T  0226:2
            SBS ← SBS + WH1;                                                        02988600 T  0231:1
            NFCI ← NFCI + 1;                                                        02988620 T  0232:2
            GO TO NLR;                                                              02988640 T  0233:3
        SUBD:                                                                       02988660 T  0234:1
            INDX ← INDX + SBS;                                                      02988680 T  0234:1
            T1 ← NMSCN;                                                             02988700 T  0235:2
        END SUBEVUL;                                                                02988720 T  0237:2
COMMENT * * * * * * * * * * END OF DECLARATIONS * * * * * * * * * * * * ;          02988740 T  0237:3
NFL←1 ;                                                                             02988745 T  0237:3
            FILX[NOT 4] ← EOFL;   FILX[NOT 3] ← PARL;                               02988760 T  0248:0
FIB ← FILX[NOT 2];      % OPEN FILE IF NOT OPEN                                     02988780 T  0251:2
IF FIB[5].[43:2] ≠ (T1 ← 2 ) THEN                                                   02988800 T  0253:1
    P(MKS,0,T1,FILX,1,SELECT);                                                      02988820 T  0255:1
            % SET/CHECK FOR MIXED FORMATTED - UNFORMATTED I/O                      02988860 T  0257:1
             CKPB;                                                                  02988880 T  0257:1
            IF FIB[0] = 0 THEN FIB[0] := 1 ELSE                                     02988900 T  0258:0
            IF FIB[0] NEQ 1 THEN P(MKS,FIB[6],FILX.[33:15],4,FORTERR);              02988920 T  0260:3
    NMLST:                                                                          02989000 T  0264:3
            NLE ← FMTA[FI].[2:10];                                                  02989020 T  0264:3
            NLT ← TRUE;                                                             02989040 T  0266:1
            SKPC; NCR ← NCR + 1;                                                    02989060 T  0267:0
            WHILE NMSCN ≠ SPC OR NBC ≠ "$" DO BEGIN READS; SKPC; END;               02989080 T  0269:1
            NCR ← NCR + 1;                                                          02989100 T  0274:2
            IF NMSCN ≠ ID THEN GO TO NLERR;                                         02989120 T  0275:3
            IF FMTA[FI].[12:36] ≠ NBC THEN                                          02989140 T  0278:0
              BEGIN READS; GO TO NMLST; END;                                        02989160 T  0279:2
NLP: IF NMSCN ≠ ID THEN GO TO NLERR;                                                02989180 T  0281:2
NLPA:                                                                               02989200 T  0284:0
            NLI ← FI+1; T1 ← NLE;                                                   02989220 T  0284:0
            WHILE T1 >0 DO                                                          02989240 T  0286:0
BEGIN                                                                               02989260 T  0287:1
            IF NBC = FMTA[NLI].[12:36] THEN GO TO NLPB;                             02989280 T  0287:1
            T1 ← T1 - 1;                                                            02989300 T  0289:1
            NLI ← NLI + 2 + FMTA[NLI+1].[1:6];                                      02989320 T  0290:2
END;                                                                                02989340 T  0293:2
                % NOT FOUND                                                        02989360 T  0294:0
            WHILE (T1← NMSCN) ≠ SPC OR(NBC ≠ "," AND NBC ≠ "$") DO                  02989380 T  0294:0
              IF T1 = NUM THEN NLCONV;                                              02989400 T  0298:2
            IF NBC = "$" THEN BEGIN DONETOG ← TRUE; READS END;                      02989420 T  0301:2
            GO TO NLP;                                                              02989440 T  0305:0
NLPB:                                                                               02989460 T  0305:2
            ATOG ← CTOG ← FALSE;                                                    02989480 T  0305:2
            LISTYPE ← FMTA[NLI].[2:10];                                             02989500 T  0306:3
            IF (T1←FMTA[NLI+1].[18:30]) ≠0 THEN                                     02989520 T  0308:1
BEGIN                                                                               02989540 T  0310:3
            SIZE←(INDX←T1.BASEF)+T1.SIZEF ;                                         02989560 T  0311:1
            ATOG ← TRUE;                                                            02989620 T  0314:0
END;                                                                                02989640 T  0314:3
            IF (T1 ← FMTA[NLI+1].[7:11]) < 1024 THEN                                02989660 T  0314:3
                 LISTADR ←  [PRTBASE[T1]]                                           02989680 T  0317:1
             ELSE LISTADR ← IF T1.[39:1] THEN  [MEM[LISX-T1,[41:7]]]                02989700 T  0318:1
                                         ELSE  [MEM[LISX+T1,[40:8]]];               02989720 T  0324:0
            IF ATOG THEN TWDT←NOT P(*(LISTADR←MEM[(*[LISTADR]).[18:15]]),           02989740 T  0326:3
                                    TOP,XCH,DEL)                                    02989745 T  0330:1
```

```
     ELSE W1←LISTADR ;                                                                  02989750 T  0331:0
     IF LGTG=0 THEN T1←NMSCN ELSE BEGIN NBC←LGTG; LGTG←0 END;                           02989760 T  0333:0
     IF NBC ="(" THEN                                                                  02989780 T  0337:2
BEGIN                                                                                  02989800 T  0338:1
     IF NOT ATOG THEN GO TO NLERR;                                                     02989820 T  0338:3
     SUBEVUL;                                                                          02989840 T  0339:2
END;                                                                                   02989860 T  0341:0
     IF NBC ≠ "=" THEN GO TO NLERR;                                                    02989880 T  0341:0
     T1 ← NMSCN;                                                                       02989900 T  0342:1
NRP: IF T1 =NUM THEN                                                                   02989920 T  0343:2
BEGIN                                                                                  02989940 T  0344:1
     NLCONV;                                                                           02989960 T  0344:3
     IF NMSCN ≠SPC THEN GO TO NLERR;                                                   02989980 T  0346:0
     IF NBC ≠ "*" THEN GO TO NLISRT;                                                   02990000 T  0348:0
     RPT ← WH1;                                                                        02990020 T  0349:1
     IF (D2≠0) OR (EXP≠0)THEN GO TO NLERR;                                             02990040 T  0350:0
     IF NMSCN = NUM THEN GO TO NRP;                                                    02990060 T  0352:2
END;                                                                                   02990080 T  0355:0
NRPL:                                                                                  02990100 T  0355:0
     IF NBC = "," THEN                                                                 02990120 T  0355:0
BEGIN                                                                                  02990140 T  0355:3
     IF ELMTYP ≠ LOGV THEN                                                             02990160 T  0356:1
BEGIN                                                                                  02990180 T  0357:2
     NLCONV; T1 ← NMSCN; GO TO NLISRT;                                                 02990200 T  0358:0
END;                                                                                   02990220 T  0361:0
     T1 ← NMSCN;                                                                       02990240 T  0361:0
NLL: WH1 ← (NBC.[12:6]= "T");                                                          02990260 T  0362:2
     WHILE (T1←NMSCN) ≠ SPC OR (NBC≠"$" AND NBC ≠ ",")                                 02990280 T  0364:1
       DO IF T1= NUM THEN GO TO NLERR;                                                 02990300 T  0367:1
     GO TO NLISRT;                                                                     02990320 T  0370:1
END;                                                                                   02990340 T  0370:3
     IF NBC ="(" THEN                                                                  02990360 T  0370:3
BEGIN                                                                                  02990380 T  0371:2
     IF NMSCN ≠ NUM THEN GO TO NLERR;                                                  02990400 T  0372:0
     NLCONV; JUNK1 ←WH1;                                                               02990420 T  0374:0
     CTOG ← TRUE;                                                                      02990440 T  0375:3
     IF NMSCN ≠ SPC OR NBC ≠ "," THEN GO TO NLERR;                                     02990460 T  0376:2
     IF NMSCN ≠NUM THEN GO TO NLERR;                                                   02990480 T  0380:1
     NLCONV;                                                                           02990500 T  0382:0
     WH2 ← WH1; WH1 ← JUNK1;                                                           02990520 T  0383:0
     IF NMSCN ≠ SPC OR NBC ≠ ")" THEN GO TO NLERR;                                     02990540 T  0384:2
     GO TO NLISRT;                                                                     02990560 T  0388:1
END;                                                                                   02990580 T  0388:3
     IF ELMTYP ≠LOGV THEN GO TO NLERR;                                                 02990600 T  0388:3
     GO TO NLL;                                                                        02990620 T  0390:2
NLISRT:                                                                                02990640 T  0391:0
     IF ATOG THEN                                                                      02990660 T  0391:0
BEGIN                                                                                  02990680 T  0391:1
     IF INDX ≥ SIZE THEN GO TO NLERR;                                                  02990700 T  0391:3
     W1 ← NXTELM;                                                                      02990720 T  0393:0
     INDX ← INDX+ (DLN OR CMPLX);                                                      02990740 T  0397:2
     INDX ← INDX + 1;                                                                  02990760 T  0401:1
END;                                                                                   02990780 T  0402:2
     IF ELMTYP = INTEGV THEN W1[0]←WH1 DIV 1 ELSE                                      02990800 T  0402:2
     W1[0] ← WH1;                                                                      02990820 T  0405:2
     IF (DLN OR CMPLX) THEN W1[1] ←WH2;                                                02990840 T  0406:3
     IF NOT (CTOG EQV CMPLX) THEN GO TO NLERR;                                         02990860 T  0411:2
```

```
        IF ATOG THEN IF (RPT ← RPT-1) > 0 THEN GO TO NLISRT;                    02990880 T  0413:3
        NFL←0;  WHILE NBC≠"," AND NBC≠"$" DO P(NMSCN,DEL);  NFL←1 ;             02990900 T  0416:3
        IF NBC = "$" THEN BEGIN DONETOG ← TRUE; READS; END;                     02990920 T  0422:2
        IF NOT ATOG THEN GO TO NLP;                                             02990940 T  0426:0
        IF ELMTYP ≠ LOGV THEN IF (NMSCN =ID )                                   02990960 T  0426:3
            THEN GO TO NLPA ELSE IF NBC≠"," THEN GO TO NRP                      02990980 T  0430:1
                ELSE BEGIN WH1←WH2←0; GO TO NLISRT END;                         02991000 T  0431:3
        IF (T1 ← NMSCN) = NUM THEN GO TO NRP;                                   02991020 T  0434:0
        IF NBC = "." THEN GO TO NRPL;                                           02991040 T  0436:2
        WH1 ← NBC; T1 ← NMSCN;                                                  02991060 T  0437:3
        IF NBC ≠ "," THEN                                                       02991080 T  0440:2
BEGIN                                                                           02991100 T  0441:1
        LGTG ← NBC; NBC ← WH1; GO TO NLPA;                                      02991120 T  0441:3
END;                                                                            02991140 T  0443:3
        WH1 ← (WH1,[12:6] = "T");                                               02991160 T  0443:3
        GO TO NLISRT;                                                           02991180 T  0445:2
NLERR:                                                                          02991200 T  0446:0
        P(MKS,FIB[6],FILX,[33:15],1,FORTERR);                                   02991220 T  0446:0
END FINNAME;                                                                    02991240 T  0448:0
                                                                                      SIZE= 0449 WORDS
```

```
PROCEDURE FOUTNAME;         %155                                                02991260 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00442
          BEGIN                                                                 02991280 T  0000:0
COMMENT FILX        FILE TOP IO DESCRIPTOR                                      02991300 T  0000:0
        FMTA        FORMAT OR NAMELIST OR 0                                     02991320 T  0000:0
        LISX        ACCIDENTAL ENTRY DESC, OR 0                                 02991340 T  0000:0
                                                ;                               02991460 T  0000:0
REAL     FORTERR    = 24,                                                       02991500 T  0000:0
         LISX       =  -1,                                                      02991520 T  0000:0
         FI         =  -3,                                                      02991540 T  0000:0
         DKADR      =  -4;                                                      02991560 T  0000:0
ARRAY    FMTA       =  -2[*], FPB = 3[*] ;                                      02991580 T  0000:0
NAME     FILX       =  -5,                                                      02991600 T  0000:0
         MEM        =    2;                                                     02991620 T  0000:0
REAL        ALGOLWRITE  = 12,                                                   02991640 T  0000:0
            SELECT      = 14;                                                   02991660 T  0000:0
INTEGER     LSTRN       = 19;                                                   02991680 T  0000:0
REAL        LISTYPE     = 20,                                                   02991700 T  0000:0
            ARRAYSTUFF  = 18,                                                   02991720 T  0000:0
            NAMEV       = 21;                                                   02991740 T  0000:0
ARRAY       PRTBASE     = 10[*],                                                02991760 T  0000:0
            TEN         = 22[*],                                                02991780 T  0000:0
            TPAR        = 23[*],                                                02991800 T  0000:0
            FIB[*];                                                             02991820 T  0000:0
NAME        LISTADR;                                                            02991840 T  0000:0
REAL        BUFF                        , %  FIRST BUFFER POSITION              02991860 T  0000:0
            BSIZE                       , %  ARGUMENTS                          02991880 T  0000:0
            FLG             , % TRUE FOR SERIAL I/O                             02991900 T  0000:0
            WH1,                                                                02991920 T  0000:0
            WH2                         , %                                     02991940 T  0000:0
            W1                          , %                                     02991960 T  0000:0
            W2                          , %                                     02991980 T  0000:0
            NFCI                        ; %  NEXT FORMAT CHAR LOCATION          02992000 T  0000:0
```

```
INTEGER     DH1                    , %   CONV-                                  02992020 T  0000:0
            DH2                    , %      ERTED NU-                           02992040 T  0000:0
            DH3                    , %                MBER                      02992060 T  0000:0
            W                      , %   FIELD                                  02992080 T  0000:0
            WT                     , %        WIDTH                             02992100 T  0000:0
            T1                     , %                                          02992120 T  0000:0
            D                      , %   DEC-                                   02992140 T  0000:0
            DT                     , %       IMAL P-                           02992160 T  0000:0
            D1                     , %           LA-                           02992180 T  0000:0
            D2                     , %             CE-                         02992200 T  0000:0
            D3                     , %               S                         02992220 T  0000:0
            ZEROS                  , %   TRAILING ZEROES                        02992240 T  0000:0
            EXP                    , %   EXPONENT                               02992260 T  0000:0
            SHFT                   , %   INTEGER PART OF SHIFT                  02992280 T  0000:0
            CODE                   , %   EDITING FUNCTION                       02992300 T  0000:0
            SKP                    , %   REDUNDANT POSITIONS                    02992320 T  0000:0
            NCR                    , %   CURRENT BUFFER POSITION                02992340 T  0000:0
            LCR                    , %   BUFFER SIZE IN CHARACTERS              02992360 T  0000:0
            CHR                    ; %   CURRENT CHAR FROM FORMAT               02992380 T  0000:0
BOOLEAN     DONETOG                , %   RETURN AFTER WRITE                     02992400 T  0000:0
            SGN                    , %   SIGN                                   02992420 T  0000:0
            PRNTR          ,% TRUE IF PRINTER OUT PUT                           02992440 T  0000:0
            DTOG                   , %   DOUBLE PRECISION TOG                   02992460 T  0000:0
            CTOG                   ; %   COMPLEX NUMBER TOG                     02992480 T  0000:0
DEFINE LOGV       = 4#,                                                         02992500 T  0000:0
       INTEGV      =1#,                                                         02992520 T  0000:0
       DBLV       = 5#,                                                        02992540 T  0000:0
       CMPLXV     = 6#,                                                        02992560 T  0000:0
       ETYPE      = 3#,                                                        02992580 T  0000:0
       DTYPE      = 4#,                                                        02992600 T  0000:0
       ITYPE      = 5#,                                                        02992620 T  0000:0
       LTYPE      = 6#,                                                        02992640 T  0000:0
       ELMTYP      = LISTYPE , [44:4]#,                                        02992660 T  0000:0
       DLN         = (LISTYPE.[44:4] =DBLV)#,                                  02992680 T  0000:0
       CMPLX       = (LISTYPE.[44:4] = CMPLXV)#,                               02992700 T  0000:0
       TWOD        = LISTYPE.[38:1]#,                                          02992720 T  0000:0
       ENDLIST     = (LSTRN = (-1))#,                                          02992740 T  0000:0
       SIZEF       = [33:15]#;                                                 02992760 T  0000:0
LABEL  ERTN,E,DC,I,L,AST,COMM,NM1,NM2,FMERR;                                   02992780 T  0000:0
COMMENT  * * * * * START OF SUBROUTINE DECLARATIONS * * * * * * * * * ;       02992800 T  0000:0
SUBROUTINE CKPB;                                                               02992820 T  0000:0
BEGIN COMMENT   INITIALIZE FILE AND ACQUIRE RECORD SIZE;                       02992840 T  0001:0
LCR ← 8×(BSIZE ← P(MKS,FLG,DKADR,0,(-1),FILX,ALGOLWRITE));                     02992860 T  0001:0
IF PRNTR←PRNTR&(((T1←FIB[4].[8:4])=1 OR T1=7 OR T1=12) AND FPB[FIB[4]          02992880 T  0004:2
        .[13:11]+3].[43:5]<20)[47:47:1] THEN                                   02992885 T  0009:0
   IF BSIZE GEQ 17 THEN BEGIN LCR := 132; BSIZE := 17 END;                     02992900 T  0013:0
IF PRNTR AND BSIZE=17 THEN LCR←120 ;                                           02992920 T  0016:1
BUFF :=(IF T1 := PRNTR THEN TPAR ELSE *FILX).[33:15];                          02992960 T  0018:3
IF ((NOT T1) OR PRNTR.[46:1]) THEN                                             02993020 T  0022:2
STREAM(P2 ← (BSIZE-1).[36:6],                                                  02993040 T  0024:0
          P3←T1.[47:1]+BSIZE-1,P4←BUFF) ;                                      02993060 T  0026:0
   BEGIN  DI ← P4;   DS ← 8 LIT " ";                                           02993080 T  0028:1
          SI ← P4;                                                             02993100 T  0029:3
          P2(DS ← 32 WDS;  DS ← 32 WDS);                                       02993120 T  0030:0
          DS ← P3 WDS;                                                         02993140 T  0031:1
   END;                                                                        02993160 T  0031:3
NCR ← 0;                                                                       02993200 T  0032:0
```

```
END CKPB;                                                                                    02993220 T  0032:3
SUBROUTINE PRNT;                                                                             02993240 T  0033:0
BEGIN COMMENT     GENERATE A CALL FOR CAR, CONT, AND FOR OUTPUT;                             02993260 T  0033:0
IF PRNTR THEN                                                                                02993280 T  0033:0
BEGIN;                                                                                       02993300 T  0033:1
    NCR ← 0;                                                                                 02993320 T  0033:3
    STREAM(P1←0:P2←TPAR);                                                                    02993340 T  0034:2
    BEGIN SI ←P2; DI ← LOC P1; DI ←DI + 7; DS ←CHR;                                          02993360 T  0036:0
          DI ← P2; DS ← LIT " ";END;                                                         02993380 T  0037:0
         NCR ← P;                                                                            02993400 T  0038:0
         IF NCR = " " THEN D2 ← 16 ELSE                                                      02993420 T  0038:2
         IF NCR = "0" THEN D2 ← 32 ELSE                                                      02993440 T  0040:2
         IF NCR = "+" THEN D2 ← 0 ELSE                                                       02993460 T  0043:0
              IF (D2 ← NCR) > 9 THEN D2 ← 16;                                                02993480 T  0045:2
         IF NOT PRNTR.[46:1] THEN FIB[17]←FIB[17]+BSIZE ;                                    02993520 T  0048:2
         P(MKS,D2.[42:2],D2.[44:4],PRNTR.[46:1],BSIZE,FILX,ALGOLWRITE) ;                     02993540 T  0052:0
         FIB[6]←FIB[6]"(D2=0) ;                                                              02993560 T  0055:1
         IF NOT (*FILX).[19:1] THEN P(FILX,@2000000000,2,COM,DEL,DEL);                       02993580 T  0057:3
         PRNTR←1;  CKPB ;                                                                    02993600 T  0061:0
         STREAM(P1←TPAR,P2←*FILX,P3←BSIZE.[36:6],P4←BSIZE);                                  02993620 T  0063:0
         BEGIN                                                                               02993640 T  0065:2
              SI ← P1; DI ← P2; DS ← P4 WDS;                                                 02993660 T  0065:2
              P3(DS ←32 WDS; DS ← 32 WDS);                                                   02993680 T  0066:2
              DI←P1; P4(DS←8LIT" ") ;                                                        02993700 T  0067:3
         END;                                                                                02993720 T  0070:0
    FIB[17]←FIB[17]"BSIZE;  IF DONETOG THEN P(XIT) ;                                         02993740 T  0070:1
    END ELSE BEGIN P(MKS,FLG,DKADR,0,BSIZE,FILX,ALGOLWRITE);                                 02993760 T  0073:1
    IF DONETOG THEN P(XIT);                                                                  02993780 T  0076:3
    CKPB END ;                                                                               02993800 T  0077:3
END PRNT;                                                                                    02993820 T  0079:0
        % PARAMETERS FOR LIST CONTROL                                                       02993840 T  0079:1
BOOLEAN ATOG,TWDT;                                                                           02993860 T  0079:1
ARRAY AR1 = LISTADR[*];                                                                      02993880 T  0079:1
REAL  INDX,SIZE,NLI,NLE;                                                                     02993900 T  0079:1
LABEL RTNLST,SRT;                                                                            02993920 T  0079:1
DEFINE NXTELM = IF TWDT THEN P(*[AR1[INDX.[33:7]]],INDX.[40:8],COC)                          02993940 T  0079:1
                        ELSE AR1[INDX]#;                                                     02993960 T  0079:1
SUBROUTINE GETNMLST;                                                                         02993980 T  0079:1
    BEGIN                                                                                    02994000 T  0080:0
        IF (NLE ← NLE " 1) <0 THEN LSTRN ← " 1 ELSE                                          02994020 T  0080:0
    BEGIN                                                                                    02994040 T  0083:1
        NAMEV ← FMTA[NLI← NLI+1].[12:36];                                                    02994060 T  0083:3
        LISTYPE ← FMTA [NLI].[2:10];                                                         02994080 T  0086:1
        ARRAYSTUFF ← FMTA[NLI←NLI+1].[18:30];                                                02994100 T  0087:3
        IF (T1 ← FMTA[NLI].[7:11] ) < 1024                                                   02994120 T  0090:1
          THEN LISTADR←  [PRTBASE[T1]] ELSE                                                  02994140 T  0091:3
       IF T1.[39:1] THEN LISTADR ← [MEM[LISX"(T1.[41:7])]]                                   02994160 T  0093:3
                    ELSE LISTADR ← [MEM[LISX+(T1.[40:8])]];                                  02994180 T  0099:0
        NLI ← NLI + FMTA[NLI].[1:6];                                                         02994200 T  0102:1
    END;                                                                                     02994220 T  0104:1
    END GETNMLST;                                                                            02994240 T  0104:1
SUBROUTINE GETLIST;                                                                          02994260 T  0104:2
    BEGIN                                                                                    02994280 T  0105:0
    SRT: IF ATOG THEN                                                                        02994300 T  0105:0
    BEGIN                                                                                    02994320 T  0105:1
        IF DLN THEN                                                                          02994340 T  0105:3
    BEGIN                                                                                    02994360 T  0107:0
```

```
     WH1 ← NXTELM;                                                    02994380 T  0107:2
     INDX ← INDX + 1;                                                 02994400 T  0112:0
     WH2 ← NXTELM;                                                    02994420 T  0113:1
END ELSE                                                              02994440 T  0117:3
BEGIN                                                                 02994460 T  0117:3
     WH1 ← NXTELM;                                                    02994480 T  0118:1
     WH2 ← 0;                                                         02994500 T  0122:3
END;                                                                  02994520 T  0123:2
     IF (INDX ←INDX + 1) ≥ SIZE THEN                                  02994540 T  0123:2
BEGIN                                                                 02994560 T  0125:1
     ARRAYSTUFF ← 0;                                                  02994580 T  0125:3
     ATOG ← FALSE;                                                    02994600 T  0126:2
END;                                                                  02994620 T  0127:1
     GO TO RTNLST;                                                    02994640 T  0127:1
END;                                                                  02994660 T  0127:3
     IF CTOG THEN                                                     02994680 T  0127:3
BEGIN     % IMAGINARY PART OF COMPLEX                                 02994700 T  0128:0
     WH1 ← LISTADR[1];                                                02994720 T  0128:2
     WH2 ← 0;                                                         02994740 T  0130:0
     CTOG ← FALSE;                                                    02994760 T  0130:3
     GO TO RTNLST;                                                    02994780 T  0131:2
  END;                                                                02994800 T  0132:0
          GETNMLST;                                                   02994820 T  0132:0
          IF ENDLIST THEN GO TO RTNLST;                               02994840 T  0133:0
     IF ARRAYSTUFF ≠ 0 THEN                                           02994860 T  0134:2
BEGIN                                                                 02994880 T  0135:1
     ATOG ← TRUE;                                                     02994900 T  0135:3
     SIZE←(INDX←ARRAYSTUFF.[18:15])+ARRAYSTUFF.SIZEF ;                02994910 T  0136:2
          TWDT←NOT P(*(LISTADR←MEM[(*[LISTADR]).[18:15]]),TOP);  P(DEL) ;  02994920 T  0139:1
          GO TO SRT;                                                  02994970 T  0143:1
     END;                                                             02994980 T  0143:3
          IF NOT P(*LISTADR,TOP,XCH,DEL) THEN LISTADR←P(*LISTADR) ;   02994985 T  0143:3
          WH1 ← LISTADR[0];                                           02995000 T  0146:3
          WH2 ← IF DLN THEN LISTADR[1] ELSE 0;                        02995020 T  0147:2
          CTOG ← CMPLX;                                               02995040 T  0151:2
     RTNLST:                                                          02995060 T  0153:1
     END GETLIST;                                                     02995080 T  0153:1
SUBROUTINE NMSZ;                                                      02995100 T  0153:2
     BEGIN;                                                           02995120 T  0154:0
          STREAM(P1←[NAMEV]:P2←0);                                    02995140 T  0154:0
     BEGIN                                                            02995160 T  0155:1
               SI ←P1; SI ← SI + 2;                                   02995180 T  0155:1
               6(IF SC = " " THEN JUMP OUT;                           02995200 T  0155:3
                 SI ←SI + 1; TALLY ← TALLY + 1);                      02995220 T  0157:0
               P1 ←TALLY;                                             02995240 T  0157:3
          END;                                                        02995260 T  0158:0
          NFCI ← P;                                                   02995280 T  0158:1
     END NMSZ;                                                        02995300 T  0158:3
SUBROUTINE PUT;                                                       02995320 T  0159:0
     BEGIN;                                                           02995340 T  0159:0
          STREAM(P1←[NAMEV]:P2←NFCI,P3←BUFF);                         02995360 T  0159:0
          BEGIN                                                       02995380 T  0160:2
               SI ← P1; SI ←SI + 2; DS ←P2 CHR;                       02995400 T  0160:2
               P1 ← DI;                                               02995420 T  0161:2
          END;                                                        02995440 T  0161:3
          BUFF ← P;                                                   02995460 T  0162:0
     END PUT;                                                         02995480 T  0162:2
```

```
SUBROUTINE FUNNYZERO;
    BEGIN
         SKP ← W - (D+6+SGN);
         STREAM(P1← BUFF:P2←SKP,P3←SGN,P4←(D+4));
         BEGIN
              DI ←P1; DI ← DI + P2;
              P3(DS ← LIT "-"; JUMP OUT TO L);
         L:   DS ← 2 LIT "0.";
              P4(DS ← LIT " ");
              P1 ← DI;
         END;
         BUFF ← P;
    END FUNNYZERO;
SUBROUTINE FINDE;
BEGIN         IF DTOG THEN
              DOUBLE(TEN[0],0,WH1,WH2,×,←,WH1,WH2)
                   ELSE WH1 ← TEN[0] × WH1;
              EXP ← ((0&WH1[42:3:6]&WH1[1:2:1]+12)×.9039) +.5;
              W2 ← 0;
              IF DTOG THEN
              IF EXP ≥ 0 THEN DOUBLE(TEN[EXP],TEN[69+EXP],←,W1,W2)
               ELSE DOUBLE(1,0,TEN[-EXP],TEN[69-EXP],/,←,W1,W2)
              ELSE W1 ← IF EXP ≥ 0 THEN TEN[EXP] ELSE 1/TEN[-EXP];
              IF WH1 > W1 THEN GO TO ERTN;
              IF WH1 = W1 THEN
               IF WH2 ≥ W2 THEN GO TO ERTN;
              EXP ← EXP-1;
    ERTN:
END FINDE;
SUBROUTINE NUMCONVERT;
BEGIN
              IF D1 > 0 THEN
         BEGIN
              DOUBLE(WH1,WH2,TEN[16],TEN[85],/,←,W1,W2);
              DH1 ← W1 DIV 1.0;
         END;
              IF D2 > 0 THEN
         BEGIN     IF DTOG  THEN
           BEGIN
              DOUBLE(WH1,WH2,DH1,0,TEN[16],TEN[85],x,-,
                     TEN[ 8],TEN[77],/,←,W1,W2);
              DH2 ← W1 DIV 1;
           END
              ELSE DH2 ← WH1 DIV TEN[8];
         END;
              IF DTOG THEN
         BEGIN
              DOUBLE(WH1,WH2,DH1,0,TEN[16],TEN[85],x,
                             DH2,0,TEN[ 8],TEN[77],x,+,-,←,W1,W2);
              DH3 ← W1 DIV 1;
         END
              ELSE DH3 ← WH1 DIV 1;
              EXP ← EXP+1;
END NUMCONVERT;
SUBROUTINE SETD;
BEGIN
              IF DLN AND DT > 23 THEN
```

```
02995500 T  0162:3
02995520 T  0163:0
02995540 T  0163:0
02995560 T  0165:1
02995580 T  0167:2
02995600 T  0167:2
02995620 T  0168:1
02995640 T  0170:0
02995660 T  0170:2
02995680 T  0171:3
02995700 T  0172:0
02995720 T  0172:1
02995740 T  0172:3
02995760 T  0173:0
02995780 T  0173:0
02995800 T  0173:1
02995820 T  0176:1
02995840 T  0178:2
02995860 T  0182:3
02995880 T  0183:2
02995900 T  0183:3
02995920 T  0188:0
02995940 T  0194:3
02995960 T  0199:2
02995980 T  0200:3
02996000 T  0201:2
02996020 T  0203:1
02996040 T  0204:2
02996060 T  0204:2
02996080 T  0204:3
02996100 T  0205:0
02996120 T  0205:0
02996140 T  0205:3
02996160 T  0206:1
02996180 T  0209:1
02996200 T  0210:2
02996220 T  0210:2
02996240 T  0211:1
02996260 T  0212:0
02996280 T  0212:2
02996300 T  0215:1
02996320 T  0217:3
02996340 T  0219:0
02996360 T  0219:0
02996380 T  0222:2
02996400 T  0222:2
02996420 T  0222:3
02996440 T  0223:1
02996460 T  0225:3
02996480 T  0229:1
02996500 T  0230:2
02996520 T  0230:2
02996540 T  0232:1
02996560 T  0233:2
02996580 T  0233:3
02996600 T  0234:0
02996620 T  0234:0
```

```
            BEGIN                                                                    02996640 T  0236:1
                 ZEROS←DT-23; DT ← 23; D1 ← 7; D2 ← D3 ← 8;                          02996660 T  0236:3
            END ELSE IF DT>12 AND NOT DLN THEN                                       02996680 T  0240:3
            BEGIN                                                                    02996700 T  0243:2
                 ZEROS←DT-12; DT ← 12; D1←0; D2 ← 4; D3 ← 8;                         02996720 T  0244:0
            END ELSE IF DT>16 THEN                                                   02996740 T  0248:1
            BEGIN                                                                    02996760 T  0249:2
                 D1←DT-16; D2←D3←8;                                                  02996780 T  0250:0
            END ELSE IF DT > 8 THEN                                                  02996800 T  0252:2
            BEGIN                                                                    02996820 T  0253:3
                 D1←0;D2←DT-8; D3←8;                                                 02996840 T  0254:1
            END ELSE                                                                 02996860 T  0257:0
            BEGIN                                                                    02996880 T  0257:0
                 D1←D2←0;D3←DT;                                                      02996900 T  0257:2
            END;                                                                     02996920 T  0259:2
END SETD;                                                                            02996940 T  0259:2
SUBROUTINE RNDOFF;                                                                   02996960 T  0259:3
BEGIN       IF DTOG THEN                                                             02996980 T  0260:0
            IF T1 ≥ 0 THEN                                                           02997000 T  0260:1
             DOUBLE(WH1,WH2,,5,TEN[T1],TEN[T1+69],×,+,←,WH1,WH2) ELSE                02997020 T  0261:2
             DOUBLE(WH1,WH2,,5,TEN[-T1],TEN[69-T1],/,+,←,WH1,WH2)                    02997040 T  0266:1
           ELSE WH1 ← WH1 + (IF T1≥0 THEN 5×TEN[T1] ELSE 5/TEN[-T1]);                02997060 T  0271:0
END RNDOFF;                                                                          02997080 T  0276:0
SUBROUTINE SCALE;                                                                    02997100 T  0277:0
BEGIN       IF DTOG THEN                                                             02997120 T  0277:0
    BEGIN   IF T1 ≥ 0                                                            02997140 T  0277:1
             THEN DOUBLE(WH1,WH2,TEN[T1],TEN[T1+69],×,←,WH1,WH2)                     02997160 T  0278:0
             ELSE DOUBLE(WH1,WH2,TEN[-T1],TEN[69-T1],/,←,WH1,WH2);                   02997180 T  0282:1
            IF WH1 ≥ TEN[DT] THEN                                                    02997200 T  0286:3
        BEGIN                                                                        02997220 T  0287:3
            EXP ← EXP + 1;                                                           02997240 T  0288:1
            DOUBLE(WH1,WH2,TEN[1],0,/,←,WH1,WH2);                                    02997260 T  0289:2
        END                                                                          02997280 T  0292:1
    END ELSE WH1 ← IF T1 ≥ 0 THEN WH1×TEN[T1] ELSE WH1/TEN[-T1];                     02997300 T  0292:1
END SCALE;                                                                           02997320 T  0297:1
        %************** S T A R T  O F   EDIT-CONTROL*****************%              02997340 T  0297:2
SUBROUTINE CONVERT;                                                                  02997360 T  0297:2
BEGIN                                                                                02997380 T  0298:0
DTOG := FALSE;                                                                       02997400 T  0298:0
SGN ←WH1,[1:1]; IF CODE < LTYPE THEN WH1 ← ABS(WH1); WT ← W; DT ← D;                 02997420 T  0298:3
DH1 ← DH2 ← DH3 ← ZEROS ←EXP ← SKP ← SHFT ← D1 ← D2 ← D3 ← 0;                        02997440 T  0303:3
GO TO P(CODE,DUP,ADD);                                                               02997460 T  0309:0
GO TO FMERR;                                                                         02997480 T  0310:0
GO TO FMERR;                                                                         02997500 T  0310:2
GO TO FMERR;                                                                         02997520 T  0311:0
GO TO E;                                                                             02997540 T  0311:2
GO TO DC;                                                                            02997560 T  0312:0
GO TO I;                                                                             02997580 T  0312:2
GO TO L;                                                                             02997600 T  0313:0
    L:   COMMENT LOGICIAL CONVERSION;                                                02997620 T  0313:2
            IF W >1 THEN SKP←W-(WT←1);                                               02997640 T  0313:2
            WH1← 0&(IF WH1 THEN "T" ELSE "F")[12:42:6];                              02997660 T  0316:2
      STREAM(P1 := BUFF:P2 := WH1,P3 := SKP, P4 := WT);                              02997680 T  0319:3
       BEGIN DI := P1; DI := DI + P3;                                                02997700 T  0321:2
             SI := LOC P2;  SI := SI + 2;                                            02997720 T  0322:1
             DS := P4 CHR; P1 := DI;                                                 02997740 T  0322:3
       END;                                                                          02997760 T  0323:2
```

```
    BUFF := P;                                                                  02997780 T  0323:3
    GO TO COMM;                                                                 02997800 T  0324:1
     I:   COMMENT INTEGER CONVERSION;                                           02997820 T  0324:3
               IF WH1=0 AND WH2=0 THEN DT ← D3 ← 1 ELSE                         02997840 T  0324:3
          BEGIN     IF DTOG THEN                                              02997860 T  0328:1
               DOUBLE(WH1,WH2,,,5,+,←,WH1,WH2)  % ROUND OFF                     02997880 T  0329:0
               ELSE WH1 ← T1 ← WH1;                                             02997900 T  0331:3
               FINDE;                                                           02997920 T  0335:1
               IF EXP < 0 THEN DT ← D3 ← 1 ELSE                                 02997940 T  0336:0
          BEGIN                                                                 02997960 T  0338:2
               IF (DLN AND EXP≥24) OR (NOT DLN AND EXP≥12) THEN GO AST;         02997980 T  0339:0
               DT ← EXP+1; SETD; NUMCONVERT;                                    02998000 T  0344:2
          END;                                                                  02998020 T  0348:0
          END;                                                                  02998040 T  0348:0
               IF DT + SGN > W THEN GO TO AST;                                  02998060 T  0348:0
               IF W > DT + SGN THEN SKP ← W - DT - SGN;                         02998080 T  0349:3
STREAM(P1←0:P2 ← D1,P3 ← DH1,P4 ← D2,P5 ← DH2,                                  02998100 T  0353:1
       P6 ← D3,P7 ← DH3,P8 ← SGN,P9 ← SKP,P10 ← BUFF);                          02998120 T  0355:0
   BEGIN  DI ← P10;     P9(DI ← DI + 1);                                        02998140 T  0356:2
          P8(DS ← LIT "-");                                                     02998160 T  0357:3
          SI ←LOC P3;  DS ← P2 DEC;                                             02998180 T  0359:0
          SI ← LOC P5;    DS ← P4 DEC;                                          02998200 T  0359:3
          SI ← LOC P7;    DS ← P6 DEC;                                          02998220 T  0360:2
          P1 ← DI;                                                              02998240 T  0361:1
   END;                                                                         02998260 T  0361:2
BUFF ← P;                                                                       02998280 T  0361:3
GO TO COMM;                                                                     02998300 T  0362:1
    DC:  COMMENT DOUBLE PRECISION CONVERT,SAME AS E CONVERT;                    02998320 T  0362:3
    E:   COMMENT E CONVERSION;                                                  02998340 T  0362:3
               DTOG ← TRUE;                                                     02998360 T  0362:3
               SETD;                                                            02998380 T  0363:2
               IF WH1=0 AND WH2 = 0 THEN                                        02998400 T  0365:0
          BEGIN                                                                 02998420 T  0366:3
               IF W < (D+6+ SGN) THEN GO TO AST;                                02998460 T  0367:1
               FUNNYZERO; GO TO COMM;                                           02998480 T  0369:2
          END  ELSE                                                             02998500 T  0371:2
          BEGIN                                                                 02998520 T  0371:2
               FINDE;                                                           02998540 T  0372:0
               IF (SKP ← W - D - 5 - SGN) < 0 THEN GO TO AST; SETD;             02998560 T  0373:0
               IF DT LSS 0 THEN DT := 0;                                        02998580 T  0377:0
               T1 ← EXP - DT; RNDOFF;                                           02998600 T  0379:0
               SETD;                                                            02998620 T  0381:0
               T1←DT-1-EXP; SCALE;                                              02998640 T  0382:0
               NUMCONVERT;                                                      02998660 T  0385:0
          END;                                                                  02998680 T  0386:0
STREAM(P1 ← 0:P2 ← SKP,P3 ← SGN,P4 ← D1,P5 ← DH1,                               02998700 T  0386:0
       P6 ← D2,P7 ← DH2,P8 ← D3,P9 ← DH3,P10 ← (DLN),                           02998720 T  0387:3
       P11 ← (EXP < 0),P12 ← ABS(EXP),P13 ← SHFT,P14 ← ZEROS,P15 ←BUFF);        02998740 T  0390:0
   BEGIN  DI ← P15;   DI ← DI + P2;   P3(DS ← LIT "-");                         02998760 T  0392:0
          P2 ← DI;   DS ← LIT ".";                                              02998780 T  0394:1
          SI ← LOC P5;   DS ← P4 DEC;                                           02998800 T  0395:0
          SI ← LOC P7;   DS ← P6 DEC;                                           02998820 T  0395:3
          SI ← LOC P9;   DS ← P8 DEC;                                           02998840 T  0396:2
          P14(DS ← LIT " ");  DS ← LIT "E";                                     02998860 T  0397:1
          P10(DI ← DI - 1;  DS ← LIT "D");                                      02998880 T  0399:0
          DS ← LIT " " ;                                                        02998900 T  0400:2
          P11(DI ← DI - 1;  DS ← LIT "-");                                      02998920 T  0401:0
```

```
            SI ← LOC P12;   DS ← 2 DEC;                                          02998940 T  0402:2
            P1 ← DI;                                                             02998960 T  0403:0
            P13(DI ← P2;  SI ← P2;  SI ← SI + 1;                                 02998980 T  0403:1
                DS ← P13 CHR;  DS ← LIT ".";  JUMP OUT TO X);  X:                02999000 T  0404:2
   END;                                                                          02999020 T  0406:1
BUFF ← P;                                                                        02999040 T  0406:2
            GO TO COMM;                                                          02999060 T  0407:0
AST:                                                                             02999080 T  0407:2
STREAM(P1 ← 0;P2 ← BUFF,P3 ← W);                                                 02999100 T  0407:2
   BEGIN  DI ← P2;     P3(DS ← LIT "*");   P1 ← DI;   END;                       02999120 T  0409:0
BUFF ← P;                                                                        02999140 T  0411:0
COMM:                                                                            02999160 T  0411:2
END CONVERT;                                                                     02999180 T  0411:2
COMMENT * * * * * * * * * * END OF DECLARATIONS * * * * * * * * * * * ;         02999200 T  0411:3
FIB ← FILX[NOT 2];      % OPEN FILE IF NOT OPEN                                 02999220 T  0411:3
IF DKADR < 0 THEN BEGIN FLG ← 1; DKADR ←0 END;                                   02999240 T  0423:3
IF FIB[5].[43:1] THEN P(MKS,0,0,FILX,1,SELECT);                                  02999260 T  0426:2
PRNTR←2×(FIB[5].[41:2]≠0) ; %%% IFF FILE IS CLOSED, SETS PRNTR,[46:1]=1.         02999280 T  0429:2
IF PRNTR THEN STREAM(TPAR); DS←8LIT" " ;                                         02999290 T  0432:0
CKPB; ARRAYSTUFF ← 0;                                                            02999300 T  0435:1
        IF FIB[0] = 0 THEN FIB[0] := 1 ELSE                                      02999320 T  0436:3
        IF FIB[0] NEQ 1 THEN P(MKS,FIB[6],FILX,[33:15],4,FORTERR);               02999340 T  0439:2
        LSTRN ←0; CHR ← "      "; NLI ← FI;                                      02999420 T  0443:2
        NAMEV ← CHR&"$"[18:42:6];                                                02999430 T  0445:3
        NCR ←NCR + (NFCI ← 2); PUT;                                              02999440 T  0447:2
        NAMEV ← FMTA[NLI].[12:36];                                               02999450 T  0450:0
        NLE ← FMTA[NLI].[2:10];                                                  02999460 T  0451:2
        NMSZ; PUT; NCR ← NCR + NFCI;                                             02999470 T  0453:0
        NAMEV ← CHR; NCR ← NCR + (NFCI ← 1); PUT;                                02999480 T  0456:1
                                                                                 02999490 T  0460:0
NM1: GETLIST; IF ENDLIST THEN                                                    02999500 T  0460:0
BEGIN;                                                                           02999510 T  0462:0
     STREAM(P1←BUFF);                                                            02999520 T  0462:2
     BEGIN                                                                       02999530 T  0463:1
          DI ← P1; DI ← DI - 3; DS ← LIT "$";                                    02999540 T  0463:1
     END;                                                                        02999550 T  0464:1
     DONETOG ← TRUE; PRNT;                                                       02999560 T  0464:2
END;                                                                             02999570 T  0466:0
     IF PRNTR THEN PRNT;                                                         02999580 T  0466:0
 CODE←NAMEV; NAMEV←CHR; NCR←NCR+(NFCI←2); PUT; NAMEV←CODE;                        02999590 T  0468:0
     NMSZ;                                                                       02999600 T  0472:3
     IF ELMTYP = INTEGV THEN                                                     02999610 T  0474:0
      BEGIN W ←12; D ←0; CODE ← ITYPE END ELSE                                   02999620 T  0475:1
     IF ELMTYP = LOGV THEN                                                       02999630 T  0478:0
      BEGIN W ←1; CODE ← LTYPE END ELSE                                          02999640 T  0481:1
     IF ELMTYP = DBLV THEN                                                       02999650 T  0483:1
      BEGIN W ←29; D ←23; CODE ← DTYPE END ELSE                                  02999660 T  0485:0
      BEGIN W ←18; D ← 12; CODE ← ETYPE END;                                     02999670 T  0487:3
     IF (6 + W + NFCI + ( IF CMPLX THEN (W+3) ELSE 0) + NCR)                     02999680 T  0490:2
       ≥ LCR THEN PRNT;                                                          02999690 T  0495:1
     PUT;          % NAME                                                       02999700 T  0498:0
     NCR ← NCR + NFCI + 3;                                                       02999710 T  0499:0
     NAMEV ← CHR&" ="[12:36:12]; NFCI ←3; PUT;                                   02999720 T  0500:3
NM2: IF ELMTYP = CMPLXV THEN                                                     02999740 T  0504:0
BEGIN                                                                            02999750 T  0505:1
     IF (NCR+W+W+6) ≥ LCR THEN PRNT;                                             02999760 T  0505:3
     NCR ← NCR + (NFCI ←1); NAMEV ← CHR&"("[12:42:6]; PUT;                       02999770 T  0510:0
```

```
            NCR ← NCR + W; CONVERT;
            NCR ← NCR + (NFCI+1); NAMEV ← CHR&","[12:42:6]; PUT;
            CTOG ← TRUE;
            GETLIST;
            NCR ← NCR + W; CONVERT;
            NCR ← NCR + (NFCI +4);
            NAMEV ← CHR&"),"[12:36:12]; PUT;
        END
            ELSE
        BEGIN
            IF (NCR + W + 3) ≥ LCR THEN PRNT;
            NCR ← NCR + W; CONVERT;
            NCR ← NCR + (NFCI ← 3);
            NAMEV ← CHR&","[12:42:6];  PUT;
        END;
            IF NOT ATOG THEN GO TO NM1;
            GETLIST; GO TO NM2;
    FMERR:
            P(MKS,FIB[6],FILX.[33:15],0,FORTERR);
        END FOUTNAME;
```

```
02999780 T  0515:0
02999790 T  0517:0
02999800 T  0522:0
02999810 T  0522:3
02999820 T  0524:0
02999830 T  0526:0
02999840 T  0527:3
02999850 T  0531:0
02999860 T  0531:0
02999870 T  0531:0
02999880 T  0534:0
02999890 T  0537:0
02999900 T  0539:0
02999910 T  0540:3
02999920 T  0544:0
02999930 T  0544:0
02999940 T  0544:3
02999950 T  0546:2
02999960 T  0546:2
02999970 T  0548:2
          SIZE= 0549 WORDS
```

```
PROCEDURE DABS  ;          % 052

COMMENT ABSOLUTE VALUE OF A DOUBLE PRECISION NUMBER;  %  PF  JUNE 67
BEGIN REAL X    = -1,
           XL = -2,
           JUNK = 17;
P(X,SSP,,JUNK,STD,XL,RTN);
END DABS;
```

```
03000000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00461
03000100 T  0000:0
03000200 T  0000:0
03000300 T  0000:0
03000400 T  0000:0
03000500 T  0000:0
03000600 T  0001:2
          SIZE= 0002 WORDS
```

```
PROCEDURE CABS  ;          % 053

COMMENT COMPLEX ABSOLUTE INTRINSIC;   %  PF  JUNE 67
BEGIN REAL X = -1,
           Y = -2,
           SQRT=+1 ;
P(INTDESC(SQRTI)) ;
IF (X ← ABS(X)) = 0 OR (Y ← ABS(Y)) = 0 THEN P(X,Y,ADD,RTN)
ELSE IF X > Y THEN P(MKS,1,Y,X,/,DUP,MUL,ADD,SQRT,X,MUL)
             ELSE P(MKS,1,X,Y,/,DUP,MUL,ADD,SQRT,Y,MUL);
P(RTN);
END CABS;
```

```
03100000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00462
03100100 T  0000:0
03100200 T  0000:0
03100300 T  0000:0
03100400 T  0000:0
03100410 T  0000:0
03100500 T  0001:2
03100600 T  0006:1
03100700 T  0010:3
03100800 T  0014:0
03100900 T  0014:1
          SIZE= 0015 WORDS
```

```
PROCEDURE AINT  ;          % 054
```

```
03200000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00463
```

```
BEGIN REAL X = -1;
P(X,1,DIV,RTN);
END AINT;                                                                  03200100 T  0000:0
                                                                           03200200 T  0000:0
                                                                           03200300 T  0001:0
                                                                                  SIZE= 0002 WORDS



PROCEDURE MATH;              % 055                                          03300000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00464
COMMENT MATHEMATIC MANIPULATION INTRINSIC          % PF  JUNE 67           03300100 T  0000:0
            CODE = 3×TYPE(OP 1) + 9×OPERATOR + TYPE(OP 2)                  03300200 T  0000:0
                    TYPE   VALUE          OPERATOR  VALUE                  03300300 T  0000:0
                    REAL     0               +        0                    03300400 T  0000:0
                    DOUBLE   1               -        1                    03300500 T  0000:0
                    COMPLEX  2               *        2                    03300600 T  0000:0
                                             /        3;                   03300700 T  0000:0
BEGIN REAL CODE = -1,                                                      03300800 T  0000:0
               A = -3,                                                     03300900 T  0000:0
               B = -4,                                                     03301000 T  0000:0
               C = -5,                                                     03301100 T  0000:0
               D = -6,                                                     03301200 T  0000:0
               ERR = 24,                                                   03301250 T  0000:0
               T;                                                          03301300 T  0000:0
     LABEL RPLUSD,RPLUSC,DPLUSC,CPLUSD,CPLUSC,RLESSD,RLESSC,DLESSC,        03301400 T  0000:0
           CLESSD,CLESSC,RTIMED,RTIMEC,DTIMEC,CTIMED,CTIMEC,RDIVDD,        03301500 T  0000:0
           RDIVDC,DDIVDC,CDIVDD,CDIVDC,CTIMER,CDIVDR,INLINE;               03301600 T  0000:0
GO TO P(CODE,DUP,ADD);                                                     03301700 T  0000:0
GO TO INLINE;      %  REAL + REAL          0                              03301800 T  0001:1
GO TO RPLUSD;      %  REAL + DOUBLE        1                              03301900 T  0001:3
GO TO RPLUSC;      %  REAL + COMPLEX       2                              03302000 T  0002:1
GO TO INLINE;      %  DOUBLE + REAL        3                              03302100 T  0002:3
GO TO INLINE;      %  DOUBLE + DOUBLE      4                              03302200 T  0003:1
GO TO DPLUSC;      %  DOUBLE + COMPLEX     5                              03302300 T  0003:3
GO TO INLINE;      %  COMPLEX + REAL       6                              03302400 T  0004:1
GO TO CPLUSD;      %  COMPLEX + DOUBLE     7                              03302500 T  0004:3
GO TO CPLUSC;      %  COMPLEX + COMPLEX    8                              03302600 T  0005:1
GO TO INLINE;      %  REAL - REAL          9                              03302700 T  0005:3
GO TO RLESSD;      %  REAL - DOUBLE       10                              03302800 T  0006:1
GO TO RLESSC;      %  REAL - COMPLEX      11                              03302900 T  0006:3
GO TO INLINE;      %  DOUBLE - REAL       12                              03303000 T  0007:1
GO TO INLINE;      %  DOUBLE - DOUBLE     13                              03303100 T  0007:3
GO TO DLESSC;      %  DOUBLE - COMPLEX    14                              03303200 T  0008:1
GO TO INLINE;      %  COMPLEX - REAL      15                              03303300 T  0008:3
GO TO CLESSD;      %  COMPLEX - DOUBLE    16                              03303400 T  0009:1
GO TO CLESSC;      %  COMPLEX - COMPLEX   17                              03303500 T  0009:3
GO TO INLINE;      %  REAL * REAL         18                              03303600 T  0010:1
GO TO RTIMED;      %  REAL * DOUBLE       19                              03303700 T  0010:3
GO TO RTIMEC;      %  REAL * COMPLEX      20                              03303800 T  0011:1
GO TO INLINE;      %  DOUBLE * REAL       21                              03303900 T  0011:3
GO TO INLINE;      %   DOUBLE * DOUBLE    22                              03304000 T  0012:1
GO TO DTIMEC;      %  DOUBLE × COMPLEX    23                              03304100 T  0012:3
GO TO CTIMER;      %  COMPLEX × REAL      24                              03304200 T  0013:1
GO TO CTIMED;      %  COMPLEX × DOUBLE    25                              03304300 T  0013:3
GO TO CTIMEC;      %  COMPLEX × COMPLEX   26                              03304400 T  0014:1
GO TO INLINE;      %  REAL / REAL         27                              03304500 T  0014:3
GO TO RDIVDD;      %  REAL / DOUBLE       28                              03304600 T  0015:1
```

```
GO TO RDIVDC;        %  REAL / COMPLEX        29                    03304700 T  0015:3
GO TO INLINE;        %  DOUBLE / REAL         30                    03304800 T  0016:1
GO TO INLINE;        %  DOUBLE / DOUBLE       31                    03304900 T  0016:3
GO TO DDIVDC;        %  DOUBLE / COMPLEX      32                    03305000 T  0017:1
GO TO CDIVDR;        %  COMPLEX / REAL        33                    03305100 T  0017:3
GO TO CDIVDD;        %  COMPLEX / DOUBLE      34                    03305200 T  0018:1
GO TO CDIVDC;        %  COMPLEX / COMPLEX     35                    03305300 T  0018:3
RPLUSD:  P(0,C,B,A,DLA,.B,STD,.C,STD,XIT);                          03305400 T  0019:1
RPLUSC:  P(A,C,ADD,B,.C,STD,.B,STD,XIT);                            03305500 T  0021:3
DPLUSC:  P(A,C,ADD,.C,STD,B,.D,STD,XIT);                            03305600 T  0024:0
CPLUSD:  P(A,C,ADD,.C,STD,XIT);                                     03305700 T  0026:1
CPLUSC:  P(A,C,ADD,.C,STD,B,D,ADD,.D,STD,XIT);                      03305800 T  0027:3
RLESSD:  P(0,C,B,A,DLS,.B,STD,.C,STD,XIT);                          03305900 T  0030:2
RLESSC:  P(C,A,SUB,B,CHS,.C,STD,.B,STD,XIT);                        03306000 T  0033:0
DLESSC:  P(C,A,SUB,.C,STD,B,CHS,.D,STD,XIT);                        03306100 T  0035:2
CLESSD:  P(C,A,SUB,.C,STD,XIT);                                     03306200 T  0038:0
CLESSC:  P(D,B,SUB,.D,STD,C,A,SUB,.C,STD,XIT);                      03306300 T  0039:2
RTIMED:  P(0,C,B,A,DLM,.B,STD,.C,STD,XIT);                          03306400 T  0042:1
RTIMEC:  P(C,DUP,B,MUL,.C,STD,A,MUL,.B,STD,XIT);                    03306500 T  0044:3
DTIMEC:  P(C,DUP,A,MUL,.C,STD,B,MUL,.D,STD,XIT);                    03306600 T  0047:2
CTIMER:  P(A,DUP,B,MUL,.B,STD,C,MUL,.C,STD,XIT);                    03306700 T  0050:1
CTIMED:  P(A,DUP,C,MUL,.C,STD,D,MUL,.D,STD,XIT);                    03306800 T  0053:0
CTIMEC:  P(A,C,MUL,D,B,MUL,SUB,C,B,MUL,A,D,MUL,ADD,.D,STD,.C,STD,XIT);  03306900 T  0055:3
RDIVDD:  P(0,C,B,A,DLD,.B,STD,.C,STD,XIT);                          03307000 T  0060:2
RDIVDC:  P(C,A,DUP,MUL,B,DUP,MUL,ADD,/,DUP,B,MUL,CHS,               03307100 T  0063:0
           .C,STD,A,MUL,.B,STD,XIT);                                03307200 T  0066:1
DDIVDC:  P(C,A,DUP,MUL,B,DUP,MUL,ADD,/,DUP,B,MUL,CHS,               03307300 T  0068:0
           .D,STD,A,MUL,.C,STD,XIT);                                03307400 T  0071:1
CDIVDR:  P(B,A,/,.B,STD,C,A,/,.C,STD,XIT);                          03307500 T  0073:0
CDIVDD:  P(C,A,/,.C,STD,D,A,/,.D,STD,XIT);                          03307600 T  0075:3
CDIVDC:  P(A,C,MUL,B,D,MUL,ADD,A,DUP,MUL,B,DUP,MUL,ADD,.T,STN,/,    03307700 T  0078:2
           A,D,MUL,B,C,MUL,SUB,T,/,.D,STD,.C,STD,XIT);              03307800 T  0082:3
INLINE: P(MKS,10,ERR);    %  COMPILER WRITERS ERROR                 03307900 T  0086:1
END MATH;                                                           03308000 T  0087:0
                                                                              SIZE= 0088 WORDS
```

```
PROCEDURE XTOI;              % 056                                   03400000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00467
COMMENT VARIOUS COMBINATIONS OF X TO THE I    % PF  JUNE 67         03400100 T  0000:0
            CODE = 3×TYPE(OP 1) + TYPE(OP 2)                        03400200 T  0000:0
                 TYPE       VALUE                                   03400300 T  0000:0
                 REAL         0                                     03400400 T  0000:0
                 DOUBLE       1                                     03400500 T  0000:0
                 COMPLEX      2;                                    03400600 T  0000:0
BEGIN REAL CODE = -1,                                               03400700 T  0000:0
           A = -3,                                                  03400800 T  0000:0
           B = -4,                                                  03400900 T  0000:0
           C = -5,                                                  03401000 T  0000:0
           D = -6,                                                  03401100 T  0000:0
           JUNK = 17,                                               03401200 T  0000:0
           T=+1,V=+2,ERR=+3,BOOL=+4,CDTOG=+5 ;                      03401300 T  0000:0
      REAL EXPINT=27, LNINT=29 ;                                    03401350 T  0000:0
      INTEGER J=+6,I=J,R=CDTOG ;                                    03401400 T  0000:0
      REAL EXP=+7,LN=+8,DEXP=EXP,DLOG=LN,CABS=+9,ATAN2=+10,SQRT=+11,  03401500 T  0000:0
```

```
          COS=+12,SIN=+13 ;                                                    03401510 T  0000:0
     DEFINE DF(DF1)=FLAG(DF1 OR T) # ;                                         03401520 T  0000:0
     LABEL REXPOR,DEXPOR,REXPOD,DEXPOD,CEXPOD,CEXPOR,REXPOC,DEXPOC,L1,         03401600 T  0000:0
           L2,L3,L4,CEXPOC,CDENT,RDENT,TOPI,TOPIL,PI2,TPI2,HAF,PI,MAX,         03401700 T  0000:0
           F096,L5,PIT,CREL,PICK,RX1,RX2,REXPOR1,CEXPOD2 ;                     03401710 T  0000:0
P(0&85[1:41:7],0,DF(FORTERRI),0,0,0);  IF CODE=0 THEN GO REXPOR ;             03401800 T  0000:0
IF CODE<5 THEN IF CODE≠2 THEN BEGIN P(DF(DEXPI),DF(DLOGI)); GO PICK END;      03401900 T  0004:2
P(DF(EXPI),DF(LNI),DF(CABSI),DF(ATAN2I),DF(SQRTI),DF(COSI),DF(SINI)) ;        03401910 T  0009:2
PICK:    T←0 ;                                                               03401920 T  0016:2
GO TO P(CODE,DUP,ADD);                                                       03402000 T  0017:1
GO TO REXPOR;        %  REAL ** REAL          0                             03402100 T  0018:1
GO TO REXPOD;        %  REAL ** DOUBLE        1                             03402200 T  0018:3
GO TO REXPOC;        %  REAL ** COMPLEX       2                             03402300 T  0019:1
GO TO DEXPOR;        %  DOUBLE ** REAL        3                             03402400 T  0019:3
GO TO DEXPOD;        %  DOUBLE ** DOUBLE      4                             03402500 T  0020:1
GO TO DEXPOC;        %  DOUBLE ** COMPLEX     5                             03402600 T  0020:3
GO TO CEXPOR;        %  COMPLEX ** REAL       6                             03402700 T  0021:1
GO TO CEXPOD;        %  COMPLEX ** DOUBLE     7                             03402800 T  0021:3
GO TO CEXPOC;        %  COMPLEX ** COMPLEX    8                             03402810 T  0022:1
DEXPOC:  R←P(,C);  I←P(,D);  C←C+0&C[1:1:8]&D[47:9:1];  GO L3 ;               03402815 T  0022:3
REXPOC:  R←P(,B);  I←P(,C) ;                                                 03402820 T  0028:0
L3:      IF C=0 THEN BEGIN P(0); GO L1 END ;                                 03402825 T  0029:2
         IF B=0 THEN                                                         03402830 T  0031:2
              BEGIN                                                          03402835 T  0032:1
              IF A=0 THEN BEGIN P(1); GO L1 END ;                            03402837 T  0032:3
              IF C>0 THEN GO L5 ;                                            03402840 T  0034:3
              IF ABS(A) LEQ P(MAX) THEN IF P(A,,BOOL,ISN)=A THEN             03402845 T  0036:0
L4:                BEGIN A←BOOL ;                                           03402850 T  0038:3
L5:                P(ABS(C),A,MKS,,EXP,LOD,INTCALL(*P(,LN),XTOII),DEL) ;    03402855 T  0040:0
                   IF B≠0 THEN GO L2;  IF BOOL THEN P(CHS) ;                 03402860 T  0043:3
L1:                P(R,STD,0,I,STD,XIT) ;                                    03402865 T  0046:0
                   END ;                                                     03402870 T  0047:2
              T←P(PI)×A;  P(MKS,MKS,ABS(C),LN,A,×,EXP);  GO L2 ;             03402875 T  0047:2
              END ;                                                          03402880 T  0051:1
         T←(V←P(MKS,ABS(C),LN))×B ;                                          03402885 T  0051:1
         IF A=0 THEN                                                         03402890 T  0053:3
              BEGIN                                                          03402895 T  0054:2
              IF C>0 THEN P(MKS,T,COS,R,STD,MKS,T,SIN,I,STD,XIT) ;           03402900 T  0055:0
              P(MKS,(-P(PI))×B,EXP) ;                                        03402905 T  0059:0
L2:           P(V←P,MKS,T,COS,×,R,STD,MKS,T,SIN,V,×,I,STD,XIT) ;             03402910 T  0060:2
              END ;                                                          03402915 T  0064:2
         IF C<0 THEN                                                         03402920 T  0064:2
              BEGIN P(MKS,V×A-B×P(PI),EXP); T←T+P(PI)×A; GO L2 END ;          03402925 T  0065:1
         IF ABS(A) LEQ 1023 THEN IF P(A,,BOOL,ISN)=A THEN GO L4 ;            03402930 T  0070:1
         P(MKS,A×V,EXP);  GO L2 ;                                            03402950 T  0073:2
CEXPOC:  IF B=0 THEN GO CEXPOD2 ;                                            03402965 T  0075:1
         R←P(,C);  I←P(,D);  IF D=0 THEN GO L3 ;                             03402970 T  0076:2
         IF C=0 THEN BEGIN T←ABS(D); P(PIT); IF D<0 THEN P(SSN); V←P END     03402975 T  0079:1
         ELSE BEGIN T←P(MKS,D,C,CABS); V←P(MKS,D,C,ATAN2) END ;              03402980 T  0083:1
         T←(BOOL←P(MKS,T,LN))×B+V×A;  P(MKS,BOOL×A-V×B,EXP);  GO L2 ;        03402985 T  0087:1
PI:::    3,14159265359 ;                                                     03402990 T  0093:1
MAX:::   @0007777777777777 ;                                                 03402991 T  0095:0
PIT:::   @1141444176652104 ;                                                 03402992 T  0096:0
REXPOR:  IF B = 0 OR B = 1 THEN P(XIT);                                      03403000 T  0097:0
        IF A = 0 THEN P(1,,B,STD,XIT);                                       03403100 T  0099:2
          IF ABS(A)<4096 THEN IF (J←A)=A THEN                                03403200 T  0101:3
REXPOR1:  BEGIN IF BOOL←J<0 THEN J←-J ;                                      03403300 T  0104:2
```

```
                        P(J,.T,STD,B);
                        WHILE (T ← (J ← T).[36:11]) ≠ 0 DO
                        BEGIN P(DUP);
                              IF J THEN
                              BEGIN V ← V + 1;
                                    P(DUP);
                              END;
                              P(MUL);
                        END;
                        WHILE (V ← V - 1) ≥ 0 DO P(MUL);
                        IF BOOL THEN P(1,XCH,/);
                        IF CDTOG≠0 THEN
                           BEGIN
                           IF CDTOG>2 THEN P(,C,,D) ELSE P(,B,,C) ;
                           J←(J←A) AND 3;
                           IF BOOL AND J THEN J←(J+2) AND 3;
                           IF C=0 THEN RX1: P(0,XCH,STD,STD,XIT) ;
                           IF J = 0 THEN GO RX1;
                           IF J=1 THEN BEGIN P(XCH); GO RX1 END ;
                           IF J=2 THEN RX2: P(0,XCH,STD,XCH,CHS,XCH,STD,XIT) ;
                           P(XCH);  GO RX2 ;
                           END ;
                        P(,B,STD,XIT);
                     END;
                  IF B>0
                     THEN P(MKS,MKS,B,LNINT,A,X,EXPINT,,B,STD,XIT); %FORTRAN ONLY
                  P(MKS,11,ERR);
REXPOD:  P(0,,B,,C);  GO RDENT ;
DEXPOR:  P(C,,B,,C);  C←B;  B←0;  GO RDENT ;
DEXPOD:  P(D,,C,,D) ;
RDENT:   CODE←P;  R←P;  JUNK←P;  IF C=0 THEN P(0,CODE,STD,XIT) ;
         IF A=0 THEN P(1,R,STD,0,CODE,STD,XIT) ;
        IF B = 0 THEN
              IF ABS(A)<P(F096) THEN IF (J←A)=A THEN
              BEGIN IF BOOL ← J < 0 THEN J ← -J;
                    P(J,,T,STD,JUNK,C) ;
                      WHILE (T ← (J ← T).[36:11]) ≠ 0 DO
                      BEGIN P(,A,STD,DUP,A,XCH,A);
                            IF J THEN
                            BEGIN V ← V + 1;
                                  P(,A,STD,DUP,A,XCH,A);
                            END;
                            P(DLM);
                      END;
                      WHILE (V ← V - 1) ≥ 0 DO P(DLM);
                      IF BOOL THEN P(,T,STD,0,XCH,1,XCH,T,DLD) ;
                      P(R,STD,CODE,STD,XIT) ;
              END;
         IF C>0 THEN P(MKS,MKS,JUNK,C,DLOG,JUNK,B,A,DLM,DEXP,CODE,STD,
                       JUNK,R,STD,XIT) ;
         P(MKS,11,ERR);
CEXPOR:  IF B =0 THEN IF C = 0 THEN P(XIT);
         IF A = 0 THEN P(1,,B,STD,0,,C,STD,XIT);
CDENT:    IF ABS(A)<P(F096) THEN IF (J←A)=A THEN
          BEGIN
          IF C=0 OR B=0 THEN
              BEGIN CDTOG←CDTOG OR 2; IF B=0 THEN B←C; GO REXPOR1 END ;
```

```
03403400 T  0107:3
03403500 T  0108:3
03403600 T  0111:2
03403700 T  0111:3
03403800 T  0112:0
03403900 T  0113:3
03404000 T  0114:0
03404100 T  0114:0
03404200 T  0114:1
03404300 T  0116:0
03404400 T  0119:0
03404410 T  0120:2
03404415 T  0121:1
03404420 T  0121:3
03404422 C  0124:2
03404424 C  0126:1
03404425 T  0129:1
03404430 P  0131:3
03404435 T  0133:0
03404440 T  0135:0
03404445 T  0138:1
03404450 T  0139:0
03404500 T  0139:0
03404600 T  0139:3
03404700 T  0139:3
03404710 T  0140:0
03404800 T  0143:2
03404900 T  0144:1
03405000 T  0145:2
03405100 T  0148:1
03405200 T  0149:0
03408100 T  0152:3
03408200 T  0155:3
03408300 T  0156:2
03408400 T  0159:3
03408500 T  0163:0
03408600 T  0164:1
03408700 T  0167:0
03408800 T  0168:2
03408900 T  0168:3
03409000 T  0170:2
03409100 T  0172:0
03409200 T  0172:0
03409300 T  0172:1
03409400 T  0172:3
03409500 T  0175:3
03409600 T  0178:2
03409700 T  0179:3
03409800 T  0179:3
03409900 T  0184:0
03410000 T  0185:0
03410100 T  0185:3
03410200 T  0188:2
03410300 T  0191:2
03410305 T  0194:1
03410310 T  0194:3
03410315 T  0196:2
```

```
                IF BOOL←J<0 THEN J←-J ;                                          03410400 T  0200:3
                   GO CREL;  F096::: 4096; CREL:                                 03410450 T  0203:2
                   P(J,,T,STD,C,B);                                              03410500 T  0205:0
                   WHILE (T ← (J ← T).[36:11]) ≠ 0 DO                            03410600 T  0206:1
                   BEGIN P(,A,STD,DUP,A,XCH,A);                                  03410700 T  0209:0
                         IF J THEN                                               03410800 T  0210:2
                         BEGIN V ← V + 1;                                        03410900 T  0210:3
                               P(,A,STD,DUP,A,XCH,A);                            03411000 T  0212:2
                         END;                                                    03411100 T  0214:0
                         P(DUP,MUL,XCH,DUP,MUL,SUB,,A,STD,MUL,DUP,ADD,A);        03411200 T  0214:0
                   END;                                                          03411300 T  0217:0
                   WHILE (V ← V - 1) ≥ 0 DO                                      03411400 T  0217:2
                P(,A,STD,,B,STD,,C,STD,DUP,A,MUL,B,C,MUL,ADD,                    03411500 T  0219:3
                         XCH,CHS,B,MUL,A,C,MUL,ADD);                             03411600 T  0223:0
                   IF BOOL THEN P(,A,STD,CHS,DUP,DUP,MUL,A,DUP,                  03411700 T  0225:2
                                  MUL,ADD,,B,STN,/,A,B,/);                       03411800 T  0228:1
                   IF CDTOG THEN P(,C,STD,,D,STD,XIT)                            03411900 T  0230:1
                            ELSE P(,B,STD,,C,STD,XIT);                           03412000 T  0232:1
             END;                                                                03412100 T  0234:0
          C ← (V ← P(MKS,MKS,MKS,C,B,CABS,LN,A,MUL,EXP))                         03412200 T  0234:0
             ×(A ← P(MKS,MKS,C,B,ATAN2,A,MUL,TOPI,MOD,,T,STN,SIN));              03412300 T  0236:2
          P(MKS,1,A,DUP,MUL,SUB,SQRT);                                           03412400 T  0241:1
          IF T > P(PI2) THEN IF T < P(TPI2) THEN P(CHS);                         03412500 T  0243:0
          P(V,MUL,,B,STD);                                                       03412600 T  0245:3
          IF CDTOG THEN P(C,,D,STD,B,,C,STD);                                    03412700 T  0246:3
          P(XIT);                                                                03412800 T  0249:0
CEXPOD:   A←A+0&A[1:1:8]&B[47:9:1] ;                                             03412900 T  0249:1
CEXPOD2:  IF C=0 THEN IF D=0 THEN P(XIT) ;                                       03413000 T  0252:2
          IF A=0 THEN P(1,,C,STD,0,,D,STD,XIT) ;                                 03413050 T  0255:1
          B ← C;                                                                 03413100 T  0258:1
          C ← D;                                                                 03413200 T  0259:0
          CDTOG ← TRUE;                                                          03413300 T  0259:3
          GO TO CDENT;                                                           03413400 T  0260:2
TOPI ::: @1146220773250420;     TOPIL ::: @0005506043230461;                     03413500 T  0261:0
HAF  ::: @1154000000000000;                                                      03413600 T  0263:0
PI2  ::: @1141444176652104;                                                      03413700 T  0264:0
TPI2 ::: @1144554574376314;                                                      03413800 T  0265:0
                                                                                 03413900 T  0266:0
END XTOI;                                                                        03414000 T  0266:0
                                                                                        SIZE= 0267 WORDS


PROCEDURE IDINT ;          % 057                                                 03500000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00476
COMMENT DOUBLE TO INTEGER CONVERT;   %  PF  JULY 67                             03501000 T  0000:0
BEGIN REAL X = -1,                                                               03501001 T  0000:0
           XL = -2;                                                              03501002 T  0000:0
P(X + 0&X[1:1:8]&XL[47:9:1],1,DIV,RTN);                                          03501004 T  0000:0
END IDINT;                                                                       03501005 T  0003:2
                                                                                        SIZE= 0004 WORDS


PROCEDURE FLOAT ;          % 060                                                 03600000 T  0000:0
```

```
BEGIN REAL X = -1;
        P(X, RTN);
END FLOAT;
```

```
START OF REL SEGMENT; DISK ADDRESS = 00477
                03600100 T  0000:0
                03600200 T  0000:0
                03600300 T  0000:2
                        SIZE= 0001 WORDS
```

```
PROCEDURE SNGL  ;           % 061

COMMENT SNGL INTRINSIC (DOUBLE TO SINGLE CONVERT);   %  PF  JUNE 67
BEGIN REAL X = -1,
           XL= -2;
P(X + 0&X[1:1:8]&XL[47:9:1],RTN);

END SNGL;
```

```
                03700000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00478
                03700100 T  0000:0
                03700200 T  0000:0
                03700300 T  0000:0
                03700400 T  0000:0
                03700500 T  0003:0
                03700600 T  0003:0
                        SIZE= 0004 WORDS
```

```
PROCEDURE DBLE  ;           % 062

COMMENT DBLE INTRINSIC (SINGLE TO DOUBLE);  %  PF  JUNE 67
BEGIN REAL X = -1,
           JUNK = 17;
P(X,,JUNK,STD,0,RTN);
END DBLE;
```

```
                03800000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00479
                03800100 T  0000:0
                03800200 T  0000:0
                03800300 T  0000:0
                03800400 T  0000:0
                03800500 T  0001:1
                        SIZE= 0002 WORDS
```

```
PROCEDURE AMOD  ;           % 063

BEGIN    REAL X = -2, Y = -1;
         P(X MOD Y, RTN);
END AMOD;
```

```
                03900000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00480
                03900100 T  0000:0
                03900200 T  0000:0
                03900300 T  0001:0
                        SIZE= 0002 WORDS
```

```
PROCEDURE TIME ;            % 064

COMMENT FORTRAN TIME INTRINSIC (LIKE ALGOL);  %  PF  JULY 67
BEGIN REAL X =-1;
P(X,1,COM,RTN);
END TIME;
```

```
                04000000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00481
                04001000 T  0000:0
                04001002 T  0000:0
                04001003 T  0000:0
                04001004 T  0001:0
                        SIZE= 0002 WORDS
```

```
PROCEDURE DMOD ;   % DOUBLE PRECISION MOD INTRINSIC # @065.
```

```
                04100000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00482
```

```
    BEGIN
    REAL H=+2, B=-1, BL=-2, A=-3, AL=-4, E=17;  LABEL G,Q ;                                  04100100 T  0000:0
    IF B=0 THEN IF BL=0 THEN P(MKS,INTCALL(13,FORTERRI)) ;                                   04100200 T  0000:0
    IF P(AL,ABS(A),BL,NABS(B),DLA,DUP)=0 THEN GO Q ;                                         04100300 T  0000:0
    IF P<0 THEN P(AL,A,.E,←,RTN) ;                                                           04100400 T  0004:2
    IF (E←P(AL,A,BL,B,DLD,DUP).[3:6])>13 THEN P(.E,ISD) ;                                    04100500 T  0007:2
    P(XCH) ;                                                                                 04100550 T  0009:3
    IF E=0 OR H.[2:1] THEN BEGIN P(DEL,0,XCH,E);::P(.G,+,LOD,LND,XCH)END                     04100600 T  0013:3
    ELSE BEGIN P(13-E); :: P(.G,+,LOD,LND) END ;                                             04100700 T  0014:0
    IF P(DUP,ABS(H),0,1,DLA,BL,NABS(B),DLM,AL,ABS(A),DLA)≥0 THEN                              04100750 T  0019:1
Q:  P(E←0,RTN) ;                                                                             04100800 T  0022:0
    P(DEL,XCH,BL,B,DLM,CHS,AL,A,DLA,.E,←,RTN) ;                                              04100850 T  0026:0
G:::@3777777777777777, % DYNAMIC MASK CONSTANTS.                                             04100900 T  0027:2
@3777777777777770,@3777777777777700,@3777777777777000,@3777777777770000,                    04101000 T  0030:2
@3777777777700000,@3777777777000000,@3777777770000000,@3777777700000000,                    04101050 T  0032:0
@3777777000000000,@3777770000000000,@3777700000000000,@3777000000000000;                    04101100 T  0036:0
    END OF DMOD ;                                                                            04101200 T  0040:0
                                                                                             04101300 T  0044:0
                                                                                                   SIZE= 0045 WORDS
```

```
PROCEDURE DMAX1 ;          % 066                                                  04200000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00484
COMMENT DOUBLE PRCISION MAX ROUTINE;    %  PF    JUNE 67                         04200100 T  0000:0
BEGIN REAL X = -1,                                                               04200200 T  0000:0
          XL = -2,                                                               04200300 T  0000:0
          JUNK = 17,                                                             04200400 T  0000:0
          RCW = +0, SIZE = +1, NEW = +2, NEWL = +3, JUNKL = +4;                  04200500 T  0000:0
P([RCW] INX 0,0,RCW,FCX,1,INX,SUB,0,0,XL,X,,JUNK,STD);                           04200600 T  0000:0
WHILE (SIZE ← SIZE - 2) > 0 DO                                                   04200700 T  0003:3
IF P(NEWL ← *P(.X,SIZE,ADD,,NEW,STN,1,ADD),NEW ← *P(NEW),                        04200800 T  0006:0
     JUNKL,JUNK,DLS,XCH,DEL) > 0 THEN                                            04200900 T  0009:2
   BEGIN JUNKL ← NEWL;                                                           04201000 T  0011:1
         JUNK ← NEW;                                                             04201100 T  0012:2
   END;                                                                          04201200 T  0013:1
P(JUNKL,RTN);                                                                    04201300 T  0013:3
END DMAX1;                                                                       04201400 T  0014:1
                                                                                       SIZE= 0015 WORDS
```

```
PROCEDURE DMIN1 ;          % 067                                                  04300000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00485
COMMENT DOUBLE PRCISION MIN ROUTINE;    %  PF    JUNE 67                         04300100 T  0000:0
BEGIN REAL X = -1,                                                               04300200 T  0000:0
          XL = -2,                                                               04300300 T  0000:0
          JUNK = 17,                                                             04300400 T  0000:0
          RCW = +0, SIZE = +1, NEW = +2, NEWL = +3, JUNKL = +4;                  04300500 T  0000:0
P([RCW] INX 0,0,RCW,FCX,1,INX,SUB,0,0,XL,X,,JUNK,STD);                           04300600 T  0000:0
WHILE (SIZE ← SIZE - 2) > 0 DO                                                   04300700 T  0003:3
IF P(NEWL ← *P(.X,SIZE,ADD,,NEW,STN,1,ADD),NEW ← *P(NEW),                        04300800 T  0006:0
     JUNKL,JUNK,DLS,XCH,DEL) < 0 THEN                                            04300900 T  0009:2
   BEGIN JUNKL ← NEWL;                                                           04301000 T  0011:1
         JUNK ← NEW;                                                             04301100 T  0012:2
   END;                                                                          04301200 T  0013:1
```

```
P(JUNKL,RTN);
END DMIN1;
```

```
04301300 T  0013:3
04301400 T  0014:1
        SIZE= 0015 WORDS
```

```
PROCEDURE SIGNV ;          % 070

COMMENT SIGN INTRINSIC;    %  PF  JUNE 67
BEGIN REAL S = -1,
           X = -2;
P(X);
IF S.[1:1] THEN P(SSN,RTN) ELSE P(SSP,RTN);
END SIGN;
```

```
04400000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00486
04400100 T  0000:0
04400200 T  0000:0
04400300 T  0000:0
04400400 T  0000:0
04400500 T  0000:1
04400600 T  0003:0
        SIZE= 0004 WORDS
```

```
PROCEDURE DSIGN ;          % 071

COMMENT COMPLEX DOUBLE SIGN INTRINSIC;   %  PF  JUNE 67
BEGIN REAL S  = -1,
           X  = -3,
           XL = -4,
           JUNK = 17;
P(X);
IF S.[1:1] THEN P(SSN) ELSE P(SSP);
P(,JUNK,STD,XL,RTN);
END DSIGN;
```

```
04500000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00487
04500100 T  0000:0
04500200 T  0000:0
04500300 T  0000:0
04500400 T  0000:0
04500500 T  0000:0
04500600 T  0000:0
04500700 T  0000:1
04500750 T  0002:2
04500800 T  0003:2
        SIZE= 0004 WORDS
```

```
PROCEDURE DIIM  ;          % 072

BEGIN     REAL X = -2, Y = -1;
          P(X -(IF X ≤ Y THEN X ELSE Y),RTN);
END DIIM;
```

```
04600000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00488
04600100 T  0000:0
04600200 T  0000:0
04600300 T  0003:0
        SIZE= 0004 WORDS
```

```
PROCEDURE REALP ;          % 073

COMMENT COMPLEX TO REAL INTRINSIC;   %  PF  JUNE 67
BEGIN REAL X = -1;
P(X,RTN);
END REALP;
```

```
04700000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00489
04700100 T  0000:0
04700200 T  0000:0
04700300 T  0000:0
04700400 T  0000:2
        SIZE= 0001 WORDS
```

```
PROCEDURE AIMAG ;          % 074
```

```
04800000 T  0000:0
```

```
COMMENT IMAGINARY PART OF COMPLEX NUMBER;    %  PF  JULY 67
BEGIN REAL Y = =2;
P(Y,RTN);
END AIMAG;
```

```
                                                                START OF REL SEGMENT; DISK ADDRESS = 00490
                                                                        04801000 T  0000:0
                                                                        04801010 T  0000:0
                                                                        04801020 T  0000:0
                                                                        04801030 T  0000:2
                                                                                SIZE= 0001 WORDS
```

```
PROCEDURE CMPLX ;          % 075

COMMENT  TWO REALS TO A COMPLEX;    %   PF  JULY 67
BEGIN REAL Y = =1,
           X = =2,
           JUNK = 17;
P(X,,JUNK,STD,Y,RTN);
END CMPLX;
```

```
                                                                        04900000 T  0000:0
                                                                START OF REL SEGMENT; DISK ADDRESS = 00491
                                                                        04900100 T  0000:0
                                                                        04900200 T  0000:0
                                                                        04900250 T  0000:0
                                                                        04900300 T  0000:0
                                                                        04900400 T  0000:0
                                                                        04900500 T  0001:1
                                                                                SIZE= 0002 WORDS
```

```
PROCEDURE CONJG ;          % 076

COMMENT CONJUGATE INTRINSIC;    %  PF  JUNE 67
BEGIN REAL X  = =1,
           XL = =2,
           JUNK = 17;
P(X,,JUNK,STD,XL,CHS,RTN);
END CONJG;
```

```
                                                                        05000000 T  0000:0
                                                                START OF REL SEGMENT; DISK ADDRESS = 00492
                                                                        05000100 T  0000:0
                                                                        05000200 T  0000:0
                                                                        05000300 T  0000:0
                                                                        05000400 T  0000:0
                                                                        05000500 T  0000:0
                                                                        05000600 T  0001:2
                                                                                SIZE= 0002 WORDS
```

```
PROCEDURE DEXP  ;          % 077
                                                                START OF REL SEGMENT; DISK ADDRESS = 00493
COMMENT DOUBLE PRECISION EXPONENTIAL INTRINSIC;    %  PF  JUNE 67
BEGIN REAL X  = =1,
           XL = =2,
           JUNK = 17,
           T,TL;
      BOOLEAN SIG,HUGE;
      INTEGER N;
      LABEL AT13,LG2,LG2L,EMAX,HAF,A,AL,B,BL,C,CL,D,DL,E,EL,F,FL,G,GL,
                    CLGL,H,HL,I,IL,J,JL,K,KL,L,LL,M,ML;
      DEFINE TIMES = NOP,DLA,XL,X,NOP,DLM#;
IF SIG ← X.[1:1] THEN X ← ABS(X);
IF HUGE←X>27 THEN IF X>P(EMAX) THEN P(MKS,INTCALL(14,FORTERRI)) ;

P(XL,X,LG2L,LG2,DLD,.X,STD,DUP,X,XCH,X,0,AT13,DLA,0,AT13,DLS,
  .JUNK,STN,DLS,.X,STD,.XL,STD);
T ← 1;  IF HUGE THEN WHILE (N ← N + 1) ≤ JUNK DO P(TL,DUP,T,XCH,T,DLA,
                                                   .T,STD,.TL,STD)
                ELSE WHILE (N ← N + 1) ≤ JUNK DO T ← P(T,DUP,ADD);
P(ML,M,XL,X,DLM);
```

```
                                                                        05100000 T  0000:0
                                                                        05100100 T  0000:0
                                                                        05100200 T  0000:0
                                                                        05100300 T  0000:0
                                                                        05100400 T  0000:0
                                                                        05100500 T  0000:0
                                                                        05100600 T  0000:0
                                                                        05100700 T  0000:0
                                                                        05100800 T  0000:0
                                                                        05100900 T  0000:0
                                                                        05101000 T  0000:0
                                                                        05101100 T  0000:0
                                                                        05101150 T  0004:0
                                                                        05101250 T  0009:0
                                                                        05101300 T  0009:0
                                                                        05101400 T  0013:1
                                                                        05101500 T  0015:0
                                                                        05101501 T  0020:1
                                                                        05101502 T  0021:1
                                                                        05101600 T  0026:1
```

```
:: P(LL,L,TIMES,KL,K,TIMES,JL,J,TIMES,IL,I,TIMES,HL,H,TIMES,GL,G,TIMES,                05101700 T  0027:2
     FL,F,TIMES,EL,E,TIMES,DL,D,TIMES,CL,C,TIMES,BL,B,TIMES,AL,A,TIMES,               05101800 T  0040:0
     CLGL,LG2,TIMES,0,1,DLA,TL,T,DLM,.JUNK,STD);                                     05101900 T  0052:0
IF SIG THEN P(0,XCH,1,XCH,JUNK,DLD,.JUNK,STD);                                       05102000 T  0056:0
P(RTN);                                                                              05102100 T  0058:3
AT13 ::: @0151000000000000;                                                          05102200 T  0059:0
HAF  ::: @1154000000000000;                                                          05102300 T  0060:0
EMAX ::: @1122360000000000;     CLGL  ::: @0007173632567030;                         05102400 T  0061:0
LG2  ::: @1155427102775750;     LG2L  ::: @0007173632571165;                         05102500 T  0063:0
M    ::: @1333302330351773;     ML    ::: @0005405676153645;                         05102600 T  0065:0
L    ::: @1325447251503330;     LL    ::: @0003745760641244;                         05102700 T  0067:0
K    ::: @1301616647307714;     KL    ::: @0002676025700645;                         05102800 T  0069:0
J    ::: @1273641733265077;     JL    ::: @0006664462403121;                         05102900 T  0071:0
I    ::: @1267446477210572;     IL    ::: @0003454166117342;                         05103000 T  0073:0
H    ::: @1241552224137002;     HL    ::: @0007263626741044;                         05103100 T  0075:0
G    ::: @1232613001073174;     GL    ::: @0002061610334651;                         05103200 T  0077:0
F    ::: @1223777137704414      FL    ::: @0000407415212622;                         05103300 T  0079:0
E    ::: @1215030221137052;     EL    ::: @0005757400272176;                         05103400 T  0081:0
D    ::: @1205354177170511;     DL    ::: @0002237577766326;                         05103500 T  0083:0
C    ::: @1174731253373511;     CL    ::: @0001657523134265;                         05103600 T  0085:0
B    ::: @1163432604327011;     BL    ::: @0002027630376772;                         05103700 T  0087:0
A    ::: @1151727757377602;     AL    ::: @0006130725275347;                         05103800 T  0089:0
END DEXP;                                                                            05103900 T  0091:0
                                                                                                 SIZE= 0092 WORDS
```

```
PROCEDURE CEXP  ;          % 100                                                     05200000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00497
COMMENT COMPLEX EXPONENTIAL INTRINSIC;   %  PF  JUNE 67                             05200100 T  0000:0
BEGIN REAL X = -1,                                                                   05200200 T  0000:0
          Y = -2,                                                                    05200300 T  0000:0
       JUNK = 17 ;                                                                   05200400 T  0000:0
     LABEL EMAX,TOPI,PI2,TPI2;                                                       05200800 T  0000:0
IF ABS(X)>P(EMAX) THEN P(MKS,INTCALL(15,FORTERRI)) ;                                 05200900 T  0000:0
P(MKS,INTCALL(X,EXPI),,X,STN,MKS,1,MKS,Y,TOPI,MOD,DUP,SSP,,Y,STD,                    05201000 T  0003:2
  CALLINT(SINI),DUP,X,MUL,,JUNK,STD,DUP,MUL,SUB,CALLINT(SQRTI),MUL) ;                05201100 T  0008:2
IF Y > P(PI2) THEN IF Y < P(TPI2) THEN P(CHS);                                       05201200 T  0014:1
P(JUNK,XCH,,JUNK,STD,RTN);                                                           05201300 T  0017:0
EMAX ::: @1122360000000000;                                                          05201400 T  0018:1
TOPI ::: @1146220773250421;                                                          05201500 T  0020:0
PI2  ::: @1141444176652104;                                                          05201600 T  0021:0
TPI2 ::: @1144554574376314;                                                          05201700 T  0022:0
END CEXP;                                                                            05201800 T  0023:0
                                                                                                 SIZE= 0024 WORDS
```

```
PROCEDURE DLOG  ;          % 101                                                     05300000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00498
COMMENT DOUBLE PRECISION NATURAL LOG INTRINSIC;   %  PF  JUNE 67                   05300100 T  0000:0
BEGIN REAL X  = -1,                                                                  05300200 T  0000:0
          XL = -2,                                                                   05300300 T  0000:0
          JUNK = 17,                                                                 05300400 T  0000:0
          T,TL;                                                                      05300500 T  0000:0
```

```
      INTEGER N;                                                                05300600 T  0000:0
      BOOLEAN LESS1;                                                            05300700 T  0000:0
      LABEL HAF,LG2,LG2L,SQ2,SQ2L,A,AL,B,BL,C,CL,D,DL,E,EL,F,FL,G,GL,           05300800 T  0000:0
            H,HL,I,IL,J,JL;                                                     05300900 T  0000:0
DEFINE TIMES = NOP,DLA,XL,X,NOP,DLM#;                                           05301000 T  0000:0
IF X LEQ 0 THEN P(MKS,INTCALL(16+(X#0),FORTERRI)) ;                             05301100 T  0000:0
IF LESS1 ← X < 1 THEN P(0,1,XL,X,DLD,,X,STD,,XL,STD);                           05301200 T  0005:1
P(1,,N,STN,,JUNK,STD);                                                          05301300 T  0009:1
WHILE (JUNK ← P(JUNK,DUP,ADD)) ≤ X DO N ← N + 1;                                05301400 T  0010:2
IF P(XL,X,0,JUNK,DLD,,JUNK,STD,,T,STN,JUNK,SQ2L,SQ2,DLS,XCH,DEL) < 0            05301500 T  0014:2
THEN                                                                            05301550 T  0018:1
    BEGIN N ← N - 1;                                                            05301600 T  0018:3
          P(HAF,,TL,STD);                                                       05301700 T  0020:2
    END ELSE TL ← 1;                                                            05301800 T  0021:1
P(T,JUNK,0,TL,DLS,T,JUNK,0,TL,DLA,DLD,,JUNK,STD,,T,STN,DUP,JUNK,XCH,            05301900 T  0022:2
  JUNK,DLM,,X,STD,,XL,STN,X,JL,J,DLM);                                          05301950 T  0027:0
:: P(IL,I,TIMES,HL,H,TIMES,GL,G,TIMES,FL,F,TIMES,EL,E,TIMES,                    05302000 T  0029:2
     DL,D,TIMES,CL,C,TIMES,BL,B,TIMES,AL,A,TIMES,                               05302100 T  0040:0
    0,2,DLA,T,JUNK,DLM,0,N,LG2L,LG2,DLM,DLA,,JUNK,STD);                         05302200 T  0048:0
IF LESS1 THEN P(JUNK,CHS,,JUNK,STD);                                            05302300 T  0051:2
P(RTN);                                                                         05302400 T  0053:1
HAF  ::: @1154000000000000;                                                     05302500 T  0053:2
LG2  ::: @1155427102775750;    LG2L  ::: @0007173632571165;                     05302600 T  0055:0
SQ2  ::: @1155520236314774;    SQ2L  ::: @0007363110213136;                     05302700 T  0057:0
J    ::: @1167100510467432;    JL    ::: @0002164460474016;                     05302800 T  0059:0
I    ::: @1166521204435224;    IL    ::: @0007651024467003;                     05302900 T  0061:0
H    ::: @1167420605757260;    HL    ::: @0002135500773125;                     05303000 T  0063:0
G    ::: @1151042101275720;    GL    ::: @0000102676565544;                     05303100 T  0065:0
F    ::: @1151166116643351;    FL    ::: @0001161621531011;                     05303200 T  0067:0
E    ::: @1151350564271710;    EL    ::: @0007071635510300;                     05303300 T  0069:0
D    ::: @1151616161616162;    DL    ::: @0006643172311051;                     05303400 T  0071:0
C    ::: @1152222222222222;    CL    ::: @0002176633022026;                     05303500 T  0073:0
B    ::: @1153146314631463;    BL    ::: @0001463176726243;                     05303600 T  0075:0
A    ::: @1155252525252525;    AL    ::: @0002525252507053;                     05303700 T  0077:0
END DLOG;                                                                       05303800 T  0079:0
                                                                                        SIZE= 0080 WORDS
```

```
PROCEDURE CLOG  ;          % 102                                                05400000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00501
COMMENT COMPLEX LOG INTRINSIC;   %  PF  JUNE 67                                 05400100 T  0000:0
BEGIN REAL X = -1,                                                              05400200 T  0000:0
          Y = -2,                                                               05400300 T  0000:0
        JUNK = 17 ;                                                             05400400 T  0000:0
IF Y=0 THEN                                                                     05400900 T  0000:0
    IF X=0 THEN P(MKS,INTCALL(18,FORTERRI))                                     05400950 T  0000:3
    ELSE IF X>0 THEN P(MKS,INTCALL(X,LNI),,JUNK,STD,0,RTN) ;                    05400975 T  0004:2
JUNK←P(MKS,INTCALL(P(MKS,X,INTCALL(Y,CABSI)),LNI)) ;                            05401000 T  0009:1
P(MKS,Y,INTCALL(X,ATAN2I),RTN) ;                                                05401100 T  0013:3
END CLOG;                                                                       05401200 T  0016:1
                                                                                        SIZE= 0017 WORDS
```

```
PROCEDURE ALOG10;          % 103                                                05500000 T  0000:0
```

```
                                                                                                                START OF REL SEGMENT; DISK ADDRESS = 00502
COMMENT  LOG BASE 10 INTRINSIC;  %  PF  JUNE 67                                                                05500100 T  0000:0
BEGIN REAL X=-1 ;                                                                                               05500200 T  0000:0
           LABEL LGI;                                                                                           05500400 T  0000:0
IF X LEQ 0 THEN P(MKS,INTCALL(19+(X#0),FORTERRI)) ;                                                             05500500 T  0000:0
P(MKS,INTCALL(X,LNI),LGI,MUL,RTN) ;                                                                             05500600 T  0004:1
LGI  ::: @1153362675425116;                                                                                     05500700 T  0007:0
END ALOG10;                                                                                                     05500800 T  0008:0
                                                                                                                        SIZE= 0009 WORDS


PROCEDURE DLOG10;          % 104                                                                                 05600000 T  0000:0
                                                                                                                START OF REL SEGMENT; DISK ADDRESS = 00503
COMMENT DOUBLE PRECISION COMMON LOG INTRINSIC;   %  PF  JUNE 67                                                05600100 T  0000:0
BEGIN REAL X  = -1,                                                                                             05600200 T  0000:0
           XL = -2,                                                                                             05600300 T  0000:0
         JUNK = 17 ;                                                                                            05600400 T  0000:0
           LABEL LGI,LGIL;                                                                                      05600600 T  0000:0
IF X LEQ 0 THEN P(MKS,INTCALL(21+(X#0),FORTERRI)) ;                                                             05600700 T  0000:0
P(MKS,XL,INTCALL(X,DLOGI),JUNK,LGIL,LGI,DLM,,JUNK,STD,RTN) ;                                                    05600800 T  0004:1
LGI  ::: @1153362675425115;    LGIL  ::: @0006241614523261;                                                     05600900 T  0008:1
END DLOG10;                                                                                                     05601000 T  0011:0
                                                                                                                        SIZE= 0012 WORDS


PROCEDURE DSIN  ;          % 105                                                                                05700000 T  0000:0
                                                                                                                START OF REL SEGMENT; DISK ADDRESS = 00504
COMMENT DOUBLE PRECISION SINE INTRINSIC;   %  PF  JUNE 67                                                       05700100 T  0000:0
BEGIN REAL X  = -1,                                                                                             05700200 T  0000:0
           XL = -2,                                                                                             05700300 T  0000:0
           JUNK = 17,                                                                                           05700400 T  0000:0
           T;                                                                                                   05700600 T  0000:0
      BOOLEAN SIG;                                                                                              05700700 T  0000:0
      LABEL TOPI,TOPIL,PI,PIL,PI2,PI2L,TPI2,TPI2L,                                                              05700800 T  0000:0
              A,AL,B,BL,C,CL,D,DL,E,EL,F,FL,G,GL,H,HL,I,IL,J,JL;                                                05700900 T  0000:0
DEFINE ADDER = NOP,DLA,T,JUNK,NOP,DLM#,                                                                         05701000 T  0000:0
       SUBER = NOP,DLS,T,JUNK,NOP,DLM#;                                                                         05701050 T  0000:0
IF SIG ← X.[1:1] THEN X ← P(X,SSP);                                                                             05701100 T  0000:0
IF P(MKS,XL,X,TOPIL,INTCALL(TOPI,DMODI),JUNK,,X,STD,,XL,STN,X,PI2L,PI2,                                         05701200 T  0003:1
     DLS,XCH,DEL)>0                                                                                             05701250 T  0008:0
 THEN IF P(XL,X,PIL,PI,DLS,XCH,DEL) ≤ 0                                                                         05701300 T  0008:3
     THEN P(PIL,PI,XL,X,DLS)                                                                                    05701400 T  0011:2
     ELSE BEGIN SIG ← NOT SIG;                                                                                  05701500 T  0013:3
               IF P(XL,X,TPI2L,TPI2,DLS,XCH,DEL) ≤ 0                                                            05701600 T  0015:1
               THEN P(XL,X,PIL,PI,DLS)                                                                          05701700 T  0017:0
               ELSE P(TOPIL,TOPI,XL,X,DLS);                                                                     05701800 T  0019:1
          END                                                                                                   05701900 T  0021:0
 ELSE P(XL,X);                                                                                                  05702000 T  0021:0
P(,X,STD,,XL,STN,DUP,X,XCH,X,DLM,,JUNK,STD,,T,STN,JUNK,JL,J,DLM);                                               05702100 T  0022:0
:: P(IL,I,SUBER,HL,H,ADDER,GL,G,SUBER,FL,F,ADDER,EL,E,SUBER,DL,D,ADDER,                                         05702200 T  0026:1
     CL,C,SUBER,BL,B,ADDER,AL,A,SUBER,0,1,DLA,XL,X,DLM,,JUNK,STD);                                             05702300 T  0039:0
IF SIG THEN P(JUNK,CHS,,JUNK,STD);                                                                              05702400 T  0047:0
P(RTN);                                                                                                         05702500 T  0048:3
```

```
PI2  ::: @1141444176652104;     PI2L  ::: @0001321410646113;
PI   ::: @1143110375524210;     PIL   ::: @0002643021514230;
TPI2 ::: @1144554574376314;     TPI2L ::: @0004164432362343;
TOPI ::: @1146220773250420;     TOPIL ::: @0005506043230461;
J    ::: @1421317506616043;     JL    ::: @0004106341505647;
I    ::: @1371136261610121;     IL    ::: @0001561406721354;
G    ::: @1323271771732327;     GL    ::: @0001122361440352;
F    ::: @1271302221411627;     FL    ::: @0002101305056316;
E    ::: @1253271442547752;     EL    ::: @0002347333135765;
D    ::: @1235616743512533;     DL    ::: @0000704703144000;
C    ::: @1216400640064006;     CL    ::: @0004006354436671;
B    ::: @1174210421042104;     BL    ::: @0002104210366543;
A    ::: @1151252525252525;     AL    ::: @0002525252525234;
H    ::: @1356251301236324;;    HL    ::: @0007344376112457;
END DSIN;
```

```
05702600 T  0049:0
05702700 T  0051:0
05702800 T  0053:0
05702900 T  0055:0
05703000 T  0057:0
05703100 T  0059:0
05703200 T  0061:0
05703300 T  0063:0
05703400 T  0065:0
05703500 T  0067:0
05703600 T  0069:0
05703700 T  0071:0
05703800 T  0073:0
05703900 T  0075:0
05704000 T  0077:0
                SIZE= 0078 WORDS
```

```
PROCEDURE CSIN  ;          % 106

COMMENT COMPLEX SINE INTRINSIC;   %  PF  JUNE 67
BEGIN REAL X = -1,
           Y = -2,
           JUNK = 17,
           T;
      LABEL EMAX,HAF,TOPI,PI2,TPI2;
IF ABS(Y)>P(EMAX) THEN P(MKS,INTCALL(23,FORTERRI)) ;
P(MKS,INTCALL(Y,EXPI),DUP,DUP,1,XCH,/,,Y,STN,SUB,HAF,MUL,,T,STD,
  Y,ADD,HAF,MUL,MKS,X,TOPI,MOD,DUP,SSP,,X,STD,CALLINT(SINI),,Y,STN,MUL,
  MKS,1,Y,DUP,MUL,SUB,CALLINT(SQRTI),T,MUL) ;
IF X > P(PI2) THEN IF X < P(TPI2) THEN P(CHS);
P(XCH,,JUNK,STD,RTN);
EMAX ::: @1122360000000000;
HAF  ::: @1154000000000000;
TOPI ::: @1146220773250421;
PI2  ::: @1141444176652104;
TPI2 ::: @1144554574376314;
END CSINE;
```

```
05800000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00507
05800100 T  0000:0
05800200 T  0000:0
05800300 T  0000:0
05800400 T  0000:0
05800800 T  0000:0
05800900 T  0000:0
05801000 T  0000:0
05801100 T  0003:3
05801200 T  0008:3
05801300 T  0014:1
05801400 T  0018:0
05801500 T  0020:3
05801600 T  0021:3
05801700 T  0023:0
05801800 T  0024:0
05801900 T  0025:0
05802000 T  0026:0
05802100 T  0027:0
                SIZE= 0028 WORDS
```

```
PROCEDURE DCOS  ;          % 107

COMMENT DOUBLE PRECISION COSINE INTRINSIC;   %  PF  JUNE 67
BEGIN REAL X  = -1,
           XL = -2,
           LOW = -4 ;
           LABEL PI2,PI2L;
P(MKS,XL,X,PI2L,PI2,DLA,CALLINT(DSINI),RTN) ;
PI2  ::: @1141444176652104;     PI2L  ::: @0001321410646113;
END DCOS;
```

```
05900000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00508
05900100 T  0000:0
05900200 T  0000:0
05900300 T  0000:0
05900400 T  0000:0
05900600 T  0000:0
05900700 T  0000:0
05900800 T  0003:2
05900900 T  0006:0
                SIZE= 0007 WORDS
```

```
PROCEDURE CCOS  ;          % 110

COMMENT COMPLEX COSINE INTRINSIC;   %  PF  JUN 67
BEGIN REAL X = -1,
           Y = -2,
           JUNK = 17,
           T;
           LABEL EMAX,HAF,TOPI,TPI2,PI2,MHAF;
IF ABS(Y)>P(EMAX) THEN P(MKS,INTCALL(24,FORTERRI)) ;
P(MKS,INTCALL(Y,EXPI),DUP,DUP,1,XCH,/,.Y,STN,SUB,MHAF,MUL,MKS,X,TOPI,
  MOD,DUP,SSP,.X,STD,CALLINT(SINI),.T,STN,MUL,.JUNK,STD,Y,ADD,HAF,MUL,
  MKS,1,T,DUP,MUL,SUB,CALLINT(SQRTI),MUL) ;
IF X > P(PI2) THEN IF X < P(TPI2) THEN P(CHS);
P(JUNK,XCH,,JUNK,STD,RTN);
EMAX :::: @1122360000000000;
HAF  :::: @1154000000000000;
TOPI :::: @1146220773250421;
TPI2 :::: @1144554574376314;
PI2  :::: @1141444176652104;
MHAF :::: @3154000000000000;
END CCOS;
```

```
06000000 T  0000:0
                    START OF REL SEGMENT; DISK ADDRESS = 00509
06000100 T  0000:0
06000200 T  0000:0
06000300 T  0000:0
06000400 T  0000:0
06000800 T  0000:0
06000900 T  0000:0
06001000 T  0000:0
06001100 T  0003:3
06001200 T  0009:0
06001300 T  0014:1
06001400 T  0017:3
06001500 T  0020:2
06001600 T  0021:3
06001700 T  0023:0
06001800 T  0024:0
06001900 T  0025:0
06002000 T  0026:0
06002100 T  0027:0
06002200 T  0028:0
                    SIZE= 0029 WORDS
```

```
PROCEDURE TANF  ;          % 111

COMMENT TANGENT INTRINSIC;                  %  PF  MAY 67
BEGIN REAL RSQ;
      REAL X = -1;
      INTEGER Q; BOOLEAN S;
      LABEL L1,L2,PMAX,MMAX,PI,PI2,PI4;
IF S ← X.[1:1] THEN X ← P(X,SSP);
IF (Q ← P(X,PI,MOD,,X,STN,PI4,DIV)) ≠ 0 THEN X ←
IF Q=1 THEN P(PI2,X,SUB) ELSE IF Q=2 THEN P(X,PI2,SUB) ELSE P(PI,X,SUB);
IF X = 0 THEN BEGIN IF Q = 1 THEN P(PMAX) ELSE IF Q = 2 THEN P(MMAX)
                    ELSE P(0);   GO TO L1;
               END;
P(1,X,DUP,MUL,,RSQ,STN,DUP,,.0097433825958,MUL,CHS,1,ADD,MUL,
                    16,248537744,SUB,
                48,7456132319,RSQ,/,6,2497075488,SUB,RSQ,DUP,
                .000361003565256,MUL,CHS,,136381360679,ADD,MUL,ADD,
               /,ADD);
IF Q = 0 THEN P(X,XCH,/)
        ELSE IF Q = 1 THEN P(X,/)
                      ELSE IF Q = 2 THEN P(X,/,CHS)
                                    ELSE P(X,XCH,/,CHS);
L1:
IF S THEN P(CHS);
P(RTN);
PMAX :::: @0777777777777777;
MMAX :::: @2777777777777777;
PI   :::: @1143110375524210;
PI2  :::: @1141444176652104;
```

```
06100000 T  0000:0
                    START OF REL SEGMENT; DISK ADDRESS = 00510
06100100 T  0000:0
06100200 T  0000:0
06100300 T  0000:0
06100400 T  0000:0
06100500 T  0000:0
06100600 T  0000:0
06100700 T  0003:2
06100800 T  0006:3
06100900 T  0012:2
06101000 T  0017:3
06101100 T  0019:0
06101200 T  0019:0
06101300 T  0022:1
06101400 T  0022:3
06101500 T  0024:2
06101600 T  0026:1
06101700 T  0026:3
06101800 T  0028:3
06101900 T  0037:3
06102000 T  0040:1
06102100 T  0041:3
06102200 T  0041:3
06102300 T  0042:3
06102400 T  0043:0
06102500 T  0044:0
06102600 T  0045:0
06102700 T  0046:0
```

```
PI4  ::: @1156220773250421;                                                    06102800 T  0047:0
END TAN;                                                                       06102900 T  0048:0
                                                                                      SIZE= 0049 WORDS



PROCEDURE COTAN ;          % 112                                               06200000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00512
COMMENT COTANGENT INTRINSIC;                          %  PF  MAY 67           06200100 T  0000:0
BEGIN REAL T ;                                                                 06200200 T  0000:0
      REAL X = -1;                                                             06200300 T  0000:0
      LABEL PMAX;                                                              06200400 T  0000:0
      IF (T←P(MKS,INTCALL(X,TANI)))=0 THEN P(PMAX)                             06200500 T  0000:0
      ELSE P(1,T,/);                                                           06200600 T  0004:0
P(RTN);                                                                        06200700 T  0005:1
PMAX ::: @0777777777777777;                                                    06200800 T  0005:2
END COTAN;                                                                     06200900 T  0007:0
                                                                                      SIZE= 0008 WORDS



PROCEDURE DATAN ;          % 113                                               06300000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00513
COMMENT DOUBLE PRECISION ARC TANGENT INTRINSIC;  %  PF  JUNE 67               06300100 T  0000:0
BEGIN REAL X  = -1,                                                            06300200 T  0000:0
          XL = -2,                                                             06300300 T  0000:0
          JUNK = 17,                                                           06300400 T  0000:0
          T;                                                                   06300500 T  0000:0
      BOOLEAN S,U,Y;                                                           06300600 T  0000:0
      LABEL SR3,SR3L,PI6,PI6L,PI2,PI2L,A,AL,B,BL,C,CL,D,DL,E,EL,F,FL,          06300700 T  0000:0
            G,GL,H,HL,I,IL,J,JL,K,KL,L,LL,M,ML;                                06300800 T  0000:0
DEFINE AM = NOP,DLA,T,JUNK,NOP,DLM#,    SM = NOP,DLS,T,JUNK,NOP,DLM#;          06300900 T  0000:0
IF S ← X.[1:1] THEN X ← P(X,SSP);                                              06301000 T  0000:0
IF Y ← X > 1 THEN P(0,1,XL,X,DLD,.X,STD,.XL,STD);                              06301100 T  0003:3
IF U ← X > .2679491924311 THEN P(SR3L,SR3,0,4,SR3L,SR3,XL,X,DLA,               06301200 T  0007:3
                          DLD,DLS,.X,STD,.XL,STD);                             06301300 T  0011:3
P(XL,DUP,X,XCH,X,DLM,.JUNK,STD,.T,STN,JUNK,ML,M,DLM);                          06301400 T  0013:1
:: P(LL,L,AM,KL,K,SM,JL,J,AM,IL,I,SM,HL,H,AM,GL,G,SM,FL,F,AM,EL,E,SM,          06301500 T  0016:3
    DL,D,AM,CL,C,SM,BL,B,AM,AL,A,SM,0,1,DLA,XL,X,DLM,.JUNK,STD);               06301600 T  0033:0
IF U THEN P(JUNK,PI6L,PI6,DLA,.JUNK,STD);                                      06301700 T  0043:0
IF Y THEN P(JUNK,PI2L,PI2,DLS,CHS,.JUNK,STD);                                  06301800 T  0045:1
IF S THEN P(JUNK,CHS,.JUNK,STD);                                               06301900 T  0047:3
P(RTN);                                                                        06302000 T  0049:2
SR3  ::: @1141566636564130;     SR3L  ::: @0002312516354455;                   06302100 T  0049:3
PI6  ::: @1154140522160265;     PI6L  ::: @0006331302145566;                   06302200 T  0052:0
PI2  ::: @1141444176652104;     PI2L  ::: @0001321410646113;                   06302300 T  0054:0
M    ::: @1161401124046414;     ML    ::: @0004072764260344;                   06302400 T  0056:0
L    ::: @1162303273323564;     LL    ::: @0001630262103372;                   06302500 T  0058:0
K    ::: @1162605113035023;     KL    ::: @0004301553367304;                   06302600 T  0060:0
J    ::: @1163027321345406;     JL    ::: @0003326323362544;                   06302700 T  0062:0
I    ::: @1163274446267506;     IL    ::: @0001464411354576;                   06302800 T  0064:0
H    ::: @1163607415673413;     HL    ::: @0001424256207512;                   06302900 T  0066:0
G    ::: @1164210421020314;     GL    ::: @0001737716236562;                   06303000 T  0068:0
F    ::: @1164730473047014;     FL    ::: @0006266260505571;                   06303100 T  0070:0
E    ::: @1165642721350561;     EL    ::: @0006467443753240;                   06303200 T  0072:0
```

```
D     ::: @1167070707070707;     DL    ::: @0000552165603175;
C     ::: @1151111111111111;     CL    ::: @0001111051232710;
B     ::: @1151463146314631;     BL    ::: @0004631463070633;
A     ::: @1152525252525252;     AL    ::: @0005252525252470;
END DATAN;
```

```
                                                                    06303300 T  0074:0
                                                                    06303400 T  0076:0
                                                                    06303500 T  0078:0
                                                                    06303600 T  0080:0
                                                                    06303700 T  0082:0
                                                                          SIZE= 0084 WORDS
```

```
PROCEDURE ATAN2 ;          % 114
```

```
                                                                    06400000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00516
COMMENT ARC TANGENT OF A/B INTRINSIC;          %  PF  MAY 67       06400100 T  0000:0
BEGIN                                                               06400200 T  0000:0
      REAL A = -2, B = -1;                                          06400300 T  0000:0
      LABEL PI,PI2,MPI2;                                            06400400 T  0000:0
IF B > 0 THEN                                                       06400500 T  0000:0
     IF A≠0 THEN P(MKS,INTCALL(A/B,ARCTANI))                        06400550 T  0000:3
     ELSE P(0)                                                      06400600 T  0005:0
ELSE                                                                06400650 T  0005:3
     IF B < 0 THEN                                                  06400700 T  0005:3
          IF A>0 THEN P(MKS,INTCALL(A/B,ARCTANI),PI,ADD)            06400750 T  0007:0
          ELSE                                                      06400800 T  0011:3
               IF A<0 THEN P(MKS,INTCALL(A/B,ARCTANI),PI,SUB)       06400850 T  0011:3
               ELSE P(PI)                                           06400900 T  0016:2
     ELSE                                                           06400950 T  0017:1
               IF A < 0 THEN P(MPI2)                                06401000 T  0017:1
               ELSE P(PI2);                                         06401050 T  0019:1
P(RTN);                                                             06401100 T  0020:0
PI   ::: @1143110375524210;                                         06401200 T  0020:1
PI2  ::: @1141444176652104;                                         06401300 T  0022:0
MPI2 ::: @3141444176652104;                                         06401400 T  0023:0
END ATAN2;                                                          06401500 T  0024:0
                                                                          SIZE= 0025 WORDS
```

```
PROCEDURE DATAN2;          % 115
```

```
                                                                    06500000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00517
COMMENT DOUBLE PRECISION ARC TANGENT OF A/B INTRINSIC; %  PF  JUNE 67  06500100 T  0000:0
BEGIN REAL B  = -1,                                                 06500200 T  0000:0
          BL = -2,                                                  06500300 T  0000:0
          A  = -3,                                                  06500400 T  0000:0
          AL = -4,                                                  06500500 T  0000:0
          JUNK = 17 ;                                               06500600 T  0000:0
     LABEL PI,PIL,PI2,MPI2,PI2L;                                    06500800 T  0000:0
IF A ≠ 0 AND B ≠ 0 THEN                                             06500900 T  0000:0
    BEGIN P(MKS,AL,A,BL,B,DLD,CALLINT(DATANI)) ;                    06501000 T  0001:3
        IF B.[1:1] THEN IF A > 0 THEN P(JUNK,PIL,PI,DLA,,JUNK,STD,RTN)  06501100 T  0005:2
                   ELSE P(JUNK,PIL,PI,DLS,,JUNK,STD,RTN);           06501300 T  0009:3
    END ELSE                                                        06501500 T  0012:0
IF B = 0 THEN  IF A.[1:1] THEN P(MPI2,,JUNK,STD,PI2L,RTN)           06501600 T  0012:0
                          ELSE P( PI2,,JUNK,STD,PI2L,RTN)           06501700 T  0016:1
               ELSE IF B.[1:1] THEN P(PI,,JUNK,STD,PIL,RTN)         06501800 T  0018:0
                               ELSE P(0,,JUNK,STN,RTN);             06501900 T  0021:0
P(RTN);                                                             06502000 T  0022:2
```

```
PI   ::: @11431103755242l0;     PIL   ::: @0002643021514230;                    06502100 T  0022:3
PI2  ::: @1141444176652104;     PI2L  ::: @0001321410646113;                    06502200 T  0025:0
MPI2 ::: @3141444176652104;                                                     06502300 T  0027:0
END DATAN2;                                                                     06502400 T  0028:0
                                                                                        SIZE= 0029 WORDS


PROCEDURE ARSIN ;          % 116                                                06600000 T  0000:0
                                                                        START OF REL SEGMENT; DISK ADDRESS = 00518
COMMENT ARC SINE INTRINSIC;                        %  PF  MAY 67               06600100 T  0000:0
BEGIN REAL X =-1,                                                               06600200 T  0000:0
          XSQ;                                                                  06600300 T  0000:0
      BOOLEAN S,U;                                                              06600400 T  0000:0
      LABEL PI2,HAF,A,B,C,D,E,F,G,H,I,J,K,L,M,N;                                06600500 T  0000:0
      DEFINE TIMES = ADD,XSQ,MUL#;                                              06600600 T  0000:0
IF S ← X,[1:1] THEN X ← P(X,SSP);                                               06600700 T  0000:0
IF X>1 THEN P(MKS,INTCALL(26,FORTERRI)) ;                                       06600800 T  0003:2
IF U←X>P(HAF) THEN X←P(MKS,1,X,SUB,HAF,MUL,,XSQ,STN,CALLINT(SQRTI))             06600900 T  0006:3
ELSE XSQ ← X×X;                                                                 06601000 T  0012:1
:: P(NOP,A,XSQ,MUL,B,TIMES,C,TIMES,D,TIMES,E,TIMES,F,TIMES,G,TIMES,             06601100 T  0014:2
          H,TIMES,I,TIMES,J,TIMES,K,TIMES,L,TIMES,M,TIMES,N,TIMES,              06601200 T  0022:0
          1,ADD,X,MUL);                                                         06601300 T  0029:0
IF U THEN P(DUP,ADD,CHS,PI2,ADD);                                               06601400 T  0030:0
IF S THEN P(CHS);                                                               06601500 T  0032:0
P(RTN);                                                                         06601600 T  0033:0
PI2 ::: @1141444176652104;    HAF ::: @1154000000000000;                        06601700 T  0033:1
  A ::: @1172506721410650;      B ::: @1172740556641135;                        06601800 T  0036:0
  C ::: @1173232061727030;      D ::: @1173574736467510;                        06601900 T  0038:0
  E ::: @1174227363636371;      F ::: @1174776745032742;                        06602000 T  0040:0
  G ::: @1175724170360740;      H ::: @1177114631463146;                        06602100 T  0042:0
  I ::: @1161070473047305;      J ::: @1161335056427214;                        06602200 T  0044:0
  K ::: @1161743434343434;      L ::: @1162666666666667;                        06602300 T  0046:0
  M ::: @1164631463146315;      N ::: @1151252525252526;                        06602400 T  0048:0
END ARSIN;                                                                      06602500 T  0050:0
                                                                                        SIZE= 0051 WORDS


PROCEDURE ARCOS ;          % 117                                                06700000 T  0000:0
                                                                        START OF REL SEGMENT; DISK ADDRESS = 00520
COMMENT ARC COSINE INTRINSIC;                      %  PF  MAY 67               06700100 T  0000:0
BEGIN REAL X = -1 ;                                                             06700200 T  0000:0
      LABEL PI2;                                                                06700400 T  0000:0
IF ABS(X)>1 THEN P(MKS,INTCALL(25,FORTERRI)) ;                                  06700500 T  0000:0
      P(PI2,MKS,INTCALL(X,ARSINI),SUB) ;                                        06700600 T  0003:2
      P(RTN);                                                                   06700700 T  0006:0
PI2  ::: @1141444176652104;                                                     06700800 T  0006:1
END ARCOS;                                                                      06700900 T  0008:0
                                                                                        SIZE= 0009 WORDS


PROCEDURE SINH  ;          % 120                                                06800000 T  0000:0
```

```
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00521
COMMENT HYPERBOLIC SINE INTRINSIC;                  %  PF  MAY 67                  06800100 T  0000:0
BEGIN REAL X = -1 ;                                                                 06800200 T  0000:0
      BOOLEAN S;                                                                    06800400 T  0000:0
      LABEL EMAX;                                                                   06800500 T  0000:0
DEFINE S1AM = /,1,ADD,MUL#;                                                         06800600 T  0000:0
IF S ← X.[1:1] THEN X ← P(X,SSP);                                                   06800700 T  0000:0
IF X>P(EMAX) THEN P(MKS,INTCALL(29,FORTERRI)) ;                                     06800800 T  0003:0
IF X ≤ .5 THEN                                                                      06800900 T  0006:1
  P(X,DUP,DUP,MUL,DUP,DUP,DUP,72,S1AM,42,S1AM,20,S1AM,6,S1AM) ELSE                  06801000 T  0007:0
  P(MKS,INTCALL(X,EXPI),DUP,1,XCH,/,SUB,.5,MUL) ;                                   06801100 T  0014:1
IF S THEN P(CHS);                                                                   06801200 T  0019:3
P(RTN);                                                                             06801300 T  0020:3
EMAX ::: @1122360000000000;                                                         06801400 T  0021:0
END SINH;                                                                           06801500 T  0022:0
                                                                                          SIZE= 0024 WORDS


PROCEDURE COSH  ;          % 121                                                    06900000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00522
COMMENT HYPERBOLIC COSINE INTRINSIC;                %  PF  MAY 67                  06900100 T  0000:0
BEGIN REAL X = -1,                                                                  06900200 T  0000:0
           T;                                                                       06900300 T  0000:0
      LABEL EMAX;                                                                   06900400 T  0000:0
DEFINE S1AM = /,1,ADD,MUL#;                                                         06900500 T  0000:0
IF (T←ABS(X))>P(EMAX) THEN P(MKS,INTCALL(30,FORTERRI)) ;                            06900600 T  0000:0
IF T ≤ .75 THEN                                                                     06900700 T  0004:1
  P(X,DUP,MUL,DUP,DUP,DUP,DUP,90,S1AM,56,S1AM,30,S1AM,12,S1AM,                      06900800 T  0005:0
    .5,MUL,1,ADD) ELSE P(MKS,INTCALL(X,EXPI),DUP,1,XCH,/,ADD,.5,MUL) ;              06900900 T  0012:1
P(RTN);                                                                             06901000 T  0019:3
EMAX ::: @1122360000000000;                                                         06901100 T  0020:0
END COSH;                                                                           06901200 T  0021:0
                                                                                          SIZE= 0023 WORDS


PROCEDURE TANH  ;          % 122                                                    07000000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00523
COMMENT HYPERBOLIC TANGENT INTRINSIC;               %  PF  MAY 67                  07000100 T  0000:0
BEGIN REAL T ;                                                                      07000200 T  0000:0
      REAL X = -1;                                                                  07000300 T  0000:0
      BOOLEAN S;                                                                    07000400 T  0000:0
IF S ← X.[1:1] THEN X ← P(X,SSP);                                                   07000600 T  0000:0
IF X > 27 THEN IF S THEN P(1,CHS,RTN) ELSE P(1,RTN);                                07000700 T  0003:1
IF X ≤ .14 THEN                                                                     07000800 T  0007:0
  P(X,DUP,DUP,MUL,DUP,DUP,DUP,6.8888888889,MUL,17,SUB,                              07000900 T  0007:3
                        MUL,21,/,2,ADD,MUL,5,/,1,SUB,MUL,3,/,1,ADD,MUL)             07001000 T  0011:0
ELSE P(MKS,INTCALL(X,EXPI),DUP,.T,STN,1,XCH,/,DUP,T,ADD,.T,STD,SUB,T,/);            07001100 T  0015:0
IF S THEN P(CHS);                                                                   07001200 T  0023:2
P(RTN);                                                                             07001300 T  0024:2
END TANH;                                                                           07001500 T  0024:3
                                                                                          SIZE= 0025 WORDS
```

```
PROCEDURE DSQRT ;          % 123
COMMENT DOUBLE PRECISION SQUARE ROOT INTRINSIC;  %  PF  JUNE 67
BEGIN REAL X  = -1,
           XL = -2,
           JUNK = 17 ;
      LABEL HAF;
IF X LEQ 0 THEN IF X=0 THEN P(0,.JUNK,STN,RTN)
                ELSE P(MKS,INTCALL(27,FORTERRI)) ;
P(XL,X,0,MKS,INTCALL(X,SQRTI),.JUNK,STN,DLD,0,JUNK,DLA,0,HAF,DLM,.JUNK,
  STD,RTN) ;
HAF  ::: @1154000000000000;
END DSQRT;
```

```
                                                        07100000 T  0000:0
                                                START OF REL SEGMENT; DISK ADDRESS = 00524
                                                        07100100 T  0000:0
                                                        07100200 T  0000:0
                                                        07100300 T  0000:0
                                                        07100400 T  0000:0
                                                        07100600 T  0000:0
                                                        07100700 T  0000:0
                                                        07100710 T  0003:2
                                                        07100800 T  0006:0
                                                        07100810 T  0011:1
                                                        07100900 T  0011:3
                                                        07101000 T  0013:0
                                                              SIZE= 0014 WORDS
```

```
PROCEDURE CSQRT ;          % 124
COMMENT COMPLEX SQUARE ROOT INTRINSIC;   %   PF  JUNE 67
BEGIN REAL X = -1,
           Y = -2,
           JUNK = 17 ;
           LABEL HAF;
IF X = 0 THEN IF Y = 0 THEN P(0,.JUNK,STN,RTN);
P(MKS,INTCALL(P(MKS,X,INTCALL(Y,CABSI),X,SSP,ADD,HAF,MUL),SQRTI)) ;
IF X ≥ 0 THEN P(.JUNK,STN,DUP,ADD,Y,XCH,/,RTN)
         ELSE BEGIN IF Y.[1:1] THEN P(CHS);
                    P(DUP,DUP,ADD,Y,XCH,/,.JUNK,STD,RTN);
              END;
HAF  ::: @1154000000000000;
END CSQRT;
```

```
                                                        07200000 T  0000:0
                                                START OF REL SEGMENT; DISK ADDRESS = 00525
                                                        07200100 T  0000:0
                                                        07200200 T  0000:0
                                                        07200300 T  0000:0
                                                        07200400 T  0000:0
                                                        07200700 T  0000:0
                                                        07200800 T  0000:0
                                                        07200900 T  0003:2
                                                        07201000 T  0008:3
                                                        07201100 T  0012:0
                                                        07201200 T  0014:0
                                                        07201300 T  0016:1
                                                        07201400 T  0016:1
                                                        07201500 T  0018:0
                                                              SIZE= 0019 WORDS
```

```
PROCEDURE ERF   ;          % 125
COMMENT THE ERROR FUNCTION INTRINSIC;            %  PF  MAY 67
BEGIN REAL X = -1,
           XSQ,T,W;
      LABEL A,B,C,D,E,F,G,H,I,J,K,L,M,N,OVER,MSRTPI;
DEFINE MORE = ADD,XSQ,MUL#,  LESS = SUB,XSQ,MUL#;
IF (XSQ ← X×X) < 2.22 THEN
  :: P(NOP,A,XSQ,MUL,B,LESS,C,MORE,D,LESS,E,MORE,F,LESS,G,MORE,H,LESS,
       I,MORE,J,LESS,K,MORE,L,LESS,M,MORE,N,LESS,OVER,ADD,X,MUL,RTN);
IF XSQ < 24 THEN
BEGIN W ← (XSQ + 14.5)×(T ← XSQ + 6.6267867473) - 39.1779586414;
      T ← (XSQ + 12.5)×W - 45.5×T;
      W ← (XSQ + 10.5)×T - 33×W;
      T ← (XSQ + 8.5)×W - 22.5×T;
      W ← (XSQ + 6.5)×T - 14×W;
      T ← (XSQ + 4.5)×W - 7.5×T;
      W ← (XSQ + 2.5)×T - 3×W;
      T ← (XSQ + .5)×W - .5×T;
```

```
                                                        07300000 T  0000:0
                                                START OF REL SEGMENT; DISK ADDRESS = 00526
                                                        07300100 T  0000:0
                                                        07300200 T  0000:0
                                                        07300400 T  0000:0
                                                        07300500 T  0000:0
                                                        07300600 T  0000:0
                                                        07300700 T  0000:0
                                                        07300800 T  0002:2
                                                        07300900 T  0011:0
                                                        07301000 T  0018:1
                                                        07301100 T  0019:0
                                                        07301200 T  0022:3
                                                        07301300 T  0025:2
                                                        07301400 T  0028:1
                                                        07301500 T  0031:0
                                                        07301600 T  0033:3
                                                        07301700 T  0036:2
                                                        07301800 T  0039:1
```

```
    P(ABS(X),W,MUL,MKS,INTCALL(XSQ,EXPI),MSRTPI,MUL,T,MUL,/,1,ADD) ;                07301850 T  0042:0
END ELSE P(1);                                                                     07301900 T  0046:3
IF X.[1:1] THEN P(CHS);                                                            07302000 T  0062:1
P(RTN);                                                                            07302100 T  0063:3
A ::: @1321164756260433;      B ::: @1313314675626043;                             07302200 T  0064:0
C ::: @1306316666647563;      D ::: @1261242725431173;                             07302300 T  0066:0
E ::: @1251771347130371;      F ::: @1242575635313531;                             07302400 T  0068:0
G ::: @1233347466027367;      H ::: @1223723222675344;                             07302500 T  0070:0
I ::: @1213746431157302;      J ::: @1203400555500006;                             07302600 T  0072:0
K ::: @1172531336320715;      L ::: @1161560263430450;                             07302700 T  0074:0
M ::: @1167161362064016;      N ::: @1153004472153007;                             07302800 T  0076:0
OVER ::: @1141101565650103;     MSRTPI:::@3141613376110665;                        07302900 T  0078:0
END ERF;                                                                           07303000 T  0080:0
                                                                                          SIZE= 0081 WORDS



PROCEDURE GAMMA ;          % 126                                                   07400000 T  0000:0
                                                                START OF REL SEGMENT; DISK ADDRESS = 00529
COMMENT GAMMA INTRINSIC;                          %  PF  MAY 67                   07400100 T  0000:0
BEGIN REAL X = -1,                                                                 07400200 T  0000:0
          E,V,Y;                                                                   07400400 T  0000:0
      BOOLEAN S; INTEGER K;                                                        07400500 T  0000:0
      LABEL L1,PMAX,MMAX,PI,MPI;                                                   07400600 T  0000:0
DEFINE SUBMUL = SUB,E,MUL#, ADDMUL = ADD,E,MUL#;                                   07400700 T  0000:0
IF S ← X ≤ 0  THEN X ←  P(X,SSP);                                                  07400800 T  0000:0
IF X>52 THEN P(MKS,INTCALL(28,FORTERRI)) ;                                         07400900 T  0004:0
IF (E ← P(X,DUP,.Y,SND,.5,SUB,.K,ISN,SUB)) = 0 THEN                                07401000 T  0007:1
    BEGIN IF S THEN IF K.[47:1] THEN P(MMAX)ELSE P(PMAX)                           07401100 T  0010:2
                ELSE IF K ≤ 2 THEN P(1)                                            07401200 T  0015:1
                              ELSE GO TO L1;                                       07401300 T  0017:1
          P(RTN);                                                                  07401400 T  0017:1
    END;                                                                           07401500 T  0017:2
IF K ≤ 2 THEN V ← (IF K = 0 THEN P(1,X,DUP,1,ADD,MUL,/) ELSE                       07401600 T  0017:2
                   IF K = 1 THEN 1/X ELSE 1)                                       07401700 T  0022:1
         ELSE  L1: BEGIN X ← X - (V ← 1);                                          07401800 T  0025:0
                         DO V ← X×V UNTIL  (X ← X - 1) < 2;                        07401900 T  0027:3
                         IF E = 0 THEN P(V,RTN);                                   07402000 T  0031:1
                   END;                                                            07402100 T  0033:0
:: P(NOP,E,.00006771057117,MUL,                                                    07402200 T  0033:0
       .00034423420456,SUBMUL,                                                     07402300 T  0034:0
       .00153976810472,ADDMUL,                                                     07402400 T  0035:0
       .00246674798054,SUBMUL,                                                     07402500 T  0036:0
       .0109736958417,ADDMUL,                                                      07402600 T  0037:0
       .00021090746731,SUBMUL,                                                     07402700 T  0038:0
       .074237907606,ADDMUL,                                                       07402800 T  0039:0
       .081578218785,ADDMUL,                                                       07402900 T  0040:0
       .411840251796,ADDMUL,                                                       07403000 T  0041:0
       .422784336962,ADDMUL,                                                       07403100 T  0042:0
       .99999999999,ADD,V,MUL,.V,STN);                                             07403200 T  0043:0
IF S THEN P(DEL,MPI,MKS,INTCALL(P(PI)×Y,SINI),V,MUL,Y,MUL,/) ;                     07403300 T  0044:2
P(RTN);                                                                            07403400 T  0049:2
PI   ::: @1143110375524210;                                                        07403500 T  0049:3
MPI  ::: @3143110375524210;                                                        07403600 T  0051:0
PMAX ::: @0777777777777777;                                                        07403700 T  0052:0
MMAX ::: @2777777777777777;                                                        07403800 T  0053:0
```

```
END GAMMA;                                                                  07403900 T  0054:0
                                                                                  SIZE= 0066 WORDS



PROCEDURE ALGAMA;          % 127                                             07500000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00532
COMMENT LOG GAMMA INTRINSIC;                        %  PF  MAY 67           07500100 T  0000:0
BEGIN REAL X = -1,                                                          07500200 T  0000:0
          T;                                                                07500400 T  0000:0
DEFINE SUBMUL = SUB,T,MUL#, ADDMUL = ADD,T,MUL#;                            07500500 T  0000:0
IF X LEQ 0 THEN P(MKS,INTCALL(31+(X#0),FORTERRI)) ;                         07500600 T  0000:0
IF X<3.28 THEN P(MKS,INTCALL(P(MKS,INTCALL(X,GAMMAI)),LNI),RTN) ;           07500700 T  0004:2
              P(1,X,DUP,MUL,/,.T,SND);                                      07500800 T  0009:3
         :: P(NOP,1.392432216906,CHS,MUL,                                   07500900 T  0011:2
                  .179644372369,ADDMUL,                                     07501000 T  0013:0
                  .0295506535948,SUBMUL,                                    07501100 T  0014:0
                  .0064102564103,ADDMUL,                                    07501200 T  0015:0
                  .00191752691753,SUBMUL,                                   07501300 T  0016:0
                  .00084175084175,ADDMUL,                                   07501400 T  0017:0
                  .00059523809524,SUBMUL,                                   07501500 T  0018:0
                  .00079365079365,ADDMUL,                                   07501600 T  0019:0
                  .00277777777778,SUBMUL,                                   07501700 T  0020:0
                  .083333333333,ADD,X,/,.91893853321,ADD,                   07501800 T  0021:0
                X,DUP,.5,SUB,MKS,INTCALL(X,LNI),MUL,XCH,SUB,ADD) ;          07501900 T  0022:2
P(RTN);                                                                     07502000 T  0026:2
END ALGAMA;                                                                 07502100 T  0026:3
                                                                                  SIZE= 0040 WORDS



PROCEDURE ANDI  ;          % 130                                            07600000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00534
        BEGIN                                                               07600100 T  0000:0
        REAL A = - 1,B = -2;                                                07600200 T  0000:0
        P(A AND B,RTN);                                                     07600300 T  0000:0
        END ANDI;                                                           07600400 T  0001:0
                                                                                  SIZE= 0002 WORDS



PROCEDURE ORI   ;          % 131                                            07700000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00535
        BEGIN                                                               07700100 T  0000:0
        REAL A =-1,B = -2;                                                  07700200 T  0000:0
        P(A OR B,RTN);                                                      07700300 T  0000:0
        END ORI;                                                            07700400 T  0001:0
                                                                                  SIZE= 0002 WORDS



PROCEDURE CMPL  ;          % 132                                            07800000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00536
```

```
        BEGIN
        REAL A = - 1;
        P( (NOT A),RTN);
        END CMPL;
```

```
07800100 T  0000:0
07800200 T  0000:0
07800300 T  0000:0
07800400 T  0000:3
        SIZE= 0001 WORDS
```

```
PROCEDURE EQUIVP;       % 133

    BEGIN
        REAL A=-1,B = -2;
        P(A EQV B,RTN);
        END EQUIVP;
```

```
07900000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00537
07900100 T  0000:0
07900200 T  0000:0
07900300 T  0000:0
07900400 T  0001:0
        SIZE= 0002 WORDS
```

```
PROCEDURE FORTERR;          % 134     RUN-TIME ERRORS

    BEGIN
COMMENT  PROGRAM GENERATING VARIOUS ERROR MESSAGES WITH A DS,
    CODES 0 THRU 3 ARE USED BY THE FORMATING INTRINSICS,  CODES
    10 THRU 32 ARE USED BY VARIOUS MATH INTRINSICS;
        REAL CODE = -1,FID,MFID,IND,BUFF,A=-2,B=-3,C=-4,D=-5;
        ARRAY TPAR[*],FIB[*],FPB=3[*] ;
        NAME MEM = 2;
LABEL CD, F095, CD1, CD2, DC, DC1, DC2, G095 ;
LABEL CPLR,XTOI,CSSC,DMOD,DEXP,CEXP,DLGZ,DLGM,CLOG,ALTZ,ALTM,DLTZ,DLTM,
      CSIN,CCOS,ACOS,ASIN,DSQR,GAMA,SINH,COSH,ALGZ,ALGM,MAXN,ZERO,NGTV,
      L0,L1,L2,L3,LX,WRAPUP,FIGER;
LABEL L4, L5, L6;
SWITCH SW1 ← L0, L1, L2, L3, L4, L5, L6;
SWITCH SW2 ← CPLR,XTOI,CSSC,DMOD,DEXP,CEXP,DLGZ,DLGM,CLOG,ALTZ,ALTM,
     DLTZ,DLTM,CSIN,CCOS,ACOS,ASIN,DSQR,GAMA,SINH,COSH,ALGZ,ALGM;
DEFINE STREM = STREAM(D ← [TPAR[#, STO = STREM 0]])#, ST2 = STREM 2]])#;
DEFINE CC55(CC551) = CC551(DS←LIT"<"; SI←A1; DS←A13 CHR; DS←LIT">") #,
       NAS(NAS1,NAS2,NAS3) = SI←LOC NAS1; DS←NAS2 DEC; NAS3(DI←DI-4 ;
                             DS←LIT"*") #,
       CD5(CD51,CD52,CD53,CD54,CD55) = CC55(CD51); CD52(NAS(CD53,CD54,
                                       CD55)) #,
       SAVW=F #, SAVD=E #, WH2=B #, WH1=C #, R=G #;
SUBROUTINE GETFILE;
    BEGIN
        FIB ←  MEM[(NOT 2) INX A];
        MFID ← FPB[IND ← FIB[4].[13:11]];
        FID ← FPB[IND+1];
        B ← B + 1;
    END GETFILE ;
    REAL T1,T2,T3,T4,T5,E=-6,F=-7,G=-8,H=-9,I=-10,J=-11,K=-12 ;
    INTEGER IT2=T2 ;
    ARRAY TEN=22[*] ;
    LABEL LOOP, ALFA ;
    REAL SUBROUTINE SIZ ;
        BEGIN
```

```
07900410 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00538
07900500 T  0000:0
07900600 T  0000:0
07900700 T  0000:0
07900800 T  0000:0
07900900 T  0000:0
07901000 T  0000:0
07901100 T  0000:0
07901110 T  0000:0
07901200 T  0000:0
07901300 T  0000:0
07901400 T  0000:0
07901410 T  0000:0
07901500 T  0000:0
07901600 T  0000:0
07901700 T  0000:0
07901800 T  0000:0
07901810 T  0000:0
07901820 T  0000:0
07901825 T  0000:0
07901830 T  0000:0
07901835 T  0000:0
07901850 T  0000:0
07901900 T  0000:0
07902000 T  0001:0
07902100 T  0001:0
07902200 T  0003:1
07902300 T  0005:2
07902400 T  0007:0
07902405 T  0008:1
07902410 T  0008:2
07902415 T  0008:2
07902417 T  0008:2
07902420 T  0008:2
07902425 T  0008:2
07902430 T  0009:0
```

```
          TEN[T3]←TEN[68];  T1←0 ;                                                    07902435 T  0009:0
LOOP:     IF TEN[T1←T1+1]≤T2 THEN GO LOOP;   SIZ←T1 ;                                 07902440 T  0011:1
          END OF SIZ ;                                                                07902445 T  0014:1
% * * * * * * * * * * * * * * PROGRAM STARTS * * * * * * * * * * * * * * * *        07902447 T  0014:2
     TPAR←P([TPAR[1]],CFX,SFB)&17[8:38:10] ;                                         07902448 T  0014:2
     IF CODE=(-2) THEN                                                               07902450 T  0020:1
          BEGIN T3←5;  T2←B ;                                                        07902455 T  0021:1
          STREAM(E,D,C,B,A,N3←P(ALFA),N1←SIZ,T2←T2←A,N2←SIZ,TPAR) ;                   07902460 T  0023:1
               BEGIN                                                                 07902465 T  0028:3
               DS←15LIT"-DATA STMT ERR#"; SI←LOC E; DS←DEC ;                         07902470 T  0028:3
               DS←4LIT",LT="; SI←LOC N3; SI←SI+D; DS←CHR ;                           07902475 T  0031:2
               DS←4LIT",DT="; SI←LOC N3; SI←SI+C; DS←CHR ;                           07902480 T  0033:1
               DS←3LIT",L="; SI←LOC B; DS←N1 DEC ;                                   07902485 T  0035:0
               DS←3LIT",D="; DS←N2 DEC; DS←2LIT":←" ;                                07902490 T  0036:2
               END ;                                                                 07902495 T  0038:1
          GO WRAPUP ;                                                                07902500 T  0038:2
          END ;                                                                      07902505 T  0039:0
     IF CODE=(-1) THEN                                                               07902510 T  0039:0
          BEGIN                                                                      07902515 T  0040:0
          STREAM(TPAR); DS←33LIT"-MIXD UNFMT/ALPHA-MODE TAPE I/O:←" ;                 07902520 T  0040:2
          GO WRAPUP ;                                                                07902523 T  0046:1
          END ;                                                                      07902525 T  0046:3
     IF CODE=(-3) THEN                                                               07902530 T  0046:3
          BEGIN T3←4; T2←I; FID←(MFID←P(G095))-1 ;                                   07902535 T  0047:3
          STREAM(J,K,I,F1←SIZ,D←T2←H,F2←SIZ,F3←G>10 AND 15>G,F,                      07902540 T  0051:2
               N3←P(ALFA),C1←IT2←E,C←SIZ,R1←IT2←D+1,R←SIZ,                           07902545 T  0058:0
               BUFF←C,V←B.[42:1] AND B.[46:2]=0,A4←T2←B.[6:12],A5←SIZ                 07902548 T  0062:0
               ×I←T2≠1,A55←J←(K←T2)=MFID,A2←(T5←B AND 15)=12 OR T5=8,                 07902551 T  0066:2
               A3←T5=12 OR T5=4,CD←P(DC),CD1←P(DC1),Z←0,CD2←P(DC2),                   07902554 T  0073:0
               A6←(T4←B.[1:5])=2,A7←T2←B.[18:12],A8←SIZ×((T4≠30 OR                    07902557 T  0075:3
               T2≠FID) AND (T4≠9 OR T2≠0)),A85←MFID=T2,A9←T4≥11 AND                   07902560 T  0080:3
               T4≤14 OR (T4=30 AND (T2←B.[30:12])≠FID),A10←T2,A11←SIZ,                07902563 T  0085:2
               A115←T2=MFID,A12←T2←IF J THEN A.[18:15] ELSE K,                        07902566 T  0091:0
               A13←SIZ×I,TPAR) ;                                                     07902568 T  0094:2
               BEGIN DS←11LIT"-DATA ERR #"; SI←LOC J; SI←SI+7; DS←CHR ;               07902569 T  0097:1
               DS←2LIT"= "; V(NAS(A4,A5,A55)); A2(DS←LIT"K"); A3(DS←LIT               07902572 P  0099:3
               "$"); SI←LOC CD; SI←SI+A6; DS←CHR; NAS(A7,A8,A85) ;                    07902575 T  0105:0
               A9(DS←LIT","; NAS(A10,A11,A115)); DS←4LIT" => "; JUMP OUT              07902578 T  0108:3
               TO L1); DS←7LIT"FMT IS "; L1: SI←LOC A12; DS←A13 DEC ;                 07902581 T  0113:0
               A2(DS←LIT"K"); A3(DS←LIT"$"); SI←LOC K; SI←SI+7; DS←CHR ;              07902584 T  0115:3
               DS←F1 DEC; F3(DS←LIT"."; SI←LOC D; DS←F2 DEC); DS←6LIT                 07902587 T  0119:0
               "  TYP="; SI←LOC N3; SI←SI+F; DS←CHR; DS←6LIT", COL#" ;                07902590 T  0121:3
               DS←C DEC; DS←6LIT", CHR="; SI←BUFF; SI←SI-1; DS←CHR ;                  07902593 T  0124:2
               DS←6LIT", REC#"; SI←LOC R1; DS←R DEC; DS←3LIT" :←" ;                   07902596 T  0126:3
               END OF STREAM ;                                                       07902599 T  0129:1
          GO WRAPUP ;                                                                07902602 T  0129:2
ALFA:::   @2531625143242300 ;                                                        07902605 T  0130:0
DC:::     "PXTAOLJ";  DC1::: @3127262524230000;  DC2::: "(V000" ;                   07902606 T  0131:0
G095:::   4095 ;                                                                     07902607 T  0134:0
          END ;                                                                      07902608 T  0135:0
     IF CODE=(-4) THEN                                                               07902610 T  0135:0
          BEGIN                                                                      07902614 T  0136:0
          IF CODE.[2:1] THEN                                                         07902615 T  0136:2
               BEGIN                                                                 07902616 T  0137:1
               STREAM(TPAR); DS←24LIT"-UNINITIALIZED POINTER:←" ;                    07902617 T  0137:3
               GO WRAPUP ;                                                           07902618 T  0142:1
               END ;                                                                 07902619 T  0142:3
```

```
      STREAM(TPAR); DS←32LIT"=BINARY TAPE REC HAS < 3 WORDS:←" ;                    07902620 T  0142:3
      GO WRAPUP ;                                                                   07902625 T  0148:1
      END ;                                                                         07902630 T  0148:3
IF CODE=(-5) THEN                                                                   07902631 T  0148:3
      BEGIN T3←8;  CODE←H ;                                                         07902632 T  0149:3
      IF D=2 THEN BEGIN CODE←30; IF WH1>63 OR WH1<10 THEN D←12 END ;                07902633 T  0151:3
      IF A.[1:5]<5 THEN BEGIN A.[6:12]←A.[18:12]; R←SAVW; D←10 END ;                07902634 T  0156:3
      IF D>9 THEN                                                                   07902635 T  0162:1
           BEGIN FIB←P([FIB[1]],CFX,SFB)&5[8:38:10]; FIB[0]←0 ;                     07902636 T  0163:0
           BUFF←((BUFF←EDITIT(FID←FIB.[33:15],0,2,WH2,WH1)),[33:15]=                07902637 T  0167:1
                FID)×8+BUFF.[30:3] ;                                                07902638 T  0173:1
           END ;                                                                    07902639 T  0175:3
      FID←(MFID←P(F095))=1 ;                                                        07902640 T  0175:3
      IND←CODE>14 AND (CODE≠30 OR A.[30:12]=FID) ;                                  07902641 T  0177:2
      STREAM(A1←FIB,A2←(T5←A AND 15)=12 OR T5=8,A3←T5=12 OR T5=4,                   07902642 T  0181:1
            A4←T2←A.[6:12],A5←SIZ×T5←T2≠1,A55←T2=MFID,A6←(T4←A.[1:5]                 07902643 T  0186:2
            )-2,A7←T2←A.[18:12],A8←SIZ×  (T4≠29 AND T4≠3 AND T4≠4                   07902645 T  0191:1
            AND (T4≠30 OR T2≠ FID) AND (T4≠9  OR T2≠0)),A85←(T2=                    07902647 T  0197:0
            MFID)   ,A9←T4≥11 AND T4≤14 OR (T4=30 AND (T2←A.[30:12])                07902650 T  0202:1
            ≠ FID),A10←T2,A11←SIZ,A115←T2=MFID,A12←T4=3 AND A.[41:1]                07902655 T  0205:1
            ,R10←D=10,R←D≠10,R1←T2←R,R2←SIZ×T5,                                     07902660 T  0210:2
            V12←D=12,VV←D=2 AND WH1≠31,V←D>2 AND D≠12,                              07902665 T  0215:2
            V1←CODE=2,SKPWD←CODE=3 OR CODE=29 OR (CODE=30 AND                       07902670 T  0219:3
            SAVW= FID) OR CODE=4 OR (CODE=9 AND SAVW=0),W14←D=14,                   07902672 T  0223:0
            W←D≠14,WW←SAVW=FID,W5←SAVW=MFID,SKPD←CODE                               07902675 T  0228:0
            <11 OR IND,D16←D=16,D←D≠16,DD←                                          07902680 T  0230:1
            SAVD=FID,D5←SAVD=MFID,D1←T2←SAVD,D2←SIZ,W1←T2←SAVW,                      07902685 T  0233:0
            W2←SIZ,WH1,A13←BUFF,CD←P(CD),CD1←P(CD1),R5←0,                           07902690 T  0236:3
              CD2←P(CD2),TPAR) ;                                                    07902692 T  0239:1
            BEGIN DS←16LIT"=VARBL FMT ERR= "; A12(DS←LIT"=") ;                      07902695 P  0240:1
            NAS(A4,A5,A55); A2(DS←LIT"K"); A3(DS←LIT"$") ;                          07902700 T  0243:3
            SI←LOC CD; SI←SI+A6; DS←CHR; NAS(A7,A8,A85) ;                           07902705 T  0248:2
            A9(DS←LIT"."; NAS(A10,A11,A115)); DS←4LIT" => " ;                       07902710 T  0251:3
            A12(DS←LIT"="); CD5(R10,R,R1,R2,R5); A2(DS←LIT"K"); A3(DS←              07902720 T  0256:0
            LIT"$"); CC55(V12); V(SI←LOC CD; SI←SI+V1; DS←CHR); VV(DS←              07902725 T  0264:2
            LIT"<"; SI←LOC WH1; SI←SI+7; DS←CHR; DS←LIT">");SKPWD(JUMP              07902730 T  0270:0
            OUT TO XX);WW(DS←11LIT"<MISSING W>"; SKPD(JUMP OUT 2 TO XX              07902735 T  0272:2
            ); DI←DI-1; DS←7LIT" AND D>";JUMP OUT TO XX); CD5(W14,W,W1               07902740 T  0276:2
            ,W2,W5); GO XV; XX: GO XT; XV: SKPD(JUMP OUT 1 TO XT); DS←               07902745 T  0279:1
            LIT","; DD(DS←11LIT"<MISSING D>"; JUMP OUT 1 TO XT) ;                   07902750 T  0286:2
            CD5(D16,D,D1,D2,D5);   XT:  DS←3LIT" ;←" ;                              07902755 T  0290:0
            END OF STREAM ;                                                         07902760 T  0296:1
      GO WRAPUP ;                                                                   07902765 T  0296:2
F095:::  4095 ;                                                                     07902770 T  0297:0
CD:::    "PXTAOLJ";  CD1:::  @3127262524230000;  CD2::: "(V000" ;                   07902771 T  0298:0
      END ;                                                                         07902775 T  0301:0
      IF CODE<10 THEN GO SW1[CODE] ;                                                07902780 T  0301:0
      GO TO SW2[CODE - 10];                                                         07902800 T  0306:3
   L0:  %                                    0                                     07902900 T  0319:3
      STREAM(P1←0:P2 ← [TPAR[0]]);                                                  07903000 T  0319:3
      BEGIN                                                                         07903100 T  0321:1
           DS ← 14 LIT "=FORMAT ERROR ";                                            07903200 T  0321:1
           P1 ← DI;                                                                 07903300 T  0323:1
      END;                                                                          07903400 T  0323:2
      BUFF ← P;                                                                     07903500 T  0323:3
      GO TO LX;                                                                     07903600 T  0324:1
   L1:  %                                    1                                     07903700 T  0324:3
```

```
     STREAM(P1←0:D ← [TPAR[0]]);                                                    07903800 T  0324:3
     BEGIN                                                                          07903900 T  0326:1
          DS ← 16 LIT "-NAMELIST ERROR ";                                           07904000 T  0326:1
          P1 ← DI;                                                                  07904100 T  0328:2
     END;                                                                           07904200 T  0328:3
     BUFF ← P;                                                                      07904300 T  0329:0
     GO TO LX;                                                                      07904400 T  0329:2
L2:  %                                        2                                   07904500 T  0330:0
     STREAM(P1←0:D ← [TPAR[0]]);                                                    07904600 T  0330:0
     BEGIN                                                                          07904700 T  0331:2
          DS ← 12 LIT "-TYPE ERROR ";                                               07904800 T  0331:2
          P1 ← DI;                                                                  07904900 T  0333:1
     END;                                                                           07905000 T  0333:2
     BUFF ← P;                                                                      07905100 T  0333:3
LX:                                                                                 07905200 T  0334:1
     GETFILE;                                                                       07905300 T  0334:1
     STREAM(MFID,FID,B,BUFF);                                                       07905400 T  0335:0
     BEGIN                                                                          07905500 T  0336:2
          DI←BUFF;  DS←8LIT"ON FILE " ;                                             07905600 T  0336:2
          SI ← LOC MFID; SI ← SI + 1; DS ← 7 CHR; DS ← LIT "/";                     07905700 T  0338:0
     SI←LOC FID;  SI←SI+1;  DS←7CHR;  DS←7LIT", REC #" ;                            07905800 T  0339:1
     SI ← LOC B; DS ← 8 DEC; DS ← 2 LIT "←:";                                       07905900 T  0341:1
     END;                                                                           07906000 T  0342:1
     GO TO WRAPUP;                                                                  07906100 T  0342:2
L3:  %                                        3                                   07906200 T  0343:0
     STO;                                                                           07906300 T  0343:0
          DS ← 18 LIT "-DATA STMT ERROR:←";                                         07906500 T  0344:0
     GO TO WRAPUP;                                                                  07906700 T  0346:3
L4:  %                                        4                                   07906710 T  0347:1
     STREAM(P1←0:D←[TPAR[0]]);                                                      07906720 T  0347:1
     BEGIN                                                                          07906730 T  0348:3
          DS←26LIT"-MIXED FMT/UNFMT TAPE I/O " ;                                    07906740 T  0348:3
% VOID                                                                            07906750 T  0352:1
          P1 ← DI;                                                                  07906760 T  0352:1
     END;                                                                           07906770 T  0352:2
     BUFF ← P;                                                                      07906780 T  0352:3
     GO TO LX;                                                                      07906790 T  0353:1
L5:  %                                   5                                       07906800 T  0353:3
     STREAM(P1←0:D←[TPAR[0]]);                                                      07906810 T  0353:3
     BEGIN                                                                          07906820 T  0355:1
          DS:=18 LIT "-LIST SIZE ERROR  ";                                          07906830 P  0355:1
          P1 ← DI;                                                                  07906840 T  0357:3
     END;                                                                           07906850 T  0358:0
     BUFF ← P; GO TO LX;                                                            07906860 T  0358:1
L6:  STO; %                                   6                                   07906861 T  0359:1
     DS ← 21 LIT "-INVALID ARG CONCAT:←";                                           07906862 T  0360:1
     GO TO WRAPUP;                                                                  07906863 T  0363:2
CPLR:STO;  %                             10                                       07906890 T  0364:0
     DS ← 31 LIT "-EXPRESSION COMPILATION ERROR:←";                                 07906900 T  0365:0
     GO TO FIGER;                                                                   07907000 T  0369:2
XTOI:STO;   %                                 11                                  07907100 T  0370:0
     DS ← 21 LIT "-NEGATIVE BASE XTOI:←";                                           07907200 T  0371:0
     GO TO FIGER;                                                                   07907300 T  0374:1
CSSC:STO;   %                                 12                                  07907400 T  0374:3
     DS ← 24 LIT "-COMPLEX EXPONENT XTOI:←";                                        07907500 T  0375:3
     GO TO FIGER;                                                                   07907600 T  0379:1
DMOD:STO;   %                                 13                                  07907700 T  0379:3
```

```
                DS ← 20 LIT "=ZERO MODULUS DMOD:←";
                GO TO FIGER;
DEXP:           ST2;   %                                        14
                DS ← 6 LIT "DEXP:←";
                GO TO MAXN;
CEXP:           ST2;   %                                        15
                DS ← 6 LIT "CEXP:←";
                GO TO MAXN;
DLGZ:           BUFF ← TRUE;   %                                16
DLGM:           ST2;   %                                        17
                DS ← 6 LIT "DLOG:←";
                IF BUFF THEN GO TO ZERO ELSE GO TO NGTV;
CLOG:           ST2;   %                                        18
                DS ← 6 LIT "CLOG:←";
                GO TO ZERO;
ALTZ:           BUFF ← TRUE;   %                                19
ALTM:           ST2;   %                                        20
                DS ← 8 LIT "ALOG10:←";
                IF BUFF THEN GO TO ZERO ELSE GO TO NGTV;
DLTZ:           BUFF ← TRUE;   %                                21
DLTM:           ST2;   %                                        22
                DS ← 8 LIT "DLOG10:←";
                IF BUFF THEN GO TO ZERO ELSE GO TO NGTV;
CSIN:           ST2;   %                                        23
                DS ← 6 LIT "CSIN:←";
                GO TO MAXN;
CCOS:           ST2;   %                                        24
                DS ← 6 LIT "CCOS:←";
                GO TO MAXN;
ACOS:           BUFF ← TRUE;   %                                25
ASIN:           STREAM(B ← BUFF,D ← [TPAR[0]]);  %              26
                BEGIN DS ← 19 LIT "=ABS(ARG) .GT. 1 AR";
                      SI ← LOC B;  SI ← SI + 7;
                      IF SC = "1" THEN DS ← 5 LIT "COS:←"
                                  ELSE DS ← 5 LIT "SIN:←";
                END;
                GO TO FIGER;
DSQR:           ST2;   %                                        27
                DS ← 7 LIT "DSQRT:←";
NGTV:           STO;
                DS ← 16 LIT "=NEGATVE ARGMNT ";
                GO TO FIGER;
GAMA:           ST2;   %                                        28
                DS ← 7 LIT "GAMMA:←";
                GO TO MAXN;
SINH:           ST2;   %                                        29
                DS ← 6 LIT "SINH:←";
                GO TO MAXN;
COSH:           ST2;   %                                        30
                DS ← 6 LIT "COSH:←";
MAXN:           STO;
                DS ← 16 LIT "=ARGMT .GT. MAX ";
                GO TO FIGER;
ALGZ:           BUFF ← TRUE;   %                                31
ALGM:           ST2;   %                                        32
                DS ← 8 LIT "ALGAMA:←";
                IF NOT BUFF THEN GO TO NGTV;
```

```
07907800 T  0380:3
07907900 T  0383:3
07908000 T  0384:1
07908100 P  0385:1
07908200 T  0386:2
07908300 T  0387:0
07908400 P  0388:0
07908500 T  0389:1
07908600 T  0389:3
07908700 T  0390:2
07908800 T  0391:2
07908900 T  0392:3
07909000 T  0394:0
07909100 T  0395:0
07909200 T  0396:1
07909300 T  0396:3
07909400 T  0397:2
07909500 T  0398:2
07909600 T  0400:0
07909700 T  0401:1
07909800 T  0402:0
07909900 T  0403:0
07910000 T  0404:2
07910100 T  0405:3
07910200 T  0406:3
07910300 T  0408:0
07910400 T  0408:2
07910500 T  0409:2
07910600 T  0410:3
07910700 T  0411:1
07910800 T  0412:0
07910900 T  0413:1
07911000 T  0416:0
07911100 T  0416:2
07911200 T  0418:0
07911400 T  0419:1
07911500 T  0419:2
07911600 T  0420:0
07911700 T  0421:0
07911800 T  0422:2
07911900 T  0423:2
07912000 T  0426:0
07912100 T  0426:2
07912200 T  0427:2
07912300 T  0429:0
07912400 T  0429:2
07912500 T  0430:2
07912600 T  0431:3
07912700 T  0432:1
07912800 T  0433:1
07912900 T  0434:2
07913000 T  0435:2
07913100 T  0438:0
07913200 T  0438:2
07913300 T  0439:1
07913400 T  0440:1
07913500 T  0441:3
```

```
ZERO:     STO;
          DS ← 16 LIT "=ZERO ARGUMENT  ";                    07913600 T  0442:2
                                                             07913700 T  0443:2
     WRAPUP: FIGER:                                          07913800 T  0446:0
          P([TPAR[0]].[33:15],34,COM);                       07913900 T  0446:0
     END FORTERR;                                            07914000 T  0447:2
                                                                    SIZE= 0448 WORDS


PROCEDURE MAX;                                        % 135          08000000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00553
COMMENT MAX INTRINSIC RETURNING INTEGERS;   %  PF JULY 67  08000100 T  0000:0
BEGIN REAL X = -1, RCW = +0, SIZE = +1, JUNK = +2;         08000200 T  0000:0
P(0,RCW,FCX,[RCW] INX NOT 0 INX 0,XCH,SUB,0,X);            08000300 T  0000:0
WHILE (SIZE ← SIZE - 1) > 0 DO                             08000400 T  0003:1
BEGIN P(DUP);                                              08000500 T  0005:2
      JUNK ← *(P(,X) + SIZE);                              08000600 T  0005:3
      IF P < JUNK THEN P(DEL,DUP);                         08000700 T  0007:1
END;                                                       08000800 T  0008:3
P(1,DIV,RTN);                                              08000900 T  0009:1
END IMAX;                                                  08001000 T  0010:0
                                                                    SIZE= 0011 WORDS


PROCEDURE MIN;                                        % 136          08100000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00554
COMMENT MIN INTRINSIC RETURNING INTEGERS;   %  PF JULY 67  08100100 T  0000:0
BEGIN REAL X = -1, RCW = +0, SIZE = +1, JUNK = +2;         08100200 T  0000:0
P(0,RCW,FCX,[RCW] INX NOT 0 INX 0,XCH,SUB,0,X);            08100300 T  0000:0
WHILE (SIZE ← SIZE - 1) > 0 DO                             08100400 T  0003:1
BEGIN P(DUP);                                              08100500 T  0005:2
      JUNK ← *(P(,X) + SIZE);                              08100600 T  0005:3
      IF P > JUNK THEN P(DEL,DUP);                         08100700 T  0007:1
END;                                                       08100800 T  0008:3
P(1,DIV,RTN);                                              08100900 T  0009:1
END IMIN;                                                  08101000 T  0010:0
                                                                    SIZE= 0011 WORDS


PROCEDURE IMOD;                                       % 137          08200000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00555
COMMENT INTEGER MOD INTRINSIC;   %  PF  JULY 67          08200100 T  0000:0
BEGIN INTEGER X = -2,                                      08200200 T  0000:0
              Y = -1;                                      08200300 T  0000:0
P(X MOD Y,1,DIV,RTN);                                      08200400 T  0000:0
END IMOD;                                                  08200500 T  0001:2
                                                                    SIZE= 0002 WORDS


PROCEDURE CONCAT ;              % INTRINSIC NUMBER @140,    08300000 T  0000:0
```

```
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00556
BEGIN % FORTRAN CONCATENATE INTRINSIC: CONCAT=Y&Z[A:B:X].                                08300100 T  0000:0
REAL Y=-5, Z=-4, ERR=24 ;                                                                08300200 T  0000:0
INTEGER A=-3, B=-2, X=-1 ;                                                               08300250 T  0000:0
DEFINE R= @0055005570267022 #, % NOP,DIA,OPDC Y,OPDC,Z.                                  08300260 T  0000:0
       S= @0055006100650235 #; % NOP,DIB,TRB,RTN.                                        08300270 T  0000:0
IF (A←A)<1 OR (B←B)<1 OR (X←X)<1 OR (P(48-X,DUP)<A OR P(XCH)<B)                          08300300 T  0000:0
   THEN P(MKS,6,ERR) ;                                                                   08300400 T  0006:0
GO P(P(R)&(B DIV 6)[12:45:3]&(B MOD 6)[15:9:3],P(S),A MOD 6,TRB 3,                       08300500 T  0008:1
    P&(A DIV 6)[12:45:3]&X[24:42:6],,B,←,,A,←,[A]) ;                                     08300600 T  0012:3
END OF CONCAT ;                                                                          08300700 T  0016:3
                                                                                                  SIZE= 0019 WORDS


PROCEDURE FORTRANMEMHANDLER(A,H); VALUE H; REAL H; ARRAY A[*]; %@164                     08301000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00557
    BEGIN % H=0 => VARYING,  H=6 => FIXED,  H=-1 => RELEASE.                             08301100 T  0000:0
    REAL I ;                                                                             08301200 T  0000:0
    P(*A,TOP) ;                                                                          08301300 T  0000:0
    IF H≥0 THEN                                                                          08301400 T  0001:1
         IF P THEN P(A&H[3:45:3],(*2)&(A)[33:18:15],←)                                   08301500 T  0002:0
         ELSE FOR I←A,[8:10]-1 STEP -1 UNTIL 0 DO P([A[I]],DUP,LOD,                      08301600 T  0006:0
                                                   P&H[3:45:3],XCH,←)                    08301700 T  0012:2
    ELSE IF P THEN P(A,38,COM,DEL)                                                       08301800 T  0014:0
         ELSE FOR I←A,[8:10]-1 STEP -1 UNTIL 0 DO P(*[A[I]],38,COM,DEL);                 08301900 T  0016:3
    END OF FORTRANMEMHANDLER ;                                                           08301950 T  0023:3
                                                                                                  SIZE= 0025 WORDS


PROCEDURE SISO;              % 35                                                       08400000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00558
    BEGIN                                                                                08400200 T  0000:0
    COMMENT STRING ISOLATE. INVOKED AS REAL(PTR,N). N COUNTS CHARS.;                     08400400 T  0000:0
    DEFINE CSIZE=[31:02]#,     COMMENT CHAR-SIZE FIELD OF PTR;                           08400600 T  0000:0
           EIGHT=01#;          COMMENT VALUE OF CSIZE FOR 8 BITS-CHAR;                   08400800 T  0000:0
    INTEGER                                                                              08401000 T  0000:0
       SOFF;                   %BIT OFFSET TO BIT 1 OF S FOR 8-BIT CHAR                  08401200 T  0000:0
    INTEGER                                                                              08401400 T  0000:0
       PTR       =-3,                                                                    08401600 T  0000:0
       RCW       =-2,          %SISO IS REALLY A REAL PROC:VALUE IN PTR                  08401800 T  0000:0
       N         =-1;                                                                    08402000 T  0000:0
    REAL  RESULT   =PTR;                                                                 08402010 T  0000:0
    ARRAY                                                                                08402200 T  0000:0
       STRING[*];              %UNINDEXED DD FOR SOURCE CHARS                            08402400 T  0000:0
    NAME                                                                                 08402600 T  0000:0
       M=2;                                                                              08402800 T  0000:0
    IF PTR=0 THEN P(MKS,INTCALL((-4)&1[2:47:1],FORTERRI)) ;                              08402900 T  0000:0
    IF PTR.[01:01] THEN                                                                  08402925 T  0005:0
       P(M&1[14:47:01],PTR.[09:22]+(PTR.[33:15]≠0),CHS,CDC,DEL);                         08402950 T  0005:3
    STRING←M[PTR];                                                                       08403000 T  0010:2
    N←ABS(N);                                                                            08403100 T  0012:0
    IF PTR.CSIZE=EIGHT THEN                                                              08403200 T  0013:0
       BEGIN                                                                             08403400 T  0014:1
```

```
      IF N>6 THEN POLISH((STRING)&6[08:38:10],N,CDC,DEL);                    08403600 T  0014:3
      SOFF←0&PTR[32:18:13];   N←0&N[09:12:36];  COMMENT BIT INDICES;         08403800 T  0018:1
      POLISH([STRING[(SOFF+N-8) DIV 48]],DEL);                               08404000 T  0021:3
      STREAM(RESULT←0:S←[STRING[SOFF DIV 48]],SKS←(SOFF←SOFF MOD 48),        08404200 T  0024:0
         SKD←48-N,N);                                                        08404400 T  0027:0
         BEGIN                                                               08404600 T  0028:1
         SI←S;                  SKIP SKS SB;                                08404800 T  0028:1
         DI←LOC RESULT;         SKIP SKD DB;                                08405000 T  0029:0
         N(IF SB THEN DS←1 SET ELSE DS←1 RESET; SKIP 1 SB);                  08405200 T  0029:3
         END;                                                                08405400 T  0032:0
         RESULT := P(DUP);              % SAVE IT                           08405600 T  0032:1
      END ELSE                                                               08405800 T  0033:0
      BEGIN                                                                  08406000 T  0033:0
      COMMENT SOURCE HAS 6 BITS/CHAR;                                        08406200 T  0033:2
      IF N>8 THEN POLISH((STRING)&8[08:38:10],N,CDC,DEL);                    08406400 T  0033:2
      POLISH([STRING[(PTR.[18:13]+N-1).[35:10]]],DEL);                       08406600 T  0037:0
      STREAM(RESULT←0:S←[STRING[PTR.[18:10]]],SKS←PTR.[28:03],               08406800 T  0039:3
         N,SKD←8-N);                                                         08407000 T  0042:1
         BEGIN                                                               08407200 T  0043:2
         SI←S;                  SI←SI+SKS;                                  08407400 T  0043:2
         DI←LOC RESULT;         DI←DI+SKD;                                  08407600 T  0044:1
         DS←N CHR;                                                          08407800 T  0045:0
         END;                                                               08408000 T  0045:2
         RESULT := P(DUP);              % SAVE IT                           08408200 T  0045:3
      END;                                                                   08408400 T  0046:2
         IF NOT (P(TOP)) THEN           % IT IS BAD                         08408500 T  0046:2
                        P([RCW]&1[8:38:10],0,COC);% FLAG IT                 08408510 T  0047:0
END SISO;                                                                    08408600 T  0049:1
                                                                    SIZE= 0050 WORDS
```

```
PROCEDURE SCAN(UPDPDD,PTR,UPDCDD,HISCOUNT,CASECODE,CHAR);                    08410000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00560
   VALUE PTR, HISCOUNT, CASECODE, CHAR;                                      08410050 T  0000:0
   NAME UPDPDD, UPDCDD;                                                      08410100 T  0000:0
   INTEGER PTR, HISCOUNT, CASECODE, CHAR;                                    08410150 T  0000:0
                                                                             08410200 T  0000:0
   BEGIN                                                                     08410250 T  0000:0
   COMMENT        RELATION    WHILE          UNTIL                          08410300 T  0000:0
                  ≤           0              20                             08410350 T  0000:0
                  ≥           4              16                             08410400 T  0000:0
                  ≠           8              12                             08410450 T  0000:0
                  =           12             8                              08410500 T  0000:0
                  <           16             4                              08410550 T  0000:0
                  >           20             0                              08410600 T  0000:0
                  IN ALPHA    24             29                             08410650 T  0000:0
                  IN NUMERIC  25             30                             08410660 C  0000:0
                  IN TRUTHSET 26             31                             08410670 C  0000:0
            NOTE: THE TRUTH SET IS THE 64 BITS BEGINNING AT BIT 1            08410680 C  0000:0
                  OF THE WORD POINTED TO BY THE DESCRIPTOR IN CHAR,          08410690 C  0000:0
                  ;                                                          08410700 T  0000:0
                                                                             08410750 T  0000:0
   NAME     M=2;                                                             08410800 T  0000:0
   ARRAY STRINGDESC[*];                                                      08410850 T  0000:0
   INTEGER OURCOUNT, WOFSET, CHOFSET, N, N1, JUNK=17;                        08410900 T  0000:0
```

```
INTEGER AWHILE; % "IN" SCAN IS FOR WHILE.                                        08410925 C  0000:0
BOOLEAN MORE;                                                                    08410950 T  0000:0
DEFINE   PW=[18:10]#,      PC=[28:03]#,   CSIZE=[31:02]#,                        08411000 T  0000:0
         SIX=00#,          ALONE=@777777#,POFSET=[18:13]#;                       08411050 T  0000:0
                                                                                 08411100 T  0000:0
SUBROUTINE CHARSCAN;                                                             08411150 T  0000:0
   BEGIN;                                                                        08411200 T  0001:0
   COMMENT SCAN FOR CONDITIONS OTHER THAN ALPHA MEMBERSHIP.;                     08411250 T  0001:0
   STREAM(N, CHOFSET, CHAR  : DD1←[STRINGDESC[0]], CASECODE,                     08411300 T  0001:0
      DD←[STRINGDESC[WOFSET]]);                                                  08411350 T  0003:0
      BEGIN                                                                      08411400 T  0003:3
      SI←DD;               SI←SI+CHOFSET;                                        08411450 T  0003:3
      DI←LOC CHAR;         DI←DI+6;                                              08411500 T  0004:2
      CI←CI+CASECODE;                                                            08411550 T  0005:0
         GO TO LE;         %00                                                   08411600 T  0005:2
         GO TO GE;         %01                                                   08411650 T  0005:3
         GO TO NE;         %02                                                   08411700 T  0006:0
         GO TO EQ;         %03                                                   08411750 T  0006:1
         GO TO LS;         %04                                                   08411800 T  0006:2
         %GO TO GR;        %05                                                   08411850 T  0006:3
      GR:N(IF SC≤DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08411900 T  0006:3
      LS:N(IF SC≥DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08411950 T  0009:0
      EQ:N(IF SC≠DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08412000 T  0011:1
      NE:N(IF SC=DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08412050 T  0013:2
      GE:N(IF SC<DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08412100 T  0015:3
      LE:N(IF SC>DC THEN JUMP OUT TO XX; DI←DI-1); GO TO XY;                     08412150 T  0018:0
      XY:TALLY←1;          SI←SI+1;                                              08412200 T  0020:1
      XX:N←TALLY;          SI←SI-1;       CHAR←SI;                               08412250 T  0020:3
         SI←DD1;                 CHOFSET←SI;                                     08412300 T  0021:2
         END;                                                                    08412350 T  0022:0
      CHOFSET←POLISH(SUB,DUP).[18:15];                                           08412400 T  0022:1
      %WE ONLY NEED [30:03], BUT REST OF FIELD IS 0 AND ESPOL KNOWS TO           08412450 T  0024:0
      %OPTIMIZE [18:15] TO AN "FTC" OPERATOR.                                    08412500 T  0024:0
      WOFSET←POLISH.[33:15];                                                     08412550 T  0024:0
      MORE←POLISH;                                                               08412600 T  0025:0
      END CHARSCAN;                                                              08412650 T  0025:2
                                                                                 08412700 T  0025:3
   SUBROUTINE ALFSCAN;                                                           08412750 T  0025:3
      BEGIN;                                                                     08412800 T  0026:0
      COMMENT SCAN . . . WHILE/UNTIL IN ALPHA, NUMERIC, TRUTHSETID;              08412850 P  0026:0
STREAM (CHOFSET, SWITCHER:=CASECODE, N : TSET:=[CHAR], AWHILE,                   08412900 P  0026:0
       DD1:=[STRINGDESC[0]], DD:=[STRINGDESC[WOFSET]]);                          08412950 P  0027:3
BEGIN                                                                            08413000 P  0029:0
   SI:=DD; SI:=SI+CHOFSET;                                                       08413010 C  0029:0
   CI:=CI+AWHILE; GO UCASE; % GO WCASE;                                          08413020 C  0029:3
WCASE:                                                                           08413030 C  0030:2
   CI:=CI+SWITCHER; GO ATESTW; GO NTESTW; % GO TTESTW;                           08413040 C  0030:2
TTESTW: DI:=LOC SWITCHER;                                                        08413050 P  0031:2
   N ( DD:=SI; DI:=DI-1; DS:=CHR;                                                08413060 C  0031:3
       SI:=TSET; SKIP CHOFSET SB; SKIP SB;                                       08413070 C  0033:0
       IF SB THEN; SI:=DD;                                                       08413080 C  0034:0
       IF TOGGLE THEN SI:=SI+1 ELSE JUMP OUT TO TSTOPW;                          08413090 C  0034:3
     );                                                                          08413100 P  0036:0
   GO AWAYW;                                                                     08413110 C  0036:1
NTESTW:                                                                          08413120 C  0036:2
   N ( IF SC GEQ "0" THEN IF SC LEQ "9" THEN;                                    08413130 C  0036:2
       IF TOGGLE THEN SI:=SI+1 ELSE JUMP OUT TO TSTOPW;                          08413140 C  0038:0
```

```
      );
    GO AWAYW;
ATESTW:
    N ( IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT TO TSTOPW );
AWAYW:
    TALLY:=1;
TSTOPW: GO TO DONE;
UCASE:
    CI:=CI+SWITCHER; GO ATESTU; GO NTESTU; % GO TTESTU;
TTESTU: DI:=LOC SWITCHER;
    N ( DD:=SI; DI:=DI-1; DS:=CHR;
        SI:=TSET; SKIP CHOFSET SB; SKIP SB;
        IF SB THEN; SI:=DD;
        IF TOGGLE THEN JUMP OUT TO TSTOPU; SI:=SI+1;
      );
    GO AWAYU;
NTESTU:
    N ( IF SC GEQ "0" THEN IF SC LEQ "9" THEN JUMP OUT TO TSTOPU;
        SI:=SI+1;
      );
    GO AWAYU;
ATESTU:
    N ( IF SC=ALPHA THEN JUMP OUT TO TSTOPU; SI:=SI+1 );
AWAYU:
    TALLY:=1;
TSTOPU: DONE:
    N:=TALLY; SWITCHER:=SI; SI:=DD1; CHOFSET:=SI;
          END;
    MORE←POLISH;
    CHOFSET←POLISH(SUB,DUP).[18:15];%OPTIMIZED [30:03] ISOLATE.
    WOFSET←POLISH.[33:15];
    END ALFSCAN;

IF PTR=0 THEN P(MKS,INTCALL((-4)&1[2:47:1],FORTERRI)) ;
IF PTR.[01:01] THEN
    P(M&1[14:47:01],PTR.[09:22]+(PTR.[33:15]≠0),CHS,CDC,DEL);
IF PTR.CSIZE≠SIX THEN POLISH(M&1[14:47:01],8686,CDC,DEL);
STRINGDESC←M[PTR];
IF (OURCOUNT←0&(STRINGDESC)[35:08:10]-PTR.POFSET) <0 THEN
    POLISH([STRINGDESC[PTR.POFSET]]);
IF HISCOUNT ≤ 0 THEN
  BEGIN UPDCDD[0]←0;UPDPDD[0]←PTR+0&WOFSET[18:35:13];P(XIT);END;
IF (HISCOUNT←(JUNK←HISCOUNT).[33:15])<OURCOUNT THEN
    OURCOUNT←HISCOUNT;
WOFSET←PTR.PW;              CHOFSET←PTR.PC;
N←N1←OURCOUNT;              MORE←TRUE;
    IF CASECODE GEQ 24 THEN % IN ALPHA, NUMERIC, TRUTHSET
    BEGIN
    IF AWHILE:=(CASECODE LEQ 26) THEN % CONDITION IS "WHILE IN"
    ELSE CASECODE:=CASECODE-1;        % CONDITION IS "UNTIL IN"
    CASECODE:=CASECODE.[46:2]; % 0,1,2
    IF N1>63 THEN
       BEGIN
       N←63;
       DO ALFSCAN UNTIL (N1←N1-63)≤63 OR NOT MORE;
       N←N1;
       END;
```

```
08413150 P  0039:1
08413160 C  0039:2
08413170 C  0039:3
08413180 C  0039:3
08413190 C  0042:0
08413200 P  0042:0
08413210 C  0042:1
08413220 C  0042:2
08413230 C  0042:2
08413240 C  0043:2
08413250 P  0043:3
08413260 C  0045:0
08413270 C  0046:0
08413280 C  0046:3
08413290 C  0047:3
08413300 P  0048:0
08413310 C  0048:1
08413320 C  0048:1
08413330 C  0050:1
08413340 C  0050:2
08413350 P  0050:3
08413360 C  0051:0
08413370 C  0051:0
08413380 C  0053:0
08413390 C  0053:0
08413400 P  0053:1
08413410 C  0053:1
08413500 T  0054:1
08413550 T  0054:2
08413600 T  0055:0
08413650 T  0056:3
08413700 T  0057:3
08413750 T  0058:0
08413755 T  0058:0
08413760 T  0064:2
08413770 T  0065:1
08413800 T  0070:0
08413850 T  0073:3
08413900 T  0075:1
08413950 T  0078:3
08413960 T  0080:1
08413970 T  0081:0
08414000 T  0084:3
08414050 T  0087:0
08414100 T  0088:1
08414200 T  0090:3
08414250 P  0092:3
08414300 T  0093:2
08414350 P  0094:0
08414360 C  0095:1
08414370 C  0099:1
08414400 T  0100:2
08414450 T  0101:1
08414500 T  0101:3
08414550 T  0102:2
08414600 T  0107:0
08414650 T  0107:3
```

```
      IF N>0 AND MORE THEN ALFSCAN;                                                    08414700 T  0107:3
      END ELSE                                                                         08414750 T  0111:0
      BEGIN                                                                            08414800 T  0111:0
      CASECODE:=CASECODE.[43:3]; CHAR:=0&CHAR[36:42:6];                                08414825 C  0111:2
      IF N1>63 THEN                                                                    08414850 T  0114:2
         BEGIN                                                                         08414900 T  0115:1
         N←63;                                                                         08414950 T  0115:3
         DO CHARSCAN UNTIL (N1←N1-63)≤63 OR NOT MORE;                                  08415000 T  0116:2
         N←N1;                                                                         08415050 T  0121:0
         END;                                                                          08415100 T  0121:3
      IF N>0 AND MORE THEN CHARSCAN;                                                   08415150 T  0121:3
      END;                                                                             08415200 T  0125:0
   IF HISCOUNT>OURCOUNT AND MORE THEN                                                  08415250 T  0125:0
      POLISH([STRINGDESC[CHOFSET&WOFSET[30:33:15]]]);                                  08415300 T  0126:1
   IF POLISH(.UPDPDD,LOD,RFB,.UPDCDD,LOD,RFB,OR)≠0 THEN                                08415350 T  0128:1
      BEGIN                                                                            08415400 T  0130:2
      WOFSET←CHOFSET&WOFSET[30:33:15]-PTR.[18:13];                                     08415450 T  0131:0
      UPDCDD[0] ←HISCOUNT-WOFSET;                                                      08415500 T  0133:3
      UPDPDD[0]←PTR+0&WOFSET[18:35:13];                                                08415550 T  0135:0
      END;                                                                             08415600 T  0137:1
   END SCAN;                                                                           08415650 T  0137:1
                                                                                                SIZE= 0138 WORDS
```

```
PROCEDURE REPL;                                                                    08420000 T  0000:0
                                                                    START OF REL SEGMENT; DISK ADDRESS = 00565
BEGIN                                                                              08420020 T  0000:0
COMMENT STRING REPLACE INTRINSIC FOR B5500 TS ALGOL                                08420040 T  0000:0
MARCH 1968, RATCHFORD                                                              08420060 T  0000:0
8-BIT CHARS, WORD XFERS & UNCONDITIONAL XFER ADDED APRIL 1968 HJR;                  08420080 T  0000:0
DEFINE                                                                             08420100 T  0000:0
   CSIZE=[31:02]#, EIGHT=01#, TCOND=45#, DOT=[18:13]#,                             08420120 T  0000:0
   DECNVRT=(-32)#,                                                                 08420130 T  0000:0
   AMPER=[18:35:13]#,                                                              08420140 T  0000:0
   POTZ=IF SB THEN DS←1 SET ELSE DS←1 RESET; SKIP 1 SB;#;                          08420160 T  0000:0
ARRAY                                                                              08420180 T  0000:0
   SORC[*]      ,  COMMENT DESC FOR SOURCE STRING;                                 08420200 T  0000:0
   DEST[*]      ;  COMMENT DATA DESC FOR DESTINATION STRING;                       08420220 T  0000:0
NAME                                                                               08420240 T  0000:0
   UPDPDD       =-08, COMMENT DESC FOR UPDATE DEST POINTER;                        08420260 T  0000:0
   UPSPDD       =-06, COMMENT DESC FOR UPDATE SOURCE POINTER;                      08420280 T  0000:0
   UPCTDD       =-04, COMMENT DESC FOR UPDATE COUNT VARIABLE;                      08420300 T  0000:0
   M            = 02;                                                             08420320 T  0000:0
INTEGER                                                                            08420340 T  0000:0
   DPTR         =-07, COMMENT DESTINATION POINTER;                                 08420360 T  0000:0
   SPTR         =-05, COMMENT SOURCE POINTER OR 1 TO 8 LITERAL CHRS,               08420380 T  0000:0
                      8 ONLY IF LITERAL IS ARITHMETIC;                             08420400 T  0000:0
   HISCNT       =-03, COMMENT CALLER"S IDEA OF HOW BIG MAXCOUNT IS;                08420420 T  0000:0
   RELATION     =-02, COMMENT SWITCH INDEX FOR SCAN CODE.  THE INDEX               08420440 T  0000:0
                      VALUES ARE SUPPOSED TO BE THE SAME FOR REPL                  08420460 T  0000:0
                      AND SCAN.  RELATION IS <0 IF THE SOURCE IS                   08420480 T  0000:0
                      A LITERAL AND IS ≥0 IF SOURCE IS A POINTER;                  08420500 T  0000:0
                      COMMENT COMPARE USES THE SAME VALUES OF RELAT;               08420520 T  0000:0
   CHAR         =-01, COMMENT THE WHILE/UNTIL COMPARISON CHAR;                     08420540 T  0000:0
   CHAR1            , COMMENT  SDA-FORMAT ADDR OF 1ST XFERRED CHAR;               08420560 T  0000:0
```

```
   CHARN          , COMMENT SDA FORMAT ADDR OF LAST XFERRED CHAR;
   SORCL       =CHAR1,COMMENT LENGTH OF SOURCE CALCULATED BY US;
   DESTL       =CHARN,COMMENT REMAINING CHARS IN DEST STRING;
   SWI=CHAR1, DWI=CHARN, ITERC,
   OURCNT           , COMMENT SAFE MAX LENGTH FOR REPLACE,
                      FOR POINTER-SOURCE, MIN(HISCNT,SORCL,DESTL),
                      FOR LITERAL SOURCE, MIN(DESTL,HISCNT);
   SOFF             , COMMENT CHARACTER OFFSET IN SOURCE;
   SSIZE       =SOFF,COMMENT SOURCE CHAR SIZE (6 OR 8 BITS/CHAR);
   DOFF             , COMMENT CHARACTER OFFSET IN DESTINATION;
   DSIZE       =DOFF,COMMENT DEST CHAR SIZE;
   UPDTOG           , COMMENT TRUE IF ANY UPDATE(S) REQUESTED;
   JUNK        =17,   COMMENT USED FOR BUILDING CONCATENATED LITRL;
   TOGL             , COMMENT "TOGGLE" FOR REPLACE WHILE/UNTIL;
   REFETCH     =TOGL,COMMENT THE "INVALIDATOR" FOR REPL UNTIL;
   INITIAL     =TOGL;COMMENT LOCAL 4 USE BY REPL FROM LITERAL;
INTEGER AWHILE; % CONDITION IS "WHILE IN"
BOOLEAN MORE;                %CONDITIONAL REPLACE ISN"T DONE YET.
ARRAY NAME
   SORCI       =SORC,COMMENT INDEXED DESC FOR POINTER-SOURCE;
   DESTI       =DEST;COMMENT INDEXED DATA DESC FOR DEST STRING;
SUBROUTINE CREPL;
   BEGIN;
   STREAM(DOFF,CHAR,SOFF,ITERC,MORE←0;
      D1←(DEST[0]],S1←(SORC[0]],RELATION,T←0,S2←(SORC[SWI]],
      D2←(DEST[DWI]]);
      BEGIN
      DI←DI+DOFF;          D2←DI;          DI←LOC CHAR;
      DI←DI+6;             T←DI;           SI←S2;
      SI←SI+SOFF;
      ITERC(CI←CI+RELATION;
         GO TO LE;
         GO TO GE;
         GO TO NE;
         GO TO EQ;
         GO TO LS;
         %GO TO GR;
      GR:IF SC≤DC THEN;    GO TO XX;
      LS:IF SC≥DC THEN;    GO TO XX;
      EQ:IF SC≠DC THEN;    GO TO XX;
      NE:IF SC=DC THEN;    GO TO XX;
      GE:IF SC<DC THEN;    GO TO XX;
      LE:IF SC>DC THEN;   %GO TO XX;
      XX:IF TOGGLE THEN JUMP OUT TO XY;
      SI←SI-1;             DI←D2;          DS←CHR;
      D2←DI;               DI←T;
      );
      TALLY←1;             SI←SI+1;
      XY:MORE←TALLY;       SI←SI-1;        CHAR←SI;
      SI←S1;               DOFF←SI;        DI←D2;
      ITERC←DI;            DI←D1;          SOFF←DI;
      END;
   MORE←POLISH;
   DOFF←POLISH(SUB,DUP).[18:15];   %OPTIMIZED [30:03] ISOLATE.
   DWI←POLISH.[33:15];
   SOFF←POLISH(SUB,DUP).[18:15];
   SWI←POLISH.[33:15];
```

```
08420580 T  0000:0
08420600 T  0000:0
08420620 T  0000:0
08420630 T  0000:0
08420640 T  0000:0
08420660 T  0000:0
08420680 T  0000:0
08420700 T  0000:0
08420720 T  0000:0
08420740 T  0000:0
08420760 T  0000:0
08420780 T  0000:0
08420800 T  0000:0
08420820 T  0000:0
08420840 T  0000:0
08420860 T  0000:0
08420865 C  0000:0
08420870 T  0000:0
08420880 T  0000:0
08420900 T  0000:0
08420920 T  0000:0
08420921 T  0000:0
08420922 T  0001:0
08420923 T  0001:0
08420924 T  0002:2
08420925 T  0004:3
08420926 T  0005:2
08420927 T  0005:2
08420928 T  0006:2
08420929 T  0007:1
08420930 T  0007:3
08420931 T  0008:3
08420932 T  0009:0
08420933 T  0009:1
08420934 T  0009:2
08420935 T  0009:3
08420936 T  0010:0
08420937 T  0010:0
08420938 T  0010:3
08420939 T  0011:2
08420940 T  0012:1
08420941 T  0013:0
08420942 T  0013:3
08420943 T  0014:1
08420944 T  0015:0
08420945 T  0015:3
08420946 T  0016:1
08420947 T  0016:2
08420948 T  0017:0
08420949 T  0017:3
08420950 T  0018:2
08420951 T  0019:1
08420952 T  0019:2
08420953 T  0020:0
08420954 T  0021:3
08420955 T  0022:3
08420956 T  0024:2
```

```
      END CONDITIONAL REPLACE;                                                    08420957 T  0025:2
    SUBROUTINE CRA;                                                               08420958 T  0025:3
      BEGIN; COMMENT REPLACE , WHILE/UNTIL IN ALPHA, NUMERIC, TRUTHSET;           08420959 P  0026:0
STREAM(DOFF, T1:=0, SOFF, N:=ITERC, MORE:=0 : TSET:=[CHAR], AWHILE,               08420960 P  0026:0
      D1:=[DEST[0]], S1:=[SORC[0]], RELATION,                                     08420961 P  0028:1
      S2:=[SORC[SWI]], D2:=[DEST[DWI]]);                                          08420962 P  0029:2
BEGIN                                                                             08420963 P  0030:3
    DI:=DI+DOFF; SI:=S2; SI:=SI+SOFF;                                             08420964 P  0030:3
    CI:=CI+AWHILE; GO UCASE; % GO WCASE;                                          08420965 P  0032:0
WCASE:                                                                            08420966 P  0032:3
    CI:=CI+RELATION; GO ATESTW; GO NTESTW; % GO TTESTW;                           08420967 P  0032:3
TTESTW: D2:=DI; DI:=LOC T1; DI:=DI-1; T1:=DI; DI:=D2;                             08420968 P  0033:3
    N ( D2:=DI; S2:=SI; DI:=T1; DS:=CHR;                                          08420969 P  0035:0
        SI:=TSET; SKIP DOFF SB; SKIP SB;                                          08420970 P  0036:2
        IF SB THEN; SI:=S2; DI:=D2;                                               08420971 P  0037:2
        IF TOGGLE THEN DS:=CHR ELSE JUMP OUT TO TSTOPW;                           08420972 P  0038:2
      );                                                                          08420973 P  0039:3
    GO AWAYW;                                                                     08420974 P  0040:0
NTESTW:                                                                           08420975 P  0040:1
    N ( IF SC GEQ "0" THEN IF SC LEQ "9" THEN;                                    08420976 P  0040:1
        IF TOGGLE THEN DS:=CHR ELSE JUMP OUT TO TSTOPW;                           08420977 P  0041:3
      );                                                                          08420978 P  0043:0
    GO AWAYW;                                                                     08420979 P  0043:1
ATESTW:                                                                           08420980 P  0043:2
    N ( IF SC=ALPHA THEN DS:=CHR ELSE JUMP OUT TO TSTOPW );                       08420981 P  0043:2
AWAYW:                                                                            08420982 P  0045:3
    TALLY:=1;                                                                     08420983 P  0045:3
TSTOPW: GO TO DONE;                                                               08420984 P  0046:0
UCASE:                                                                            08420985 P  0046:1
    CI:=CI+RELATION; GO ATESTU; GO NTESTU; % GO TTESTU;                           08420986 C  0046:1
TTESTU: D2:=DI; DI:=LOC T1; DI:=DI-1; T1:=DI; DI:=D2;                             08420987 C  0047:1
    N ( D2:=DI; S2:=SI; DI:=T1; DS:=CHR;                                          08420988 C  0048:2
        SI:=TSET; SKIP DOFF SB; SKIP SB;                                          08420989 C  0050:0
        IF SB THEN; SI:=S2; DI:=D2;;                                              08420990 P  0051:0
        IF TOGGLE THEN JUMP OUT TO TSTOPU; DS:=CHR;                               08420991 C  0052:0
      );                                                                          08420992 C  0053:0
    GO AWAYU;                                                                     08420993 C  0053:1
NTESTU:                                                                           08420994 C  0053:2
    N ( IF SC GEQ "0" THEN IF SC LEQ "9" THEN JUMP OUT TO TSTOPU;                 08420995 P  0053:2
        DS:=CHR;                                                                  08420996 C  0055:2
      );                                                                          08420997 C  0055:3
    GO AWAYU;                                                                     08420998 C  0056:0
ATESTU:                                                                           08420999 C  0056:1
    N ( IF SC=ALPHA THEN JUMP OUT TO TSTOPU; DS:=CHR );                           08421000 P  0056:1
AWAYU:                                                                            08421001 C  0058:1
    TALLY:=1;                                                                     08421002 C  0058:1
TSTOPU: DONE;                                                                     08421003 C  0058:2
    MORE:=TALLY; N:=SI; T1:=DI; SI:=S1; SOFF:=SI; DI:=D1; DOFF:=DI;               08421004 C  0058:2
END;                                                                              08421005 C  0060:1
      MORE:=P; SOFF:=P(SUB,DUP).[18:15]; SWI:=P.[33:15];                          08421006 C  0060:2
               DOFF:=P(SUB,DUP).[18:15]; DWI:=P.[33:15];                          08421007 C  0063:3
      END C@NDITIONAL ALPHA REPLACE;                                              08421008 C  0066:2
    IF DPTR.[01:01] THEN                                                          08421009 C  0066:3
      P(M&1[14:47:01],DPTR.[09:22]+(DPTR.[33:15]#0),CHS,CDC,DEL);                 08421010 C  0070:3
    IF (SPTR=0 AND RELATION.[1:1]=0) OR DPTR=0                                    08421011 C  0075:2
        THEN P(MKS,INTCALL((-4)&1[2:47:1],FORTERRI)) ;                            08421012 C  0078:0
    DEST:=M[DPTR];                                                                08421013 C  0082:2
```

```
      DSIZE:=IF DPTR.CSIZE=EIGHT THEN 8 ELSE 6;                                  08421014 C  0084:0
      IF (TOGL:=RELATION.[01:01]=0) THEN                                         08421015 C  0087:1
         SSIZE:=IF SPTR.CSIZE=EIGHT THEN 8 ELSE 6                                08421016 C  0089:0
         ELSE SSIZE←6;           COMMENT LITERAL OR AEXP SOURCE;                 08421020 T  0092:0
      IF TOGL AND DSIZE≠SSIZE THEN                                               08421040 T  0094:0
         POLISH(DEST&1[08:38:10],8686,CDC,DEL);                                  08421060 T  0095:1
      UPDTOG←                                                                    08421080 T  0098:0
         POLISH(.UPDPDD,LOD,RFB,.UPSPDD,LOD,RFB,OR,.UPCTDD,LOD,RFB,OR)≠0;        08421100 T  0098:0
      IF (HISCNT←HISCNT) ≤ 0 THEN                                                08421120 P  0101:3
        BEGIN UPCTDD[0]←0;UPDPDD[0]←DPTR;UPSPDD[0]←SPTR;P(XIT);END;               08421130 T  0103:0
      IF DSIZE=8 THEN                                                            08421140 T  0106:0
         BEGIN                                                                   08421160 T  0106:3
 $ SET OMIT = NOT EIGHTBIT                                                       08421170 C  0107:1
COMMENT CAUSE INVALID INDEX, 9898 GEQ ASZ IF 8 BIT REPLACE IS DONE;              08423273 C  0107:1
          POLISH(DEST&1[8:38:10],9898,CDC,DEL);                                  08423275 C  0107:1
 $ RESET OMIT                                                                    08423276 C  0109:2
         END ELSE %8-BIT DEST FINISHED                                           08423280 T  0109:2
         COMMENT IF WE GET THIS FAR, DSIZE≠EIGHT & SSIZE=DSIZE, SO               08423300 T  0109:2
            SPTR CAN"T BE 8 BITS/CHAR;                                           08423320 T  0109:2
         IF RELATION.[42:06]=TCOND THEN                                          08423340 T  0109:2
            BEGIN                                                                08423360 T  0113:1
            COMMENT UNCONDITIONAL XFER OF 6-BIT CHARS OR WORDS;                  08423380 T  0113:3
            IF RELATION.[40:01]=1 THEN                                           08423400 T  0113:3
               BEGIN                                                             08423420 T  0115:0
               COMMENT WORD TRANSFER;                                            08423440 T  0115:2
               DPTR.DOT←0&(DOFF←(0&DPTR[35:18:13]+7),[35:10])[35:38:10];         08423460 T  0115:2
               OURCNT←HISCNT.[38:10];                                            08423480 T  0120:3
               IF (DOFF+OURCNT)>(DEST).[08:10] THEN                              08423500 T  0122:0
                  POLISH([DEST[(DEST.[8:10])]]);                                 08423520 T  0124:0
IF TOGL THEN                                                                     08423540 T  0125:3
   BEGIN                                                                         08423560 T  0126:0
   IF SPTR.[01:01] THEN                                                          08423570 T  0126:2
      P(M&1[14:47:01],SPTR.[09:22]+(SPTR.[33:15]≠0),CHS,                         08423571 T  0127:1
         CDC,DEL);                                                               08423572 T  0131:2
   COMMENT POINTER SOURCE;                                                       08423580 T  0132:0
   SORC←M[SPTR];                                                                 08423600 T  0132:0
SPTR.DOT←0&(SOFF←(0&SPTR[35:18:13]+7),[35:10])[35:38:10];                        08423620 T  0133:2
   IF (SOFF+OURCNT)>(SORC).[08:10] THEN                                          08423640 T  0138:3
   POLISH([SORC[(SORC.[8:10])]]);                                                08423660 T  0140:3
   IF OURCNT>0 THEN                                                              08423670 T  0142:2
   STREAM(SOURCE←[SORC[SOFF]],                                                   08423680 T  0143:1
      N1←OURCNT,N2←OURCNT.[38:04],                                               08423700 T  0144:2
      DESTAD←[DEST[DOFF]]);                                                      08423720 T  0145:2
      BEGIN                                                                      08423740 T  0146:1
      SI←SOURCE;                                                                 08423760 T  0146:1
      DS←N1 WDS;  N2(2(DS←32 WDS));                                              08423780 T  0146:2
      END;                                                                       08423800 T  0148:2
   END ELSE %6-BIT POINTER SOURCE FINISHED FOR WD XFER                           08423820 T  0148:3
   BEGIN                                                                         08423840 T  0148:3
   COMMENT LITERAL/AEXP SOURCE;                                                  08423860 T  0149:1
   SORCL←HISCNT.[18:15];                                                         08423880 T  0149:1
   IF SORCL=0 THEN SORCL←HISCNT;                                                 08423900 T  0150:2
   INITIAL←IF OURCNT>0 THEN 1 ELSE 0;                                            08423920 T  0152:2
   OURCNT←OURCNT-INITIAL;                                                        08423940 T  0155:1
   STREAM(START←SPTR,SORCL,INITIAL,                                              08423960 T  0156:2
      SOFSET←8-SORCL,IN1←(JUNK←8 DIV SORCL),                                     08423980 T  0157:2
      IN2←8-JUNK×SORCL,N1←OURCNT,N2←OURCNT.[38:04],                              08424000 T  0159:2
```

```
      DESTAD←[DEST[DOFF]],                                                  08424020 T  0161:3
      SETDI←[JUNK]);                                                        08424040 T  0162:1
      BEGIN                                                                 08424060 T  0162:3
      SI←LOC START;                          SI←SI+SOFSET;                  08424080 T  0162:3
      SOFSET←SI;                                                            08424100 T  0163:2
      IN1(DS←SORCL CHR;                      SI←SOFSET);                    08424120 T  0163:3
      DS←IN2 CHR; SI←SETDI;                  DI←DESTAD;                     08424140 T  0165:1
      START←DI;                                                             08424160 T  0166:1
      DS←INITIAL WDS;                        SI←START;                      08424180 T  0166:2
      DS←N1 WDS;  N2(2(DS←32 WDS));                                         08424200 T  0167:1
      END;                                                                  08424220 T  0169:1
   OURCNT←OURCNT+INITIAL;                                                   08424240 T  0169:2
   END;%OF WORD-XFER FROM LIT-AEXP SOURCE                                   08424260 T  0170:3
   IF UPDTOG THEN CHAR←0&OURCNT[32:35:13];                                  08424280 T  0170:3
END ELSE %WORD TRANSFER DONE                                                08424300 T  0173:1
BEGIN                                                                       08424320 T  0173:1
COMMENT CHAR XFER FROM 6-BIT SOURCES;                                       08424340 T  0173:3
DOFF←DPTR.DOT;    OURCNT←HISCNT.[35:13];                                    08424360 T  0173:3
IF (DOFF+OURCNT)>0&(DEST)[35:08:10] THEN                                    08424380 T  0176:1
   POLISH([DEST[(DEST.[8:10])]]);                                           08424400 T  0178:3
IF TOGL THEN                                                                08424420 T  0180:2
   BEGIN                                                                    08424440 T  0180:3
   COMMENT SOURCE IS A POINTER;                                             08424460 T  0181:1
   IF SPTR.[01:01] THEN                                                     08424470 T  0181:1
      P(M&1[14:47:01],SPTR.[09:22]+(SPTR.[33:15]≠0),CHS,                    08424471 T  0182:0
         CDC,DEL);                                                          08424472 T  0186:1
   SORC←M[SPTR];                                                            08424480 T  0186:3
   SOFF←0&SPTR[35:18:13];                                                   08424500 T  0188:1
IF (SOFF + OURCNT) > 0 & (SORC)[35:8:10] THEN                               08424503 C  0190:0
  BEGIN                                                                     08424504 C  0192:2
    INITIAL := HISCNT.[35:13] - HISCNT.[20:13];                             08424506 C  0193:0
    STREAM(START:=SPTR.[28:3], FINISH:=DPTR.[28:3],                         08424508 C  0195:1
      N1:=INITIAL, N2:=INITIAL.[37:5],                                      08424510 C  0197:0
      N3:=INITIAL.[35:2], INITIAL:= HISCNT.[20:13],                         08424511 C  0198:0
      IN1:= HISCNT.[22:5], IN2:= HISCNT.[20:2],                             08424512 C  0199:2
      SOURCE:= [SORC[SPTR.[18:10]]],                                        08424513 C  0201:0
      DESTAD:= [DEST[DPTR.[18:10]]]);                                       08424514 C  0202:0
    BEGIN                                                                   08424516 C  0203:1
    SI:=SOURCE; SI:=SI+START; DI:=DI+FINISH;                                08424518 C  0203:1
      SOURCE:= DI;                                                          08424520 P  0204:2
      DS:= INITIAL CHR;     IN1(2(DS:=32 CHR ));                            08424522 C  0204:3
      IN2(2(32(DS:= 32 CHR )));                                             08424524 C  0206:3
      SI := SOURCE;                                                         08424526 C  0208:3
      DS:= N1 CHR;  N2(2(DS:= 32 CHR ));                                    08424530 C  0209:0
      N3(2(32(DS:= 32 CHR )));                                              08424540 P  0211:0
    END;                                                                    08424544 C  0213:0
END ELSE                                                                    08424546 C  0213:1
            IF OURCNT>0 THEN                                                08424550 T  0213:1
            STREAM(START←SPTR.[28:03],FINISH←DPTR.[28:03],                  08424560 T  0214:2
               N1←OURCNT,N2←OURCNT.[37:05],N3←OURCNT.[35:02],                08424580 T  0216:3
               SOURCE←[SORC[SPTR.[18:10]]],                                 08424600 T  0218:2
               DESTAD←[DEST[DPTR.[18:10]]]);                                08424620 T  0219:2
               BEGIN                                                        08424640 T  0220:3
              SI←SOURCE;  SI←SI+START;                                      08424660 T  0220:3
               DI←DI+FINISH;                                                08424680 T  0221:2
               DS←N1 CHR;  N2(2(DS←32 CHR));                                08424700 T  0222:0
               N3(2(32(DS←32 CHR)));                                        08424720 T  0224:0
```

```
   END;                                                                                    08424740 T  0226:0
END ELSE %POINTER SOURCE FINISHED                                                          08424760 T  0226:1
BEGIN                                                                                      08424780 T  0226:1
COMMENT LITERAL/AEXP SOURCE, UNCOND XFER, 6-BIT DEST;                                      08424800 T  0226:3
SORCL←HISCNT.[18:15];                                                                      08424820 T  0226:3
IF SORCL=0 THEN SORCL←HISCNT;                                                              08424840 T  0228:0
INITIAL←IF OURCNT>7 THEN 8 ELSE OURCNT;                                                    08424860 T  0230:0
OURCNT←OURCNT-INITIAL;                                                                     08424880 T  0232:3
IF INITIAL>0 THEN                                                                          08424885 T  0234:0
IF INITIAL≤SORCL THEN                                                                      08424890 T  0234:3
   STREAM(INITIAL,SPTR,SSKP← 8-SORCL,                                                      08424891 T  0236:0
      DOFSET←DPTR.[28:03],D←[DEST[DPTR.[18:10]]]);                                         08424892 T  0238:0
      BEGIN                                                                                08424893 T  0240:0
      DI←DI+DOFSET;              SI←LOC SPTR;                                              08424894 T  0240:0
      SI←SI+SSKP;                DS←INITIAL CHR;                                           08424895 T  0240:3
      END ELSE                                                                             08424896 T  0241:3
STREAM(START←SPTR,SORCL,INITIAL,                                                           08424900 T  0242:0
   SOFSET←8-SORCL,IN1←(JUNK←8 DIV SORCL),                                                  08424920 T  0243:2
   IN2←8-JUNK×SORCL,N1←OURCNT,N2←OURCNT.[37:05],                                           08424940 T  0245:2
   N3←OURCNT.[35:02],DOFSET←DPTR.[28:03],                                                  08424960 T  0247:3
   DESTAD←[DEST[DPTR.[18:10]]],SETDI←[JUNK]);                                              08424980 T  0249:1
   BEGIN                                                                                   08425000 T  0250:3
   SI←LOC START;                          SI←SI+SOFSET;                                   08425020 T  0250:3
   SOFSET←SI;  IN1(DS←SORCL CHR;          SI←SOFSET);                                     08425040 T  0251:2
   DS←IN2 CHR; SI←SETDI;                  DI←DESTAD;                                      08425060 T  0253:1
   DI←DI+DOFSET;                          START←DI;                                       08425080 T  0254:1
   DS←INITIAL CHR;                        SI←START;                                       08425100 T  0255:0
   DS←N1 CHR;  N2(2(DS←32 CHR));                                                          08425120 T  0255:3
               N3(2(32(DS←32 CHR)));                                                      08425140 T  0257:3
               END;                                                                       08425160 T  0259:3
            OURCNT←OURCNT+INITIAL;                                                        08425180 T  0260:0
            END;%OF LITERAL/AEXP 6-BIT SOURCE                                             08425200 T  0261:1
            CHAR←OURCNT;                                                                  08425220 T  0261:1
         END % OF 6-BIT UNCONDITIONAL XFER                                                08425240 T  0262:0
      END ELSE% UNCONDITIONAL XFER FINISHED                                               08425260 T  0262:0
IF RELATION=DECNVRT THEN                                                                   08425264 T  0262:0
   BEGIN                                                                                   08425265 T  0263:2
   DOFF←DPTR.[28:03];        DWI←DPTR.[18:10];                                             08425266 T  0264:0
   IF (HISCNT← (JUNK← HISCNT).[33:15])>8 THEN                                              08425267 T  0266:2
      POLISH(M&8[08:38:10],HISCNT,CDC,DEL);                                                08425268 T  0268:3
   IF    (0&(DEST) [35:08:10]-DPTR.DOT)-HISCNT<0 THEN                                      08425269 T  0271:1
      POLISH([DEST[DPTR.DOT+HISCNT]],DEL);                                                 08425270 T  0274:3
   SPTR←SPTR;  STREAM(SPTR,DOFF,HISCNT,D←[DEST[DWI]]);                                     08425271 T  0277:0
      BEGIN                                                                                08425272 T  0279:2
      DI← DI + DOFF;         SI← LOC SPTR;                                               08425273 T  0279:2
      DS←HISCNT DEC;                                                                       08425274 T  0280:1
      END;                                                                                 08425275 T  0280:3
   CHAR←HISCNT;                                                                            08425276 T  0281:0
   END ELSE                                                                                08425277 T  0281:3
   BEGIN                                                                                   08425280 T  0281:3
   COMMENT CONDITIONAL XFER W/ SOURCE & DEST BOTH 6-BIT POINTERS;                          08425300 T  0282:1
   IF SPTR.[01:01] THEN                                                                    08425312 T  0282:1
      P(M&1[14:47:01],SPTR.[09:22]+(SPTR.[33:15]≠0),CHS,CDC,DEL);                          08425314 T  0283:0
   SORC←M[SPTR];                                                                           08425320 T  0287:3
   SORCL←0&(SORC)[35:08:10]-SPTR.DOT;                                                      08425340 T  0289:1
   DESTL←0&(DEST)[35:08:10]-DPTR.DOT;                                                      08425360 T  0292:1
   OURCNT←IF SORCL>DESTL THEN                                                              08425380 T  0295:1
```

```
      (IF HISCNT>DESTL THEN DESTL ELSE HISCNT) ELSE                          08425400 T  0296:0
      IF HISCNT>SORCL THEN SORCL ELSE HISCNT;                                08425420 T  0299:1
   SOFF←SPTR.[28:03];       DOFF←DPTR.[28:03];                               08425440 T  0302:0
   SWI←SPTR.[18:10];        DWI←DPTR.[18:10];                                08425460 T  0304:2
   MORE←TRUE;                                                                08425480 T  0307:0
   IF RELATION GEQ 24 THEN % IN ALPHA, NUMERIC, TRUTHSET                     08425500 P  0307:3
      BEGIN                                                                  08425520 T  0308:2
      IF AWHILE:=(RELATION LEQ 26) THEN % "WHILE IN"                         08425540 P  0309:0
      ELSE RELATION:=31-RELATION;       % "UNTIL IN" (MUST INVERT)            08425545 C  0310:1
      RELATION:=RELATION.[46:2]; % 0,1,2                                     08425550 C  0312:2
      IF (ITERC←@URCNT)>63 THEN                                              08425560 T  0313:3
         BEGIN                                                               08425580 T  0315:0
         TOGL←OURCNT;       ITERC←63;                                       08425600 T  0315:2
         DO CRA UNTIL (TOGL←TOGL-63)≤63 OR NOT MORE;                         08425620 T  0317:0
         ITERC←TOGL;                                                         08425640 T  0321:0
         END;                                                                08425660 T  0321:3
      IF MORE AND ITERC>0 THEN CRA;                                          08425680 T  0321:3
      END ELSE                                                               08425700 T  0325:0
      BEGIN                                                                  08425720 T  0325:0
      CHAR:=0&CHAR[36:42:6]; RELATION:=RELATION.[43:3];                      08425740 P  0325:2
      IF (ITERC←OURCNT)>63 THEN                                              08425760 T  0328:2
         BEGIN                                                               08425780 T  0329:3
         TOGL←OURCNT;       ITERC←63;                                       08425800 T  0330:1
      DO CREPL UNTIL (TOGL←TOGL-63)≤63 OR NOT MORE;                          08425820 T  0331:3
         ITERC←TOGL;                                                         08425840 T  0336:0
         END;                                                                08425860 T  0336:3
      IF MORE AND ITERC>0 THEN CREPL;                                        08425880 T  0336:3
      END;                                                                   08425900 T  0340:0
   IF MORE AND HISCNT>OURCNT THEN                                            08425920 T  0340:0
      POLISH([DEST[DOFF&DWI[30:33:15]]],DEL,                                 08425940 T  0341:1
         [SORC[SOFF&SWI[30:33:15]]],DEL);                                    08425960 T  0343:2
      IF UPDTOG THEN CHAR←DOFF&DWI[30:33:15]-DPTR.DOT;                       08425980 T  0345:1
      END;% CONDITIONAL XFER OF 6-BIT CHARS DONE                             08427840 T  0348:3
      IF UPDTOG THEN                                                         08427860 T  0348:3
         BEGIN                                                               08427880 T  0349:0
         UPCTDD[0]←HISCNT.[35:13]-CHAR;                                      08427900 T  0349:2
         UPDPDD[0]←DPTR&( DPTR.DOT+CHAR )AMPER;                              08427920 T  0351:1
         UPSPDD[0]←SPTR&( SPTR.DOT+CHAR )AMPER;                              08427940 T  0354:0
         END;                                                                08427960 T  0356:3
   END REPL;                                                                 08427980 T  0356:3
                                                                                    SIZE= 0357 WORDS
```

```
PROCEDURE COMPARE;                                          %043            08430000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00577
   BEGIN                                                                    08430020 T  0000:0
   COMMENT STRING/POINTER COMPARISON INTRINSIC FOR B5500 TS ALGOL.           08430040 T  0000:0
      MARCH 1968.  POINTER UPDATES ADDED FOR STRING CMPR JUNE 1968.         08430060 T  0000:0
      MAJOR REWRITE TO CORRECT BAD ALGORITHM--OCT 69.                       08430062 T  0000:0
   RATCHFORD;                                                               08430080 T  0000:0
   COMMENT  THERE ARE FOUR FLAVORS OF STRING/POINTER COMPARE:               08430100 T  0000:0
   1. <AEXP> IN ALPHA.          AEXP IS IN [42:06] OF F=7, RELATION=29,     08430120 T  0000:0
   2. <PEXP1>=<PEXP2>  OR  <PEXP1> ≠ <PEXP2>,                               08430140 T  0000:0
      RELATION=65 FOR =  &  66 FOR ≠                                        08430160 T  0000:0
   3. <PUP1>:<PEXP1> <RELATION> <PUP2>:<PEXP2> FOR <COUNT>,                 08430180 T  0000:0
```

```
      4. <PEXP1> <RELATION> <LITERAL> FOR <COUNT>.                              08430200 T  0000:0
         VALUES OF <RELATION> ARE SAME AS FOR SCAN & REPLACE.  FOR #4,          08430220 T  0000:0
         F-FIELD OF F-2 IS LENGTH OF LITERAL STRING & C-FIELD IS VALUE          08430240 T  0000:0
         OF <COUNT> IN SOURCE STMT (8192 IF OMITTED),                           08430260 T  0000:0
         RELATION  HAS SIGN-BIT=1 FOR CASE #4.                                  08430280 T  0000:0
    ;                                                                           08430300 T  0000:0
      DEFINE                                                                    08430320 T  0000:0
         DOT=[18:13]#,                                                          08430340 T  0000:0
         AMPER=[18:35:13]#,                                                     08430360 T  0000:0
         CSIZE=[31:02]#,       SIX=00#;                                         08430380 T  0000:0
      INTEGER                                                                   08430400 T  0000:0
         RELATION     =-01,       %SAME CODES AS FOR SCAN/REPLACE.              08430420 T  0000:0
         HISCNT       =-02,       %LENGTH OF LITERAL IN F-FIELD,                08430440 T  0000:0
                                  %LENGTH OF COMPARE IN C-FIELD.                08430460 T  0000:0
         P2           =-03,       %SOURCE STMT IS "P1 RELATION P2 FOR           08430480 T  0000:0
                                  %HISCNT"                                      08430500 T  0000:0
         LITERAL      =P2 ,       %P2 MAY BE A LITERAL OF 1->8 CHARS,           08430520 T  0000:0
         P1           =-07,                                                     08430540 T  0000:0
         CHAR         =P1 ,       %LOC"N OF CHAR FOR "CHAR IN ALPH A"           08430560 T  0000:0
         R1C, R1W,                %CHAR & WORD OFFSET FOR P1                    08430580 T  0000:0
         R2C, R2W,                                                              08430600 T  0000:0
         N,                       %LENGTH OF COMPARE FOR CURRENT CALL OF        08430620 T  0000:0
                                  %FINDIT.                                      08430640 T  0000:0
         LOOPCOUNT    =P2,        %USED FOR LITERAL COMPARISONS,                08430660 T  0000:0
         JUNK         = 17;                                                     08430680 T  0000:0
      REAL RJUNK=JUNK;                                                          08430690 T  0000:0
      BOOLEAN                                                                   08430700 T  0000:0
         RESULT       =P1,        %THIS IS ALSO ONE OF THE POINTER ARGS,        08430720 T  0000:0
         DONE;                    %SOMETIMES MEANS WE FOUND SOME ≠ CHARS,       08430740 T  0000:0
      NAME                                                                      08430760 T  0000:0
         UPP2DD       =-04,       %DD FOR UPDATE OF P2 POINTER,                 08430780 T  0000:0
         UPP1DD       =-05,                                                     08430800 T  0000:0
         M            = 02;                                                     08430820 T  0000:0
      ARRAY                                                                     08430840 T  0000:0
         ROW1[*], ROW2[*];        %ARRAY ROWS REFERENCED BY P1/P2,              08430860 T  0000:0
      SUBROUTINE FINDIT;    %FIND BLOCK OF 64 CONTAINING 1ST ≠ CHARS,           08430880 T  0000:0
         BEGIN;                                                                 08430890 T  0001:0
         STREAM(N:R1C,R2C,R1←[ROW1[R1W]],R2←[ROW2[R2W]]);                       08430900 T  0001:0
            BEGIN                                                               08430910 T  0003:2
            SI←R1;                SI←SI+R1C;                                    08430920 T  0003:2
            DI←DI+R2C;                                                          08430930 T  0004:1
            IF N SC≠DC THEN TALLY←1;                                            08430940 T  0004:3
            N←TALLY;                                                            08430950 T  0005:3
            END;                                                                08430960 T  0006:0
         IF NOT (DONE←POLISH) THEN                                              08430970 T  0006:1
            BEGIN             %SET UP WORD & CHAR OFFSET FOR NEXT CALL,         08430980 T  0007:0
            R1C←POLISH(R1C&R1W[35:38:10]+N,DUP),[45:03];                        08430990 T  0007:2
            R1W←POLISH,[35:10];                                                 08431000 T  0010:2
            R2C←POLISH(R2C&R2W[35:38:10]+N,DUP),[45:03];                        08431010 T  0011:2
            R2W←POLISH,[35:10];                                                 08431020 T  0014:2
            END UPDATE OF CHAR AND WORD INDICES;                                08431030 T  0015:2
         END FINDIT;                                                            08431040 T  0015:2
      SUBROUTINE COMP;         %COMPARE 2 ≠ CHARS FOR < OR >,                   08431050 T  0015:3
         BEGIN;                                                                 08431060 T  0016:0
         STREAM(RELATION:R1C,R2C,R1←[ROW1[R1W]],R2←[ROW2[R2W]]);                08431070 T  0016:0
            BEGIN                                                               08431080 T  0018:2
            SI←R1;                SI←SI+R1C;                                    08431090 T  0018:2
```

```
      DI←DI+R2C;                                                                  08431100 T  0019:1
      COMMENT COMP SHOULD ONLY BE CALLED IF FINDIT FINDS 2 ≠ CHARS.;              08431110 T  0019:3
      63(IF SC≠DC THEN JUMP OUT);                                                 08431120 T  0019:3
      SI←SI-1;              DI←DI-1;                                              08431130 T  0021:1
      CI←CI+RELATION;                                                           08431140 T  0021:3
         GO GR;                                                                 08431150 T  0022:1
        %GO LS;                                                                 08431160 T  0022:2
      LS:IF SC<DC THEN ;    GO XX;                                              08431170 T  0022:2
      GR:IF SC>DC THEN ; % GO XX;                                               08431180 T  0023:1
      XX:IF TOGGLE THEN TALLY←1;                                                08431190 T  0023:3
      RELATION←TALLY;                                                           08431200 T  0024:1
      END COLLATING SEQ COMPARE;                                                08431210 T  0024:2
   DONE←POLISH;                                                                 08431220 T  0024:3
   END COMP;                                                                    08431230 T  0025:1
HISCNT ← (HISCNT≥ 0) × HISCNT;                                                  08431520 P  0025:2
IF RELATION.[43:5] GEQ 29 THEN % IN ALPHA, NUMERIC, TRUTHSET                    08431540 P  0029:3
   BEGIN;                                                                       08431560 T  0031:0
   COMMENT CHAR IN ALPHA TEST;                                                  08431580 T  0031:2
   STREAM(TALLIE:=0 : CHAR, ITS:=RELATION.[46:2]-1, TSET:=[HISCNT]);            08431600 P  0031:2
      BEGIN                                                                     08431620 T  0034:0
      SI←LOC CHAR;          SI←SI+7;                                            08431640 T  0034:1
      CI:=CI+ITS; GO TO ALP; GO TO NMR; % GO TO TSET;                           08431660 P  0034:3
      TST: SI:=TSET; SKIP CHAR SB; SKIP SB;                                     08431665 C  0035:3
           IF SB THEN BEGIN TALLY:=1; TALLIE:=TALLY END; GO DUN;                08431670 C  0036:3
      NMR: IF SC GEQ "0" THEN IF SC LEQ "9" THEN %                              08431675 C  0038:0
           BEGIN TALLY:=1; TALLIE:=TALLY END; GO DUN;                           08431680 P  0039:0
      ALP: IF  SC=ALPHA THEN BEGIN TALLY:=1; TALLIE:=TALLY END;                 08431685 C  0039:3
      DUN:                                                                      08431690 C  0040:3
      END;                                                                      08431695 C  0040:3
   RESULT←POLISH;                                                               08431700 T  0041:0
   END ELSE                                                                     08431720 T  0041:2
IF RELATION>64 THEN                                                             08431740 T  0041:2
   COMMENT P1=P2  OR P1≠P2:COMPARE THE ABSOLUTE ADDRESSES THAT THE              08431760 T  0042:3
   2 POINTERS REFERENCE WITHOUT LOOKING AT THE CONTENTS OF THESE                08431780 T  0042:3
   LOCATIONS.  NOTE THAT UNINITIALIZED POINTERS COMPARE EQUAL.;                 08431800 T  0042:3
   RESULT←(RELATION=65) EQV   (P1.[18:30]=P2.[18:30]) ELSE                      08431820 T  0042:3
   BEGIN                                                                        08431840 T  0046:2
   COMMENT A RELATIONAL COMPARISON OF TWO STRINGS.;                             08431860 T  0047:0
   COMMENT NOTE THAT THE 5500 SIMULATIONS USE THE BCL COLLATING                 08431880 T  0047:0
   SEQUENCE FOR RELATIONAL COMPARISONS, WHEREAS THE 6500 WILL                   08431900 T  0047:0
   COMPARE THE MAGNITUDES OF THE TWO CHARACTERS AS 4- 6- OR 8-BIT               08431920 T  0047:0
   INTEGERS.  THE 5500 SIMULATION ALSO ONLY ALLOWS 6-BIT BCL CHARS;             08431940 T  0047:0
   IF (P2=0 AND RELATION.[1:1]=0) OR P1=0                                       08431945 T  0047:0
    THEN P(MKS,INTCALL((-4)&1[2:47:1],FORTERRI)) ;                              08431946 T  0049:2
   IF P1.[01:01] THEN                                                           08431950 T  0054:0
      P(M&1[14:47:01],P1.[09:22]+(P1.[33:15]≠0),CHS,CDC,DEL);                   08431951 T  0054:3
   IF P1.CSIZE≠SIX THEN POLISH(M&1[14:47:01],8686,CDC,DEL);                     08431960 T  0059:2
   ROW1←M[P1];              R1C←P1.[28:03];                                     08431980 T  0063:1
   R1W←P1.[18:10];          HISCNT←ABS(HISCNT);                                 08432000 T  0066:0
   IF (JUNK←HISCNT.[35:13]+P1.DOT)>0&(ROW1)[35:08:10] THEN                      08432020 T  0068:1
      POLISH([ROW1[JUNK]],DEL);                                                 08432040 T  0072:1
   IF RELATION.[01:01]=0 THEN                                                   08432060 T  0073:2
      BEGIN                                                                     08432080 T  0074:3
      COMMENT BOTH P1&P2 ARE POINTERS;                                          08432100 T  0075:1
      IF P2.[01:01] THEN                                                        08432110 T  0075:1
         P(M&1[14:47:01],P2.[09:22]+(P2.[33:15]≠0),CHS,CDC,DEL);                08432111 T  0076:0
      IF P2.CSIZE≠SIX THEN                                                      08432120 T  0080:3
```

```
   POLISH(M&1[14:47:01], 8686,CDC,DEL);                                  08432140 T  0082:0
ROW2←M[P2];            R2C←P2.[28:03];                                   08432160 T  0084:2
R2W←P2.[18:10];                                                          08432180 T  0087:1
IF (HISCNT←P2.DOT)>0&(ROW2)[35:08:10] THEN                               08432200 T  0088:2
   POLISH([ROW2[HISCNT←P2.DOT]],DEL);                                    08432220 T  0091:2
IF (JUNK←HISCNT)>63 THEN                                                 08432240 T  0093:3
   BEGIN                                                                 08432260 T  0095:0
   N←63;                                                                 08432280 T  0095:2
   DO FINDIT UNTIL ((JUNK←JUNK-63)≤63) OR DONE;                          08432300 T  0096:1
   END;                                                                  08432320 T  0099:3
IF (NOT DONE) AND (N←JUNK)≠0 THEN FINDIT;                                08432340 T  0099:3
END ELSE                                                                 08432380 T  0103:0
BEGIN                                                                    08432400 T  0103:0
COMMENT P2 IS A LITERAL STRING;                                          08432420 T  0105:0
IF (N←HISCNT.[18:15])=0 THEN N←HISCNT;                                   08432440 T  0105:0
RJUNK←P2;              ROW2←[RJUNK]&1[17:47:01];                         08432460 T  0108:0
R2W←0;                                                                   08432480 T  0110:2
COMMENT IF HISCNT.[18:15]≠0 THEN [18:15]=STRING LENGTH &                 08432481 T  0111:1
   [33:15]=EXPLICIT LENGTH OF COMPARE,  OTHERWISE, [33:15] IS            08432482 T  0111:1
   BOTH LENGTH OF COMPARE AND LENGTH OF LITERAL.;                        08432483 T  0111:1
IF (HISCNT←HISCNT.[33:15])≤N THEN                                        08432500 T  0111:1
   BEGIN                                                                 08432520 T  0113:0
   COMMENT LENGTH OF COMPARE IS ≤ LENGTH OF LITERAL: WE DON"T            08432540 T  0113:2
      HAVE TO REPEAT THE LITERAL;                                        08432560 T  0113:2
   R2C←8-N;            FINDIT;                                           08432580 T  0113:2
   END ELSE                                                              08432600 T  0116:0
   BEGIN;                                                                08432620 T  0116:0
   COMMENT LITERAL MUST BE DUPLICATED TO FILL 8 CHARS ;                  08432640 T  0116:2
   STREAM(P2,N,SOFF←8-N,N1←8 DIV N,N2←(JUNK←8 MOD N),                    08432660 T  0116:2
   D←[JUNK]);                                                            08432680 T  0120:0
      BEGIN                                                              08432700 T  0120:2
      SI←LOC P2;      SI←SI+SOFF;     D←SI;                            08432720 T  0120:2
      N1(DS←N CHR;    SI←D);                                           08432740 T  0121:2
      DS←N2 CHR;                                                         08432760 T  0123:0
      END;                                                               08432780 T  0123:2
   N←IF HISCNT≤8 THEN HISCNT ELSE 8;                                     08432800 T  0123:3
   LOOPCOUNT←HISCNT-N;                R2C←0;                             08432820 T  0126:2
   FINDIT;                                                               08432840 T  0128:2
   IF NOT DONE THEN                                                      08432860 T  0130:0
      BEGIN                                                              08432880 T  0130:2
      R2C←P1.[28:03];          R2W←P1.[18:10];                           08432900 T  0131:0
      ROW2←ROW1;                                                         08432920 T  0133:2
      IF LOOPCOUNT>63 THEN                                               08432940 T  0134:2
         BEGIN                                                           08432960 T  0135:1
         N←63;                                                           08432980 T  0135:3
         DO FINDIT UNTIL DONE OR (LOOPCOUNT←LOOPCOUNT-63)≤63;            08433000 T  0136:2
         END;                                                            08433020 T  0140:3
      IF (NOT DONE) AND (N←LOOPCOUNT)≠0 THEN FINDIT;                     08433040 T  0140:3
      END;                                                               08433060 T  0144:0
   END                                                                   08433120 T  0144:0
END;                                                                     08433140 T  0144:0
BEGIN                                                                    08433141 T  0144:0
UPP1DD[0] ← P1 +0&HISCNT[18:35:13];                                      08433142 T  0144:0
UPP2DD[0] ← P2 +0&HISCNT[18:35:13]; END;                                 08433143 T  0146:1
IF (RELATION←ABS(RELATION))=8 THEN RESULT←(NOT DONE).[47:01] ELSE        08433145 T  0148:2
IF RELATION=12 THEN RESULT←DONE ELSE                                     08433147 T  0151:2
IF DONE THEN                                                             08433149 T  0154:2
```

```
            BEGIN    %FINDIT DISCOVERED A ≠ CHAR IN THE TWO STRINGS,          08433151 T  0155:1
            RELATION←RELATION.[45:01];    %0 & 16 TEST >, 4 & 20 TEST <      08433153 T  0155:3
            COMP;                  RESULT←DONE;                              08433155 T  0157:0
            END ELSE                                                         08433157 T  0158:3
            COMMENT STRINGS WERE = AND RELATION IS NOT = OR ≠, PLUCK THE      08433159 T  0158:3
               "=" HALF OF "≤" OR "≥" OUT OF RELATION.;                      08433161 T  0158:3
            RESULT←RELATION.[43:01];                                         08433163 T  0158:3
         END COLLATING SEQUENCE COMPARES;                                    08433320 T  0160:2
      END COMPARE;                                                           08433340 T  0160:2
                                                                                    SIZE= 0161 WORDS


      PROCEDURE BASICPRINT(TYPE);                                            08500000 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00583
      VALUE    TYPE;                                                         08500100 T  0000:0
      REAL     TYPE;                                                         08500200 T  0000:0
                                                                             08500300 T  0000:0
      BEGIN REAL    ALGOLWRITE = 12,                                         08500400 T  0000:0
                    ALGOLSELECT = 14;                                        08500450 T  0000:0
         REAL       RCW = +0;                                               08500500 T  0000:0
         ARRAY      POT = 25[*];                                            08500600 T  0000:0
         NAME       M = 2;                                                  08500700 T  0000:0
         REAL       T,                                                      08500800 T  0000:0
                    WH1,                                                    08500900 T  0000:0
                    WH2;                                                    08501000 T  0000:0
         BOOLEAN    THISTYPE;                                               08501100 T  0000:0
         INTEGER    BSIZE,                                                  08501200 T  0000:0
                    BUFF,                                                   08501300 T  0000:0
                    BUFFLOAD,                                               08501400 T  0000:0
                    COL,                                                    08501500 T  0000:0
                    COUNTER,                                                08501600 T  0000:0
                    E,                                                      08501700 T  0000:0
                    ESIGN,                                                  08501800 T  0000:0
                    EXPCHR,                                                 08501900 T  0000:0
                    I,                                                      08502000 T  0000:0
                    ITEMS,                                                  08502100 T  0000:0
                    NUMCHR,                                                 08502200 T  0000:0
                    NUMROWS,                                                08502300 T  0000:0
                    ROW,                                                    08502400 T  0000:0
                    ROWLENGTH,                                              08502500 T  0000:0
                    SIGN,                                                   08502600 T  0000:0
                    SKIP,                                                   08502700 T  0000:0
                    TAB,                                                    08502800 T  0000:0
                    WRITESTMT;                                              08502810 T  0000:0
         NAME       POINTER;                                                08502900 T  0000:0
         NAME       FILX;                                                   08502950 T  0000:0
         NAME       STRING = T;                                             08503000 T  0000:0
         ARRAY      FIB[*],                                                 08503100 T  0000:0
                    MATRIX[*],                                              08503200 T  0000:0
                    MATRIXROW[*];                                           08503300 T  0000:0
             BOOLEAN DATACOM, FIRSTIME;                                     08503325 P  0000:0
         LABEL      COMMON,                                                 08503400 T  0000:0
                    LOGEIGHT,                                               08503500 T  0000:0
                    MAXINT,                                                 08503600 T  0000:0
                    MINVALUE,                                               08503700 T  0000:0
```

```
                TENSEVEN,                                                                  08503800 T  0000:0
                TENSIX,                                                                    08503900 T  0000:0
                CONVERTED,                                                                 08504000 T  0000:0
                NORMAL,                                                                    08504100 T  0000:0
                                DUMMYLABEL;                                                08504200 T  0000:0
DEFINE          LOG8 = P(LOGEIGHT)#,                                                       08504300 T  0000:0
                MAX  = P(MAXINT)  #,                                                       08504400 T  0000:0
                DELTA= P(MINVALUE)#,                                                       08504500 T  0000:0
                TEN6 = P(TENSIX)  #,                                                       08504600 T  0000:0
                TEN7 = P(TENSEVEN)#;                                                       08504700 T  0000:0
DEFINE          COMMA = 1#,                                                                08504800 T  0000:0
                SEMICOLON = 2#,                                                            08504900 T  0000:0
                ENDLINE = 3#;                                                              08505000 T  0000:0
                                                                                           08505100 T  0000:0
                                                                                           08505200 T  0000:0
REAL SUBROUTINE GETNEXT;                                                                   08505300 T  0000:0
BEGIN THISTYPE ← (TYPE ← 0&TYPE[6:7:40]),[6:1];                                            08505400 T  0001:0
     ITEMS ← ITEMS-1;  TAB ← *(1 INX POINTER);                                             08505500 T  0003:3
     P(*POINTER);  POINTER ← 2 INX POINTER;                                                08505600 T  0006:2
     GETNEXT ← POLISH                                                                      08505700 T  0008:1
END GETNEXT;                                                                               08505800 T  0008:1
BOOLEAN SUBROUTINE DIMENSION;                                                              08505900 T  0008:3
BEGIN COMMENT TRUE FOR SINGLE DIMENSIONED (INCLUDES STRINGS);;                             08506000 T  0009:0
     STREAM(T←POLISH(XCH, 0, CDC):A←0);                                                    08506100 T  0009:0
     BEGIN SI←T;  DI←T;  SI←SI-16;  SKIP 2 SB;                                             08506200 T  0010:3
          IF SB THEN ELSE TALLY←1;  T ← TALLY;                                             08506300 T  0011:3
     END STREAM;                                                                           08506400 T  0013:0
     DIMENSION ← THISTYPE OR POLISH;                                                       08506500 T  0013:1
END DIMENSION;                                                                             08506600 T  0014:0
SUBROUTINE SETUPANDEXIT;                                                                   08506700 T  0014:1
BEGIN IF DATACOM THEN                                                                      08506800 T  0015:0
     IF COUNTER NEQ 0 THEN                                                                 08506810 T  0015:1
     BEGIN;STREAM(BUFF);  DS:=LIT "←";                                                     08506820 T  0016:2
          POLISH(MKS, 16, 0, 0, BSIZE, FILX, ALGOLWRITE);                                  08506830 T  0018:2
          BUFF := =(*FILX).[CF];                                                           08506850 T  0020:1
     END;                                                                                  08506860 T  0022:0
     FIB[20]:=BUFF&COUNTER[3:33:15]&BSIZE[18:38:10]&1[29:47:1];                            08506900 T  0022:0
     POLISH(XIT);                                                                          08507000 T  0026:1
END SETUPANDEXIT;                                                                          08507100 T  0026:2
SUBROUTINE PRINT;                                                                          08507200 T  0026:3
BEGIN P(MKS,1,0,0,BSIZE+((NOT DATACOM).[47:1]),FILX,ALGOLWRITE);                           08507300 T  0027:0
     IF NOT DATACOM THEN                                                                   08507310 T  0030:0
     STREAM(A:="10",B:=[FIB[0]]);                                                          08507320 T  0030:2
     BEGIN SI:=LOC A; DS:=8 ADD; END STREAM;                                               08507330 T  0032:1
END  PRINT ROUTINE;                                                                        08507340 T  0033:0
SUBROUTINE CLEAR;                                                                          08507400 T  0033:1
BEGIN;STREAM(A←BSIZE-1+DATACOM:D←*FILX);                                                   08507450 T  0034:0
     BEGIN DS:=8 LIT "  "; SI:=D;  DS:=A WDS;  DI:=D;  A:=DI;  END;                       08507500 T  0036:2
     BUFF:=POLISH;  BUFFLOAD:=BSIZE×8;                                                     08507550 T  0039:0
END  CLEAR ROUTINE;                                                                        08507600 T  0041:0
SUBROUTINE CHECKPRESENCE;                                                                  08507700 T  0041:1
BEGIN  FIB[20]←(*P(DUP))&(COUNTER←0)[3:33:15];                                             08507800 T  0042:0
       BSIZE←P(MKS,1,0,0,(-1),FILX,ALGOLWRITE);                                            08507825 T  0045:0
       IF DATACOM←FIB[4].[8:4]=10 OR FIB[4].[8:4]=13 THEN                                  08507850 T  0047:2
          BSIZE ← 9 ELSE                                                                   08507855 T  0051:1
          BEGIN   BSIZE←BSIZE-1;                                                           08507860 T  0052:2
                  STREAM(A←FIB[0],B←BSIZE INX (*FILX));                                    08507870 T  0054:1
```

```
                    BEGIN  SI←LOC A;  DS←WDS;  END;                                  08507880 T  0056:1
          END;                                                                       08507890 T  0057:0
        CLEAR;                                                                       08507895 T  0057:0
END CHECKPRESENCE;                                                                   08507900 T  0058:0
SUBROUTINE PRINTEXIT;                                                                08508000 T  0058:1
BEGIN PRINT;                                                                         08508100 T  0059:0
     COUNTER ← 0;                                                                    08508200 T  0060:0
     SETUPANDEXIT;                                                                   08508300 T  0060:3
END PRINTEXIT;                                                                       08508400 T  0062:0
SUBROUTINE PRINTRETURN;                                                              08508500 T  0062:1
BEGIN PRINT;                                                                         08508600 T  0063:0
     CHECKPRESENCE;                                                                  08508700 T  0064:0
END PRINTRETURN;                                                                     08508800 T  0065:0
SUBROUTINE FINDE;                                                                    08508900 T  0065:1
BEGIN COMMENT DETERMINE THE EXPONENT OF A REAL NUMBER.  THE NUMBER                   08509000 T  0066:0
     IS POSITIVE AND PASSED IN T.  WHEN DONE, STORE THE                              08509100 T  0066:0
     EXPONENT IN E, AND ROUND T TO  10*6 ≤ T < 10*7;                                 08509200 T  0066:0
     E ← ((0&T[42:3:6]&T[1:2:1]+12)×LOG8)+0.5;                                       08509300 T  0066:0
     WHILE T<(IF E≥0 THEN POT[E] ELSE 1/POT[-E]) DO E ← E-1;                         08509400 T  0070:1
     T ← IF (6-E)≥0 THEN T×POT[6-E] ELSE T/POT[E-6];                                 08509500 T  0078:0
END FIND EXPONENT AND ROUND NUMBER;                                                  08509600 T  0083:3
SUBROUTINE CONVERT;                                                                  08509700 T  0084:0
BEGIN                                                                                08509800 T  0084:0
     IF THISTYPE THEN                                                                08509900 T  0084:0
     BEGIN;  STREAM(A←0:S←STRING);                                                   08510000 T  0084:1
          BEGIN SI←S; SI←SI+15; DI←LOC A; DI←DI+7; DS←CHR; END;                      08510100 T  0086:0
               IF (I ← POLISH+6)>15 THEN I ← I-20;                                   08510200 T  0087:2
               NUMCHR := I + 3 × WRITESTMT;                                          08510250 T  0090:3
          IF(COUNTER + NUMCHR) GTR BUFFLOAD THEN PRINTRETURN;                        08510300 T  0092:2
          STREAM(I:STRING,WRITESTMT,BUFF);                                           08510400 T  0095:0
          BEGIN SI:=STRING; WRITESTMT(DS←LIT """"); DS←I CHR;                        08510500 T  0096:3
               WRITESTMT(DS←LIT """"; DS←LIT ","); I←DI;                             08510550 T  0098:3
          END  STREAM;                                                               08510575 T  0100:3
          BUFF := POLISH;  COUNTER := COUNTER + NUMCHR;                              08510600 T  0101:0
                                                                                     08510650 P  0102:3
          GO TO CONVERTED;                                                           08510700 T  0102:3
LOGEIGHT::: @1157163034761674;                                                       08510750 T  0103:1
     END STRING HANDLING;                                                            08510800 T  0105:0
     COMMENT THAT WAS EASY -- NOW FOR THE NUMERICAL STUFF;                           08510900 T  0105:0
     ESIGN ← EXPCHR ← SIGN ← SKIP ← NUMCHR ← 0;                                      08511000 T  0105:0
     SIGN ← T≠(T ← ABS(T));                                                          08511100 T  0107:3
     IF T<MAX THEN                                                                   08511200 T  0109:3
       IF (IF T=0 THEN TRUE ELSE (ABS(((I ← T)-T)/T)<DELTA)) THEN                    08511300 T  0110:2
          BEGIN;  COMMENT INTEGER, OR NEAR ENOUGH;                                   08511400 T  0115:2
               STREAM(I←I ← T:T←[WH1]);                                              08511500 T  0116:0
               BEGIN SI←LOC I;  DS←8 DEC;  SI←T;                                     08511600 T  0117:3
                    7(IF SC="0" THEN SI←SI+1);  I←SI;                                08511700 T  0118:2
               END STREAM;                                                           08511800 T  0120:0
               NUMCHR ← 8-(SKIP ← P(XCH),[30:3]);                                    08511900 T  0120:1
               GO TO COMMON;                                                         08512000 T  0122:2
          END INTEGER CASE;                                                          08512100 T  0123:0
     T ← 1.0×T;  FINDE;                                                              08512200 T  0123:0
     IF (I ← T)≥TEN7 THEN                                                            08512300 T  0125:0
     BEGIN I ← TEN6;  E ← E+1;  END;                                                 08512400 T  0126:1
     IF E<0 AND E≥(-7) THEN                                                          08512500 T  0128:3
       IF I=((I DIV (T ← POT[ABS(E+1)]))×T) THEN                                     08512600 T  0130:3
     BEGIN I ← I DIV T;                                                              08512700 T  0134:2
```

```
     STREAM(P1←0;P2←P(ABS(E+1),DUP),                                    08512800 T  0136:1
          P3←7-P(XCH),P4←P(DUP)-1,P5←I,P6←[WH1]);                        08512900 T  0138:1
     BEGIN DS←2 LIT "0.";  P2(DS ← LIT "0");                            08513000 T  0140:2
          SI←LOC P5;  DS←P3 DEC;  P1←DI;  DI←P6;  SI←P1;                08513100 T  0142:1
          SI←SI-1;  P4(IF SC≠"0" THEN JUMP OUT;                         08513200 T  0143:3
                         TALLY←TALLY+1;  SI←SI-1);                      08513300 T  0145:2
          P1←TALLY;                                                     08513400 T  0146:1
     END STREAM;                                                        08513500 T  0146:2
     NUMCHR ← 9-P(XCH);                                                 08513600 T  0146:3
     GO TO COMMON;                                                      08513700 T  0148:0
END F TYPE STUFF;                                                       08513800 T  0150:0
IF E≥0 AND E<7 THEN                                                     08513810 T  0150:0
BEGIN COMMENT THE OTHER HALF OF F-FORMATTING;;                          08513820 T  0151:3
     STREAM(P0←0; P1←P(E+1, DUP),                                       08513830 T  0152:1
                    P2←7-P(XCH), P3←I, P4←[WH1]);                       08513840 T  0154:0
     BEGIN DI←DI+1;  SI←LOC P3;  DS←7 DEC;                              08513845 T  0155:2
          DI←P4;  SI←P4;  SI←SI+1;  DS←P1 CHR;                          08513850 T  0156:1
          DS←LIT ".";  SI←SI+P2;  SI←SI-1;                              08513855 T  0157:2
          P2(IF SC≠"0" THEN JUMP OUT;                                   08513860 T  0158:3
               TALLY←TALLY+1;  SI←SI-1);  P0←TALLY;                     08513865 T  0160:1
     END  STREAM;                                                       08513870 T  0161:1
     NUMCHR ← 8-P(XCH);  GO TO COMMON;                                  08513880 T  0161:2
END OTHER HALF FORMATTING;                                              08513890 T  0163:1
STREAM(P1←ABS(E);P2←I,P3←[E],P4←[WH1]);                                 08513900 T  0163:1
BEGIN DI←DI+1;  SI←LOC P2;  DS←7 DEC;  DI←P4;  SI←P4;                   08514000 T  0165:1
     SI←SI+1;  DS←CHR;  DS←LIT ".";  SI←SI+5;                           08514100 T  0166:2
     6(IF SC≠"0" THEN JUMP OUT;  SI←SI-1;  TALLY←TALLY+1);              08514200 T  0167:3
     DI←P3;  SI←LOC P1;  DI←DI+6;  DS←2 DEC;  P1←TALLY;                  08514300 T  0169:3
     END STREAM;                                                        08514400 T  0171:0
     EXPCHR ← 1+(ABS(E)>9);  ESIGN ← E<0;  NUMCHR ← 8-P(XCH);           08514500 T  0171:1
COMMON::  T ← 1+NUMCHR+EXPCHR+((EXPCHR≠0)+P(DUP))+WRITESTMT;             08514600 T  0175:3
     IF (COUNTER+T)>BUFFLOAD THEN PRINTRETURN;                          08514700 T  0179:3
     STREAM(P1←SKIP;NUMCHR,SIGN,EXPCHR,P2←P(DUP)≠0,                     08514800 T  0183:0
          ESIGN,P3←[WH1],P4←[E],WRITESTMT,BUFF);                        08514900 T  0185:1
     BEGIN DS←LIT " ";  SIGN(DI←DI-1;  DS←LIT "-");                     08515000 T  0186:3
          SI←P3;  SI←SI+P1;  DS←NUMCHR CHR;                             08515100 T  0188:3
          P2(DS←2 LIT "E+";  ESIGN(DI←DI-1;  DS←LIT "-");               08515200 T  0190:0
             SI←P4;  SI←SI+8;  SI←SI-EXPCHR; DS←EXPCHR CHR);            08515300 T  0192:2
          WRITESTMT(DS←LIT ",");  P1←DI;                                08515400 T  0194:1
     END STREAM;                                                        08515500 T  0195:3
     BUFF ← POLISH;  COUNTER ← COUNTER+T;                               08515600 T  0196:0
     GO TO CONVERTED;                                                   08515700 T  0197:3
                                                                        08515800 T  0198:1
MAXINT  ::: @1045753603774000;                                          08515900 T  0198:1
MINVALUE::: @1256553762465363;                                          08516000 T  0200:0
TENSEVEN::: @1054611320000000;                                          08516100 T  0201:0
TENSIX  ::: @1063641100000000;                                          08516200 T  0202:0
CONVERTED:                                                              08516300 T  0203:0
   END CONVERT;                                                         08516400 T  0203:0
   SUBROUTINE TABCONTROL;                                               08516500 T  0203:1
   BEGIN COMMENT DOES ROUTINE TABBING OPERATIONS;                       08516600 T  0204:0
     IF TAB<0 THEN    COMMENT TAB CONTROL GIVEN EXPLICITLY;             08516700 T  0204:0
       BEGIN TAB:= (ABS(TAB)-1) MOD BUFFLOAD;                           08516800 P  0204:3
         IF FIRSTIME THEN BEGIN FIRSTIME:= FALSE;                       08516900 P  0207:1
         IF TAB LSS COUNTER THEN PRINTRETURN; END;                      08516910 P  0208:3
       BEGIN COMMENT SPACE FWD;%                                        08517000 T  0211:0
          STREAM(A:=BUFF:TAB:=TAB-COUNTER,                              08517050 T  0211:0
```

```
            T:=P(DUP).[36:6]);%                                                  08517060 T  0212:2
  BEGIN                                                                          08517100 P  0213:2
      SI:= LOC TAB; SKIP SB;                                                     08517110 C  0213:2
      IF SB THEN                                                                 08517120 C  0214:0
        BEGIN                COMMENT SPACE BACKWARD;                           08517130 C  0214:2
        SI:= A; T(SI:= SI- 32; SI:= SI - 32);                                   08517140 C  0214:2
        SI:= SI - TAB;                                                          08517150 C  0216:0
        END ELSE                                                                08517160 C  0216:2
        BEGIN                COMMENT SPACE FORWARD;                             08517170 C  0216:3
        SI:= A; T(SI:= SI + 32; SI:= SI + 32);                                  08517180 C  0216:3
        SI := SI + TAB;                                                         08517190 C  0218:1
        END;                                                                    08517200 P  0218:3
      A:= SI;                                                                   08517210 C  0218:3
          END STREAM;%                                                          08517300 T  0219:0
          BUFF:=POLISH; COUNTER:=TAB;%                                          08517400 T  0219:1
    END SPACE FWD THRU BUFFER;%                                                 08517410 T  0220:2
END ELSE BEGIN    COMMENT NORMAL TAB CONTROL FUNCTION;                          08517500 T  0220:2
    IF TAB=ENDLINE THEN PRINTEXIT;                                              08517600 T  0221:0
    IF WRITESTMT = 0 THEN BEGIN                                                 08517650 T  0223:0
    IF TAB=COMMA THEN                                                           08517700 T  0224:1
        T ← COUNTER-(COUNTER ← ((COUNTER+14) DIV 15)×15)                        08517800 T  0225:0
    ELSE IF TAB=SEMICOLON THEN                                                  08517900 T  0227:1
          T := COUNTER-(COUNTER :=((COUNTER+5) DIV 3) × 3);                     08518000 T  0230:0
        IF COUNTER>BUFFLOAD THEN PRINTRETURN ELSE                               08518100 T  0233:3
        IF TAB≠0 THEN BEGIN;                                                    08518200 T  0236:0
            STREAM(BUFF:T);                                                     08518300 T  0237:3
            BEGIN SI←BUFF;  SI←SI+T;  BUFF←SI;  END;                            08518400 T  0239:0
            BUFF ← POLISH;                                                      08518500 T  0240:1
END END END END TABCONTROL;                                                     08518600 T  0240:3
COMMENT*********START OF CODE******;                                           08518700 T  0241:0
ITEMS ← TYPE.[1:6];  WRITESTMT ← TYPE.[46:1];                                   08518800 T  0241:0
POINTER ← 1 INX ([RCW]&RCW[FTC]);                                               08518900 T  0250:3
FILX ← *POINTER;  FILX[NOT 4] ← *(1 INX POINTER);                               08518910 T  0252:2
POINTER ← 2 INX POINTER;                                                        08518920 T  0256:0
FIB ← FILX[NOT 2];                                                              08519000 T  0257:1
IF FIB[5].[43:1] THEN P(MKS, 0, 0, FILX, 1, ALGOLSELECT);                       08519100 T  0259:0
IF FIB[0]=0 THEN BEGIN FIB[0]:="1000"; THISTYPE:=TRUE; END;                     08519150 T  0262:0
DATACOM ←    FIB[4].[8:4] = 10 OR FIB[4].[8:4] = 13;                            08519175 T  0265:2
IF (COUNTER ← (T ← FIB[20]).[3:15])=0 THEN CHECKPRESENCE ELSE                   08519200 T  0269:1
BEGIN BUFF ← T.[30:18];  BUFFLOAD ← 8×(BSIZE ← T.[18:10]) END;                  08519300 T  0273:0
IF THISTYPE AND FIB[4].[8:4]=4 THEN                                             08519320 T  0278:2
  P(*[FIB[14]],7,CDC,1,SSN,XCH,STD);                                            08519330 T  0280:2
IF DATACOM THEN CLEAR                                                           08519350 T  0283:1
  ELSE IF FIB[21] NEQ 0 THEN P(FILX,8,11,COM);                                  08519360 T  0284:3
      TAB:= *POINTER; POINTER:= 1 INX POINTER;                                  08519400 P  0288:0
      FIRSTIME := TRUE; TABCONTROL;                                             08519410 C  0290:1
IF ITEMS=0 THEN SETUPANDEXIT;                                                   08519500 T  0292:0
IF NOT TYPE THEN GO TO NORMAL;                                                  08519600 T  0294:0
DO BEGIN     COMMENT MATRIX PRINT ROUTINE;                                      08519700 T  0294:3
    POLISH(MATRIX ← GETNEXT);                                                   08519800 T  0294:3
    IF P(TAB, DUP)=ENDLINE THEN TAB ← COMMA;  P(XCH);                           08519900 T  0296:2
    IF DIMENSION THEN                                                           08520000 T  0299:0
    BEGIN COL ← THISTYPE+1;  ROWLENGTH ← MATRIX.[8:10];                         08520100 T  0300:0
        DO BEGIN                                                                08520200 T  0303:1
            T ← [MATRIX[COL]];                                                  08520300 T  0303:1
            CONVERT;  TABCONTROL;                                               08520400 T  0304:1
        END UNTIL (COL ← COL+THISTYPE+1)=ROWLENGTH;                             08520500 T  0306:0
```

```
                    IF (TAB ← POLISH)=ENDLINE THEN PRINTEXIT;
               END ELSE BEGIN
                    NUMROWS ← MATRIX,[8:10];
                    ROWLENGTH ← (*[MATRIX[ROW ← 1]]),[8:10];
                    DO BEGIN
                         MATRIXROW ← *[MATRIX[ROW]];  COL ← 1;
                         DO BEGIN
                              T ← [MATRIXROW[COL]];
                              CONVERT;  TABCONTROL;
                         END UNTIL (COL ← COL+1)=ROWLENGTH;
                         IF COUNTER≠0 THEN PRINTRETURN;
                    END UNTIL (ROW ← ROW+1)=NUMROWS;
                    IF (TAB ← POLISH)=ENDLINE THEN SETUPANDEXIT;
               END;
          END UNTIL (POINTER INX 0)=[TYPE],[CF];
          SETUPANDEXIT;
NORMAL:   DO BEGIN T ← GETNEXT;
               CONVERT;  TABCONTROL;
          END UNTIL (POINTER INX 0)=[TYPE],[CF];
          SETUPANDEXIT;
END BASIC PRINT ROUTINE;
```

```
08520550 T  0308:3
08520600 T  0311:0
08520700 T  0311:2
08520800 T  0313:0
08520900 T  0315:1
08521000 T  0315:1
08521100 T  0317:1
08521200 T  0317:1
08521300 T  0318:1
08521400 T  0320:0
08521450 T  0322:1
08521500 T  0325:0
08521600 T  0327:1
08521700 T  0330:0
08521800 T  0330:0
08521900 T  0332:1
08522000 T  0333:0
08522100 T  0334:2
08522200 T  0337:0
08522300 T  0339:1
08522400 T  0340:0
                    SIZE= 0341 WORDS
```

```
PROCEDURE READATA(TYPE);
VALUE   TYPE;
REAL    TYPE;
BEGIN
   ARRAY     DATA = 21[*],
             COMPANION = 22[*];
   INTEGER   PTR = 23,
             ENDATA = 24;
   INTEGER   COL,
             COUNT,
             D,
             NUMROWS,
             R,
             ROW,
             ROWLENGTH,
             T;
   BOOLEAN   THISTYPE;
   ARRAY     MATRIX[*],
             MATRIXROW[*],
             DATAROW[*],
             COMPROW[*];
   NAME      N;
   LABEL     NORMAL;
   REAL SUBROUTINE GETNEXT;
    BEGIN COMMENT GET NEXT ITEM FROM STACK, AND DO
             SOME ROUTINE HOUSEKEEPING OPERATIONS;
        P(*(P(.TYPE)+COUNT));
        THISTYPE ← (TYPE ← 0&TYPE[6:7:40]).[6:1];
        GETNEXT ← POLISH
   END GETNEXT;
```

```
08600000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00595
08600100 T  0000:0
08600200 T  0000:0
08600300 T  0000:0
08600400 T  0000:0
08600500 T  0000:0
08600600 T  0000:0
08600700 T  0000:0
08600800 T  0000:0
08600900 T  0000:0
08601000 T  0000:0
08601100 T  0000:0
08601200 T  0000:0
08601300 T  0000:0
08601400 T  0000:0
08601500 T  0000:0
08601600 T  0000:0
08601700 T  0000:0
08601800 T  0000:0
08601900 T  0000:0
08602000 T  0000:0
08602100 T  0000:0
08602200 T  0000:0
08602300 T  0000:0
08602400 T  0001:0
08602500 T  0001:0
08602600 T  0001:0
08602700 T  0002:0
08602800 T  0004:3
08602900 T  0004:3
```

```
BOOLEAN SUBROUTINE DIMENSION;                                          08603000 T  0005:1
BEGIN COMMENT TRUE FOR SINGLE DIMENSIONED (INCLUDES STRINGS);;         08603100 T  0006:0
     STREAM(T←POLISH(XCH, 0, CDC):A←0);                                08603200 T  0006:0
     BEGIN SI←T;  DI←T;  SI←SI-16;  SKIP 2 SB;                         08603300 T  0007:3
          IF SB THEN ELSE TALLY←1;  T ← TALLY;                         08603400 T  0008:3
     END STREAM;                                                       08603500 T  0010:0
     DIMENSION ← THISTYPE OR POLISH;                                   08603600 T  0010:1
END DIMENSION;                                                         08603700 T  0011:0
SUBROUTINE PUT;                                                        08603800 T  0011:1
BEGIN COMMENT GETS AND STORES NEXT DATUM;                              08603900 T  0012:0
     IF (R×256+D)=ENDATA THEN POLISH((-48), 26, COM);                  08604000 T  0012:0
     IF DATAROW=0 THEN                                                 08604100 T  0015:1
     BEGIN DATAROW ← *[DATA[R]];                                       08604200 T  0016:1
          COMPROW ← *[COMPANION[R]];                                   08604300 T  0018:0
     END;                                                              08604400 T  0019:1
     STREAM(A←T.[43:5]:B←[COMPROW[T.[40:3]]]);                         08604500 T  0019:1
     BEGIN SI←B;  SI←SI+2;  SKIP 4 SB;  SKIP A SB;                     08604600 T  0021:3
          IF SB THEN TALLY←1;  A←TALLY;                                08604700 T  0023:0
     END STREAM;                                                       08604800 T  0024:0
     IF POLISH+THISTYPE THEN P((-44), 26, COM);                        08604900 T  0024:1
     IF THISTYPE THEN                                                  08605000 T  0026:1
     BEGIN COMMENT STRING;;                                            08605100 T  0026:2
          STREAM(S←[DATAROW[D]],N);                                    08605200 T  0027:0
          BEGIN SI←S;  DS←2 WDS;  END;                                 08605300 T  0028:1
     END ELSE COMMENT NUMERICAL STUFF (OR WE ARE IN TROUBLE);           08605400 T  0029:0
       P(DATAROW[D],[N],←);                                            08605500 T  0029:0
     IF (D ← D+THISTYPE+1)≥256 THEN                                    08605600 T  0030:2
     BEGIN COMMENT ROW OVERFLOW;                                       08605700 T  0032:3
          R := R+1;  T := D := DATAROW := 0;                           08605800 T  0033:1
     END ELSE T ← T+1;                                                 08605900 T  0036:1
END PUT;                                                               08606000 T  0038:0
SUBROUTINE EXIT;                                                       08606100 T  0038:1
BEGIN COMMENT FUTZ UP PTR AND GO BACK TO THE REAL WORLD;               08606200 T  0039:0
     PTR ← R&D[CTF]&T[9:39:9];                                         08606300 T  0039:0
     POLISH(XIT);                                                      08606400 T  0041:1
END EXIT;                                                              08606500 T  0041:2
     COMMENT**********START OF CODE**********;                        08606600 T  0041:3
     COUNT ← TYPE.[1:6];                                               08606700 T  0041:3
     R ← PTR.[CF];  D ← PTR.[FF];  T ← PTR.[9:9];                      08606800 T  0046:3
     IF NOT TYPE THEN GO TO NORMAL;                                    08606900 T  0050:2
     DO BEGIN                                                          08607000 T  0051:1
          POLISH(MATRIX ← GETNEXT);                                    08607100 T  0051:1
          IF DIMENSION THEN                                            08607200 T  0052:2
          BEGIN COL ← THISTYPE+1;  ROWLENGTH ← MATRIX.[8:10];           08607300 T  0054:0
               DO BEGIN                                                08607400 T  0057:1
               N ← [MATRIX[COL]];                                      08607500 T  0057:1
                    PUT;                                               08607600 T  0058:1
               END UNTIL (COL ← COL+THISTYPE+1)=ROWLENGTH;             08607700 T  0059:0
          END ELSE BEGIN                                               08607800 T  0061:3
               NUMROWS ← MATRIX.[8:10];                                08607900 T  0062:1
               ROWLENGTH ← (*[MATRIX[ROW ← 1]]).[8:10];                08608000 T  0063:3
               DO BEGIN                                                08608100 T  0066:0
                    MATRIXROW ← *[MATRIX[ROW]];  COL ← 1;              08608200 T  0066:0
                    DO BEGIN                                           08608300 T  0068:0
                         N ← [MATRIXROW[COL]];                         08608400 T  0068:0
                         PUT;                                          08608500 T  0069:0
                    END UNTIL (COL ← COL+1)=ROWLENGTH;                 08608600 T  0070:0
```

```
                    END UNTIL (ROW ← ROW+1)=NUMROWS;
               END;
          END UNTIL (COUNT ← COUNT-1)=0;
          EXIT;
     NORMAL:
          DO BEGIN
               N ← GETNEXT;  PUT;
          END UNTIL (COUNT ← COUNT-1)=0;
          EXIT;
     END READATA;



PROCEDURE BASICINPUT(TYPES);

VALUE   TYPES;
REAL    TYPES;

BEGIN REAL    ROW = +0,
              ALGOLREAD = 13,
              ALGOLSELECT = 14;
   ARRAY      POT = 25[*];
   INTEGER    BSIZE,
              BUFF,
              CHAR,
              COL,
              COUNT,
              COUNTER,
              DECADES,
              E,
              ESIGN,
              NUMBER,
              NUMROWS,
              ROW,
              ROWLENGTH,
              SIGN;
   BOOLEAN    GOTDIGIT,
              READSTMT,
              STOG,
              THISTYPE;
   ARRAY      FIB[*],
              MATRIX[*],
              MATRIXROW[*];
   NAME       ADDRESS,
              FILX,
              POINTER,
              STRING = ADDRESS;
   LABEL      LOOK,
              SIGNED,
              PASTPOINT,
              AT,
              EXPSIGNED,
              DECIMAL,
              ERROR,
              STRUNG,
```

```
08608700 T  0072:1
08608800 T  0074:2
08608900 T  0074:2
08609000 T  0076:3
08609100 T  0078:0
08609200 T  0078:0
08609300 T  0078:0
08609400 T  0081:0
08609500 T  0083:1
08609600 T  0084:0
          SIZE= 0085 WORDS



08700000 T  0000:0
START OF REL SEGMENT; DISK ADDRESS = 00598
08700100 T  0000:0
08700200 T  0000:0
08700300 T  0000:0
08700400 T  0000:0
08700500 T  0000:0
08700600 T  0000:0
08700700 T  0000:0
08700800 T  0000:0
08700900 T  0000:0
08701000 T  0000:0
08701100 T  0000:0
08701200 T  0000:0
08701300 T  0000:0
08701400 T  0000:0
08701500 T  0000:0
08701600 T  0000:0
08701700 T  0000:0
08701800 T  0000:0
08701900 T  0000:0
08702000 T  0000:0
08702100 T  0000:0
08702200 T  0000:0
08702250 T  0000:0
08702300 T  0000:0
08702400 T  0000:0
08702500 T  0000:0
08702600 T  0000:0
08702700 T  0000:0
08702800 T  0000:0
08702850 T  0000:0
08702900 T  0000:0
08703000 T  0000:0
08703100 T  0000:0
08703200 T  0000:0
08703300 T  0000:0
08703400 T  0000:0
08703500 T  0000:0
08703600 T  0000:0
08703700 T  0000:0
08703800 T  0000:0
```

```
          QUOTEDSTRING,                                                    08703810 T  0000:0
          SETCOUNT,                                                        08703820 T  0000:0
          NORMAL,                                                          08703900 T  0000:0
          EXIT,                                                            08704000 T  0000:0
                                                  DUMMYLABEL;              08704100 T  0000:0
REAL SUBROUTINE GETNEXT;                                                   08704200 T  0000:0
BEGIN COUNT ← COUNT-1;                                                     08704300 T  0001:0
     P(*POINTER);  POINTER ← 1 INX POINTER;                                08704400 T  0002:1
     THISTYPE ← (TYPES ← 0&TYPES[6:7:39]).[6:1];                           08704500 T  0004:0
     GETNEXT ← POLISH                                                      08704600 T  0006:3
END GETNEXT;                                                               08704700 T  0006:3
BOOLEAN SUBROUTINE DIMENSION;                                              08704800 T  0007:1
BEGIN COMMENT TRUE FOR SINGLE DIMENSIONED (INCLUDES STRINGS);;             08704900 T  0008:0
     STREAM(T←POLISH(XCH, 0, CDC):A←0);                                    08705000 T  0008:0
     BEGIN SI←T;  DI←T;  SI←SI-16;  SKIP 2 SB;                             08705100 T  0009:3
          IF SB THEN ELSE TALLY←1;  T ← TALLY;                             08705200 T  0010:3
     END STREAM;                                                           08705300 T  0012:0
     DIMENSION ← THISTYPE OR POLISH;                                       08705400 T  0012:1
END DIMENSION;                                                             08705500 T  0013:0
SUBROUTINE CHECKPRESENCE;                                                  08705600 T  0013:1
BEGIN COMMENT CALL ALGOL READ INTRINSIC TO                                 08705700 T  0014:0
          AWAIT TOP BUFFER BEING PRESENT;                                  08705800 T  0014:0
     BSIZE ← POLISH(MKS, 0, 1, FILX, ALGOLREAD);                           08705900 T  0014:0
     IF FIB[4].[8:4]=10 THEN BSIZE ← 9 ELSE BSIZE ← BSIZE-1;               08705950 T  0015:3
     BSIZE ← BSIZE×8;  BUFF ← (*FILX).[CF];                                08706000 T  0020:1
END CHECK PRESENCE BIT;                                                    08706100 T  0023:0
SUBROUTINE READIT;                                                         08706200 T  0023:1
     POLISH(MKS, 0, 0, FILX, ALGOLREAD);                                   08706300 T  0024:0
SUBROUTINE SETUPANDEXIT;                                                   08706310 T  0025:2
BEGIN FIB[21]:=BUFF&BSIZE[18:38:10]&1[29:47:1]; P(XIT); END;                08706320 T  0026:0
SUBROUTINE SCAN;                                                           08706400 T  0029:3
BEGIN COMMENT GENERAL-PURPOSE SCANNER -- CHARACTER AT A TIME;              08706500 T  0030:0
LOOK:  IF BSIZE=0 THEN BEGIN READIT;  CHECKPRESENCE;  END;                 08706600 T  0030:0
     STREAM(I←-1,BUFF,N←IF BSIZE<63 THEN BSIZE ELSE 63:STOG);              08706700 T  0033:0
     BEGIN SI←BUFF;  CI←CI+STOG;  GO TO DEBLANK;                           08706800 T  0037:0
          COMMENT SWITCH ON WHETHER WITHIN STRING OR NOT;                  08706900 T  0038:0
GNC:      TALLY←TALLY+1;  DI←LOC I;  DS←LIT "0";                           08707000 T  0038:0
          DI←DI+6;  DS←CHR;  GO TO EXIT;                                   08707100 T  0039:0
DEBLANK:  N(IF SC≠" " THEN JUMP OUT TO GNC;                                08707200 T  0039:3
               TALLY←TALLY+1;  SI←SI+1);                                   08707300 T  0041:1
EXIT:     N←TALLY;  BUFF←SI;                                               08707400 T  0042:0
     END STREAM;                                                           08707500 T  0042:2
     BSIZE ← BSIZE-P(XCH);  COMMENT UPDATE COUNT;                          08707600 T  0042:3
     BUFF ← POLISH;         COMMENT UPDATE POINTER;                       08707700 T  0044:0
     IF P(DUP)<0 THEN       COMMENT ONLY FOUND BLANKS;                    08707800 T  0044:2
     BEGIN P(DEL);                                                         08707900 T  0045:1
          IF (BSIZE=0) AND GOTDIGIT THEN P(",") ELSE GO TO LOOK;           08707920 T  0046:0
     END;                                                                  08707940 T  0048:0
     CHAR ← POLISH;                                                        08708000 T  0048:0
END SCAN ROUTINE;                                                          08708100 T  0048:2
BOOLEAN SUBROUTINE TESTCOLLECT; COMMENT PUTS CURRENT CHAR                  08708110 T  0048:3
     INTO STRING AND UPDATES CHAR COUNTER,ALSO DETECTS OVERFLOW;           08708115 T  0049:0
BEGIN                                                                      08708120 T  0049:0
     STREAM(CHAR, N:=COUNTER:=COUNTER+1, STRING);%                         08708125 T  0049:0
     BEGIN SI := LOC N; SI := SI-1;%                                       08708130 T  0051:1
          DI := DI+N; DS := CHR;%                                          08708135 T  0051:3
     END STREAM;%                                                          08708140 T  0052:2
```

```
     TESTCOLLECT:=COUNTER=15;%                                                  08708145 T  0052:3
END TESTCOLLECT;%                                                               08708150 T  0053:3
SUBROUTINE FREEREAD;                                                            08708200 T  0054:0
BEGIN COMMENT READS AND STORES NEXT DATUM, DOING APPROPRIATE                    08708300 T  0054:0
              CONVERSIONS.  HANDLES STRINGS AND NUMBERS, AND                    08708400 T  0054:0
              ACCEPTS A VARIETY OF FORMATS;                                     08708500 T  0054:0
     DECADES ← E ← ESIGN ← GOTDIGIT ← NUMBER ← STOG ← 0;  SCAN;                 08708600 T  0054:0
     IF CHAR="," THEN GO TO EXIT;                                               08708700 T  0058:0
     IF THISTYPE THEN GO TO STRUNG;                                             08708800 T  0059:1
     GOTDIGIT ← 1;                                                              08708850 T  0060:1
     IF (SIGN ← CHAR="-") OR CHAR="+" OR CHAR="&" THEN SCAN;                    08708900 T  0061:0
     IF CHAR>9 THEN GO TO DECIMAL;                                              08709000 T  0066:0
     DO BEGIN NUMBER ← 10×NUMBER+CHAR;  SCAN;                                   08709100 T  0067:1
       END UNTIL CHAR>9;                                                        08709200 T  0070:0
     IF CHAR="." THEN                                                           08709300 T  0071:1
     BEGIN SCAN;                                                                08709400 T  0072:0
PASTPOINT::  WHILE CHAR≤9 DO                                                    08709500 T  0074:0
          BEGIN NUMBER ← 10×NUMBER+CHAR;                                        08709600 T  0075:1
               DECADES ← DECADES+1;  SCAN;                                      08709700 T  0077:0
     END  END;                                                                  08709800 T  0079:0
     IF CHAR="@" OR CHAR="E" THEN                                               08709900 T  0079:2
AT:: BEGIN SCAN;                                                                08710000 T  0081:1
          IF (ESIGN ← CHAR="-") OR CHAR="+" OR CHAR="&" THEN SCAN;              08710100 T  0083:0
          IF (E ← CHAR)>9 THEN GO TO ERROR;  SCAN;                              08710200 T  0088:0
          WHILE CHAR≤9 DO BEGIN E ← 10×E+CHAR;  SCAN;  END;                     08710300 T  0091:0
          IF ESIGN THEN E ← -E;                                                 08710400 T  0095:2
     END;                                                                       08710500 T  0097:1
     WHILE CHAR≠"," DO SCAN;                                                    08710600 T  0097:1
     P(NUMBER, E-DECADES, POT[ABS(P(DUP))], XCH);                               08710700 T  0100:2
     IF P(DUP)=0 THEN P(DEL, DEL)                                               08710800 T  0102:2
       ELSE IF P<0 THEN P(/) ELSE P(×);                                         08710900 T  0104:1
     IF SIGN THEN P(CHS);  P([ADDRESS], ←);                                     08711000 T  0106:3
     GO TO EXIT;                                                                08711100 T  0108:1
DECIMAL:: IF CHAR="." THEN BEGIN SCAN;                                          08711200 T  0108:3
          IF CHAR≤9 THEN GO TO PASTPOINT ELSE GO TO ERROR;  END;                08711300 T  0111:0
     NUMBER ← 1;                                                                08711400 T  0112:3
     IF CHAR="@" OR CHAR="E" THEN GO TO AT;                                     08711500 T  0113:2
ERROR:  COMMENT ERROR TERMINATE - INVALID INPUT DATUM;%                         08711600 T  0116:0
     POLISH((-41), 26, COM);%                                                   08711610 T  0116:0
STRUNG:: COMMENT COLLECT STRING ITEM;%                                          08711700 T  0117:0
     COUNTER:=-(STOG:=1);%                                                      08711800 T  0117:0
     STREAM(STRING); DS:=16 LIT" "; % BLANK STRING                              08711900 T  0118:2
     IF CHAR ="""" THEN GO QUOTEDSTRING;%                                       08712000 T  0121:3
     WHILE (CHAR NEQ " " AND CHAR NEQ ",")                                      08712100 T  0123:0
     DO IF TESTCOLLECT THEN GO ERROR ELSE SCAN;%                                08712200 T  0124:1
     GO TO SETCOUNT;%                                                           08712210 T  0128:2
QUOTEDSTRING::                                                                  08712300 T  0129:0
     SCAN; IF CHAR="""" THEN GO SETCOUNT;%                                      08712400 T  0129:0
     IF TESTCOLLECT THEN GO ERROR ELSE GO QUOTEDSTRING;%                        08712500 T  0131:1
     COMMENT CONVERT COUNTER TO COLLATING SEQUENCE;%                            08712600 T  0133:0
SETCOUNT:                                                                       08712700 T  0133:0
     IF COUNTER LSS 0 THEN GO ERROR % NULL STRING                               08712800 T  0133:0
     ELSE IF (COUNTER:=COUNTER-5) LSS 0 THEN COUNTER:=COUNTER+20;%              08712810 T  0133:3
     COMMENT PUT IN CHAR COUNT REQUIRED BY BASIC STRINGVARB;%                   08712900 T  0137:3
     STREAM(COUNTER,STRING);%                                                   08712920 T  0137:3
     BEGIN SI:=LOC STRING; SI:=SI-1;                                            08713000 T  0138:3
          DI:=DI+15; DS:=CHR;%                                                  08713010 T  0139:1
```

```
     END STREAM;%                                                                    08713020 T  0139:3
     GOTDIGIT:=1; STOG:=0;%                                                          08713100 T  0140:0
     IF CHAR NEQ "," THEN DO SCAN UNTIL CHAR=",";%                                   08713110 T  0141:2
EXIT:;                                                                               08713200 T  0145:1
END FREE FIELD READ ROUTINE;                                                         08713300 T  0145:1
     COUNT ← TYPES.[1:6];  READSTMT ← TYPES.[46:1];                                  08713400 T  0146:1
     FILX ← *(POINTER ← 1 INX ([RCW]&RCW[FTC]));                                     08713500 T  0155:2
     FILX[NOT 4] ← *(1 INX POINTER);  FIB ← FILX[NOT 2];                             08713510 T  0158:0
     POINTER ← 2 INX POINTER;                                                        08713520 T  0162:1
     IF FIB[5].[43:2]≠2 THEN P(MKS, 0, 2, FILX, 1, ALGOLSELECT);                     08713600 T  0163:2
     IF (E:=FIB[4].[8:4])=10 OR E=13 THEN FIB[20]:=0                                 08713610 T  0167:0
       ELSE IF FIB[20] NEQ 0 THEN P(FILX,8,11,COM);                                  08713620 T  0171:0
     IF (BUFF ← (E ← FIB[21]).[30:18]) = 0                                           08713700 T  0174:3
       THEN CHECKPRESENCE ELSE BSIZE ← E.[18:10];                                    08713710 T  0176:3
     IF NOT TYPES THEN GO TO NORMAL;                                                 08713800 T  0180:3
     DO BEGIN                                                                        08713900 T  0181:2
          POLISH(MATRIX ← GETNEXT);                                                  08714000 T  0181:2
          IF DIMENSION THEN                                                          08714100 T  0183:2
          BEGIN COL ← THISTYPE+1;  ROWLENGTH ← MATRIX.[8:10];                        08714200 T  0185:0
               DO BEGIN                                                              08714300 T  0188:1
                    ADDRESS ← [MATRIX[COL]];                                         08714400 T  0188:1
                    FREEREAD;                                                        08714500 T  0189:1
               END UNTIL (COL ← COL+THISTYPE+1)=ROWLENGTH;                           08714600 T  0190:0
          END ELSE BEGIN                                                             08714700 T  0192:3
               NUMROWS ← MATRIX.[8:10];                                              08714800 T  0193:1
               ROWLENGTH ← (*[MATRIX[ROW ← 1]]).[8:10];                              08714900 T  0194:3
               DO BEGIN                                                              08715000 T  0197:0
                    MATRIXROW ← *[MATRIX[ROW]];  COL ← 1;                            08715100 T  0197:0
                    DO BEGIN                                                         08715200 T  0199:0
                         ADDRESS ← [MATRIXROW[COL]];                                 08715300 T  0199:0
                         FREEREAD;                                                   08715400 T  0200:0
                    END UNTIL (COL ← COL+1)=ROWLENGTH;                               08715500 T  0201:0
               END UNTIL (ROW ← ROW+1)=NUMROWS;                                      08715600 T  0203:1
          END;                                                                       08715700 T  0205:2
     END UNTIL COUNT=0;                                                              08715800 T  0205:2
     SETUPANDEXIT;                                                                   08715900 T  0206:3
NORMAL:                                                                              08716000 T  0208:0
     DO BEGIN                                                                        08716100 T  0208:0
          ADDRESS ← GETNEXT;                                                         08716200 T  0208:0
          FREEREAD;                                                                  08716300 T  0209:2
     END UNTIL COUNT=0;                                                              08716400 T  0211:0
     SETUPANDEXIT;                                                                   08716490 T  0212:1
END BASIC INPUT ROUTINE;                                                             08716500 T  0213:0
                                                                                                SIZE= 0214 WORDS
```

```
%*************************************************************************        08800000 T  0000:0
PROCEDURE MATRIXDIDDLER(A, B, C, TYPE);% MAT ARITH INTRINSIC                        08800050 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00606
VALUE     A, % RESULTANT MOM                                                        08800100 T  0000:0
          B, % ARG 1 MOM OR SCALAR VALUE                                            08800110 T  0000:0
          C; % ARG 2 MOM                                                            08800120 T  0000:0
 ARRAY   A[*], B[*], C[*];                                                          08800200 T  0000:0
 INTEGER TYPE;                                                                      08800300 T  0000:0
%*************************************************************************        08800310 T  0000:0
```

```
        BEGIN REAL   SCALE = B;                                                  08800400 T  0000:0
      INTEGER   I,                                                              08800500 T  0000:0
                J,                                                              08800600 T  0000:0
                LASTI,                                                          08800700 T  0000:0
                LASTJ;                                                          08800800 T  0000:0
      ARRAY     AROW[*],                                                        08800900 T  0000:0
                BROW[*],                                                        08801000 T  0000:0
                CROW[*];                                                        08801100 T  0000:0
      BOOLEAN   SINGLE;                                                         08801200 T  0000:0
      LABEL     ERROR,                                                          08801300 T  0000:0
                DIMERR,  %%                                                     08801350 T  0000:0
                CHKSI,                                                          08801355 T  0000:0
                CHKSIZE,                                                        08801360 T  0000:0
                NORMAL,                                                         08801400 T  0000:0
                SCALEFACTOR;                                                    08801500 T  0000:0
DEFINE      SF = [8:10]#;%                                                      08801600 T  0000:0
BOOLEAN SUBROUTINE DIMENSION; %                                                 08801700 T  0000:0
     BEGIN COMMENT TRUE FOR SINGLY-DIMENSIONED MATRIX;;                         08801800 T  0001:0
         STREAM(T:=P(XCH, 0, COC):A:=0);                                        08801900 T  0001:0
      BEGIN SI:=T; DI:=T; SI:=SI-16; SKIP 2 SB;%%                               08802000 T  0002:3
          IF SB THEN ELSE TALLY:=1; T:=TALLY;                                   08802100 T  0003:3
      END STREAM;                                                               08802200 T  0005:0
          DIMENSION := POLISH;                                                  08802300 T  0005:1
      END DIMENSION;                                                            08802400 T  0005:2
COMMENT  * * * * * * * * * * * START OF CODE * * * * * * * * * * *;             08802500 T  0005:3
          I := J:= 1;  POLISH(CROW := C);%                                      08802600 T  0005:3
          IF (SINGLE := DIMENSION) THEN                                         08802700 T  0010:1
      BEGIN COMMENT ROW VECTOR CASE;%                                           08802800 T  0011:2
          LASTI := 2;  LASTJ := C.SF;%                                          08802850 T  0012:0
          IF NOT POLISH(AROW :=A, DIMENSION) THEN GO ERROR; %                   08802900 T  0014:1
          IF TYPE = 2 THEN GO CHKSI; %                                          08803000 T  0016:2
          IF POLISH(BROW := B, DIMENSION) THEN                                  08803050 T  0017:3
          IF LASTJ = BROW.SF THEN GO CHKSI;%                                    08803100 T  0020:0
ERROR:    COMMENT NON CONFORMAL ARGUMENTS;                                      08803200 T  0022:2
          POLISH((-50), 26, COM); %                                             08803300 T  0022:2
CHKSI:    COMMENT CHECK DIMESION BOUNDS;                                        08803310 T  0023:2
          IF LASTJ GTR A.SF THEN                                                08803400 T  0023:2
DIMERR:   COMMENT DIMESION SIZE ERROR;                                          08803500 T  0025:0
          POLISH((-72), 26, COM);%                                              08803510 T  0025:2
     END ROW VECTOR CASE ELSE                                                   08803600 T  0026:2
     BEGIN COMMENT MATRIX CASE;%                                                08803610 T  0026:2
          LASTI := C.SF;  LASTJ := (*[C[1]]).SF;%                               08803700 T  0027:0
          IF POLISH(A, DIMENSION) THEN GO ERROR;%                               08803800 T  0030:1
          IF TYPE = 2 THEN GO CHKSIZE;%                                         08803900 T  0032:3
          IF NOT POLISH(B,DIMENSION) THEN                                       08804000 T  0034:0
          IF LASTI = B.SF THEN                                                  08804100 T  0036:1
          IF LASTJ = (*[B[1]]).SF THEN GO CHKSIZE;%                             08804200 T  0038:1
          GO TO ERROR;%                                                         08804300 T  0041:0
CHKSIZE:   COMMENT CHEK DIMENSION BOUNDS;%                                      08804310 T  0041:2
          IF LASTI GTR A.SF OR                                                  08804400 T  0041:2
          LASTJ GTR (*[A[1]]).[8:10] THEN  %%                                   08804500 T  0043:0
          GO TO DIMERR;  %%                                                     08804600 T  0045:0
     END MATRIX CASE;%                                                          08804700 T  0045:3
          IF TYPE = 2 THEN GO TO SCALEFACTOR;  %%                               08804800 T  0045:3
NORMAL::                                                                        08804900 T  0047:0
DO BEGIN                                                                        08805000 T  0047:0
          IF NOT SINGLE THEN                                                    08805100 T  0047:0
```

```
   BEGIN AROW := *[A[I]];  BROW := *[B[I]];                                  08805200 T  0047:2
         CROW := *[C[I]];  J := 1;                                           08805300 T  0050:2
   END;                                                                      08805400 T  0052:2
         IF TYPE=0 THEN                                                      08805500 T  0052:2
::         DO AROW[J] := BROW[J]+P(CROW[J], XCH)                            08805600 T  0053:1
         UNTIL (J := J+1)=LASTJ                                              08805700 T  0055:3
         ELSE                                                                08805800 T  0057:3
::         DO AROW[J] := CROW[J]-P(BROW[J], XCH)                            08805900 T  0058:3
         UNTIL (J := J+1)=LASTJ;                                             08806000 T  0061:3
            COMMENT NOTE FANCY WAY OF SAYING A=B-C;                          08806100 T  0064:3
   END UNTIL (I := I+1)=LASTI;                                               08806200 T  0064:3
         P(XIT);                                                             08806300 T  0067:0
SCALEFACTOR::                                                                08806400 T  0067:1
DO BEGIN                                                                     08806500 T  0067:1
         IF NOT SINGLE THEN                                                  08806600 T  0068:0
   BEGIN AROW := *[A[I]];  CROW := *[C[I]]; END;  %%                         08806700 T  0068:2
         IF SCALE = 1 THEN  %%                                               08806800 T  0071:2
   BEGIN; STREAM(F:=[CROW[1]],N:=LASTJ-1,                                    08806900 T  0072:1
         T:=P(DUP).[36:6],DES := [AROW[1]]);  %%                             08807000 T  0074:1
   BEGIN SI:=F; T(DS:=32 WDS; DS:=32 WDS); %%                                08807100 T  0075:3
         DS  := N WDS;  %%                                                   08807200 T  0077:1
   END STREAM;  %%                                                           08807300 T  0077:3
   END ELSE  %%                                                              08807400 T  0078:0
   BEGIN J := 1;  %%                                                         08807500 T  0078:0
::            DO AROW[J] := CROW[J]×P(SCALE, NOP, XCH)  %%                 08807600 T  0079:1
         UNTIL (J := J+1)=LASTJ;  %%                                         08807700 T  0081:3
   END;  %%                                                                  08807800 T  0084:3
   END UNTIL (I := I+1)=LASTI;  %%                                           08807900 T  0084:3
   END MATRIX DIDDLER;  %%                                                   08808000 T  0087:0
                                                                                    SIZE= 0088 WORDS
```

```
%*********************************************************                  08900000 T  0000:0
PROCEDURE TRANSPOSE(A,B);   %%%% MATRIX TRANSPOSE %%%%                       08900010 T  0000:0
                                                                  START OF REL SEGMENT; DISK ADDRESS = 00609
VALUE A,B;                        %%%%     INTRINSIC    %%%%                 08900100 T  0000:0
ARRAY A[*], % MOM DESC FOR RESULTANT MATRIX OR ROW VECTOR                    08900200 T  0000:0
      B[*]; % MOM DESC FOR ARGUMENT  "                    "                 08900210 T  0000:0
%*********************************************************                  08900220 T  0000:0
BEGIN LABEL ERR50,ERR72,NORMAL,PLACE,TRANSPOSEIT;%                           08900300 T  0000:0
    INTEGER I,J,LASTI,LASTJ;%                                                08900320 T  0000:0
    ARRAY ROW[*];%                                                           08900400 T  0000:0
    DEFINE SF=[8:10]#;%                                                      08900405 T  0000:0
    COMMENT THERE ARE THREE SPECIES OF MATRIX TRANSPOSITION;                 08900420 T  0000:0
    %     1.    ROW INTO COLUMN                                              08900430 T  0000:0
    %     2.    COLUMN INTO ROW                                              08900440 T  0000:0
    %     3.    MATRIX INTO MATRIX                                           08900450 T  0000:0
    %           IN THIS CASE TRANSPOSITION MAY BE DONE IN PLACE              08900500 T  0000:0
    COMMENT TRANPOSITION WILL BE PERFORMED WHEN THE RESULTANT                08900501 T  0000:0
    MATRIX DIMENSIONS ARE LARGE ENOUGH TO ACCOMMODATE                        08900502 T  0000:0
    THE DIMENSIONS OF THE ARGUMENT MATRIX,EVEN THO THE MATRICES              08900503 T  0000:0
    MAY NOT BE MATHEMATICALLY CONFORMABLE;%                                  08900504 T  0000:0
                                                                             08900505 T  0000:0
    BOOLEAN SUBROUTINE DIMENSION;                                            08900510 T  0000:0
    BEGIN COMMENT TRUE IF SINGLY DIMENSIONED;;                               08900515 T  0001:0
```

```
          STREAM(T:=POLISH(XCH, 0, CDC):A:=0);                                    08900520 T  0001:0
          BEGIN SI:=T; DI:=T; SI:=SI-16; SKIP 2 SB;                               08900525 T  0002:3
               IF SB THEN ELSE TALLY:=1; T:=TALLY;                                08900530 T  0003:3
          END STREAM;                                                             08900535 T  0005:0
          DIMENSION := POLISH;                                                    08900540 T  0005:1
     END;                                                                         08900545 T  0005:2
%************ START OF CODE *************                                        08900549 T  0005:3
     POLISH(A); IF DIMENSION THEN%                                                08900550 T  0005:3
     BEGIN COMMENT 1-DIM RESULTANT,ALLOW 2, ONLY;                                 08900555 T  0009:0
          POLISH(B); IF DIMENSION THEN % ERROR - ROW TO ROW                       08900560 T  0009:2
          BEGIN ERR50:P((-50),26,COM);%                                           08900600 T  0011:0
          ERR72: P((-72),26,COM);%                                                08900605 T  0012:2
          END ERRORS;%                                                            08900607 T  0013:2
          IF (LASTI:=(*[B[1]]),SF) NEQ 2 THEN GO ERR50;%                          08900610 T  0013:2
          IF (LASTJ:=B.SF) GTR A.SF THEN GO ERR72;%                               08900630 T  0016:1
          I:=1; ROW:=A;%                                                          08900650 T  0019:2
          GO TRANSPOSEIT;%                                                        08900700 T  0021:1
     END 1 DIM RESULTANT ELSE                                                     08900710 T  0021:3
     BEGIN COMMENT 2-DIM RESULTANT, ALLOW 1, OR 3,;                               08900750 T  0021:3
          POLISH(B); IF NOT DIMENSION THEN GO NORMAL;%                            08900800 T  0022:1
          IF (*[A[1]]),SF NEQ 2 THEN GO ERR50;%                                   08900810 T  0024:2
          IF (LASTJ:=B.SF) GTR A.SF THEN GO ERR72;%                               08900830 T  0026:3
     ::   DO POLISH(B[J],*[A[J]],1,CDC,STD) UNTIL (J:=J+1)=LASTJ;%                08900850 T  0030:0
          POLISH(XIT);%                                                           08900870 T  0034:1
     END 2 DIM RESULTANT;%                                                        08900890 T  0034:2
     NORMAL:                                                                      08900900 T  0034:2
          IF (LASTI:=(*[B[1]]),SF) GTR A.SF                                       08901000 T  0034:2
          OR (LASTJ:=B.SF) GTR(*[A[1]]).SF THEN GO ERR72;%                        08901010 T  0036:1
          I:=1; IF A.[FF]=B.[FF] THEN GO PLACE; % TRN IN PLACE                    08901020 T  0041:2
     ::   DO BEGIN ROW:=*[A[I]];%                                                 08901100 T  0045:0
     TRANSPOSEIT: J:=1;%                                                          08901110 T  0046:1
     ::        DO P(*[B[J]], I, CDC, [ROW[J]], ←) UNTIL (J ← J+1)=LASTJ;          08901200 T  0047:0
          END UNTIL (I ← I+1)=LASTI;                                              08901300 T  0051:1
          POLISH(XIT);                                                            08901400 T  0053:2
 PLACE:: IF LASTI=2 THEN POLISH(XIT);                                             08901500 T  0053:3
     ::   DO BEGIN ROW ← *[A[I]];  J ← I+1;                                       08901600 T  0055:2
     ::        DO P(ROW[J], *[A[J]], I, CDC, DUP, LOD, [ROW[J]], ←, ←)           08901700 T  0058:2
               UNTIL (J ← J+1)=LASTJ;                                             08901800 T  0062:1
          END UNTIL (I ← I+1)=LASTI-1;                                            08901900 T  0064:2
     END TRANSPOSE ROUTINE;                                                       08902000 T  0067:1
                                                                                        SIZE= 0068 WORDS
```

```
%*************************************************************************      09000000 T  0000:0
PROCEDURE MATRIXMULTIPLY(A,B,C); %%% MATRIX MULTIPLICATION %%%                   09000100 T  0000:0
                                                    START OF REL SEGMENT; DISK ADDRESS = 00612
VALUE A,B,C;                                  %%% INTRINSIC %%%                 09000200 T  0000:0
ARRAY A[*], % RESULTANT MAT OR ROW VECTOR MOM                                   09000300 T  0000:0
      B[*], % ARG-1          "                                                  09000400 T  0000:0
      C[*]; % ARG-2          "                                                  09000500 T  0000:0
%*************************************************************************      09000600 T  0000:0
BEGIN                                                                           09000700 T  0000:0
     LABEL ERR50,ERR72,DOTPRODUCT,CROSSPRODUCT;%                                09000800 T  0000:0
     ARRAY AROW[*], %                                                           09000900 T  0000:0
           BROW[*], %                                                           09001000 T  0000:0
```

```
          CROW[*];%                                                                    09001100 T  0000:0
    INTEGER I,J,K,LASTI,LASTJ,LASTK;%                                                  09001200 T  0000:0
    DEFINE SF = [8:10]#, X = BROW#;%                                                   09001300 T  0000:0
    DEFINE SETTOSAVEBIT(SETTOSAVEBIT1)=SI:=SETTOSAVEBIT1;                              09001400 T  0000:0
        SI:=SI-16; SKIP 2 SB; IF SB THEN DS:=SET ELSE DS:=RESET#;                      09001500 T  0000:0
    LABEL RRM,MRM,MMR,MMM; %                                                           09001600 T  0000:0
    SWITCH SWL :=ERR50,RRM,ERR72,ERR72,ERR50,                                          09001700 T  0000:0
    MRM,MMR,MMM;%                                                                      09001800 T  0000:0
REAL SUBROUTINE DIMENSIONS;                                                            09001900 T  0000:0
BEGIN COMMENT SETS BIT IN SWVAL TO INDICATE MAT;%                                      09002000 T  0001:0
    STREAM(SWVAL:=0:A1:=[A[0]],B1:=[B[0]],C1:=[C[0]]);%                                09002100 T  0001:0
    BEGIN DI:=LOC SWVAL; SKIP 45 DB;%                                                  09002200 T  0003:2
        SETTOSAVEBIT(A1); SETTOSAVEBIT(B1); SETTOSAVEBIT(C1);                          09002300 T  0004:0
    END STREAM;%                                                                       09002400 T  0010:0
    DIMENSIONS:=POLISH;%                                                               09002500 T  0010:1
END DIMENSIONS;%                                                                       09002600 T  0010:2
BOOLEAN SUBROUTINE DIMERR;%                                                            09002610 T  0010:3
    DIMERR:=(LASTI GTR A.SF) OR (LASTJ GTR AROW.SF);%                                  09002620 T  0011:0
COMMENT ********** START OF CODE **************;                                       09002700 T  0014:3
    LASTI:=B.SF; LASTJ:=(CROW:=*[C[1]]).SF; %                                          09002710 T  0014:3
    LASTK:=C.SF; AROW:=*[A[I:=1]]; BROW:=*[B[1]];%                                     09002800 T  0021:0
    GO TO SWL[DIMENSIONS];%                                                            09002900 T  0025:2
MMM: COMMENT A(A1,A2)=B(B1,B2)*C(C1,C2);%                                              09003000 T  0031:3
    IF LASTK NEQ BROW.SF THEN                                                          09003100 T  0031:3
ERR50: COMMENT NON-CONFORMAL ARGUMENT MATRICES;%                                       09003200 T  0033:1
    POLISH((-50),26,COM);%                                                             09003300 T  0033:3
    IF DIMERR THEN                                                                     09003400 T  0034:3
ERR72: COMMENT RESULTANT BOUNDS TOO SMALL - DIM ERR;%                                  09003600 T  0036:0
    POLISH((-72),26,COM);%                                                             09003700 T  0036:2
    IF LASTK=2 THEN GO CROSSPRODUCT ELSE GO DOTPRODUCT;%                               09003800 T  0037:2
::  DO BEGIN                                                                           09004000 T  0039:1
        BROW:=*[B[I]]; AROW:=*[A[I]];%                                                 09004100 T  0040:0
    DOTPRODUCT: J:=1;%                                                                 09004200 T  0042:2
    ::   DO BEGIN                                                                      09004300 T  0043:1
            K:=1; POLISH(0);%                                                          09004400 T  0044:0
         ::   DO P(*[C[K]],J,COC,BROW[K],MUL,ADD)                                      09004500 T  0045:0
            UNTIL (K:=K+1)=LASTK;%                                                     09004600 T  0047:1
            POLISH( [AROW[J]], STD);%                                                  09004700 T  0049:2
        END UNTIL (J:=J+1)=LASTJ;%                                                     09004800 T  0050:1
    END UNTIL (I:=I+1)=LASTI;%                                                         09004900 T  0052:2
    POLISH(XIT);%                                                                      09005000 T  0054:3
RRM: COMMENT A(A1)=B(B1)*C(C1,C2);%                                                    09005100 T  0055:0
    AROW:=A;                %                                                          09005200 T  0055:0
MRM: COMMENT A(A1,A2)=B(B1)*C(C1,C2);%                                                 09005400 T  0056:0
    IF LASTK NEQ (BROW:=B).SF THEN GO ERR50; %                                         09005700 T  0056:0
    LASTI:=2; IF DIMERR THEN GO TO ERR72;%                                             09005800 T  0058:2
    GO TO DOTPRODUCT;%                                                                 09005900 T  0060:3
MMR: COMMENT A(A1,A2)=B(B1,1)*C(C1);%                                                  09006100 T  0061:1
    IF BROW.SF NEQ 2 THEN GO TO ERR50;%                                                09006200 T  0061:1
    LASTJ:=LASTK; IF DIMERR THEN GO TO ERR72;%                                         09006300 T  0063:1
    CROW:=C;%                                                                          09006600 T  0065:3
CROSSPRODUCT::                                                                         09006700 T  0066:3
    DO BEGIN                                                                           09006800 T  0066:3
        AROW:=*[A[I]]; J:=1; %                                                         09006900 T  0067:0
        X:=POLISH(*[B[I]],1,COC);%                                                     09007000 T  0069:0
    ::   DO P(X,CROW[J],MUL,[AROW[J]],STD)                                             09007100 T  0070:3
        UNTIL(J:=J+1)=LASTJ;%                                                          09007200 T  0073:0
```

```
    END UNTIL(I:=I+1)=LASTI;%                                                                    09007300 T  0075:1
END MATRIXMULTIPLY;%                                                                             09007400 T  0077:2
                                                                                                          SIZE= 0078 WORDS



PROCEDURE INVERT(A, B);                                                                          09100000 T  0000:0
                                                                                        START OF REL SEGMENT; DISK ADDRESS = 00615
VALUE   A, B;                                                                                    09100100 T  0000:0
ARRAY   A[*], B[*];                                                                              09100200 T  0000:0
BEGIN REAL   BIG,                                                                                09100300 T  0000:0
             DIAG = BIG;                                                                         09100400 T  0000:0
   DEFINE    EPS = COMMENT 10@-13;  0.0000000000001#;                                          09100500 T  0000:0
   INTEGER   I,                                                                                  09100600 T  0000:0
             II,                                                                                 09100700 T  0000:0
             J,                                                                                  09100800 T  0000:0
             K,                                                                                  09100900 T  0000:0
             K2,                                                                                 09101000 T  0000:0
             L,                                                                                  09101100 T  0000:0
             N,                                                                                  09101200 T  0000:0
             N1;                                                                                 09101300 T  0000:0
   REAL      BLOCKCOUNTER = 16,                                                                  09101400 T  0000:0
             BLOCKROUTINE =  5;                                                                  09101500 T  0000:0
   ARRAY     AROW[*],                                                                            09101600 T  0000:0
             COPY[*],                                                                            09101700 T  0000:0
             PLACEHOLDER[*];                                                                     09101800 T  0000:0
   DEFINE    MOVEWORDS = N1(DS←32 WDS;  DS←32WDS));  DS←N WDS#;                                  09101900 T  0000:0
SUBROUTINE SWAPROWS;                                                                             09102000 T  0000:0
BEGIN;STREAM(A←*[A[I]],B←*[A[K2]],N←N+1,N1←P(DUP),[36:6],COPY);                                   09102100 T  0001:0
     BEGIN SI←A;  MOVEWORDS;                                                                     09102200 T  0005:0
           SI←B;  DI←A;  MOVEWORDS;                                                              09102300 T  0007:0
           SI←COPY;  DI←B;  MOVEWORDS;                                                           09102400 T  0009:1
END  END  SWAPROWS;                                                                              09102500 T  0011:2
     ;STREAM(T:=[A[0]]:T1:=[B[0]]);                                                              09102505 T  0012:0
     BEGIN SI:=T; DI:=T; SI:=SI-16; SKIP 2 SB;                                                   09102510 T  0016:3
           IF SB THEN                                                                            09102515 T  0017:3
           BEGIN SI:=T1; DI:=T1; SI:=SI-16; SKIP 2 SB;                                           09102520 T  0018:1
                 IF SB THEN ELSE TALLY:=1;                                                       09102525 T  0019:1
           END ELSE TALLY:=1;                                                                    09102530 T  0020:1
           T:=TALLY;                                                                             09102535 T  0020:3
     END STREAM;                                                                                 09102540 T  0021:0
     IF POLISH THEN POLISH((-50), 26, COM);                                                      09102545 T  0021:1
     IF (N ← A.[8:10])≠(*[A[1]]).[8:10] THEN P((-54), 26, COM);                                  09102600 T  0022:3
     N1 ← (N ← N-1)-1;                                                                           09102700 T  0027:1
     IF A.[FF]≠B.[FF] THEN                                                                       09102800 T  0029:2
     BEGIN IF B.[8:10]≠N+1 OR (*[B[1]]).[8:10]≠N+1                                               09102900 T  0031:3
       THEN POLISH((-50), 26, COM);                                                              09103000 T  0035:3
           IF N1=0 THEN P(*[B[1]], 1, COC, *[A[1]], 1, CDC, ←) ELSE                              09103100 T  0038:1
       FOR I←1 STEP 1 UNTIL N DO                                                                 09103200 T  0042:1
           STREAM(N, N1←P(DUP).[36:6], S←*[B[I]], D←*[A[I]]);                                    09103300 T  0044:0
           BEGIN SI←S; SI←SI+8; DI←D; DI←DI+8; MOVEWORDS; END;                                   09103400 T  0047:0
     END;                                                                                        09103500 T  0052:1
     IF N=0 THEN                                                                                 09103600 T  0052:1
     POLISH(*[A[1]], 1, CDC, DUP, LOD, 1, XCH, /, XCH, ←, XIT);                                  09103700 T  0053:0
     BLOCKCOUNTER ← BLOCKCOUNTER+1;                                                              09103800 T  0056:3
     POLISH(MKS, [PLACEHOLDER[P]], N+1, 1, 1, 0, BLOCKROUTINE);                                  09103900 T  0058:0
```

```
     BLOCKCOUNTER ← BLOCKCOUNTER+1;                                                  09104000 T  0060:1
     POLISH(MKS, [COPY[P]], N+1,1,1,1,BLOCKROUTINE);                                 09104100 T  0061:2
COMMENT REDUCE THE MATRIX BY ROW PIVOTS TO TRI-DIAGONAL FORM;                        09104200 T  0063:3
     FOR I←1 STEP 1 UNTIL N DO                                                       09104300 T  0063:3
     BEGIN II ← (K2 ← I)-1;  BIG ← 0;                                                09104400 T  0065:0
          FOR J←I STEP 1 UNTIL N DO                                                  09104500 T  0067:2
          BEGIN AROW ← *[A[J]];  POLISH(0);                                          09104600 T  0069:0
               FOR K←1 STEP 1 UNTIL II DO                                            09104700 T  0070:2
                 POLISH(*[A[K]], I, COC, AROW[K], ×, -);                             09104800 T  0072:0
               POLISH(AROW[I], +);                                                   09104900 T  0076:2
               IF POLISH([AROW[I]], SND, SSP, DUP)>BIG THEN                          09105000 T  0077:1
                 POLISH(.BIG, ←, J, .K2, ←) ELSE P(DEL);                             09105100 T  0079:0
          END;                                                                       09105200 T  0081:2
          IF BIG≤EPS THEN POLISH((-57), 26, COM);%  NEARLY SINGULAR                  09105300 T  0083:3
          IF (PLACEHOLDER[I] ← K2)≠I THEN SWAPROWS;                                  09105400 T  0086:0
          DIAG ← POLISH((AROW ← *[A[I]]), I, COC);                                   09105500 T  0089:0
          FOR J←I+1 STEP 1 UNTIL N DO                                                09105600 T  0091:1
          BEGIN POLISH(0);                                                           09105700 T  0095:2
               FOR K←1 STEP 1 UNTIL II DO                                            09105800 T  0095:3
                 POLISH(*[A[K]], J, COC, AROW[K], ×, -);                             09105900 T  0098:0
               POLISH(AROW[J], +, DIAG, /, [AROW[J]], ←);                            09106000 T  0102:2
     END  END;                                                                       09106100 T  0104:2
     POLISH(10, COM);     COMMENT RETURN COPY ARRAY;                                 09106200 T  0107:1
COMMENT INVERT LOWER TRIANGULAR MATRIX;                                              09106300 T  0107:3
     FOR I←1 STEP 1 UNTIL N DO                                                       09106400 T  0107:3
     BEGIN II ← I-1;  DIAG ← POLISH((AROW ← *[A[I]]), I, COC);                       09106500 T  0109:0
          FOR J←1 STEP 1 UNTIL II DO                                                 09106600 T  0112:2
          BEGIN POLISH(0);                                                           09106700 T  0114:0
               FOR K←J STEP 1 UNTIL II DO                                            09106800 T  0114:1
                 POLISH(*[A[K]], J, COC, AROW[K], ×, -);                             09106900 T  0115:0
               POLISH(DIAG, /, [AROW[J]], ←);                                        09107000 T  0119:2
          END;                                                                       09107100 T  0120:3
          AROW[I] ← 1.0/DIAG;                                                        09107200 T  0123:0
     END;                                                                            09107300 T  0124:3
COMMENT INVERT UPPER TRIANGULAR MATRIX;                                              09107400 T  0127:0
     FOR I←N1 STEP -1 UNTIL 1 DO                                                     09107500 T  0127:0
     BEGIN II ← I+1;  AROW ← *[A[I]];                                                09107600 T  0129:0
          FOR J←N STEP -1 UNTIL II DO                                                09107700 T  0131:2
          BEGIN L ← J-1;  POLISH(0);                                                 09107800 T  0133:0
               FOR K←II STEP 1 UNTIL L DO                                            09107900 T  0134:2
                 POLISH(*[A[K]], J, COC, AROW[K], ×, -);                             09108000 T  0136:0
               POLISH(AROW[J], CHS, +, [AROW[J]], ←);                                09108100 T  0140:2
     END  END;                                                                       09108200 T  0142:1
COMMENT MULTIPLY UPPER AND LOWER HALVES TO PRODUCE INVERSE;                          09108300 T  0146:3
     FOR I←1 STEP 1 UNTIL N1 DO                                                      09108400 T  0146:3
     BEGIN AROW ← *[A[I]];                                                           09108500 T  0148:0
          FOR J←1 STEP 1 UNTIL N DO                                                  09108600 T  0149:1
          BEGIN IF (K2 ← J)≤I THEN K2 ← I+1;  POLISH(0);                             09108700 T  0150:0
               FOR K←K2 STEP 1 UNTIL N DO                                            09108800 T  0153:1
                 POLISH(*[A[K]], J, COC, AROW[K], ×, +);                             09108900 T  0154:0
               IF I≥J THEN POLISH(AROW[J], +);                                       09109000 T  0158:2
               POLISH([AROW[J]], ←);                                                 09109100 T  0160:2
     END  END;                                                                       09109200 T  0161:1
COMMENT EXCHANGE COLUMN ELEMENTS TO ABSOLVE ROW PIVOTING;                            09109300 T  0165:3
     FOR J←N STEP -1 UNTIL 1 DO                                                      09109400 T  0165:3
     IF (I ← PLACEHOLDER[J])≠J THEN                                                  09109500 T  0167:0
     FOR K←1 STEP 1 UNTIL N DO                                                       09109600 T  0168:2
```

```
        BEGIN AROW ← *[A[K]];                                                    09109700 T  0170:0
            POLISH(AROW[I], [AROW[J]], DUP, LOD, [AROW[I]], ←, ←);               09109800 T  0171:1
        END;                                                                     09109900 T  0173:3
        POLISH(10, COM);    COMMENT RETURN PLACEHOLDER ARRAY;                    09110000 T  0178:1
    END INVERT;                                                                  09110100 T  0178:3
                                                                                        SIZE= 0179 WORDS
```

```
PROCEDURE FORTRANFREEREAD;                                                       09200000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00621
BEGIN REAL    PARL        = -1,     % PARITY "LABEL WORD"                      09200100 T  0000:0
              EOFL        = -2,     % END-OF-FILE "LABEL WORD"                 09200200 T  0000:0
              LISX        = -3,     % ACCIDENTAL ENTRY FOR LIST                09200300 T  0000:0
              DKADR       = -4;     % DISK ADDRESS                             09200400 T  0000:0
  NAME        FILX        = -5;     % FILE TANK DESCRIPTOR                      09200500 T  0000:0
  REAL        BLOCK       =  5,     % INTRINSIC INTRINSIC DESCRIPTOR            09200600 T  0000:0
              ALGOLREAD   = 13,     % NORMAL-STATE I/O INTRINSIC                09200700 T  0000:0
              SELECT      = 14,     % FILE STATUS INTRINSIC                     09200800 T  0000:0
              JUNK        = 17,     % ANOTHER TEMPORARY                         09200900 T  0000:0
              ARRAYSTUFF  = 18,     % USED BY LIST FOR ARRAYS                   09201000 T  0000:0
              LSTRN       = 19,     % INTERNAL LIST POINTER                     09201100 T  0000:0
              LISTYPE     = 20,     % TELLS TYPE OF LIST ITEM                   09201200 T  0000:0
              HOLTOG      = 21;     % FOR CHARACTER TRANSLATION                 09201300 T  0000:0
  ARRAY       POT         = 22[*];  % POWERS-OF-TEN TABLE                       09201400 T  0000:0
  REAL        FORTERR     = 24;     % FORTRAN ERROR MESSAGE ROUTINE             09201450 T  0000:0
                                                                                 09201500 T  0000:0
  ARRAY       ARRY[*],              % GLOBAL TEMPORARY ARRAY                    09201600 T  0000:0
              FIB[*];               % FILE INFORMATION BLOCK                    09201700 T  0000:0
                                                                                 09201800 T  0000:0
  BOOLEAN     ARRAYTOG,             % LIST ELEMENT WAS ARRAY NAME               09201900 T  0000:0
              COMPLEXTOG,           % FIRST HALF OF COMPLEX NUMBER              09202000 T  0000:0
              DBLTOG,               % LIST ELEMENT DOUBLE TYPE                  09202100 T  0000:0
              DONE,                 % FLAG FOR LIST EXHAUSTED                   09202200 T  0000:0
              ESIGN,                % EXPONENT NEGATIVE FLAG                    09202300 T  0000:0
              GOTDIGIT,             % TELLS IF CHARACTER SEEN                   09202400 T  0000:0
              SEQ,                  % TRUE IFF FILE HAS SEQ NUMBERS.            09202450 T  0000:0
              READREC,              % TRUE IFF SCANNER MAY READ A RECORD         09202460 T  0000:0
              SIGN,                 % MANTISSA NEGATIVE FLAG                    09202500 T  0000:0
              STRINGTOG,            % CONTROLS SCANNER ACTION                   09202600 T  0000:0
              TWODIMTOG;            % ON IF ARRAY IS TWO-DIMENSIONAL            09202700 T  0000:0
                                                                                 09202800 T  0000:0
  INTEGER     BSIZE,                % NUMBER OF CHARACTERS LEFT IN BUFFER       09202900 T  0000:0
              BUFF,                 % CURRENT BUFFER POSITION                   09203000 T  0000:0
              CHAR,                 % CONTAINS LAST CHARACTER SCANNED           09203100 T  0000:0
              COUNTER,              % NUMBER CHARACTERS IN STRING               09203200 T  0000:0
              DECADES,              % NUMBER OF DECIMAL PLACES                  09203300 T  0000:0
              E,                    % CONTAINS EXPONENT                         09203400 T  0000:0
              INDEX,                % INDEX INTO ARRAY IF ARRAYTOG              09203500 T  0000:0
              SIZE,                 % ARRAY SIZE IF ARRAYTOG                    09203600 T  0000:0
              TYPE;                 % TYPE OF LAST LIST ELEMENT                 09203700 T  0000:0
                                                                                 09203800 T  0000:0
  NAME        ADDRESS,              % HOLDS ADDRESS TO STORE NEXT DATUM         09203900 T  0000:0
              LISTADR = ARRY;       % HOLDS RESULT OF [LISX]                    09204000 T  0000:0
                                                                                 09204100 T  0000:0
  REAL        NUMBER,               % TEMPORARY NUMBER-HOLDER.                  09204200 T  0000:0
```

```
                NUMBERL, NUMBERH ;   % DBLPREC NUMBER BUILT BY FREEREAD,
LABEL           LISTSTART,
                LISTEXIT,
                LOOK,
                NUMERICAL,
                PASTPOINT,
                BYE,
                SCNR,
                AT,
                DECIMAL,
                ERROR,
                STRING,
                STRUNG,
                GETCOMMA,
                LOGICAL,
                EXIT;

SWITCH          SWISH := EXIT, NUMERICAL, STRING, NUMERICAL,
                         LOGICAL, NUMERICAL, NUMERICAL;

DEFINE          INTEGERV    = 1#,
                STRINGV     = 2#,
                REALV       = 3#,
                LOGICALV    = 4#,
                DOUBLEV     = 5#,
                COMPLEXV    = 6#;

DEFINE          KIND        = (FIB[4],[8:4])#,
                DATATYPE    = (LISTYPE,[44:4])#,
                TWOD        = (LISTYPE,[38:1])#,
                SIZEF       = [33:15]#,
                BASEF       = [18:15]#,
                IOD         = (*FILX)#;

SUBROUTINE CHECKPRESENCE;
BEGIN COMMENT GETS NEXT BUFFER FROM ALGOLREAD;
          BSIZE←(P(MKS,DKADR,1,FILX,ALGOLREAD)-SEQ)×8 ;
     BUFF := IOD.[33:15];
END  CHECKPRESENCE;

SUBROUTINE READIT;
BEGIN COMMENT ORDER NEXT RECORD READ FROM MEDIUM;
     P(MKS, DKADR, 0, FILX, ALGOLREAD);
     IF DONE THEN P(XIT);
     IF IOD.[27:1] THEN P(XIT);
     CHECKPRESENCE;
END  READIT;

REAL SUBROUTINE NEXT;
BEGIN COMMENT GET DESCRIPTOR POINTING INTO AN ARRAY;
     IF TWODIMTOG THEN
       P(*[ARRY[INDEX,[33:7]]], INDEX,[40:8], CDC)
     ELSE P([ARRY[INDEX]]);
     NEXT := POLISH;
END  NEXT ITEM INSIDE AN ARRAY;
```

```
09204205 T  0000:0
09204300 T  0000:0
09204400 T  0000:0
09204500 T  0000:0
09204600 T  0000:0
09204700 T  0000:0
09204800 T  0000:0
09204850 T  0000:0
09204860 T  0000:0
09204900 T  0000:0
09205000 T  0000:0
09205100 T  0000:0
09205200 T  0000:0
09205300 T  0000:0
09205400 T  0000:0
09205500 T  0000:0
09205600 T  0000:0
09205700 T  0000:0
09205800 T  0000:0
09205900 T  0000:0
09206000 T  0000:0
09206100 T  0000:0
09206200 T  0000:0
09206300 T  0000:0
09206400 T  0000:0
09206500 T  0000:0
09206600 T  0000:0
09206700 T  0000:0
09206800 T  0000:0
09206900 T  0000:0
09207000 T  0000:0
09207100 T  0000:0
09207200 T  0000:0
09207300 T  0000:0
09207400 T  0000:0
09207500 T  0000:0
09207600 T  0001:0
09207700 T  0001:0
09207800 T  0003:3
09207900 T  0005:1
09208000 T  0005:2
09208100 T  0005:2
09208200 T  0006:0
09208300 T  0006:0
09208400 T  0007:1
09208700 T  0008:1
09208800 T  0010:0
09208900 T  0011:0
09209000 T  0011:1
09209100 T  0011:1
09209200 T  0012:0
09209300 T  0012:0
09209400 T  0012:1
09209500 T  0015:0
09209600 T  0016:0
09209700 T  0016:1
09209800 T  0016:2
```

```
SUBROUTINE LISTELEMENT;                                                  09209900 T  0016:2
BEGIN COMMENT GETS ADDRESS TO STORE NEXT DATUM, AND                      09210000 T  0017:0
        DIDDLES CERTAIN TOGGLES AS REQUIRED;                             09210100 T  0017:0
LISTSTART:                                                               09210200 T  0017:0
     IF ARRAYTOG THEN                                                    09210300 T  0017:0
     BEGIN ADDRESS := NEXT;                                              09210400 T  0017:1
          IF (INDEX := INDEX+DBLTOG+1)≥SIZE THEN                         09210500 T  0019:2
            ARRAYSTUFF←ARRAYTOG←COMPLEXTOG←0 ;                           09210600 T  0021:3
          GO TO LISTEXIT;                                                09210700 T  0024:0
     END;                                                                09210800 T  0024:2
     IF COMPLEXTOG THEN                                                  09210900 T  0024:2
     BEGIN ADDRESS := [LISTADR[1]]; COMPLEXTOG := 0;                     09211000 T  0024:3
          GO TO LISTEXIT;                                                09211100 T  0027:1
     END;                                                                09211200 T  0027:3
     P(0); LISTADR := [LISX];                                            09211300 T  0027:3
     COMPLEXTOG←(TYPE←DATATYPE)=COMPLEXV; DBLTOG←TYPE=DOUBLEV ;          09211310 T  0028:3
     IF ARRAYSTUFF≠0 THEN                                                09211400 T  0032:1
     BEGIN                                                               09211500 T  0033:0
          ARRAYTOG←1; P(LISTADR←MEM[LISTADR,[18:15]]) ;                  09211600 T  0033:2
          SIZE←(INDEX←ARRAYSTUFF.BASEF)+ARRAYSTUFF.SIZEF ;               09211700 T  0036:1
          TWODIMTOG←NOT P(LOD,TOP); P(DEL) ;                             09211800 T  0039:0
          GO TO LISTSTART;                                               09211900 T  0040:2
     END;                                                                09212000 T  0041:0
     ADDRESS ← [LISTADR[0]]; P(DEL);                                     09212100 T  0041:0
LISTEXIT:                                                                09212150 T  0042:0
                                                                         09212300 T  0042:0
END  GET NEXT LIST ELEMENT;                                              09212400 T  0042:0
                                                                         09212500 T  0042:1
SUBROUTINE SCANNER ;                                                     09212600 T  0042:1
BEGIN COMMENT GENERAL PURPOSE SCANNER -- CHARACTER AT A TIME.            09212700 T  0043:0
        PURLOINED FROM BASICINPUT ROUTINE BY WWF4;                       09212800 T  0043:0
LOOK: IF BSIZE=0 THEN READIT;                                            09212900 T  0043:0
     STREAM(I:=-1, BUFF,                                                 09213000 T  0045:0
               N:=IF BSIZE<63 THEN BSIZE ELSE 63: STRINGTOG);            09213100 T  0046:0
     BEGIN SI:=BUFF; CI:=CI+STRINGTOG; GO TO DEBLANK;                    09213200 T  0049:0
          COMMENT BLANKS SIGNIFICANT WITHIN STRINGS;                     09213300 T  0050:0
GNC:      TALLY:=TALLY+1; DI:=LOC I; DS:=LIT "0";                        09213400 T  0050:0
          DI:=DI+6; DS:=CHR; GO TO EXIT;                                 09213500 T  0051:0
DEBLANK:  N(IF SC≠" " THEN JUMP OUT TO GNC;                              09213600 T  0051:3
               TALLY:=TALLY+1; SI:=SI+1);                                09213700 T  0053:1
EXIT:     N:=TALLY; BUFF:=SI;                                            09213800 T  0054:0
     END  STREAM;                                                        09213900 T  0054:2
     BSIZE := BSIZE-P(XCH);      % UPDATE CHARACTER COUNT                09214000 T  0054:3
     BUFF := POLISH;             % UPDATE BUFFER POINTER                 09214100 T  0056:0
     IF (CHAR←POLISH)<0 THEN                                             09214200 T  0056:2
          IF BSIZE=0 THEN                                                09214250 T  0057:2
               BEGIN                                                     09214275 T  0058:3
               IF GOTDIGIT THEN CHAR←","                                 09214300 T  0059:1
               ELSE IF READREC THEN GO LOOK                              09214350 T  0060:1
               END                                                       09214355 T  0061:2
          ELSE GO LOOK ;                                                 09214360 T  0062:1
END  SCANNER;                                                            09214400 T  0062:1
                                                                         09214405 T  0062:2
SUBROUTINE LOGICALCOMPARE ;                                              09214410 T  0062:2
     BEGIN COMMENT COMPARES LOGICAL TO .TRUE.,.TRU.,.TR.,.T., OR         09214415 T  0063:0
               .FALSE.,.FALS.,.FAL.,.FA.,.F. ;                           09214416 T  0063:0
     STREAM(C←P(XCH):C2←COUNTER-1,C1←8-COUNTER,E,NUMBER) ;                09214425 T  0063:0
```

```
                    BEGIN                                                                                09214430 T  0066:0
                    SI←LOC NUMBER; SI←SI+C1; DI←LOC C; DI←DI+E ;                                         09214435 T  0066:0
                    IF C2 SC=DC THEN IF SC="." THEN TALLY←1;  C←TALLY ;                                  09214440 T  0067:2
                    END ;                                                                                09214445 T  0069:1
               E←P ;                                                                                     09214450 T  0069:2
               END OF LOGICALCOMPARE ;                                                                   09214455 T  0070:0
                                                                                                         09214460 T  0070:1
          SUBROUTINE SCAN ;                                                                              09214465 T  0070:1
     SCNR:     BEGIN SCANNER ;                                                                           09214470 T  0071:0
               IF CHAR="/" THEN                                                                          09214475 T  0072:0
                    BEGIN READREC←0 ;                                                                    09214480 T  0072:3
                    WHILE CHAR≠"=" AND BSIZE>0 DO SCANNER;  READREC←1 ;                                  09214485 T  0074:0
                    IF BSIZE=0 AND GOTDIGIT THEN CHAR←"," ELSE GO SCNR ;                                 09214487 T  0078:1
                    END ;                                                                                09214490 T  0080:3
               END OF SCAN ;                                                                             09214495 T  0080:3
                                                                                                         09214500 T  0081:0
          SUBROUTINE BUILDNUMBER ;                                                                       09214505 T  0081:0
               BEGIN COMMENT BUILDS DBLPREC NUMBER NUMBERL,NUMBERH ;                                     09214510 T  0081:0
               P(NUMBERL,NUMBERH) ;                                                                      09214515 T  0081:0
               WHILE CHAR<10 DO                                                                          09214520 T  0081:2
                    BEGIN                                                                                09214525 T  0082:3
                    COUNTER←NUMBER←0 ;                                                                   09214530 T  0082:3
                    DO BEGIN COUNTER←COUNTER+1; NUMBER←NUMBER×10+CHAR;SCAN END                            09214540 T  0084:0
                    UNTIL CHAR>9 OR COUNTER=11 ;                                                         09214545 T  0087:3
                    DECADES←DECADES+COUNTER ;                                                            09214546 T  0090:1
                    IF DBLTOG THEN P(0,POT[COUNTER],DLM,0,NUMBER,DLA)                                    09214547 T  0091:2
                    ELSE P(POT[COUNTER],×,NUMBER,+) ;                                                    09214550 T  0094:0
                    END ;                                                                                09214555 T  0095:3
               NUMBERL←P(.NUMBERH,←) ;                                                                   09214560 T  0096:1
               END OF BUILDNUMBER ;                                                                      09214565 T  0097:1
                                                                                                         09214570 T  0097:2
          REAL SUBROUTINE ALFA ;                                                                         09214575 T  0097:2
               BEGIN                                                                                     09214577 T  0098:0
               STREAM(CHAR:Q←0) ;                                                                        09214579 T  0098:0
                    BEGIN                                                                                09214581 T  0099:1
                    SI←LOC CHAR; SI←SI+7; IF SC=ALPHA THEN TALLY←1; CHAR←TALLY                           09214583 T  0099:1
                    END ;                                                                                09214585 T  0100:2
               ALFA←P ;                                                                                  09214587 T  0101:0
               END OF ALFA ;                                                                             09214589 T  0101:1
                                                                                                         09214591 T  0101:2
          SUBROUTINE FREEREAD;                                                                           09214600 T  0101:2
          BEGIN COMMENT READS AND STORES NEXT DATUM, DOING APPROPRIATE                                   09214700 T  0102:0
                    CONVERSIONS.  TYPE OF SCAN IS DEPENDENT ON TYPE OF                                   09214800 T  0102:0
                    LIST ITEM.  OPERATES INDIFFERENTLY ON A VARIETY OF                                   09214900 T  0102:0
                    NUMERICAL FORMATS;                                                                   09215000 T  0102:0
               GOTDIGIT := STRINGTOG := FALSE;                                                           09215100 T  0102:0
               COUNTER←E←ESIGN←NUMBERL←NUMBERH←NUMBER←DECADES←0 ;                                        09215200 T  0103:1
               SCAN;  IF CHAR="," THEN GO TO EXIT;                                                       09215300 T  0107:0
               IF CHAR>9 THEN                                                                            09215310 T  0109:1
                    IF ALFA THEN                                                                         09215315 T  0110:0
                         BEGIN                                                                           09215320 T  0112:0
                         DO SCAN UNTIL NOT ALFA ;                                                        09215325 T  0112:2
                         IF CHAR="(" THEN                                                                09215330 T  0115:3
                              BEGIN                                                                      09215335 T  0116:2
                              DO BEGIN DO SCAN UNTIL CHAR>9 END UNTIL CHAR≠58;                           09215340 T  0117:0
                              IF CHAR≠")" THEN GO ERROR;  SCAN ;                                         09215345 T  0119:3
                              END ;                                                                      09215350 T  0123:0
```

```
                    IF CHAR="=" THEN                                                    09215355 T  0123:0
                         BEGIN                                                          09215360 T  0123:3
                         SCAN; IF NOT(CHAR="R" OR CHAR="I")THEN GO ERROR;               09215365 T  0124:1
                         SCAN ;                                                         09215370 T  0127:0
                         END ;                                                          09215375 T  0128:0
                    IF CHAR#"=" THEN GO ERROR;  SCAN ;                                  09215380 T  0128:0
                    END ;                                                               09215385 T  0130:0
BYE:     IF (DONE←CHAR="*") THEN READIT ;                                               09215400 T  0130:0
         IF CHAR  = """ THEN GO TO STRING;                                              09215405 T  0133:0
         IF CHAR="×" THEN GO GETCOMMA ;                                                 09215410 T  0134:1
         GOTDIGIT := TRUE;                                                              09215500 T  0135:2
         GO TO SWISH[TYPE];                                                             09215600 T  0136:1
    NUMERICAL::                                                                         09215700 T  0140:3
         IF (SIGN := CHAR="-") OR CHAR="+" OR CHAR="&" THEN SCAN;                       09215800 T  0140:3
         IF CHAR>9 THEN GO TO DECIMAL;                                                  09215900 T  0146:0
         BUILDNUMBER ;                                                                  09216000 T  0147:1
         DECADES←0 ;                                                                    09216100 T  0148:0
         IF CHAR="." THEN                                                               09216200 T  0148:3
         BEGIN SCAN;                                                                    09216300 T  0149:2
    PASTPOINT::                                                                         09216400 T  0151:0
              BUILDNUMBER ;                                                             09216500 T  0151:0
         END ;                                                                          09216800 T  0152:0
         IF CHAR="@" OR CHAR="E" OR CHAR="D" THEN                                       09216900 T  0152:0
    AT:: BEGIN SCAN;                                                                    09217000 T  0154:3
              IF (ESIGN := CHAR="-") OR CHAR="+" OR CHAR="&" THEN SCAN;                 09217100 T  0157:0
              IF (E := CHAR)>9 THEN GO TO ERROR;  SCAN;                                 09217200 T  0162:0
              WHILE CHAR≤9 DO                                                           09217300 T  0165:0
              BEGIN E := 10×E+CHAR;  SCAN;  END;                                        09217400 T  0166:1
              IF ESIGN THEN E := -E;                                                    09217500 T  0169:2
         END;                                                                           09217600 T  0171:1
                                                                                        09217700 T  0171:1
         IF ABS(NUMBER←E-DECADES)>69 THEN GO ERROR ;                                    09217800 T  0171:1
         P(NUMBERL,NUMBERH) ;                                                           09217900 T  0173:3
         IF NUMBER#0 THEN                                                               09218000 T  0174:1
              IF DBLTOG THEN P(POT[69+ABS(NUMBER)],POT[ABS(NUMBER)],                    09218025 T  0175:0
                             IF NUMBER<0 THEN P(DLD) ELSE P(DLM))                       09218050 T  0178:1
              ELSE P(POT[ABS(NUMBER)],IF NUMBER<0 THEN P(/) ELSE P(×)) ;                09218100 T  0180:2
         IF SIGN THEN P(CHS) ;                                                          09218400 T  0184:0
         IF DBLTOG THEN P([ADDRESS],STD,[ADDRESS[1]],STD)                               09218500 T  0185:0
         ELSE BEGIN                                                                     09218600 T  0187:1
              P(XCH,DEL,[ADDRESS]) ;                                                    09218615 T  0187:3
              IF TYPE=INTEGERV THEN                                                     09218620 T  0188:2
                   BEGIN                                                                09218625 T  0189:1
                   IF P(DUP)>@7777777777777 THEN GO ERROR ;                             09218630 T  0189:3
                   P(ISD) ;                                                             09218635 T  0191:0
                   END                                                                  09218640 T  0191:1
              ELSE P(STD) ;                                                             09218645 T  0191:1
              END ;                                                                     09218650 T  0193:1
         GO TO GETCOMMA;                                                                09218700 T  0193:1
DECIMAL::                                                                               09218800 T  0193:3
         IF CHAR="." THEN                                                               09218900 T  0193:3
         BEGIN SCAN;                                                                    09219000 T  0194:3
              IF CHAR≤9 THEN GO TO PASTPOINT ELSE GO TO ERROR;                          09219100 T  0196:0
         END;                                                                           09219200 T  0197:3
         NUMBERH←1 ;                                                                    09219300 T  0197:3
         IF CHAR="@" OR CHAR="E" OR CHAR="D" THEN GO TO AT;                             09219400 T  0198:2
ERROR::                                                                                 09219500 T  0202:0
```

```
     IF PARL≠0 THEN                                                                  09219600 T  0202:0
       P(PARL, MKS, 9, BLOCK);                                                       09219700 T  0202:3
     P(MKS, FIB[6], FILX.[33:15], 2, FORTERR);                                       09219800 T  0204:1
STRING::                                                                             09219900 T  0206:1
     IF CHAR≠""" THEN GO TO ERROR;                                                   09220000 T  0206:1
     COUNTER := 0;  STRINGTOG := 1;  NUMBER := "      ";                            09220100 T  0208:1
     DO BEGIN SCANNER ;                                                              09220200 T  0210:2
STRUNG::                                                                             09220300 T  0212:0
          IF CHAR≠""" THEN                                                           09220400 T  0212:0
          BEGIN COUNTER:=COUNTER+1;                                                  09220500 T  0212:3
               STREAM(CHAR,N:=COUNTER,T:=[NUMBER]);                                  09220600 T  0214:2
               BEGIN SI:=LOC N; SI:=SI-1;                                            09220700 T  0215:3
                    DI:=DI+1; DI:=DI+N; DS:=CHR;                                     09220800 T  0216:1
               END  STREAM;                                                          09220900 T  0217:1
          END;                                                                       09221000 T  0217:2
     END UNTIL (COUNTER=6) OR CHAR="";                                               09221100 T  0217:2
     IF COUNTER=0 THEN GO TO ERROR;                                                  09221200 T  0219:3
     P(NUMBER, [ADDRESS], STD);                                                      09221300 T  0221:0
     IF CHAR≠""" THEN                                                            09221400 T  0221:3
     BEGIN SCANNER;  IF CHAR="" THEN GO GETCOMMA ;                                   09221500 T  0222:2
          IF LSTRN=(-1) THEN GO TO ERROR;                                            09221600 T  0225:1
          LISTELEMENT;                                                               09221700 T  0226:3
          IF LSTRN=(-1) THEN GO ERROR ;                                              09221750 T  0228:0
          NUMBER := "      ";  COUNTER := 0;  GO TO STRUNG;                          09221800 T  0229:2
     END;                                                                            09221900 T  0233:0
GETCOMMA::                                                                           09222000 T  0233:0
     WHILE CHAR≠"," AND CHAR≠"*" DO SCAN;  IF CHAR="*" THEN GO BYE ;                 09222100 T  0233:0
     GO TO EXIT;                                                                     09222200 T  0237:1
LOGICAL::                                                                            09222300 T  0238:1
     IF CHAR="." THEN                                                                09222400 T  0238:1
     BEGIN COMMENT SHOULD BE ".TRUE.", ".FALSE.", OR ABBREVIATIONS;                  09222500 T  0239:3
          NUMBER := COUNTER := E := 0;                                               09222600 T  0240:1
          DO BEGIN                                                                   09222700 T  0242:0
               SCAN;  NUMBER := CHAR & NUMBER[12:18:30];                             09222800 T  0242:0
          END UNTIL (COUNTER := COUNTER+1)=6 OR                                      09222900 T  0244:3
                         CHAR="," OR CHAR=".";                                       09223000 T  0246:2
          IF NOT (E←COUNTER=2 AND NUMBER="T,") THEN                                  09223100 T  0249:0
               BEGIN E←4; P("TRUE"); LOGICALCOMPARE ;                                09223110 T  0251:2
               IF NOT E THEN                                                         09223115 T  0254:0
                    BEGIN                                                            09223120 T  0254:2
                    IF COUNTER≠2 OR NUMBER≠"F," THEN                                 09223125 T  0255:0
                         BEGIN E←3; P("FALSE"); LOGICALCOMPARE ;                     09223130 T  0256:3
                         IF NOT E THEN GO ERROR ;                                    09223135 T  0259:0
                         END ;                                                       09223140 T  0259:3
                    E←0 ;                                                            09223145 T  0259:3
                    END ;                                                            09223200 T  0260:2
               END ;                                                                 09223300 T  0260:2
     END ELSE IF (E←CHAR="T") OR CHAR="F" THEN                                       09223400 T  0260:2
                    BEGIN SCAN ;                                                     09223405 T  0267:1
                    IF NOT (CHAR="." OR CHAR=",") THEN GO ERROR ;                    09223410 T  0269:0
                    END                                                              09223420 T  0271:1
               ELSE IF NOT ((E←CHAR=1) OR CHAR=0) THEN GO ERROR ;                    09223425 T  0271:1
     P(E, [ADDRESS], STD);  GO TO GETCOMMA;                                          09223500 T  0274:2
EXIT::                                                                               09223600 T  0275:3
END  FREEREAD;                                                                       09223700 T  0275:3
                                                                                     09223800 T  0276:1
COMMENT ***** ***** START OF CODE ***** ***** *****;                                 09223900 T  0276:1
```

```
      FILX[NOT 3]←PARL; FILX[NOT 4]←EOFL;                                    09224000 T  0276:1
      FIB := FILX[NOT 2];                                                    09224100 T  0287:0
      IF FIB[5].[43:2]≠2 THEN                                                09224200 T  0288:3
        POLISH(MKS, 0, 2, FILX, 1, SELECT);                                  09224300 T  0290:1
      CHECKPRESENCE; ARRAYSTUFF := 0;                                        09224400 T  0292:1
       IF FIB[0]=0 THEN FIB[0] := 1;                                         09224500 T  0293:3
      IF FIB[0]≠1 AND KIND=2 THEN                                            09224600 T  0296:2
        POLISH(MKS, FIB[6], FILX.[33:15], 4, FORTERR);                       09224700 T  0299:1
      IF @(*[FIB[14]],TOP) THEN P(DEL)                                       09224710 T  0301:3
      ELSE BSIZE←-(SEQ←(SEQ←(*(4 INX P(XCH))).[36:6])≠0 AND SEQ≠8             09224720 T  0303:2
                                                        AND SEQ≠9)×8         09224730 T  0306:3
                +BSIZE ;                                                     09224740 T  0309:1
      LSTRN←READREC←1 ;                                                      09224800 T  0310:3
      DO BEGIN IF DONE←LSTRN=(-1) THEN READIT; LISTELEMENT ;                 09224900 T  0312:0
           IF (DONE := (LSTRN=(-1))) THEN READIT;                            09225000 T  0316:0
           FREEREAD;                                                         09225050 T  0319:0
      END UNTIL FALSE;                                                       09225100 T  0320:0
END FORTRAN FREE FIELD READ;                                                 09225300 T  0320:3
                                                                                  SIZE= 0321 WORDS
```

```
PROCEDURE COBOLDECIMALTOOCTALCONVERT(A) ;  %% INTRINSIC # @151,              09300000 T  0000:0
                                                        START OF REL SEGMENT; DISK ADDRESS = 00632
VALUE A;  NAME A ;                                                           09300100 T  0000:0
    % THIS PROCEDURE CONVERTS A STRING OF N BCD DIGITS, STARTING AT WORD      09300200 T  0000:0
    % ADDRESS A, CHARACTER OFFSET S, INTO A DOUBLE-LENGTH VALUE. THE LOW      09300300 T  0000:0
    % PART OF THIS IS STORED IN S, THE HIGH PART IN N. IF N.[1:1]=1,THEN      09300400 T  0000:0
    % THE SIGN OF THE VALUE IS OBTAINED FROM THE ZONE BITS OF THE 1-ST        09300500 T  0000:0
    % CHARACTER (BCD DIGIT), OTHERWISE FROM THE LAST. 0≤S≤7, 0≤ABS(N)≤23      09300600 T  0000:0
BEGIN                                                                        09300700 T  0000:0
REAL N=A-2, S=N-1, Q=9, C ;                                                  09300800 T  0000:0
REAL HOLD1,HOLD2,HOLD3 ; NAME A1;                                            09300850 C  0000:0
LABEL B,D,E,T8,G ;                                                           09300900 T  0000:0
Q:=N≥0; A1:=[HOLD1];                                                         09301000 P  0000:0
STREAM(A,S,JSIGN:=IF Q THEN 0 ELSE 1,NUMD:=ABS(N)-1,                         09301005 C  0003:1
       SAVSI:=0,HOLD:=[HOLD1]);                                              09301010 C  0006:3
BEGIN                                                                        09301012 C  0007:2
     SI:=A;  SI:=SI+S;  SAVSI:=SI;                                           09301015 C  0007:2
     SI:=SI+JSIGN;                                                           09301020 C  0008:2
     DI:=DI+JSIGN;                                                           09301025 C  0009:0
     DS:=NUMD NUM;                                                           09301030 C  0009:2
     JSIGN(SI:=SAVSI;  DI:=HOLD;);                                           09301050 C  0010:0
     DS:=CHR;                                                                09301065 C  0011:1
END;                                                                         09301070 C  0011:2
A:=A1;  S:=0;                                                                09301075 C  0011:3
P(DIB 1);                                                                    09301080 C  0013:1
IF (N←ABS(N))≤8 THEN                                                         09301100 T  0013:2
     BEGIN                                                                   09301200 T  0015:0
     STREAM(C:S,A,N); BEGIN SI←A; SI←SI+S; DI←LOC C; DS←N OCT END ;           09301300 T  0015:2
     IF NOT Q THEN GO D;  N←P ;                                              09301400 T  0019:0
     END                                                                     09301500 T  0020:1
ELSE BEGIN  P(0) ;                                                           09301600 T  0020:1
     IF N>16 THEN                                                            09301700 T  0021:0
          BEGIN                                                              09301800 T  0021:3
          STREAM(S,Z←0,A:N←N-16,CA←[C]) ;                                    09301900 T  0022:1
```

```
                    BEGIN                                                                                09302000 T  0024:3
                    SI←A; SI←SI+S; DI←LOC A; DS←N OCT; DI←LOC S ;                                        09302100 T  0024:3
                    DS←8 OCT; DI←CA; DS←8 OCT ;                                                          09302200 T  0026:2
                    END ;                                                                                09302300 T  0027:1
               P(0,T8,DLM,DLA) ;                                                                         09302400 T  0027:2
B:             P(0,T8,DLM,0,ABS(C),DLA) ;                                                                09302500 T  0028:2
               END                                                                                       09302600 T  0030:1
          ELSE BEGIN                                                                                     09302700 T  0030:1
               STREAM(S:A,N←N-8,CA←[C]) ;                                                                09302800 T  0030:3
                    BEGIN                                                                                09302900 T  0033:0
                    SI←A; SI←SI+S; DI←LOC S; DS←N OCT; DI←CA; DS←8 OCT ;                                  09303000 T  0033:0
                    END ;                                                                                09303100 T  0035:0
               IF P(DUP)>P(G) THEN GO B;  P(T8,x,ABS(C),+) ;                                             09303200 T  0035:1
               END ;                                                                                     09303300 T  0037:3
          IF C≠0 AND Q THEN                                                                              09303400 T  0037:3
               BEGIN                                                                                     09303500 T  0039:0
               P(C,DIA 1);  GO E ;                                                                       09303600 T  0039:2
T8:::          100000000.0 ;                                                                             09303700 T  0040:2
G:::           5496.0 ;                                                                                  09303800 T  0042:0
               END ;                                                                                     09303900 T  0043:0
          IF Q THEN S←S+N-1 ;                                                                            09304000 T  0043:0
D:        STREAM(S:A); BEGIN SI←A; SI←SI+S; S←TALLY; DI←LOC S; DI←DI+7;                                   09304100 T  0045:2
                       DS←ZON; END;                                                                      09304200 T  0048:1
          P(P=@40,DIA 47) ;                                                                              09304300 T  0048:3
E:        N←P(TRB 1) ;                                                                                   09304400 T  0049:2
          END ;                                                                                          09304500 T  0050:1
S←P ;                                                                                                    09304600 T  0050:1
END OF COBOLDECIMALTOOCTALCONVERT ;                                                                      09304700 T  0050:3
                                                                                                         SIZE= 0051 WORDS
```

```
PROCEDURE COBOLOCTOLTODECIMALCONVERT(A,L,H,S,N,R,T); % INTRINSIC # @152.          09400000 T  0000:0
                                                                                  START OF REL SEGMENT; DISK ADDRESS = 00634
VALUE L,H,R,N,S,T;  REAL L,H,R,N,S,T;  NAME A ;                                   09400100 T  0000:0
    % THIS PROCEDURE CONVERTS THE DOUBLE-LENGTH WORD (L,H) INTO A STRING          09400200 T  0000:0
    % OF N BCD DIGITS. THE STRING STARTS AT WORD ADDRESS A, CHARACTER             09400300 T  0000:0
    % OFFSET S. PRIOR TO THE CONVERSION, (L,H) IS SCALED-TO-THE-LEFT/RHT          09400400 T  0000:0
    % BY R DIGITS, I.E. (L,H) IS DIVIDED/MULTED BY 10*R. T IS A COMBINED          09400500 T  0000:0
    % TRUNCATION/J-SIGN TOGGLE: T.[2:1]=1 => PUT THE SIGN OF (L,H) IN             09400600 T  0000:0
    % 1-ST CHR OF THE STRING; T.[1:1]=1 => PUT SIGN IN THE LAST CHR;              09400700 T  0000:0
    % ABS(T).[47:1]=1 => TRUNCATE (L,H) BEFORE CONVERSION (AND AFTER              09400800 T  0000:0
    % SCALING); ABS(T).[46:1]=1 => ROUND (L,H) BEFORE CONVERSION (AND             09400900 T  0000:0
    % AFTER SCALING). NOTE THAT 0≤S≤7, 0≤N≤23.                                   09401000 T  0000:0
    BEGIN                                                                         09401100 T  0000:0
    INTEGER IR=R, IH=H, IL=L ;                                                    09401200 T  0000:0
    REAL B=17, SERR=19, WH=11, DMOD=21, Q=9 ;                                     09401300 T  0000:0
    ARRAY TEN=23[*] ;                                                             09401400 T  0000:0
    LABEL HLF,T8,T16 ;                                                            09401500 T  0000:0
    IF R<0 THEN                                                                   09401600 T  0000:0
         BEGIN                                                                    09401700 T  0000:3
         STREAM(S,N,A); BEGIN DI←DI+S; N(DS←LIT"0") END ;                         09401800 T  0001:1
         N←N+R;  R←0 ;                                                            09401900 T  0004:2
         END ;                                                                    09402000 T  0006:2
         IF T.[1:2]=0 THEN H ← ABS(H);                                            09402100 T  0006:2
    IF H.[2:1] THEN P(0,H/TEN[R]) ELSE P(L,H,TEN[R+27],TEN[R],DLD) ;              09402200 T  0009:1
```

```
L←0 ;                                                                    09402300 T  0014:2
IF P(ABS(Q←P),DUP)<P(HLF) THEN H←R←SERR←0                                09402400 T  0015:1
ELSE BEGIN                                                               09402500 T  0017:1
     IF SERR←P(DUP)>TEN[23] THEN P(TEN[27+N],TEN[N],DMOD,B,XCH) ;         09402600 T  0019:2
     IF P(DUP).[2:1] THEN                                                09402700 T  0023:3
          BEGIN IF T THEN P(HLF,-); H←(IR←P) DIV P(T8) END               09402800 T  0024:2
     ELSE BEGIN                                                          09402900 T  0027:3
          IF NOT T THEN P(0,HLF,DLA);  H←P ;                             09403000 T  0028:1
          H←P(L←P,H,0,IL←P(L,H,0,T16,DLD,HLF,-),XCH,DEL,0,T16,DLM,       09403100 T  0030:2
              DLS) ;                                                     09403200 T  0035:0
          IR←P(R←P,H,0,IH←P(R,H,0,T8,DLD,HLF,-),XCH,DEL,0,T8,DLM,        09403300 T  0035:3
               DLS,HLF,-) ;                                              09403400 T  0040:1
          END ;                                                          09403500 T  0041:2
     END ;                                                               09403600 T  0041:2
IF N≤8 THEN                                                              09403700 T  0041:2
     BEGIN P(L≠0 OR H≠0 OR R≥TEN[N] OR N=0) ;                            09403800 T  0042:1
     STREAM(R,N,S,A); BEGIN DI←DI+S; SI←LOC R; DS←N DEC END ;            09403900 T  0046:3
     END                                                                 09404000 T  0049:3
ELSE IF N≤16 THEN                                                        09404100 T  0049:3
          BEGIN P(L≠0 OR H≥TEN[N-8]) ;                                   09404200 T  0051:0
          STREAM(H,R,N←N-8,S,A) ;                                        09404300 T  0054:0
               BEGIN DI←DI+S; SI←LOC H; DS←N DEC; DS←8 DEC END ;         09404400 T  0056:1
          END                                                            09404500 T  0058:0
     ELSE BEGIN P(L≥TEN[N-16]) ;                                         09404600 T  0058:0
          STREAM(L,H,R,N←N-16,S,A) ;                                     09404700 T  0060:0
            BEGIN DI←DI+S; SI←LOC L; DS←N DEC; DS←8DEC; DS←8DEC END      09404800 T  0062:2
          END ;                                                          09404900 T  0064:1
IF P OR SERR THEN IF P(1,WH,[18:15],DUP)≠0 THEN P(DIB 0,←) ;             09405000 T  0064:2
IF Q<0 THEN                                                              09405100 T  0068:1
     BEGIN                                                               09405200 T  0069:0
     IF T>0 THEN                                                         09405300 T  0069:2
          BEGIN                                                          09405400 T  0070:1
          STREAM(N←N-2,S,A) ;                                            09405500 T  0070:3
               BEGIN                                                     09405600 T  0072:2
               DI←DI+S; DS←SET; DS←RESET; DI←DI+N; DS←RESET;DS←RESET     09405700 T  0072:2
               END ;                                                     09405800 T  0074:1
          P(XIT) ;                                                       09405900 T  0074:3
HLF:::    0.499999999999 ;                                               09406000 T  0075:0
T16:::    10000000000000000,0 ;                                          09406100 T  0076:0
T8:::     100000000,0 ;                                                  09406200 T  0077:0
          END ;                                                          09406300 T  0078:0
     STREAM(S←S+N-1,A); BEGIN DI←DI+S; DS←SET; DS←RESET END ;            09406400 T  0078:0
     END ;                                                               09406500 T  0081:1
END OF COBOLOCTALTODECIMALCONVERT ;                                      09406600 T  0081:1
                                                                                SIZE= 0082 WORDS
```

```
PROCEDURE COBOLVARSZ;                                                    09500000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00637
    BEGIN                                                                09500100 T  0000:0
    REAL                                                                 09500200 T  0000:0
         TYPE      = -1;              % 0=2: EXAMINE                    09500300 T  0000:0
                                      % 0=REPLACING FIRST              09500400 T  0000:0
                                      % 1=REP/TALLY ALL,               09500500 T  0000:0
                                      % 2=LEADING/UNTIL FIRST          09500600 T  0000:0
```

```
                                                  % 3: VARIABLE SIZE SMEAR                09500700 T  0000:0
                                                  % 4-9: VARIABLE SIZE RELATE             09500800 T  0000:0
                                                  % 4=<, 5=≥, 6=>, 7=≤,8==,9=≠            09500900 T  0000:0
                                                  % 10: VARIABLE SIZE MOVE                09501000 T  0000:0
                                                  % 11: NEG ALPHA TEST                    09501100 T  0000:0
                                                  % 12: POSITIVE ALPHA TEST               09501200 T  0000:0
ARRAY DESC      = -2[*];                          % RELATE: JUNKA DESCRIPTOR               09501300 T  0000:0
 REAL CODE      = -2,                             % MOVE,SMEAR:  =0                        09501400 T  0000:0
                                                  % EXAMINE:[47:1]=1 IF REP                09501500 T  0000:0
                                                  % [46:1]=1 IF TALLYING                   09501600 T  0000:0
                                                  % [45:1]=1 IF REPLACING OR               09501700 T  0000:0
                                                  % TALLYING UNTIL FIRST                   09501800 T  0000:0
      DLENGTH   = -3,                             % MOVE & RELATE: DEST LENGTH             09501900 T  0000:0
                                                  % SMEAR: LENGTH TO SMEAR                 09502000 T  0000:0
      LNGTH     = -3,                             % EXAMINE: LENGTH                        09502100 T  0000:0
      SLENGTH   = -4,                             % SOURCE LENGTH (SMEAR: =0)              09502200 T  0000:0
      RCHR      = -4,                             % EXAMINE: CHAR TO REPLACE               09502300 T  0000:0
      DOFSET    = -5,                             % MOVE,RELATE,SMEAR:DEST OFF             09502400 T  0000:0
      SCHR      = -5,                             % EXAMINE: CHAR SOUGHT                   09502500 T  0000:0
      SMCHR     = -6,                             % SMEAR: CHAR TO SMEAR                   09502600 T  0000:0
                                                  % EXAMINE: MKS                          09502700 T  0000:0
      SOFSET    = -6,                             % MOVE&RELATE: SOURCE OFFSET             09502800 T  0000:0
      OFFSET    = -7;                             % EXAMINE: OFFSET                        09502900 T  0000:0
 ARRAY                                                                                    09503000 T  0000:0
      DEST      = -7[*],                          % MOVE,RELATE,SMEAR:DEST                 09503100 T  0000:0
      SOURCE    = -8[*];                          % MOVE,RELATE,EXAMINE:SOURCE             09503200 T  0000:0
 REAL                                                                                     09503300 T  0000:0
      RELATE,                                                                             09503400 T  0000:0
      DIFFER,                                                                             09503500 T  0000:0
      NMOD64,                                                                             09503600 T  0000:0
      SAVOFF,                                                                             09503700 T  0000:0
      NDIV64,                                                                             09503800 T  0000:0
      N,                                                                                  09503900 T  0000:0
      NWDS;                                                                               09504000 T  0000:0
 ARRAY D[*];                                                                              09504100 T  0000:0
  DEFINE                                                                                  09504200 T  0000:0
      REPLACECHR = DI←DI-1; SI←LOC P6; SI←SI-1; DS←1 CHR#,                                09504300 T  0000:0
      LISTP1TOP6 = P1←NMOD64,P2←NDIV64,P3←(NDIV64 DIV 64),                                09504400 T  0000:0
                   P4←SCHR,P5←RCHR,P6←OFFSET#;                                            09504500 T  0000:0
  LABEL VARIEXAM,CMD,SMEAR;%                                                              09504600 T  0000:0
%****************************** START HERE ******************************               09504700 T  0000:0
      IF TYPE≤2 THEN GO TO VARIEXAM;%                                                     09504800 T  0000:0
      D ← [DEST];%                                                                        09504900 T  0003:1
      IF TYPE=3 THEN GO TO SMEAR;%                                                        09505000 T  0004:1
      IF (DIFFER ← DLENGTH-SLENGTH)<0 THEN    % VARIABLE MOVE ONLY                       09505100 T  0005:2
      IF TYPE=10 THEN%                                                                    09505200 T  0007:1
  BEGIN%                                                                                  09505300 T  0008:2
      SLENGTH ← DLENGTH;                                                                  09505400 T  0009:0
      NMOD64 ← SLENGTH.[42:6];%                                                           09505500 T  0009:3
  END;                                                                                    09505600 T  0011:0
      IF DIFFER≠0 AND TYPE≥4 AND TYPE≤9 THEN  % IF THERE IS A DIFFER-                     09505700 T  0011:0
          BEGIN % THEN MOVE SHORTER TO JUNKA&FILL OUT WITH BLANKS                         09505800 T  0013:3
              IF DIFFER<0 THEN  % INTERCHANGE TO MAKE DEST THE                            09505900 T  0014:1
                  BEGIN  % LONGER, SOURCE THE SHORTER                                     09506000 T  0015:0
                      D ← [DEST]; DEST ← [SOURCE]; SOURCE ← [D];                          09506100 T  0015:2
                      SAVOFF ← SOFSET; SOFSET ← DOFSET; DOFSET←0;                         09506200 T  0018:2
                      NWDS←DLENGTH;DLENGTH←SLENGTH;SLENGTH←NWDS;                          09506300 T  0020:1
```

```
                    DIFFER ← ABS(DIFFER);%                                    09506350 T  0023:0
                END ELSE%                                                     09506400 T  0024:0
                BEGIN%                                                        09506500 T  0024:0
                    IF TYPE<8 THEN TYPE←TYPE-TYPE.[47:1]%                     09506600 T  0024:2
                        +(TYPE.[47:1]=0);%                                    09506700 T  0026:0
                    SAVOFF ← DOFSET;%                                         09506800 T  0029:0
                    DOFSET ← 0;%                                              09506900 T  0029:3
                END;%                                                         09507000 T  0030:2
            RELATE ← TYPE;                                                    09507100 T  0030:2
            TYPE ← 10;                                                        09507200 T  0031:1
            D ← [DESC[0]];%                                                   09507300 T  0032:0
        END;%                                                                 09507400 T  0033:0
CMD:                    % TRANSFER OR COMPARE FIELDS                         09507500 T  0033:0
    IF TYPE≠10 OR DIFFER≥0 OR RELATE>0 THEN NMOD64←SLENGTH.[42:6];            09507600 T  0033:0
    NDIV64 ← SLENGTH DIV 64;%                                                 09507700 T  0037:2
    IF TYPE<8 THEN%                                                           09507800 T  0038:3
BEGIN%                                                                        09507900 T  0039:2
    STREAM(P0←0:P1←NMOD64,P2←NDIV64,P2A←NDIV64≠0,P3←(NDIV64 DIV 64)           09508000 T  0040:0
        ,P4←SOURCE,P5←SOFSET,P6←DOFSET,P7←TYPE≥6,%                            09508100 T  0042:2
         P8←TYPE.[47:1],P9←D);%                                               09508200 T  0044:2
    BEGIN                                                                     09508300 T  0046:0
        SI ← P4; SI ← SI+P5; DI ← DI+P6;                                      09508400 T  0046:0
        CI ← CI+P7; GO TO GREQ; GO TO GOLSQ;%                                 09508500 T  0047:1
    GREQ:                                                                     09508600 T  0048:1
        P3(63(P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                               09508700 T  0048:1
            BEGIN SI←P0;DI←P9;IF 63 SC>DC THEN;%                              09508800 T  0050:1
                JUMP OUT 2 TO XYT1;%                                          09508900 T  0051:1
            END);%                                                            09509000 T  0052:0
            2(P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                               09509100 T  0052:1
             BEGIN SI←P0;DI←P9;IF 63 SC>DC THEN;%                             09509200 T  0053:3
                JUMP OUT 2 TO XYT1;%                                          09509300 T  0054:3
             END);%                                                           09509400 T  0055:2
           IF SC=DC THEN ELSE%                                                09509500 T  0055:3
            BEGIN SI←SI-1;DI←DI-1;IF SC>DC THEN;%                             09509600 T  0056:2
                JUMP OUT 1 XYT1;%                                             09509700 T  0057:2
            END); GO TO L1;%                                                  09509800 T  0058:0
XYT1: GO TO XYT2;%                                                            09509900 T  0058:2
GOLSQ:GO TO LSEQ;%                                                            09510000 T  0058:3
L1:  P2   (P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                                  09510100 T  0059:0
            BEGIN SI←P0;DI←P9;IF 63 SC>DC THEN;                               09510200 T  0060:3
                JUMP OUT 1 TO XYT2;%                                          09510300 T  0061:3
            END);%                                                            09510400 T  0062:1
     P2A  (P0←SI;P9←DI;IF P2 SC=DC THEN ELSE%                                 09510500 T  0062:2
            BEGIN SI←P0;DI←P9;IF P2 SC>DC THEN;%                              09510600 T  0064:2
                JUMP OUT 1 TO XYT2;%                                          09510700 T  0065:3
            END);%                                                            09510800 T  0066:1
     IF P1 SC≥DC THEN;%                                                       09510900 T  0066:2
XYT2: GO TO XYT3;%                                                            09511000 T  0067:1
LSEQ:                                                                         09511100 T  0067:2
    P3(63(P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                                   09511200 T  0067:2
        BEGIN SI←P0;DI←P9;IF 63 SC<DC THEN;%                                  09511300 T  0069:2
            JUMP OUT 2 TO XYT3;%                                              09511400 T  0070:2
        END);%                                                                09511500 T  0071:1
        2(P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                                   09511600 T  0071:2
         BEGIN SI←P0;DI←P9;IF 63 SC<DC THEN;%                                 09511700 T  0073:0
            JUMP OUT 2 TO XYT3;%                                              09511800 T  0074:0
         END);%                                                               09511900 T  0074:3
```

```
                IF SC=DC THEN ELSE%                                                    09512000 T  0075:0
                 BEGIN SI←SI-1;DI←DI-1;IF SC<DC THEN;%                                 09512100 T  0075:3
                       JUMP OUT 1 TO XYT3;%                                            09512200 T  0076:3
                  END); GO TO L2;%                                                     09512300 T  0077:1
           XYT3: GO TO XYT;%                                                           09512400 T  0077:3
           L2:  P2   (P0←SI;P9←DI;IF 63SC=DC THEN ELSE%                                09512500 T  0078:0
                     BEGIN SI←P0;DI←P9;IF 63 SC<DC THEN;%                              09512600 T  0079:3
                       JUMP OUT 1 TO XYT;%                                             09512700 T  0080:3
                     END);%                                                            09512800 T  0081:1
                P2A  (P0←SI;P9←DI;IF P2 SC=DC THEN ELSE%                               09512900 T  0081:2
                     BEGIN SI←P0;DI←P9;IF P2 SC<DC THEN;%                              09513000 T  0083:2
                       JUMP OUT 1 TO XYT;%                                             09513100 T  0084:3
                     END);%                                                            09513200 T  0085:1
                IF P1 SC≤DC THEN;%                                                     09513300 T  0085:2
           XYT: P8(IF TOGGLE THEN TALLY←1; JUMP OUT 1 TO STOR);%                       09513400 T  0086:1
                IF TOGGLE THEN ELSE TALLY←1;%                                          09513500 T  0088:0
           STOR:P0←TALLY;%                                                             09513600 T  0088:3
           END STREAM;%                                                                09513700 T  0089:0
      END ELSE%                                                                        09513800 T  0089:1
      BEGIN%                                                                           09513900 T  0089:1
           STREAM(P0←0;P1←NMOD64,P2←NDIV64,P2A←NDIV64≠0,P3←(NDIV64 DIV 64)              09514000 T  0089:3
                 ,P4←SOURCE,P5←SOFSET,P6←DOFSET,P7←(TYPE≥10)+(TYPE>10),%                09514100 T  0092:1
                  P8←TYPE.[47:1],P9←D);%                                               09514200 T  0095:1
           BEGIN%                                                                      09514300 T  0096:3
                SI ← P4; SI ← SI+P5; DI ← DI+P6;                                       09514400 T  0096:3
                CI ← CI+P7; GO TO EQUL; GO TO TRFR; GO TO GOTAN;%                      09514500 T  0098:0
           EQUL:%                                                                      09514600 T  0099:1
                P3( 63(IF 63 SC=DC THEN ELSE JUMP OUT 2 TO XYT1);%                     09514700 T  0099:1
                2(IF 63 SC=DC THEN ELSE JUMP OUT 2 TO XYT1);%                          09514800 T  0101:3
                IF     1 SC=DC  THEN ELSE JUMP OUT 1 TO XYT1); GO TO L;%               09514900 T  0103:3
      GOTAN: GO TO TANL;%                                                              09515000 T  0105:2
      L:    P2(IF 63 SC=DC THEN ELSE JUMP OUT 1 TO XYT1);%                             09515100 T  0105:3
           P2A(IF P2 SC=DC THEN ELSE JUMP OUT 1 TO XYT1);%                             09515200 T  0107:3
           IF    P1 SC=DC  THEN; GO TO XYT1;%                                         09515300 T  0110:0
      TRFR:%                                                                           09515400 T  0111:0
           P3(63(DS←63 CHR); 2(DS←63 CHR); DS←CHR);% MOVE 64×64                        09515500 T  0111:0
           P2(DS ← 63 CHR); DS ← P2 CHR; DS ← P1 CHR; GO TO DONE1;%                    09515600 T  0113:2
      XYT1: GO TO XYT2;%                                                               09515700 T  0115:3
      TANL:%                                                                           09515800 T  0116:0
           P3(63(63(IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09515900 T  0116:0
                 JUMP OUT 3 TO XYT2 ELSE JUMP OUT 3 TO XYT2));%                        09516000 T  0118:2
               2(63(IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09516100 T  0121:1
                 JUMP OUT 3 TO XYT2 ELSE JUMP OUT 3 TO XYT2));%                        09516200 T  0123:1
                    IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09516300 T  0126:0
                 JUMP OUT 1 TO XYT2 ELSE JUMP OUT 1 TO XYT2);%                         09516400 T  0127:2
           GO TO L1;                                                                   09516500 T  0129:0
      XYT2: GO TO XYT;                                                                 09516600 T  0129:1
      DONE1: GO TO DONE;                                                               09516700 T  0129:2
      L1:     P2(63(IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09516800 T  0129:3
                 JUMP OUT 2 TO XYT ELSE JUMP OUT 2 TO XYT));%                          09516900 T  0132:0
                 P2(IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09517000 T  0134:1
                 JUMP OUT 1 TO XYT ELSE JUMP OUT  1 TO XYT);%                          09517100 T  0136:1
                 P1(IF SC=ALPHA THEN IF SC≤"Z" THEN SI←SI+1 ELSE%                      09517200 T  0137:3
                 JUMP OUT 1 TO XYT ELSE JUMP OUT 1 TO XYT);%                           09517300 T  0139:3
      XYT: P8(IF TOGGLE THEN ELSE TALLY←1; JUMP OUT 1 TO STOR);%                       09517400 T  0141:1
           IF TOGGLE THEN TALLY←1;%                                                    09517500 T  0143:1
      STOR: P0←TALLY;%                                                                09517600 T  0143:3
```

```
     DONE:%                                                                    09517700 T  0144:0
     END STREAM;%                                                              09517800 T  0144:0
END;%                                                                          09517900 T  0144:1
     IF TYPE≠10 THEN P(RTN);%                                                  09518000 T  0144:1
     IF DIFFER>0 THEN                                                          09518100 T  0145:3
BEGIN                        % FILL OUT DEST WITH BLANKS TO MAKE UP DIFF       09518200 T  0146:2
     P(SLENGTH+DOFSET,DUP,8,IDV,*P(.D),INX,.D,←,7,LND,.DOFSET,←);              09518300 T  0147:0
SMEAR::NDIV64 ←(NWDS←(((DIFFER←(DLENGTH-SLENGTH)-%                             09518500 T  0150:3
          (N←(8-DOFSET).[45:3])) DIV 8) - (DIFFER≥8))) DIV 64;%                09518600 T  0151:3
     STREAM(P1←DIFFER.[45:3],P2←DOFSET,P3←8×(DIFFER≥8)+N,P4←NWDS,               09518700 T  0157:1
          P4A←NWDS≠0,P5←NDIV64,P5A←NDIV64≠0,P6←SMCHR,P7←(TYPE=3                09518800 T  0160:2
          AND SMCHR≠" "),P8←D);%                                               09518850 T  0162:3
     BEGIN                                                                     09518900 T  0165:0
          DI ← DI+P2; P8←DI; P7(SI←LOC P7; SI←SI-1);%                          09519000 T  0165:0
          CI←CI+P7; GO TO BLNK; GO TO SMR;%                                    09519100 T  0167:0
     BLNK:P3(DS ← LIT " "); GO TO CONT;%                                       09519200 T  0168:0
     SMR: P3(DS ← 1 CHR; SI←SI-1);%                                            09519300 T  0169:2
     CONT:SI ← P8; P5(DS ← 63 WDS); P5A(DS ← P5 WDS);%                         09519400 T  0170:3
          P4A(DS ← P4 WDS);%                                                   09519450 T  0173:1
          CI←CI+P7; GO TO FINB; GO TO FINS;%                                   09519500 T  0174:2
     FINB:P1(DS ← LIT " "); GO TO XYT;%                                        09519600 T  0175:2
     FINS:P1(DS ← 1 CHR; SI←SI-1);%                                            09519700 T  0177:0
     XYT:%                                                                     09519800 T  0178:1
     END STREAM;                                                               09519900 T  0178:1
END;%                                                                          09520000 T  0178:2
     IF RELATE>0 THEN                % BLANK FILL DONE                         09520100 T  0178:2
          BEGIN                      % GO BACK AND DO COMPARE                  09520200 T  0179:1
               SOFSET ← SAVOFF;%                                               09520300 T  0179:3
               SOURCE ← [DEST];%                                               09520400 T  0180:2
               SLENGTH ← DLENGTH;%                                             09520500 T  0181:2
               TYPE ← RELATE;%                                                 09520600 T  0182:1
               DOFSET ← 0;%                                                    09520650 T  0183:0
               D ← [DESC];%                                                    09520660 T  0183:3
               GO TO CMD;%                                                     09520700 T  0184:3
          END;%                                                                09520800 T  0185:1
     P(XIT);                                                                   09520900 T  0185:1
VARIEXAM::%                                                                    09521000 T  0185:2
     NMOD64 ← LNGTH.[42:6];%                                                   09521100 T  0186:0
     NDIV64 ← LNGTH DIV 64;%                                                   09521200 T  0187:1
     IF TYPE=0 THEN                                                            09521300 T  0188:2
BEGIN                                % REPLACING FIRST                         09521400 T  0189:1
     STREAM(LISTP1TOP6,P7←SOURCE);%                                            09521500 T  0189:3
     BEGIN                                                                     09521600 T  0192:3
          DI←DI+P6; SI←LOC P5; SI←SI-1;                                        09521700 T  0192:3
          P3(63(63(IF SC=DC THEN JUMP OUT 3 TO REP; SI←SI-1));%                09521800 T  0193:3
              2(63(IF SC=DC THEN JUMP OUT 3 TO REP; SI←SI-1));%                09521900 T  0197:0
                   IF SC=DC THEN JUMP OUT 1 TO REP; SI←SI-1; );%               09522000 T  0199:3
          P2(63(IF SC=DC THEN JUMP OUT 2 TO REP; SI←SI-1));%                   09522100 T  0201:1
             P2(IF SC=DC THEN JUMP OUT 1 TO REP; SI←SI-1);%                    09522200 T  0204:0
             P1(IF SC=DC THEN JUMP OUT 1 TO REP; SI←SI-1);%                    09522300 T  0206:0
          GO TO XYT;                                                           09522400 T  0208:0
     REP: REPLACECHR;                                                          09522500 T  0208:1
     XYT:                                                                      09522600 T  0209:1
     END STREAM;                                                               09522700 T  0209:1
END ELSE IF TYPE=1 THEN                                                        09522800 T  0209:2
BEGIN                                % REP AND/OR TALLYING ALL                 09522900 T  0210:3
     STREAM(P0←0:LISTP1TOP6,P7←3-CODE)"0=REP&TALLY,1=TALLY ONLY,               09523000 T  0211:1
```

```
      2=REP ONLY"(P8←SOURCE);%                                                                  09523100 T  0214:1
BEGIN                                                                                           09523200 T  0215:2
     DI←DI+P6; SI←LOC P5; SI←SI-1;%                                                             09523300 T  0215:2
     P3(63(63(IF 1 SC=DC THEN%                                                                  09523400 T  0216:2
          BEGIN CI←CI+P7; GO TO TALL1; GO TO TALL1; GO TO REP1;                                 09523500 T  0218:0
          TALL1: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09523600 T  0219:1
                 CI←CI+P7; GO TO REP1; GO TO NXT1;%                                             09523700 T  0220:1
          REP1:  REPLACECHR; SI←LOC P5;%                                                        09523800 T  0221:1
          END;%                                                                                 09523900 T  0222:2
          NXT1:  SI←SI-1;))));%                                                                 09524000 T  0222:2
     P3( 2(63(IF 1 SC=DC THEN%                                                                  09524100 T  0223:2
          BEGIN CI←CI+P7; GO TO TALL2; GO TO TALL2; GO TO REP2;                                 09524200 T  0225:0
          TALL2: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09524300 T  0226:1
                 CI←CI+P7; GO TO REP2; GO TO NXT2;%                                             09524400 T  0227:1
          REP2:  REPLACECHR; SI←LOC P5;%                                                        09524500 T  0228:1
          END;%                                                                                 09524600 T  0229:2
          NXT2:  SI←SI-1;))));%                                                                 09524700 T  0229:2
     P3(      IF 1 SC=DC THEN%                                                                  09524800 T  0230:2
          BEGIN CI←CI+P7; GO TO TALL3; GO TO TALL3; GO TO REP3;                                 09524900 T  0231:2
          TALL3: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09525000 T  0232:3
                 CI←CI+P7; GO TO REP3; GO TO NXT3;%                                             09525100 T  0233:3
          REP3:  REPLACECHR; SI←LOC P5;%                                                        09525200 T  0234:3
          END;%                                                                                 09525300 T  0236:0
          NXT3:  SI←SI-1;));%                                                                   09525400 T  0236:0
        P2(63(IF 1 SC=DC THEN%                                                                  09525500 T  0236:2
          BEGIN CI←CI+P7; GO TO TALL4; GO TO TALL4; GO TO REP4;                                 09525600 T  0237:3
          TALL4: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09525700 T  0239:0
                 CI←CI+P7; GO TO REP4; GO TO NXT4;%                                             09525800 T  0240:0
          REP4:  REPLACECHR; SI←LOC P5;%                                                        09525900 T  0241:0
          END;%                                                                                 09526000 T  0242:1
          NXT4:  SI←SI-1;));%                                                                   09526100 T  0242:1
           P2(IF 1 SC=DC THEN%                                                                  09526200 T  0243:0
          BEGIN CI←CI+P7; GO TO TALL5; GO TO TALL5; GO TO REP5;                                 09526300 T  0244:0
          TALL5: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09526400 T  0245:1
                 CI←CI+P7; GO TO REP5; GO TO NXT5;%                                             09526500 T  0246:1
          REP5:  REPLACECHR; SI←LOC P5;%                                                        09526600 T  0247:1
          END;%                                                                                 09526700 T  0248:2
          NXT5:  SI←SI-1;);%                                                                    09526800 T  0248:2
           P1(IF 1 SC=DC THEN%                                                                  09526900 T  0249:0
          BEGIN CI←CI+P7; GO TO TALL6; GO TO TALL6; GO TO REP6;                                 09527000 T  0250:0
          TALL6: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                             09527100 T  0251:1
                 CI←CI+P7; GO TO REP6; GO TO NXT6;%                                             09527200 T  0252:1
          REP6:  REPLACECHR; SI←LOC P5;%                                                        09527300 T  0253:1
          END;%                                                                                 09527400 T  0254:2
          NXT6:  SI←SI-1;);%                                                                    09527500 T  0254:2
     END STREAM;                                                                                09527600 T  0255:0
END ELSE                                                                                        09527700 T  0255:1
BEGIN                                          %REP/TALLY UNTIL 1ST/LEADING                     09527800 T  0255:1
     STREAM(P0←0:LISTP1TOP6,P7←3-CODE.[46:2],P8←CODE.[45:1],%                                    09527900 T  0255:3
          P9←SOURCE);%                                                                          09528000 T  0260:2
     BEGIN                                                                                      09528100 T  0261:1
          DI←DI+P6; SI←LOC P5; SI←SI-1;%                                                        09528200 T  0261:1
          P3(63(63(CI←CI+P8; GO TO REPL1; GO TO REPUF1;%                                        09528300 T  0262:1
               REPL1: IF 1SC≠DC THEN JUMP OUT 3 TO XYT1;GO TO DOIT1;                            09528400 T  0264:1
               REPUF1: IF 1SC=DC THEN JUMP OUT 3 TO XYT1;%                                      09528500 T  0266:0
               DOIT1: CI←CI+P7; GO TO TALL1; GO TO TALL1;GO TO REP1;                            09528600 T  0267:2
               TALL1: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                                        09528700 T  0268:3
```

```
                CI←CI+P7; GO TO REP1; GO TO NXT1;%                                09528800 T  0269:3
          REP1:  REPLACECHR; SI←LOC  P5;%                                         09528900 T  0270:3
          NXT1:  SI←SI+1 )));  GO TO L2;       XYT1: GO TO XYT2;                  09529000 T  0272:0
L2:  P3( 2(63(CI←CI+P8; GO TO REPL2; GO TO REPUF2;%                               09529100 T  0273:2
          REPL2: IF 1SC≠DC THEN JUMP OUT 3 TO XYT2;GO TO DOIT2;                   09529200 T  0275:2
          REPUF2:IF 1SC=DC THEN JUMP OUT 3 TO XYT2;%                              09529300 T  0277:1
          DOIT2: CI←CI+P7; GO TO TALL2; GO TO TALL2;GO TO REP2;                   09529400 T  0278:3
          TALL2: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                               09529500 T  0280:0
                 CI←CI+P7; GO TO REP2; GO TO NXT2;%                               09529600 T  0281:0
          REP2:  REPLACECHR; SI←LOC  P5;%                                         09529700 T  0282:0
          NXT2:  SI←SI+1 )));  GO TO L3;       XYT2: GO TO XYT3;                  09529800 T  0283:1
L3:  P3(      CI←CI+P8; GO TO REPL3; GO TO REPUF3;%                               09529900 T  0284:3
          REPL3: IF 1SC≠DC THEN JUMP OUT 1 TO XYT3;GO TO DOIT3;                   09530000 T  0286:1
          REPUF3:IF 1SC=DC THEN JUMP OUT 1 TO XYT3;%                              09530100 T  0287:2
          DOIT3: CI←CI+P7; GO TO TALL3; GO TO TALL3;GO TO REP3;                   09530200 T  0288:2
          TALL3: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                               09530300 T  0289:3
                 CI←CI+P7; GO TO REP3; GO TO NXT3;%                               09530400 T  0290:3
          REP3:  REPLACECHR; SI←LOC  P5;%                                         09530500 T  0291:3
          NXT3:  SI←SI-1 ); GO TO L4;          XYT3: GO TO XYT4;                  09530600 T  0293:0
L4:     P2(63(CI←CI+P8; GO TO REPL4; GO TO REPUF4;%                               09530700 T  0294:0
          REPL4: IF 1SC≠DC THEN JUMP OUT 2 TO XYT4;GO TO DOIT4;                   09530800 T  0295:3
          REPUF4:IF 1SC=DC THEN JUMP OUT 2 TO XYT4;%                              09530900 T  0297:1
          DOIT4: CI←CI+P7; GO TO TALL4; GO TO TALL4;GO TO REP4;                   09531000 T  0298:2
          TALL4: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                               09531100 T  0299:3
                 CI←CI+P7; GO TO REP4; GO TO NXT4;%                               09531200 T  0300:3
          REP4:  REPLACECHR; SI←LOC  P5;%                                         09531300 T  0301:3
          NXT4:  SI←SI-1 )); GO TO L5;         XYT4: GO TO XYT5;                  09531400 T  0303:0
L5:      P2(CI←CI+P8; GO TO REPL5; GO TO REPUF5;%                                 09531500 T  0304:1
          REPL5: IF 1SC≠DC THEN JUMP OUT 1 TO XYT5;GO TO DOIT5;                   09531600 T  0305:3
          REPUF5:IF 1SC=DC THEN JUMP OUT 1 TO XYT5;%                              09531700 T  0307:0
          DOIT5: CI←CI+P7; GO TO TALL5; GO TO TALL5;GO TO REP5;                   09531800 T  0308:0
          TALL5: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                               09531900 T  0309:1
                 CI←CI+P7; GO TO REP5; GO TO NXT5;%                               09532000 T  0310:1
          REP5:  REPLACECHR; SI←LOC  P5;%                                         09532100 T  0311:1
          NXT5:  SI←SI-1 ); GO TO L6;          XYT5: GO TO XYT;                   09532200 T  0312:2
     L6:   P1(CI←CI+P8; GO TO REPL6; GO TO REPUF6;%                               09532300 T  0313:2
          REPL6: IF 1SC≠DC THEN JUMP OUT 1 TO XYT; GO TO DOIT6;                   09532400 T  0315:0
          REPUF6:IF 1SC=DC THEN JUMP OUT 1 TO XYT;%                               09532500 T  0316:1
          DOIT6: CI←CI+P7; GO TO TALL6; GO TO TALL6;GO TO REP6;                   09532600 T  0317:1
          TALL6: SI←P0; SI←SI+8; P0←SI; SI←LOC P5;%                               09532700 T  0318:2
                 CI←CI+P7; GO TO REP6; GO TO NXT6;%                               09532800 T  0319:2
          REP6:  REPLACECHR; SI←LOC  P5;%                                         09532900 T  0320:2
          NXT6:  SI←SI-1 );%                                                      09533000 T  0321:3
     XYT:%                                                                        09533100 T  0322:1
     END STREAM;%                                                                 09533200 T  0322:1
END;%                                                                             09533300 T  0322:2
     IF CODE.[46:1] THEN P(RTN);%                                                 09533400 T  0322:2
END COBOLVARSZ;%                                                                  09533500 T  0324:0
                                                                                        SIZE= 0325 WORDS


PROCEDURE COBOLIONONDSK;     % PRONOUNCED COBOL-IO-NON-DISK                       09600000 T  0000:0
                                                                                START OF REL SEGMENT; DISK ADDRESS = 00648
     BEGIN                                                                       09600100 T  0000:0
     REAL CODE      = -1;     % 0=READ,1=WRITE,6=WRTBLK                          09600200 T  0000:0
```

```
                NAME DLOC       = -2;     % POINTS TO BUFFER IO DESCRIPTOR
                REAL NUMWDS     = -3,     % # WDS TO BE WRITTEN
                     KEY        = -4,     % CARRIAGE RETURN
                     CHNNL      = -4,     % LP CHANNEL SKIP
                     LINES      = -5,     % # LINES TO BE SPACED
                     SKIPBFR    = -6;     % 1=SPACE BEFORE PRINT
INTEGER
                     LINAGE     = -7;     % LINE PRINTER: [1:1]=1 IF LINAGE
                                          % CLAUSE PRESENT,[33:15]= LINAGE LIMIT
                                          % ON NEXT END-OF-PAGE
%LOCALS
        REAL  IOMASK;
        ARRAY FIB [*];                    % FIB ARRAY
        REAL  FILECTRL   = 12,            % USED TO CALL COBOLFCR
              PERFORMGEN= 13,             % USED FOR PERFORMING USE ROUTINES
              COBOLIODSK= 15;
        NAME  FLOC;                       % POINTER TO FIB
        ARRAY FPB        = 3[*];          % FILE PARAMETER BLOCK
        NAME  MEM        =  2;            % DUMMY DATA DESC
        ARRAY PGUSE      = 24[*];         % PROGRAM USE ROUTINES
         REAL
              T,RT,                       % TEMPORARY
              TCW,                        % TECH C: NUMBER WORDS TO BE READ
              TCDIF,                      % TECH C: (ACTUAL RECORD - MIN REC)
              UNITYPE,                    % STORE UNIT TYPE FOR MANY TESTS
              ENDREEL;                    % USED ONLY ON READ
        ARRAY DEST[*];                    % DESTINATION IN MOVEREC
DEFINE
AF              = [12:12]#,               % FILE USE ROUTINE
ARR             = [36:12]#,               % REEL USE ROUTINE
ARROW           = P(0,NOT,(BUFFSIZE-WORDSLEFT),TIP,INX,STD)#,
                        % THIS INSERTS THE GROUP MARK
BCOUNT          = FIB[6]#,                % BLOCK COUNT
BINARY          = FIB[13].[24:1]#,        % 1=BINARY,0=ALPHA
BF              = [1:11]#,                % FILE USE ROUTINE
BREAK           = FIB[9] ≠ 0 #    ,       % BREAKOUT RESTART POINT
BREAKOUT        = IF(RCOUNT MOD FIB[9])=0 THEN
                    P(0,0,12,COM,DEL,DEL)#,% CALL BREAKOUT
BRR             = [24:12]#,               % REEL USE ROUTINE
BUFFNUM         = FIB[13].[1:9] #,        % # OF BUFFS REQUSTED
BUFFSIZE        = FIB[18].[3:15]#,        % BUFFER SIZE  (REQUESTED)
BUFFSZ          = FIB[18][8:8:10]#,       % SIZE FOR CONCATINATES
BUFTOP          = FIB[16]#,% COPY OF TOP IOD: POINTS TO BEG BUFF
CHECK(CHECK1)   = IF P(DUP)≠(CHECK1) THEN P(CHECK1,0,FLOC,#,
 ONERR(ONERR1)  =   ONERR1,17,COM,DEL,DEL,DEL,DEL);    P(DEL)#,
                        % THE ABOVE ARE USED ON BLOCK+REC CHKS
CLOSEANDOPEN    =P(MKS,1,0,FLOC,4,FILECTRL, %CLOSE NO RWD
                   MKS,FLOC,1,FILECTRL)#,   % OPEN INPUT
COUNT           = FIB[12] #,              % USED FOR BLOCKING TECH-A,B
DELAY           = TIP.[20:1] #,           % THIS ALLOWS ONE CYCLE DELY
DISK            = (UNITYPE←(FIB[4].[8:4]))=4#,
DONE            = TIP.[19:1] #,           % 1= IO COMPLETED
ENDFILE         = FIB[5].[40:1] #,        % ALREADY PASSED EOF
ENDPROCESS      = FIB[5].[39:2]#,         % SEE OPTIONAL AND ENDFILE
EOF             =((*DLOC).[27:1])#,       % FIRST EOF OR EOT
FNAM            = FIB[4].[13:11]#,        % FILE NAME INDEX IN FPB
FOREVER         =(NOT 0).[9:39] #,        % UNTIL END TIME
```

09600300 T 0000:0
09600400 T 0000:0
09600500 T 0000:0
09600600 T 0000:0
09600700 T 0000:0
09600800 T 0000:0
09600900 T 0000:0
09601000 T 0000:0
09601100 T 0000:0
09601200 T 0000:0
09601300 T 0000:0
09601400 T 0000:0
09601500 T 0000:0
09601600 T 0000:0
09601700 T 0000:0
09601800 T 0000:0
09601900 T 0000:0
09602000 T 0000:0
09602100 T 0000:0
09602200 T 0000:0
09602300 T 0000:0
09602400 T 0000:0
09602500 T 0000:0
09602600 T 0000:0
09602700 T 0000:0
09602800 T 0000:0
09602900 T 0000:0
09603000 T 0000:0
09603100 T 0000:0
09603200 T 0000:0
09603300 T 0000:0
09603400 T 0000:0
09603500 T 0000:0
09603600 T 0000:0
09603700 T 0000:0
09603800 T 0000:0
09603900 T 0000:0
09604000 T 0000:0
09604100 T 0000:0
09604200 T 0000:0
09604300 T 0000:0
09604400 T 0000:0
09604500 T 0000:0
09604600 T 0000:0
09604700 T 0000:0
09604800 T 0000:0
09604900 T 0000:0
09605000 T 0000:0
09605100 T 0000:0
09605200 T 0000:0
09605300 T 0000:0
09605400 T 0000:0
09605500 T 0000:0
09605600 T 0000:0
09605700 T 0000:0
09605800 T 0000:0
09605900 T 0000:0

```
HOWOPEN          = FIB[5].[41:3]#,    % 1=OPEN INPUT,0= OPEN OUTPT
                                      % 1 > CLOSED
INFILE           = FIB[13].[27:1]#,    % FILE OPEN INPUT
INVALIDUSER      = FIB[5]<0#,         % INVALID USER NOT PARITY
INXLINAGE        = P(LOCOFCTR,DUP,LOD,LINES,ADD,XCH,+)#,
IOERR(IOERR1)    = P(0,FLOC,IOERR1,17,COM,DEL,DEL,DEL)#,
                      % ABOVE CALLS IOERROR ROUTINE
LABELED          = NOT FIB[4].[2:1]#,
LABEQ            = FIB[5].[17:1] #,  % LABEL EQUATED FROM DISK
LBLPTR           = FLOC[1] #,        % LABEL POINTER
LINAGELIM        = FIB[1]#,     % LOGICAL LENGTH OF PRINTED PAGE
LINEPRINT        = FIB[20] #,         % CF=1 IS PRINTFILE
LINTOG           = LINAGE.[1:1]#,     % TRUE IF LINAGE PRESENT
LOCOFCTR         = FIB[3]#,           % PRT LOC OF LINAGE COUNTER
MABUSE           = FIB[4].[1:1]#,     % MAY BE USE RTNS PRESENT
MAXR             = FIB[18][8:38:10]#,% MAX REC SZ FOR CONCATS
MAXREC           = FIB[18].[33:15]#, % MAX REC SZ
MINREC           = FIB[18].[FF] #,    % MINIMUM RECORD SIZE
NONSTD           = FIB [5].[16: 1]#, % NON-STANDARD LABELS
NUMBUF           = FIB[13].[10: 9]#, % NUMBER OF BUFFERS ASSIGNED
NUMREC           = FIB[11] #,         % RECORDS PER BLOCK
NXTLINAGE        = LINAGE.[33:15]#,    % PRINTER: LINAGE LIMIT
NXTREEL          = P(MKS,2,1,FLOC,4, % THIS DOES REEL SWITCHING
                   FILECTRL)#,       %
OPTIONAL         = FIB[ 5].[39:1]#,  % OPTIONAL FILE NOT PRESENT
PARITY           = TIP.[28:1]#,       % PARITY BIT ON DESC
PBIT             = [2:1]#,            % PRESENCE BIT
PRESENT          =((*DLOC).[2:1])#,  % CHECKS PRESENTSBIT
PROPER           =P(CODE,P(DUP),+,P(DUP)=12,+,REVERSE,+,21,+) #,
                        % GENERATES PROPER IOERROR
      PUNCH      = UNITYPE=6#,  % UNIT IS CARD PUNCH
      RCOUNT     = FIB[7] #,          % RECORD COUNT
      RCPRT      = (FIB[20].[FF])#,  %PRT OF DESC POINTING TO REC
      READER     =(UNITYPE MOD 11=0)#,% 0=READER 11=PSUDOREADER
      READLBL    =P(DLOC INX 0,11,11  % THIS READS THE LABEL.
                   ,COM,DEL,DEL)#,    %
      RECPERBLK  = H[0].[30:12] #,    % RECORDS PER BLOCK
$ SET OMIT = TIMESHARING
      REMOTEUNIT =10#,                % DATACOM IS TYPE 14 ON TSS
      REMOTEREAD =BEGIN P(BUFFSIZE,TIP,1,(-13),COM);
                     P(TIP); MOVEREC;
                       P([DLOC[0]],:=,1,SUB,RTN);
                   END#,
      REMOTEWRIT =BEGIN P(TIP); MOVEREC;
                     P([DLOC[0]],STN);   % RESTORE TIP
                     P(NUMWDS×8,LINES&KEY[CTF],0,(-11),COM,
                     DEL,RTN);
                  END#,
$ POP OMIT OMIT
      REVERSE    = FIB[5].[44:1] #,  % 1=REVERSE
      SETPRESENCEBIT =P(TIP OR MEM ,DLOC,+)#,% SET PRESENCE BIT
$ SET OMIT = NOT(TIMESHARING)
      SLEEP      = 36 #,
$ POP OMIT
$ SET OMIT = TIMESHARING
      TAPEE      = TIP.[7:1] #,       % 1= TAPES   0=ALL ELSE
      TECHA      =(FIB[5].[46:2]=1) #,% TECHNIQUE=A
```

```
09606000 T  0000:0
09606100 T  0000:0
09606200 T  0000:0
09606300 T  0000:0
09606400 T  0000:0
09606500 T  0000:0
09606600 T  0000:0
09606700 T  0000:0
09606800 T  0000:0
09606900 T  0000:0
09607000 T  0000:0
09607100 T  0000:0
09607300 T  0000:0
09607400 T  0000:0
09607500 T  0000:0
09607600 T  0000:0
09607700 T  0000:0
09607800 T  0000:0
09607900 T  0000:0
09608000 T  0000:0
09608100 T  0000:0
09608200 T  0000:0
09608300 T  0000:0
09608400 T  0000:0
09608500 T  0000:0
09608600 T  0000:0
09608700 T  0000:0
09608800 T  0000:0
09608900 T  0000:0
09608910 T  0000:0
09609000 T  0000:0
09609100 T  0000:0
09609200 T  0000:0
09609300 T  0000:0
09609400 T  0000:0
09609500 T  0000:0
09609600 T  0000:0
09609690 T  0000:0
09610100 T  0000:0
09610200 P  0000:0
09610220 T  0000:0
09610240 P  0000:0
09610260 T  0000:0
09610300 T  0000:0
09610320 T  0000:0
09610340 T  0000:0
09610360 T  0000:0
09610380 T  0000:0
09610390 T  0000:0
09610400 T  0000:0
09610500 T  0000:0
09610600 T  0000:0
09610700 T  0000:0
09610701 T  0000:0
09610800 T  0000:0
09611000 T  0000:0
09611100 T  0000:0
```

```
        TECHC           =(FIB[5],[46:2]=3) #,% TECHNIQUE=C                    09611200 T  0000:0
        TERM(TERM1)     = P(1,FLOC,TERM1,17,COM)#,%TERMINATE I/O ERROR        09611300 T  0000:0
        TIP             = (*DLOC) #,          % LOAD I/O DESC                 09611400 T  0000:0
        TOSZF           = [8:38:10]#,                                        09611500 T  0000:0
        UNBLKD          = (FIB[5],[46:2]=0)#, % 1 RECORD PER BLOCK            09611600 T  0000:0
        WAITIO          = P(DLOC,IOMASK,      % THIS SLEEPS ON I/O            09611700 T  0000:0
                          SLEEP,COM,DEL,DEL)#,% WAITING FOR A COMPLETE        09611800 T  0000:0
        WRITEPARITY     = FIB[5].[3:1]#,    % INDICATES FORCED REELSWITCH     09611810 T  0000:0
        WORDSLEFT       = FIB[17]#;           % WORDS LEFT IN BUFFER           09611900 T  0000:0
LABEL LPRETURN,START,IMPROPER,ROVER,EOFSETCK;                                09612000 T  0000:0
SUBROUTINE GOUSE;       % CALLS USE ROUTINES                                 09612100 T  0000:0
    BEGIN P(MKS,[FIB],T,0,PERFORMGEN); END;%                                 09612200 T  0001:0
SUBROUTINE INPUTPARITY;%                                                     09612300 T  0002:3
    BEGIN%                                                                   09612400 T  0003:0
        IF (T ← RT ← PGUSE[4].BRR)≠0 THEN GOUSE; % INPUT ERROR USE RTN       09612600 T  0003:0
        IF (T ← FIB[15].BF) ≠ 0 THEN GOUSE;                                  09612700 T  0007:0
        IF NOT PRESENT THEN IF NOT (T OR RT) THEN                            09612800 T  0011:0
            IOERR(19 + 10 × REVERSE);                                        09612850 T  0013:3
        SETPRESENCEBIT;                                                      09612900 T  0018:0
    END INPUTPARITY;%                                                        09613000 T  0019:2
SUBROUTINE OUTPUTERROR;%                                                     09613100 T  0019:3
    BEGIN%                                                                   09613200 T  0020:0
        IF NOT EOF THEN % TAPE WRITE PARITY OR BLANK TAPE                    09613300 T  0020:0
            BEGIN     % OUTPUT ERROR USE ROUTINES                            09613400 T  0021:1
                IF (T ← PGUSE[5].BRR)≠0 THEN GOUSE;%                         09613500 T  0021:3
            IF (T ← FIB[15].BF) ≠ 0 THEN GOUSE;                              09613600 T  0025:0
                TERM(20);%                                                   09613700 T  0029:0
            END;%                                                            09613800 T  0030:1
        SETPRESENCEBIT;%                                                     09613900 T  0030:1
        NXTREEL;  % REEL SWITCH                                              09614000 T  0031:3
    END OUTPUTERROR;%                                                        09614100 T  0033:1
SUBROUTINE  INPUTEOFEOR;                                                     09614200 T  0033:2
        BEGIN     % EOF OR EOR                                               09614300 T  0034:0
            ENDFILE ← TRUE;%                                                 09614400 T  0034:0
            SETPRESENCEBIT;%                                                 09614500 T  0036:2
            IF READER OR REVERSE THEN P(1,RTN);%                             09614600 T  0038:0
            IF LABELED THEN%                                                 09614700 T  0041:2
                BEGIN%                                                       09614800 T  0042:3
                READLBL;%                                                    09614900 T  0043:1
                   STREAM(SENT←0,BC←0,RC←0,WP←0:L←5 INX LBLPTR);             09615000 T  0045:1
                     BEGIN          % THIS RETREIVES END OF REEL             09615100 T  0048:1
                     DI←LOC SENT;   % SENTINEL,BLOCK & REC COUNT             09615200 T  0048:2
                     DI←DI+7; SI←L; SI←SI-1;%                                09615300 T  0048:3
                     DS←CHR; DS←5 OCT; DS←7 OCT;%                            09615400 T  0049:2
                     DI←DI+7; DS← CHR;                                       09615410 T  0050:1
                     END;%                                                   09615500 T  0050:3
                IF P=1 THEN WRITEPARITY ← TRUE;                              09615510 T  0051:0
                CHECK(RCOUNT) ONERR(16);                                     09615600 T  0054:2
                CHECK(BCOUNT) ONERR(17);                                     09615700 T  0059:0
                ENDREEL ← P;          % STORE SENTINEL                      09615800 T  0063:2
                IF MABUSE THEN%                                              09615900 T  0064:0
                IF NOT WRITEPARITY THEN                                      09615950 T  0065:0
                    BEGIN             % END INPUT REEL USE RTNS               09616000 T  0066:3
                        IF (T←PGUSE[1].BRR)≠0 THEN GOUSE;%                   09616100 T  0067:1
                        IF (T←PGUSE[1].ARR)≠0 THEN GOUSE;%                   09616200 T  0071:0
                        IF NOT ENDREEL THEN%                                 09616300 T  0075:0
                            BEGIN % END INPUT FILE USE RTNS                  09616400 T  0075:2
```

```
                                    IF (T←PGUSE[1].BF)≠0 THEN GOUSE;          09616500 T  0076:0
                                    IF (T←PGUSE[1].AF)≠0 THEN GOUSE;          09616600 T  0080:0
                                    END;%                                     09616700 T  0084:0
                                IF (T←FIB[2].BRR)≠0 THEN GOUSE; % END         09616800 T  0084:0
                                IF (T←FIB[2].ARR)≠0 THEN GOUSE; % REEL        09616900 T  0088:0
                                IF NOT ENDREEL THEN%                          09617000 T  0092:0
                                    BEGIN % END FILE USE ROUTINES%            09617100 T  0092:2
                                    IF (T←FIB[2].BF)≠0 THEN GOUSE;%           09617200 T  0093:0
                                    IF (T←FIB[2].AF)≠0 THEN GOUSE;%           09617300 T  0097:0
                                    END;%                                     09617400 T  0101:0
                            END USE;%                                         09617500 T  0101:0
                        END LABELED;%                                         09617600 T  0101:0
                    IF LABELED AND NOT ENDREEL THEN P(1,RTN);%                09617700 T  0101:0
                    IF NONSTD THEN%                                           09617800 T  0104:0
                        BEGIN%                                                09617900 T  0105:0
                        ENDFILE ← FALSE;%                                     09618000 T  0105:2
                        CLOSEANDOPEN;%                                        09618100 T  0108:0
                        P(1,RTN);%                                            09618200 T  0110:2
                        END;%                                                 09618300 T  0111:0
                    NXTREEL;%                                                 09618600 T  0111:0
                    P(DEL,DEL);       % DELETE BRANCH RETURNS                 09618700 T  0112:2
                    WRITEPARITY ← FALSE;                                      09618710 T  0113:0
                    IF TECHC THEN P(.TCW,LOD,,NUMWDS,STD);                    09618750 T  0115:2
                    GO TO START;%                                             09618800 T  0118:2
    END  INPUTEOFEOR;                                                         09618900 T  0119:0
SUBROUTINE MOVEREC;      % MOVES RECORD BETWEEN WORK AREA AND BUFFER          09619000 T  0119:1
    BEGIN%                                                                    09619100 T  0120:0
        IF NOT DONE THEN WAITIO;%                                             09619200 T  0120:0
        P(*RCPRT,TIP INX 0);                                                  09619300 T  0123:1
        IF NOT PRESENT THEN                  % MAY BE ERROR OR EOF             09619400 T  0125:3
            IF CODE THEN                                                      09619500 T  0127:0
                IF EOF THEN BEGIN OUTPUTERROR; P(DEL,TIP INX 0); END          09619600 T  0127:3
                    ELSE P(XCH,P(DUP),[8:10],,NUMWDS,ISD)                     09619700 T  0132:1
                ELSE IF EOF THEN INPUTEOFEOR;                                 09619800 T  0134:1
        DEST ← IF CODE THEN P ELSE P(XCH);%                                   09619900 T  0137:0
        STREAM(FROM←P:NUMWDS,E←P(DUP),[36:6],X←DEST);%                        09620000 T  0139:0
            BEGIN%                                                            09620100 T  0141:1
                SI←FROM;E(DS←32 WDS;DS←32 WDS); DS←NUMWDS WDS;%               09620200 T  0141:1
            END;%                                                             09620300 T  0143:1
        P(DEL);%                                                              09620400 T  0143:2
        WORDSLEFT ← *P(DUP) - NUMWDS;                                         09620500 T  0143:3
        DLOC[0] ← (IF REVERSE THEN NOT(NUMWDS-1) ELSE NUMWDS) INX TIP;        09620600 T  0145:3
        RCOUNT ← *P(DUP) + 1;                                                 09620700 T  0150:1
        IF CODE THEN                                  % CHECK FOR             09620800 T  0152:1
            IF NOT PRESENT THEN OUTPUTERROR           % OUTPUT PARITY ERROR   09620900 T  0152:2
        ELSE ELSE                                                             09621000 T  0155:2
            IF NOT PRESENT THEN INPUTPARITY; % INPUT PARITY                   09621100 P  0156:2
        IF BREAK THEN BREAKOUT;                                               09621200 T  0160:0
    END MOVERECORDTOANDFROMWORKAREA;                                          09621300 T  0165:1
SUBROUTINE PREL;         % DOES ACTUAL I/O                                   09621700 T  0165:2
    BEGIN%                                                                    09621800 T  0166:0
        P(TIP,DLOC,PRL,DEL);              % DO IO                           09621900 T  0166:0
        BCOUNT ← *P(DUP) + 1;             % UP BLOCK COUNT                  09622100 T  0167:2
    END PREL;%                                                                09622300 T  0169:2
SUBROUTINE  SKIPPER;                 % DOES SPACING ON PRINTER              09622400 T  0169:3
    BEGIN                                                                     09622500 T  0170:0
        WHILE LINES > 0 DO                                                    09622600 T  0170:0
```

```
                    BEGIN                                                                09622700 T  0171:1
                    IF NOT DONE THEN WAITIO;                                             09622800 T  0171:1
                    IF NOT PRESENT THEN OUTPUTERROR;                                     09622900 T  0174:2
                    DLOC[0] ← TIP & 1[18:47:1] & 16[27:42:6];                            09623000 T  0177:0
                    IF LINES = 1 THEN                                                    09623100 T  0180:0
                         DLOC[0] ← TIP & 2[27:46:2];                                     09623200 T  0180:3
                    PREL;                                                                09623300 T  0183:1
                    LINES ← LINES - 2;                                                   09623400 T  0184:0
                    END;                                                                 09623500 T  0185:1
               END SKIPPINGALLTHOSELINES;                                                09623600 T  0185:3
          SUBROUTINE  GOLP;                    % MAKES THY PRYNTER GO                   09624600 T  0186:0
               BEGIN                                                                     09624700 T  0186:0
                    IF LINTOG THEN INXLINAGE;                                            09624800 T  0186:0
             IF NUMWDS > 17 THEN NUMWDS ← 17;                                            09624810 C  0189:1
              IF MAXREC > 17 THEN MAXREC ← 17;                                           09624820 C  0191:1
                    RT ← BUFFSIZE - WORDSLEFT;     % ≠0 MEANS DATA PRESENT              09624900 T  0195:1
                    IF NOT UNBLKD THEN                                                   09625000 T  0197:2
                         BEGIN                                                           09625100 T  0199:0
                         IF TECHC THEN                                                   09625200 T  0199:2
                              BEGIN                                                      09625300 T  0201:0
                              IF NUMWDS > MAXREC THEN NUMWDS ← MAXREC;                   09625400 T  0201:2
                              IF NUMWDS ≤ 0 THEN TERM(36);                               09625500 T  0205:0
                              END;                                                       09625600 T  0207:2
                         IF NUMWDS > WORDSLEFT THEN SKIPBFR ← TRUE                       09625700 T  0207:2
                              ELSE BEGIN MOVEREC;  GO LPRETURN;  END;                    09625800 T  0209:1
                         END;                                                            09625900 T  0211:2
                    IF CHNNL ≠ 0 THEN LINES ← 0;                                         09626000 T  0211:2
                    IF SKIPBFR THEN                                                      09626100 T  0213:2
          BEGIN                                                                          09626200 T  0213:3
               IF NOT DONE THEN WAITIO;                                                  09626300 T  0214:1
               IF NOT PRESENT THEN OUTPUTERROR;                                          09626400 T  0217:2
               DLOC[0] ← FLAG(BUFTOP & (RT = 0) [18:47:1]                                09626500 T  0220:0
                              &RT   TOSZF                                                09626550 T  0221:1
                              &(LINES>0)[27:46:2] & CHNNL[29:44:4]);                     09626600 T  0222:1
               IF LINES = 1 THEN DLOC[0]←TIP & 2[27:46:2];                               09626700 T  0225:2
               PREL;                                                                     09626800 T  0229:2
               WORDSLEFT ← BUFFSIZE;                                                     09626850 T  0231:0
               IF (LINES ← LINES - 2) > 0 THEN SKIPPER;                                  09626900 T  0233:0
               IF UNITYPE=12 THEN IF NOT DONE THEN WAITIO;                               09627000 T  0236:0
               BUFTOP.[CF] ← TIP;                                                        09627100 T  0240:2
               MOVEREC;                                                                  09627200 T  0242:3
          END ELSE                                                                       09627300 T  0244:0
          BEGIN                                                                          09627400 T  0244:0
               IF RT ≠ 0 THEN                                                            09627500 T  0244:2
                    BEGIN                                                                09627600 T  0245:1
                    DLOC[0] ← FLAG(BUFTOP & 0[27:42:6]                                   09627700 T  0245:3
                                   & RT  TOSZF);                                         09627750 T  0246:2
                    PREL;                                                                09627800 T  0249:0
                    WORDSLEFT ← BUFFSIZE;                                                09627850 T  0250:0
                    IF UNITYPE=12 THEN IF NOT DONE THEN WAITIO;                          09627900 T  0252:0
                    BUFTOP.[CF] ← TIP;                                                   09628000 T  0256:2
                    END;                                                                 09628100 T  0258:3
               MOVEREC;                                                                  09628200 T  0258:3
               DLOC[0] ← FLAG(BUFTOP & (LINES>0)[27:46:2]                                09628300 T  0260:0
                              & (BUFFSIZE-WORDSLEFT) TOSZF                               09628350 T  0261:1
                              & CHNNL [29:44:4]);                                        09628400 T  0263:3
               IF LINES = 1 THEN DLOC[0]←TIP & 2[27:46:2];                               09628500 T  0266:1
```

```
                    PREL;                                                                                09628600 T  0269:2
                    WORDSLEFT ← BUFFSIZE;                                                                09628650 T  0271:0
                    IF (LINES ← LINES - 2) > 0 THEN SKIPPER;                                             09628700 T  0273:0
                    IF UNITYPE=12 THEN IF NOT DONE THEN WAITIO;                                          09628800 T  0276:0
                    BUFTOP.[CF] ← TIP;                                                                   09628900 T  0280:2
               END;                                                                                      09629100 T  0282:3
LPRETURN:                                                                                                09629200 T  0282:3
          IF LINTOG THEN IF (*P(LOCOFCTR))≥LINAGELIM THEN                                                09629300 T  0282:3
               BEGIN                                                                                     09629400 T  0285:2
                    P(0,LOCOFCTR,STD);                                                                   09629500 T  0286:0
                    LINAGELIM ← NXTLINAGE;                                                               09629600 T  0287:0
                    P(1,RTN);                                                                            09629700 T  0288:3
               END;                                                                                      09629800 T  0289:1
          P(0,RTN);                                                                                      09629900 T  0289:1
     END GOINTOPRINTER;                                                                                  09630000 T  0289:3
SUBROUTINE WRIT;                  % WRITES A BLOCK                                                       09630900 T  0290:0
     BEGIN                                                                                               09631000 T  0290:0
          DLOC[0] ← FLAG(BUFTOP & (BUFFSIZE-WORDSLEFT) TOSZF);                                           09631100 T  0290:0
          IF TAPEE THEN IF NOT BINARY THEN ARROW ELSE                                                    09631200 T  0293:3
               ELSE IF PUNCH THEN DLOC[0] ← TIP & CHNNL[32:47:1];                                        09631300 T  0300:1
          PREL;                                                                                          09631400 T  0304:2
          WORDSLEFT ← BUFFSIZE;                                                                          09631500 T  0306:0
          BUFTOP.[CF] ← TIP;                                                                             09631600 T  0308:0
     END WRIT;%                                                                                          09631700 T  0310:1
SUBROUTINE REED;             % READS A BLOCK                                                            09631800 T  0310:2
     BEGIN%                                                                                              09631900 T  0311:0
          DLOC[0] ← FLAG(FIB[16]);%                                                                      09632100 T  0311:0
          PREL;%                                                                                         09632200 T  0312:1
          BUFTOP.[CF] ← TIP;%                                                                            09632300 T  0313:0
          WORDSLEFT ← 0;                                                                                 09632500 T  0315:1
     END REED;%                                                                                          09632700 T  0316:2
%%%%%%%%%%%%%%%%%%%%%%%%%%%% START HERE %%%%%%%%%%%%%%%%%%%%%%%%%%%%                                    09632800 T  0316:3
          FIB ← *(FLOC ← (NOT 2) INX DLOC);%                                                             09632900 T  0316:3
            IF FPB[FNAM+3].[42:6]=43 THEN      % DUMMY                                                   09632910 C  0321:3
                 IF CODE=0 THEN GO EOFSETCK ELSE P(0,RTN);                                               09632920 C  0324:2
          IF DISK THEN GO TO P(COBOLIODSK);%                                                             09633000 T  0326:3
          IOMASK ← @2000000000;%                                                                         09633100 T  0329:3
   START:IF NOT(ENDPROCESS=0 OR CODE) THEN GO TO EOFSETCK;                                               09633200 T  0330:2
          IF CODE > 1 THEN                 % SHOULD BE WRITE BLOCK                                      09633300 T  0333:0
               BEGIN                                                                                     09633400 T  0333:3
                    IF CODE ≠ 6 THEN TERM(25);     % UNRECOGNIZED CODE                                   09633500 T  0334:1
                    IF HOWOPEN ≠ 0 THEN GO IMPROPER;    % IO ERROR                                       09633600 T  0336:3
                    IF WORDSLEFT < BUFFSIZE THEN                                                         09633700 T  0338:3
                         IF LINEPRINT THEN GOLP ELSE WRIT;                                               09633800 T  0340:2
                    P(0,RTN);                                                                            09633900 T  0346:0
               END WRITEBLOCK;                                                                           09634000 T  0346:2
IMPROPER:IF (1=CODE)≠HOWOPEN THEN        % CHECK USE VS HOW OPEN                                        09634200 T  0346:2
               IF HOWOPEN>1 THEN TERM(31+CODE)     % CLOSED                                             09634300 T  0348:2
               ELSE TERM(PROPER);%                 % USAGE                                               09634400 T  0352:3
          IF UNITYPE=10 OR UNITYPE=13 THEN                                                               09634500 T  0357:3
               BEGIN                                                                                     09634600 T  0359:2
                    IF CODE THEN REMOTEWRIT;                                                             09634700 T  0360:0
                    REMOTEREAD;                                                                          09634800 T  0365:2
               END;                                                                                      09634900 T  0371:1
          IF CODE THEN                     % WRITE A RECORD                                             09635000 T  0371:1
               IF LINEPRINT THEN GOLP ELSE                                                             09635100 T  0371:2
               BEGIN                                                                                     09635200 T  0374:0
```

```
                    IF TECHC THEN                                                    09635300 T  0374:2
                         BEGIN                                                       09635400 T  0376:0
                         IF NUMWDS > MAXREC THEN NUMWDS ← MAXREC;                    09635500 T  0376:2
                         IF NUMWDS > WORDSLEFT THEN WRIT;                            09635600 T  0380:0
                         IF NUMWDS < MINREC THEN TERM(36);                           09635700 T  0383:0
                         END;                                                        09635800 T  0386:2
                    MOVEREC;                                                         09635900 T  0386:2
                    IF WORDSLEFT < MINREC THEN WRIT;                                 09636000 T  0388:0
                    P(0,RTN);                                                        09636100 T  0392:0
               END;                                                                  09636200 T  0392:2
                                        % READ A RECORD                             09636300 T  0392:2
ROVER:    IF WORDSLEFT ≤ 0 THEN                                                      09636400 T  0392:2
               BEGIN                    % A NEW BLOCK WAS READ                      09636500 T  0393:2
                    IF NOT DONE THEN WAITIO;                                         09636600 T  0394:0
                    WORDSLEFT ←                                                      09636800 T  0397:1
                         MEM[(IF REVERSE THEN 1 ELSE NOT 0) INX TIP];                09636900 T  0397:3
                    IF REVERSE THEN DLOC[0] ← NOT(MAXREC-2) INX TIP;                 09637000 T  0402:2
               END;                                                                  09637100 T  0407:0
               IF TECHC THEN                                                         09637200 T  0407:0
                    BEGIN                                                            09637300 T  0408:2
                    NUMWDS ← P(,NUMWDS,LOD,,TCW,STD,MINREC);                         09637400 T  0409:0
                    MOVEREC;                                                         09637500 T  0411:3
                    IF (TCW←TCW) > MAXREC THEN TCW ← MAXREC;                         09637600 T  0413:0
                    IF TCW < NUMWDS THEN                                             09637700 T  0417:0
                         IF (TCW=0) AND (WORDSLEFT+NUMWDS=1) THEN                    09637800 T  0417:3
                              BEGIN                                                  09637900 T  0420:3
                                   REED;                                             09638000 T  0421:1
                                   RCOUNT ← *P(DUP) - 1;                             09638100 T  0422:0
                                   GO ROVER;                                         09638200 T  0424:0
                              END ELSE TERM(26 + (TCW≠0));                           09638300 T  0424:2
                    IF (TCDIF ← TCW - NUMWDS) > 0 THEN                               09638400 T  0427:1
                         BEGIN                                                       09638500 T  0429:0
                              STREAM(TCDIF,E←P(DUP),[36:6],                          09638600 T  0429:2
                                   FROM← TIP INX 0,                                  09638700 T  0430:3
                                   DEST ← NUMWDS INX (*RCPRT));                      09638800 T  0431:3
                                   BEGIN  SI ← FROM;                                 09638900 T  0434:0
                                        E(DS←32 WDS;  DS←32 WDS);                    09639000 T  0434:1
                                        DS ← TCDIF WDS;                              09639100 T  0435:2
                                   END STREAM;                                       09639200 T  0436:0
                              DLOC[0] ← TCDIF INX TIP;                               09639300 T  0436:1
                              WORDSLEFT ← *P(DUP) - TCDIF;                           09639400 T  0437:3
                              NUMWDS ← TCW;                                          09639450 T  0439:3
                         END;                                                        09639500 T  0440:2
                    P(RCPRT,DUP,LOD,NUMWDS,DIA 38,DIB 8,TRB 10,XCH,STD);             09639550 T  0440:2
                    END                  % TECH C FILE READING                       09639600 T  0443:3
                    ELSE MOVEREC;                                                    09639700 T  0443:3
               IF WORDSLEFT ≤ 0 OR UNBLKD THEN REED;                                 09639800 T  0445:0
               P(0,RTN);                                                             09639900 T  0449:0
EOFSETCK:                                                                            09640000 T  0449:2
               IF ENDFILE THEN TERM(15);                                             09640100 T  0449:2
               ENDFILE ← TRUE;                                                       09640200 T  0452:1
               P(1,RTN);                                                             09640300 T  0454:3
END COBOLIONONDISK;                                                                  09640500 T  0455:1
                                                                                          SIZE= 0456 WORDS
```

```
PROCEDURE COBOLIODSK;                                                    09700000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00664
        BEGIN                                                            09700100 T  0000:0
        REAL RCW        = +0;     %USED TO CALL COBOLIONONDSK            09700200 T  0000:0
        REAL CODE       = -1;     % 0=READ,1=WRITE,2=SEEK,6=WRTBLK,      09700300 T  0000:0
        NAME DLOC       = -2;     % POINTS TO BUFFER I/O DESC            09700400 T  0000:0
        REAL NUMWDS     = -3;     % # WDS TO BE WRITTEN                  09700500 T  0000:0
%LOCALS                                                                  09700600 T  0000:0
        INTEGER  BS ;             % USED IN COMPUTING DISK ADDR          09700700 T  0000:0
        REAL  COBOLIONONDSK= 14;                                         09700800 T  0000:0
        REAL  DEST    ;           % DESTINATION IN RANDOM MOVE           09700900 T  0000:0
        ARRAY FIB [*];            % FIB ARRAY                           09701000 T  0000:0
        NAME  FLOC;               % POINTER TO FIB                      09701100 T  0000:0
        ARRAY FPB = 3[*];         % FILE PARAMETER BLOCK                09701200 T  0000:0
        ARRAY H[*];               % DISK FILE HEADER                    09701300 T  0000:0
        REAL  INTINT    =  5;     % INTRINSIC INTRINSIC                 09701400 T  0000:0
        NAME  MEM = 2;            % DUMMY DATA DESC                     09701500 T  0000:0
        NAME  PERFORMER = 13;     % USED FOR PERFORMING USE ROUTINES    09701600 T  0000:0
        ARRAY PGUSE=24[*];        % PROGRAM USE ROUTINES                09701700 T  0000:0
        INTEGER  RT ;             % USED IN COMPUTING DISK ADDR         09701800 T  0000:0
        REAL  T;                  % TEMPORARY                           09701900 T  0000:0
        INTEGER  DAS;             % USED TO COMPUTE DISK ADDRESS        09702000 T  0000:0
 $ SET OMIT = NOT SHAREDISK                                              09702004 T  0000:0
DEFINE                                                                   09702100 T  0000:0
        AF              = [12:12]#,       % FILE USE ROUTINE            09702200 T  0000:0
        ARR             = [36:12]#,       % REEL USE ROUTINE            09702300 T  0000:0
        BCOUNT          = FIB[6]#,        % BLOCK COUNT                 09702400 T  0000:0
        BF              = [1:11]#,        % FILE USE ROUTINE            09702500 T  0000:0
        BOUNDED         = FIB[9].[2:1]#,  % TRUE IF BOUNED FROM ABOVE    09702600 T  0000:0
        BREAK           = FIB[9] ≠ 0 #    ,  % BREAKOUT RESTART POINT    09702700 T  0000:0
        BREAKOUT        = IF(RCOUNT MOD FIB[9])=0 THEN                   09702800 T  0000:0
                            P(0,0,12,COM,DEL,DEL)#,% CALL BREAKOUT       09702900 T  0000:0
        BRR             = [24:12]#,       % REEL USE ROUTINE            09703000 T  0000:0
        BUFFNUM         = FIB[13].[1:9] #,  % # OF BUFFS REQUSTED         09703100 T  0000:0
        BUFFSIZE        = FIB[18].[3:15]#,  % BUFFER SIZE  (REQUESTED)    09703200 T  0000:0
        BUFFSZ          = FIB[18][8:8:10]#, % SIZE FOR CONCATINATES        09703300 T  0000:0
        BUFTOP          = FIB [16]#,      % USED ON I-O AND RANDOM       09703400 T  0000:0
        COUNT           = FIB[12] #,      % USED FOR BLOCKING TECH-A,B   09703500 T  0000:0
        DINXPRT         = P(*RCPRT & TIP [CTC],RCPRT,+)#,%UPDATE POINTER 09703600 T  0000:0
        DONE            = TIP.[19:1] #,   % 1= IO COMPLETED              09703700 T  0000:0
        DISK            = (UT =4)#,       % DISK IS UNIT TYPE OF 4       09703800 T  0000:0
        ERBIT           = FIB[13].[19:1] #, % IOERR 19 NOT YET SPOUTED     09703900 T  0000:0
        FLAGINWA        = 0[1:1:1]#,      % SAYS WE ARE IN WORK AREA     09704000 T  0000:0
        FNAM            = FIB[4].[13:11]#,  % FILE NAME INDEX IN FPB     09704100 T  0000:0
        ENDFILE         = FIB[5].[40:1] #,  % ALREADY PASSED EOF         09704200 T  0000:0
        ENDPROCESS      = FIB[5].[39:2]#,   % SEE OPTIONAL AND ENDFILE   09704300 T  0000:0
        EOF             =((*DLOC).[27:1])#, % FIRST EOF OR EOT           09704400 T  0000:0
        GETSEG          = P(FPB[(BS:=FNAM)+3],FPB[BS],FPB[BS+1],         09704500 T  0000:0
                            T,H,4,11,COM,DEL,DEL,DEL,DEL,DEL,DEL)#,      09704600 T  0000:0
        KEY             = FIB[15].[12:10]#, % REL PRT LOC OF ACTUAL KEY  09704700 T  0000:0
        HAVEWA          =(INWA OR FIB[20].[CF]>1)#,% TRUE IF WE ARE NOW  09704800 T  0000:0
                        %IN WORK AREA OR HAVE MADE IT PRESENT PREVIOUSLY 09704900 T  0000:0
        HOWOPEN         = FIB[5].[41:3]#,   % 1=OPEN INPUT,0= OPEN OUTPT  09705000 T  0000:0
                                            % 1 > CLOSED                 09705100 T  0000:0
        INVALIDUSER     = FIB[5]<0#,        % INVALID USER NOT PARITY     09705200 T  0000:0
        INWA            = FIB[20]≥0#,       % SAYS WE ARE IN WORK AREA    09705300 T  0000:0
        INXPRT          = P(NUMWDS INX *RCPRT,RCPRT,+)#,% UPDATE POINTER 09705400 T  0000:0
```

```
IOERR(IOERR1)    = P(0,FLOC,IOERR1,17,COM,DEL,DEL,DEL)#,
                      % ABOVE CALLS IOERROR ROUTINE
IOMASK           = 0&1[19:47:1] #,    % USED TO WAIT FOR IOFINISH
LASTDONE         = FIB[13].[21:1] #, % NOT OF LAST OPERATION DONE
LBLPTR           = FLOC[1] #,         % LABEL POINTER
LSUBL            = FIB [1]   #,       % LOWER BOUND FOR RANDOM
LSUBU            = FIB [3] #,         % UPPER BOUND FOR DISK REC
MABUSE           = FIB[4].[1:1]#,     % MAY BE USE RTNS PRESENT
MAKEPRESENTWA    = P(*RCPRT & 1 [CTC],0,CDC)#,
MAXR             = FIB[18][8:38:10]#,% MAX REC SZ FOR CONCATS
MAXREC           = FIB[18].[33:15]#, % MAX REC SZ
MINREC           = FIB[18].[CF]#,     % MINIMUM RECORD SIZE
NOAIT            = FIB[20].[3:1]#,    % AIT FOR WA WAS DESTROYED
NUMBUF           = FIB[13].[10: 9]#, % NUMBER OF BUFFERS ASSIGNED
NUMBSPC          = H[9].[43:5] #,     % NUMBER OF ROWS SPECIFIED
NUMREC           = FIB[11] #,         % RECORDS PER BLOCK
OPENIO           = FIB[13].[22:1]#,   % 1= OPEN INPUT=OUPUT (DISK)
PARITY           = FIB[13].[20:1] #, % IO ERROR OCCURED IN BLOCK
PBIT             = [2:1]#,            % PRESENCE BIT
POINTPRTTOBUF    = P((*RCPRT OR MEM) & TIP [CTC],RCPRT,←)#,
POINTPRTTOWA     = P(*RCPRT & FIB[20] [CTC],RCPRT,←)#,
PRESENT          =((*DLOC).[2:1])#,  % CHECKS PRESENTSBIT
PROPER           =REVERSE+CODE+CODE+21#,%GENERATES PROPER IOERR
RCOUNT           = FIB[7] #,          % RECORD COUNT
RCPRT            = (FIB[20].[FF])#,  %PRT OF DESC POINTING TO REC
RECPERBLK        = H[0].[30:12] #,    % RECORDS PER BLOCK
REDECWA          = P(MKS,RCPRT,MAXREC,1,1,1,INTINT)#,
                     % DECLARE SAVE ARRAY FOR WORK AREA
REEDING          = ((*DLOC).[24:1])#,%LAST IO WAS READ
RESETPANDERBIT   = FIB[13]:=*P(DUP)&0[19:19:2]#,%RESET ERR BITS
RESETREADBIT     = 0[24:24:1]#,       % USED TO TURN OFF READ BIT
REVERSE          = FIB[5].[44:1] #,   % 1=REVERSE INPUT
ROWLGTH          = H[1]#,             % ROW LGTH FROM HEADER
SAVEWADDR        = FIB[20]←*P(DUP)&*P(RCPRT)[CTC]#,% SAVE ADDRESS
SERIAL           = FIB[4].[27:3]=0 #,% FILE ACCESS = SERIAL
SEGPERBLK        = H[0].[42:6] #,     % SEGMENTS PER BLOCK
SETPANDERBIT     = FIB[13]:=*P(DUP)&3[19:46:2]#,
                     % SET PARITY AND IOERR 19 BITS
SETPRESENTSBIT   =P(TIP OR MEM ,DLOC,←)#,% SET PRESENCE BIT
SETREADBIT       = 1[24:47:1]#,       %USED TO TURN READ BIT ON
$ SET OMIT = NOT(TIMESHARING)
SLEEP            = 36 #,
$ POP OMIT
$ SET OMIT = TIMESHARING
TECHA            =(FIB[5].[46:2]=1)#,% TECHNIQUE=A
TERM(TERM1)      = P(1,FLOC,TERM1,17,COM)#,%TERMINATE I/O ERROR
TIP              = (*DLOC) #,         % LOAD I/O DESC
TOSZF            = [8:38:10]#,        % TO SIZE FIELD
TOTREC           = H[7]    #,         % TOTAL RECORDS ON FILE
UT               = (FIB[4].[8:4])#,   % HARDWARE TYPE
WA               = P(RCPRT,DIB 0,LOD)#,% LOAD WORK AREA PTR
WAITIO           = P(DLOC,IOMASK,     % THIS SLEEPS ON I/O
                   SLEEP,COM,DEL,DEL)#,% WAITING FOR A COMPLETE
WORDSLEFT        = FIB[17]#,          % WORDS LEFT IN BUFFER
WRITBACK         = FIB[13].[23:1]#;   % FLAG TO SAY WRITE BACK
LABEL  MOOVE,FLOTE,SEEKRTN,START,READREV;
LABEL  SERIALIO,SIOEOD,RNDEOD,EOFSETCK;
```

```
09705500 T  0000:0
09705600 T  0000:0
09705700 T  0000:0
09705800 T  0000:0
09705900 T  0000:0
09706000 T  0000:0
09706100 T  0000:0
09706200 T  0000:0
09706300 T  0000:0
09706400 T  0000:0
09706500 T  0000:0
09706600 T  0000:0
09706700 T  0000:0
09706800 T  0000:0
09706900 T  0000:0
09707000 T  0000:0
09707100 T  0000:0
09707200 T  0000:0
09707300 T  0000:0
09707400 T  0000:0
09707500 T  0000:0
09707600 T  0000:0
09707700 T  0000:0
09707800 T  0000:0
09707900 T  0000:0
09708000 T  0000:0
09708100 T  0000:0
09708200 T  0000:0
09708300 T  0000:0
09708400 T  0000:0
09708500 T  0000:0
09708600 T  0000:0
09708700 T  0000:0
09708800 T  0000:0
09708900 T  0000:0
09709000 T  0000:0
09709100 T  0000:0
09709200 T  0000:0
09709300 T  0000:0
09709400 T  0000:0
09709500 T  0000:0
09709600 T  0000:0
09709601 T  0000:0
09709700 T  0000:0
09709900 T  0000:0
09710000 T  0000:0
09710100 T  0000:0
09710200 T  0000:0
09710300 T  0000:0
09710400 T  0000:0
09710500 T  0000:0
09710600 T  0000:0
09710700 T  0000:0
09710800 T  0000:0
09710900 T  0000:0
09711000 T  0000:0
09711100 T  0000:0
```

```
LABEL   ERREND,MOOVERR,REREAD,LCKHANDLER;                                    09711150 T  0000:0
SUBROUTINE GOUSE;                       % THIS CALLS USE ROUTINES             09711200 T  0000:0
        BEGIN                                                                09711300 T  0001:0
                P(MKS,T,0,PERFORMER);                                        09711400 T  0001:0
        END GOUSE;                                                           09711500 T  0002:0
SUBROUTINE  ERROR;                      %THIS PROCESSES ALL ERRORS           09711600 T  0002:1
        BEGIN                                                                09711700 T  0003:0
        IF REEDING AND CODE AND (NUMREC=1)                                   09711800 T  0003:0
 $ SET OMIT = NOT SHAREDISK                                                  09711809 T  0005:1
        THEN % SKIP ERROR CODE                                               09711820 T  0005:1
        ELSE BEGIN                                                           09711900 T  0005:3
             IF  OPENIO  THEN  IF  (T:=RT:=PGUSE[4].ARR) ≠  0  THEN          09712000 T  0006:3
                 GOUSE  ELSE   ELSE        %WAS ERROR ON IO                  09712100 T  0010:3
             IF  REEDING AND (NOT CODE) THEN  %READ ERROR                    09712200 T  0012:2
                 IF (T:=RT:=PGUSE[4].BRR) ≠ 0 THEN                           09712300 T  0014:3
                     GOUSE  ELSE   ELSE                                      09712400 T  0017:3
                 IF (T:=PGUSE[5].BRR) ≠ 0  THEN GOUSE;  %WRITE ERROR         09712500 T  0019:2
             IF (T←FIB[15].BF) ≠ 0 THEN GOUSE;     % ERROR ON FILE=N          09712600 T  0024:0
 $ SET OMIT = NOT SHAREDISK                                                  09712609 T  0028:0
             IF REEDING AND (NOT CODE) THEN                                  09712700 T  0028:0
                  BEGIN                          %CHECK USE PROC FOR         09712800 T  0029:3
                  IF ERBIT  THEN                 %INPUT ERRORS               09712900 T  0030:1
                       IF (T OR RT) = 0 THEN IOERR(19);                      09713000 T  0031:1
                  ERBIT := FALSE;                                            09713100 T  0035:2
                  END  ELSE  TERM(20);           %WRITE ERR  TERM            09713200 T  0038:0
             END;                                                            09713300 T  0039:3
ERREND:                                                                      09713350 T  0039:3
        END ERROR;                                                           09713400 T  0039:3
SUBROUTINE MOVEREC;   %MOVES DATA  TO AND  FROM WORKAREA                     09713500 T  0040:0
        BEGIN                                                                09713600 T  0040:0
             IF NOT DONE THEN                  % DONT MOVE TILL IO DONE     09713700 T  0040:0
 $ SET OMIT = NOT SHAREDISK                                                  09713709 T  0041:1
              WAITIO;                                                        09713750 T  0041:1
             IF  NOT PRESENT THEN                                            09713800 T  0044:1
                  BEGIN                         %GOT AN ERROR              09713900 T  0045:2
                  SETPRESENTSBIT;                                            09714000 T  0046:0
                  SETPANDERBIT;                 %SET  ERROR FLAGS          09714100 T  0047:2
 $ SET OMIT = NOT SHAREDISK                                                  09714104 T  0050:0
                  IF NOT REEDING THEN                                        09714200 T  0050:0
                       BEGIN                    %ERROR ON OUTPUT           09714300 T  0051:1
                       DEST := WA;              %MOVE  FIRST RECORD        09714400 T  0051:3
                       P(TIP INX 1);            %TO WORK  AREA             09714500 T  0054:0
                       GO MOOVE;                                             09714600 T  0055:0
                       END;                                                  09714700 T  0055:2
 $ SET OMIT = SHAREDISK                                                      09714799 T  0055:2
                  END;                                                       09714800 T  0055:2
 $ POP OMIT                                                                  09714801 T  0055:2
 $ SET OMIT = NOT SHAREDISK                                                  09714809 T  0055:2
              P(BUFTOP INX(BS←NUMWDS×(RCOUNT MOD NUMREC)+ 1));               09714900 T  0055:2
              WA;                                  %MOVE TO/FROM WA          09715000 T  0059:0
              DEST ← IF CODE THEN P(XCH) ELSE P;   %FOR READ OR WRITE        09715100 T  0060:3
MOOVE:        STREAM(FROM←P:NUMWDS,E←P(DUP),[37:5],DEST←P(*P(,DEST)));       09715200 T  0062:3
                  BEGIN                                                      09715300 T  0065:0
                  SI:=FROM; E(DS:=32 WDS; DS:=32 WDS); DS:=NUMWDS WDS;       09715400 T  0065:0
                  END STREAM;                                                09715500 T  0067:0
              P(DEL);                                                        09715600 T  0067:1
              IF PARITY THEN                                                 09715700 T  0067:2
```

```
                    BEGIN                                                    09715710 T  0068:2
MOOVERR:            ERROR;                                                   09715730 T  0069:0
 $ SET OMIT = NOT SHAREDISK                                                  09715739 T  0070:0
                    END;                                                     09715760 T  0070:0
          END MOVEREC;                                                       09715800 T  0070:0
 $ SET OMIT = NOT SHAREDISK                                                  09715809 T  0070:1
SUBROUTINE DISKADDRESS;      %THIS COMPUTES THE DISK ADDRESS READ & WRIT     09715900 T  0070:1
    BEGIN                                                                    09716000 T  0071:0
          RT ← SEGPERBLK × DAS;                  % REL SEGMENT NO           09716100 T  0071:0
          IF P(RT DIV ROWLGTH,DUP) GEQ NUMBSPC THEN                          09716200 T  0073:0
               BEGIN                                                         09716210 T  0075:2
 $ SET OMIT = NOT SHAREDISK                                                  09716219 T  0076:0
               P(1,RTN);                                                     09716240 T  0076:0
               END;                                                          09716250 T  0076:2
          IF (BS←H[(T← P + 10)]) = 0 THEN                                    09716300 T  0076:2
               BEGIN                                                         09716400 T  0078:3
               GETSEG;                                                       09716500 T  0079:1
               IF HOWOPEN≠0 THEN IF NOT OPENIO THEN IOERR(22);               09716600 T  0086:0
               BS ← H[T];                                                    09716700 T  0091:3
               END;                                                          09716800 T  0092:3
          STREAM( A ← BS ← BS + RT MOD ROWLGTH,                              09716900 T  0092:3
                  B←T←BUFTOP,[CF]←(IF CODE THEN 0 ELSE WRITBACK));           09717000 T  0095:0
                 BEGIN SI←LOC A;  DS←8 DEC; END;                             09717100 T  0099:2
 $ SET OMIT = NOT SHAREDISK                                                  09717109 T  0100:1
    END DISKADDRESS;                                                         09717200 T  0100:1
SUBROUTINE ROTATEBUF;            %THIS ROTATES BUFFERS                       09717300 T  0100:2
          BEGIN                                                              09717400 T  0101:0
               IF  NUMBUF > 1 THEN                                           09717500 T  0101:0
                    P(NUMBUF,DLOC,13,11,COM,DEL,DEL,DEL);                    09717600 T  0102:2
               WORDSLEFT := BUFFSIZE;                                        09717700 T  0105:3
               RESETPANDERBIT;                                               09717800 T  0107:3
               FIB[16],[CF] := TIP;                                          09717900 T  0110:1
          END ROTATEBUF;                                                     09718000 T  0112:2
SUBROUTINE  PREL;                  % THIS DOES ACTUAL I/O                   09718100 T  0112:3
               BEGIN                                                         09718200 T  0113:0
                    P( TIP,DLOC);                                            09718300 T  0113:0
                    IF WRITBACK THEN                % DO SPECIAL WRITE-IO     09718400 T  0113:3
                    BEGIN                                                    09718500 T  0114:3
                              WRITBACK ← FALSE;    % TURN OFF READ BIT       09718600 T  0115:1
                              DLOC[0]← TIP&RESETREADBIT;% TO MAKE WRITE       09718700 T  0117:3
                    END;                                                     09718800 T  0119:3
                    P(PRL,DEL);                     % DO I=0                 09718900 T  0119:3
                    IF BREAK THEN BREAKOUT;                                  09719000 T  0120:2
               END PREL;                                                     09719100 T  0125:3
SUBROUTINE REED;                %THIS READS BLOCKS                           09719200 T  0126:0
          BEGIN                                                              09719300 T  0126:0
               WORDSLEFT := BUFFSIZE;                                        09719400 T  0126:0
               DLOC[0] := FLAG(BUFTOP & SETREADBIT);    %TO  RESET IOD       09719500 T  0128:0
               CODE ← P(CODE,0);                                             09719600 T  0130:1
               DISKADDRESS;                                                  09719700 T  0131:1
               CODE ← P;                                                     09719800 T  0132:0
 $ SET OMIT = NOT SHAREDISK                                                  09719804 T  0132:2
               MEM[BUFTOP INX NOT 2] ← DAS;          % SAVE BLOCK NUMBER     09719900 T  0132:2
               PREL;                                                         09720000 T  0135:0
               FIB[16],[CF] := TIP;                  %SAVE BUFF ADDRESS      09720100 T  0136:0
          END REED;                                                          09720200 T  0138:1
SUBROUTINE WRIT;                %THIS WRITES BLOCKS                          09720300 T  0138:2
```

```
        BEGIN                                                                   09720400 T  0139:0
             WORDSLEFT := BUFFSIZE;                                             09720500 T  0139:0
             WRITBACK ← FALSE;                                                  09720600 T  0141:0
             DAS := BCOUNT;                                %BLOCK ADDRESS       09720700 T  0143:2
             DLOC[0] := FLAG(BUFTOP & RESETREADBIT);       %RESET IOD           09720800 T  0144:2
             DISKADDRESS;                                                       09720900 T  0146:3
$ SET OMIT = NOT SHAREDISK                                                      09720904 T  0148:0
             PREL;                                                              09721000 T  0148:0
             FIB[16].[CF] := TIP;                    %SAVE BUFF ADDRESS         09721100 T  0149:0
             IF NOT(SERIAL) THEN BCOUNT := MEM[BUFTOP INX NOT 2];               09721120 T  0151:1
        END WRIT;                                                               09721200 T  0156:1
SUBROUTINE SEEK;            %THIS FINDS AND/OR READS BLOCKS                     09721300 T  0156:2
        BEGIN                                                                   09721400 T  0157:0
        IF (DAS ← RCOUNT DIV NUMREC) = BCOUNT THEN                              09721500 T  0157:0
             BEGIN                                                              09721510 T  0159:2
$ SET OMIT = NOT SHAREDISK                                                      09721519 T  0160:0
             GO SEEKRTN;                                                        09721570 T  0160:0
             END;                                                               09721580 T  0160:2
        IF SERIAL THEN                                                          09721600 T  0160:2
             BEGIN                                                              09721700 T  0162:0
             IF NOT HOWOPEN THEN                                                09721800 T  0162:2
                  BEGIN                    %NOT INPUT                           09721900 T  0163:3
                  IF RCOUNT < TOTREC THEN TOTREC := RCOUNT;                     09722000 T  0164:1
                  IF NUMREC > 1 THEN                                            09722100 T  0167:2
                       BEGIN               %BLOCKED OUTPUT                      09722200 T  0168:2
                       NUMBUF := 1;        %FILL ONLY ONE                       09722300 T  0169:0
   IF (WORDSLEFT<BUFFSIZE) THEN                                                 09722400 P  0171:2
     BEGIN                                                                      09722405 C  0173:1
   IF NOT OPENIO THEN                                                           09722410 C  0173:3
     BEGIN % SERIAL OUTPUT - NO ADDR IN BUFF                                    09722415 C  0175:0
     DAS:=P(DAS,BCOUNT);                                                        09722420 C  0175:2
     CODE:=P(CODE,1);                                                           09722425 C  0176:3
     DISKADDRESS;                                                               09722430 C  0177:3
     CODE:=P; DAS:=P;                                                           09722435 C  0179:0
     END;                                                                       09722440 C  0180:0
   WRITBACK:=TRUE;                                                              09722445 C  0180:0
   END;                                                                         09722450 C  0182:2
                  END;                                                          09722500 T  0182:2
             END;                                                               09722600 T  0182:2
        IF NUMBUF ≠ 1 THEN                                                      09722700 T  0182:2
        IF (DAS≤COUNT) AND (DAS>BCOUNT) THEN   %BLOCK IS PRESENT                09722800 T  0184:0
             IF MEM[DLOC[NUMBUF-1] INX NOT 2] = COUNT THEN                      09722900 T  0186:3
                  DAS := COUNT + 1;                                             09723000 T  0191:3
        COUNT := (RCOUNT DIV NUMREC) + NUMBUF - 1;                              09723100 T  0193:3
        DO   BEGIN                                                              09723200 T  0197:3
             IF NOT DONE THEN WAITIO;                                           09723300 T  0197:3
             IF NOT PRESENT THEN                                                09723400 T  0202:0
                  IF NOT REEDING                                                09723500 T  0203:1
                       THEN MOVEREC ELSE SETPRESENTSBIT;                        09723600 T  0204:3
                  IF DAS × NUMREC ≤ LSUBU                                       09723700 T  0209:0
                       THEN REED ELSE ROTATEBUF;                                09723800 T  0210:1
             END UNTIL (DAS := DAS + 1) > COUNT;                                09723900 T  0214:0
        IF NOT HOWOPEN THEN WAITIO;                                             09724000 T  0216:2
        BCOUNT := DAS ← DAS - NUMBUF;                                           09724100 T  0220:3
        NUMBUF := BUFFNUM;                                                      09724200 T  0223:3
        END ELSE                        %MUST BE RANDOM                         09724300 T  0227:0
        IF HOWOPEN OR (NUMREC > 1)                                              09724400 T  0227:0
```

```
 $ SET OMIT = NOT SHAREDISK
              THEN BEGIN                    % INPUT OR BLOCKED OR LOCK
REREAD:
                   IF NUMBUF = 1 THEN    % JUST READ DONT TRY TO FIND
                      BEGIN IF NOT DONE THEN
 $ SET OMIT = NOT SHAREDISK
                               WAITIO;
                            REED;
                            BCOUNT←DAS;
                      END
                   ELSE BEGIN
                        FOR T := 1 STEP 1 UNTIL NUMBUF -1
                      DO          %FIND BLOCK IN CORE
                      IF MEM[DLOC[T] INX NOT 2] = DAS THEN
 $ SET OMIT = SHAREDISK
                      GO FLOTE;
 $ POP OMIT
 $ SET OMIT = NOT SHAREDISK
                      IF NOT DONE THEN
 $ SET OMIT = NOT SHAREDISK
                           WAITIO;
                      REED;       %MAKE PRESENT IN CORE
FLOTE:                IF CODE < 2
 $ SET OMIT = NOT SHAREDISK
                      THEN BEGIN
                           IF WRITBACK THEN        %READ OR
                              BEGIN                %WRITE
                              WRIT;
                              DAS := RCOUNT DIV NUMREC;
                              END ELSE ROTATEBUF;
                         WHILE MEM[TIP INX NOT 2] ≠ DAS
                              DO  ROTATEBUF;
                         BCOUNT := DAS;
                         END;
                              END;
                        END;
SEEKRTN:
                   ENDFILE := FALSE;
                WORDSLEFT := BUFFSIZE - ((RCOUNT MOD NUMREC) × NUMWDS);
                LASTDONE ← FALSE;          % PREVENT SERIALIO OVERWRITE
                IF CODE = 2 THEN P(XIT);
                END SEEK;
%***%   S T A R T   Y E E   H E R E   Y E E   D I S K E R S   %***%
START:
       FIB := *(FLOC := (NOT 2) INX DLOC);
       IF NOT DISK THEN
            BEGIN
            FLOC := P(,RCW,LOD);
            FIB ← ABS(CODE);
            DEST := P(,DLOC,LOD);
            BS := NUMWDS;
            CODE := 1;
            RCW := DLOC := NUMWDS := 0;
            P([FLOC],DUP,0,XCH,CFX,STF,1,INX,STS);
            GO TO P(COBOLIONONDSK);
            END;
       H := *[FIB[14]];
```

```
09724409 T  0229:1
09724500 T  0229:1
09724550 T  0230:1
09724600 T  0230:1
09724700 T  0231:3
09724709 T  0233:2
09724750 T  0233:2
09724760 T  0236:2
09724770 T  0238:0
09724780 T  0239:1
09724800 T  0239:1
09724900 T  0239:3
09725000 T  0243:2
09725100 T  0244:3
09725109 T  0247:3
09725110 T  0247:3
09725111 T  0248:3
09725119 T  0248:3
09725300 T  0248:3
09725309 T  0250:0
09725390 T  0250:0
09725400 T  0253:0
09725500 T  0254:0
09725549 T  0254:1
09725600 T  0254:1
09725700 T  0255:1
09725800 T  0256:1
09725900 T  0256:3
09726000 T  0258:0
09726100 T  0259:3
09726200 T  0261:0
09726300 T  0263:0
09726400 T  0265:2
09726500 T  0266:3
09726600 T  0266:3
09726700 T  0266:3
09726800 T  0266:3
09726900 T  0266:3
09727000 T  0269:1
09727100 T  0273:1
09727200 T  0275:3
09727300 T  0277:1
09727400 T  0277:2
09727500 T  0277:2
09727600 T  0280:0
09727700 T  0282:1
09727800 T  0283:3
09727900 T  0284:1
09728000 T  0285:1
09728100 T  0286:1
09728200 T  0287:1
09728300 T  0288:0
09728400 T  0288:3
09728500 T  0290:2
09728600 T  0293:1
09728700 T  0293:3
09728800 T  0293:3
```

```
$ SET OMIT = NOT SHAREDISK                                                    09728804 T  0295:0
      IF CODE.[1:1] THEN                                                      09728815 T  0295:0
           BEGIN CODE←ABS(CODE);                                              09728820 T  0295:3
$ SET OMIT = NOT SHAREDISK                                                    09728824 T  0297:1
           END;                                                               09728890 T  0297:1
      IF CODE > 2 THEN IF CODE ≠ 6 THEN TERM(25) ELSE     %WRITE BLOCK        09728900 T  0297:1
           BEGIN IF HOWOPEN > 1 THEN TERM(37);                                09728930 T  0301:0
                 IF NOT OPENIO THEN IF HOWOPEN THEN TERM(34+REVERSE);         09728960 T  0304:3
$ SET OMIT = NOT SHAREDISK                                                    09728979 T  0310:2
                 GO EOFSETCK;    %WRITES BLOCK:IMMEDIATE-NO ROTATION          09729000 T  0310:2
           END;                                                               09729050 T  0311:0
      IF HOWOPEN > 1 THEN TERM(31 + CODE);                                    09729100 T  0311:0
      IF CODE = 2 THEN                                                        09729200 T  0314:3
           BEGIN  RCOUNT←(IF KEY=0 THEN 0 ELSE P(KEY,DIB 0,LOD)) - 1;         09729300 T  0315:2
             IF (RCOUNT<LSUBL) OR (RCOUNT>LSUBU) THEN                         09729400 T  0321:3
                  GO EOFSETCK          % INVALID KEY                         09729500 P  0324:2
              ELSE IF INVALIDUSER THEN TERM(0)% DS WITH INVALID USER          09729510 C  0324:2
                  ELSE SEEK;           %ONLY SEEK VALID RECORDS               09729520 C  0328:0
           END;                                                               09729600 T  0330:0
      IF NOT OPENIO THEN                                                      09729700 T  0330:0
           IF (1 - CODE) ≠ HOWOPEN THEN TERM(PROPER);                         09729800 T  0331:1
      IF SERIAL THEN          % PROCESS SERIAL FILE                          09729900 T  0337:3
           BEGIN                                                              09730000 T  0339:1
           IF OPENIO THEN GO SERIALIO;                                        09730100 T  0339:3
           IF (RCOUNT<LSUBL) OR (RCOUNT>LSUBU) THEN GO EOFSETCK;              09730200 T  0341:2
          IF CODE = 0 OR CODE = 2 THEN% READ OR SEEK (SERIAL)                 09730210 C  0345:0
            IF INVALIDUSER THEN TERM(0);    % DS WITH INVALID USER           09730220 C  0346:3
           MOVEREC;                                                           09730300 T  0350:0
           IF REVERSE THEN GO READREV;                                        09730400 T  0351:0
           IF CODE THEN IF RCOUNT > TOTREC THEN TOTREC := RCOUNT;             09730500 T  0352:3
           IF (WORDSLEFT := *P(DUP) - NUMWDS) ≤ 0 THEN                        09730600 T  0356:3
                BEGIN                    %BLOCK IS EXHAUSTED                  09730700 T  0359:1
                IF CODE THEN WRIT ELSE                                        09730800 T  0359:3
                     IF (DAS := BCOUNT + NUMBUF) × NUMREC ≤ LSUBU THEN        09730900 T  0362:0
                          BEGIN          %READ AHEAD TO KEEP                  09731000 T  0366:1
                          REED;          %BUFFERS READY                       09731100 T  0366:3
                          COUNT := DAS;                                       09731200 T  0368:0
                          END ELSE ROTATEBUF;                                 09731300 T  0369:1
                BCOUNT := * P(DUP) + 1;                                       09731400 T  0371:0
                END;                                                          09731500 T  0373:0
           RCOUNT := *P(DUP) + 1;                                             09731600 T  0373:0
           IF FALSE THEN           % THIS CODE EXECUTED ONLY FOR TAPE          09731700 T  0375:0
READREV:        BEGIN              % OPEN=REVERSE EQUATED TO DISK             09731800 T  0375:1
                IF (WORDSLEFT ← *P(DUP) - NUMWDS) ≤ 0 THEN                    09731900 T  0375:3
                     BEGIN          % BLOCK IS EXHAUSTED                       09732000 T  0378:1
                     IF (DAS ← BCOUNT - NUMBUF) × NUMREC ≥ LSUBL THEN          09732100 T  0378:3
                          BEGIN  REED;  COUNT ← DAS;                          09732200 T  0382:2
                          END  ELSE  ROTATEBUF;                               09732300 T  0385:1
                     BCOUNT ← *P(DUP) - 1;                                    09732400 T  0387:0
                     END;                                                     09732500 T  0389:0
                RCOUNT ← *P(DUP) - 1;                                         09732600 T  0389:0
                END REVINPUT;                                                 09732700 T  0391:0
           IF KEY ≠ 0 THEN P(RCOUNT,KEY,DIB 0,ISD);                           09732800 T  0391:0
           P(0,RTN);                                                          09732900 T  0395:0
   % %  E N D  O F  S E R I A L  ----  I O  N E X T  % %                      09733000 T  0395:2
SERIALIO:                                                                     09733100 T  0395:2
        RCOUNT ← *P(DUP) - (T ← (CODE AND LASTDONE));                         09733200 T  0395:2
```

```
                IF (RCOUNT<LSUBL) OR (RCOUNT>LSUBU) THEN GO EOFSETCK;                09733300 T  0399:1
                IF CODE = 0 OR CODE = 2 THEN   %READ OR SEEK (SERIAL I/O)            09733310 C  0402:3
                  IF INVALIDUSER THEN TERM(0);% DS WITH INVALID USER                 09733320 C  0404:2
                IF T THEN WORDSLEFT ← *P(DUP) + NUMWDS ELSE                          09733400 T  0407:3
                     IF WORDSLEFT ≤ 0 THEN                                           09733500 T  0410:2
                          BEGIN                    %BLOCK IS EXHAUSTED               09733600 T  0412:0
                          IF (DAS := BCOUNT + NUMBUF) × NUMREC ≤ TOTREC THEN         09733700 T  0412:2
                               BEGIN               %ANOTHER BLOCK                    09733800 T  0416:1
                               REED;               %  IN SIGHT                       09733900 T  0416:3
                               COUNT := DAS;                                         09734000 T  0418:0
                               END ELSE                                              09734100 T  0419:1
                               IF WRITBACK THEN    % WRITE CURRENT BLOCK              09734200 T  0419:1
                                    WRIT           % LESS WE FORGIT                  09734300 T  0420:3
                                    ELSE       % OR USE NEXT BUFFER                  09734400 T  0422:0
                                    ROTATEBUF;     % BECAUSE ITS THERE                09734500 T  0422:0
                           BCOUNT := *P(DUP) + 1;                                    09734600 T  0424:0
                           END;                                                      09734700 T  0426:0
                     IF RCOUNT > TOTREC THEN                                         09734800 T  0426:0
                          IF CODE THEN TOTREC ← RCOUNT ELSE                          09734900 T  0427:1
                            BEGIN  CODE←32;  GO SIOEOD;  END;                        09735000 T  0430:0
                     MOVEREC;                                                        09735100 T  0431:3
SIOEOD:              IF (WORDSLEFT := *P(DUP) - NUMWDS) ≤ 0 THEN                     09735200 T  0433:0
                          IF CODE THEN          %WROTE LAST RECORD                 09735300 T  0435:2
                               BEGIN             % OF YEE BLOCK                    09735400 T  0436:1
                               IF (DAS:=BCOUNT+NUMBUF)×NUMREC ≤ TOTREC THEN        09735500 T  0436:3
                                    BEGIN                %READ AHEAD TOO            09735600 T  0440:2
                                    WRITBACK := TRUE;    %KEEP BUFFERS FULL         09735700 T  0441:0
                                    REED;                                           09735800 T  0443:2
                                     COUNT := DAS;                                  09735900 T  0445:0
                                     END ELSE      % WRITE BLOCK NOW                 09736000 T  0446:1
                                     WRIT;         % ...A WRITE IN TIME...           09736100 T  0446:1
                                BCOUNT := *P(DUP) + 1;                              09736200 T  0448:0
                                END ELSE                                            09736300 T  0450:0
                             ELSE IF CODE THEN WRITBACK := TRUE; %NOT FULL BLK      09736400 T  0450:0
                        LASTDONE := NOT CODE;                                       09736500 T  0454:1
                        IF KEY ≠ 0 THEN P(RCOUNT,1,+,KEY,DIB 0,ISD);                09736600 T  0457:0
                        RCOUNT := *P(DUP) + 1;                                      09736700 T  0461:2
                        IF CODE=32 THEN GO EOFSETCK ELSE P(0,RTN);                  09736800 T  0463:2
                   END SERIAL;     % END OF ALL SERIAL PROCESSING                    09736900 T  0465:1
              %%% RANDOM AND RANDOM IO START HERE %%%                               09737000 T  0465:1
                   RCOUNT := (IF KEY = 0 THEN 0 ELSE P(KEY,DIB 0,LOD)) - 1;         09737100 T  0465:1
                   IF (RCOUNT<LSUBL) OR (RCOUNT>LSUBU) THEN GO EOFSETCK;            09737200 T  0471:0
                   IF CODE = 0 OR CODE = 2 THEN   %READ OR SEEK (RDM OR RDM I/O)    09737210 C  0474:2
                     IF INVALIDUSER THEN TERM(0);% DS WITH INVALID USER             09737220 C  0476:1
                   IF RCOUNT > TOTREC THEN                                          09737300 T  0479:2
                        IF CODE THEN TOTREC ← RCOUNT ELSE                           09737400 T  0480:3
                          BEGIN CODE ← 32;  GO RNDEOD;  END;                        09737500 T  0483:2
$ SET OMIT = NOT SHAREDISK                                                          09737549 T  0485:1
                   IF (DAS ← RCOUNT DIV NUMREC) ≠ BCOUNT THEN                       09737600 T  0485:1
                        IF (NUMREC≠CODE)                                            09737700 P  0487:3
                          THEN SEEK              % READ OR BLOCKED WRITE           09737710 C  0488:3
                          ELSE MEM[BUFTOP INX NOT 2]:=BCOUNT:=DAS;                  09737720 C  0490:2
                   MOVEREC;                                                         09737800 T  0495:0
RNDEOD:            WORDSLEFT := *P(DUP) - NUMWDS;                                   09737900 T  0496:0
                   IF CODE THEN                                                     09738000 T  0498:0
                        IF NUMREC = 1                %UNBLOCKED OUTPUT              09738100 T  0498:1
$ SET OMIT = NOT SHAREDISK                                                          09738109 T  0499:2
```

```
                THEN BEGIN                                                      09738120 T  0499:2
                     WRIT;                                                      09738140 T  0500:1
 $ SET OMIT = NOT SHAREDISK                                                     09738149 T  0501:0
                     END ELSE                                                   09738200 T  0501:0
                     WRITBACK ← TRUE;                                           09738300 T  0501:0
          IF CODE≠32 THEN P(0,RTN);                                             09738400 T  0504:0
EOFSETCK:                                                                       09738500 T  0505:3
          IF (WORDSLEFT < BUFFSIZE) AND                                         09738600 T  0505:3
               (NOT(HOWOPEN) OR (OPENIO AND WRITBACK))                          09738700 T  0507:2
          THEN BEGIN                                                            09738800 T  0511:0
               NUMBUF := 1;                 %WRITE LAST BUFFER               09738900 T  0512:0
               WRIT;                        %AND CHECK FOR                   09739000 T  0514:2
               WAITIO;                      %ERRORS                          09739100 T  0516:0
               NUMBUF := BUFFNUM;                                            09739200 T  0518:2
               IF NOT PRESENT THEN                                           09739300 T  0521:3
                    BEGIN                                                    09739400 T  0523:0
                    SETPRESENTSBIT;                                          09739500 T  0523:2
 $ SET OMIT = NOT SHAREDISK                                                  09739549 T  0525:0
                    ERROR;                                                   09739600 T  0525:0
               END  END;                                                     09739700 T  0526:0
          IF CODE = 6 THEN P(0,RTN);                                        09739750 T  0526:0
          IF SERIAL AND CODE ≠ 2 THEN        % ONLY 1 EOF ALLOWED            09739800 T  0527:3
               IF ENDFILE THEN TERM(15) ELSE ENDFILE ← TRUE;                09739900 T  0530:1
          IF CODE = 32 THEN                 % CLEAR WORK AREA               09740000 T  0536:2
               BEGIN  H ← WA;               % IF READ BEYOND EOF            09740100 T  0537:1
                    FOR RT ← 0 STEP 1 UNTIL (NUMWDS-1) DO H[RT] ← 0;        09740200 T  0540:0
               END;                                                         09740300 T  0546:0
          IF CODE ≠ 2 THEN                  % LET PROGRAM KNOW ITS EOF      09740400 T  0546:0
               BEGIN                                                        09740410 T  0546:3
 $ SET OMIT = NOT SHAREDISK                                                 09740419 T  0547:1
               P(1,RTN);                                                    09740430 T  0547:1
               END ELSE                                                     09740440 T  0547:3
               LASTDONE ← FALSE;            % PREVENT SERIALIO OVERWRITE    09740500 T  0547:3
   %%% END OF EOF CHECKING                                                  09740600 T  0550:3
END OF COBOL DISK INTRINSICS;                                               09740700 T  0550:3
                                                                                    SIZE= 0551 WORDS
```

```
PROCEDURE INTERRUPTER;  % EXECUTION FORCED BY SOFTWARE INTERRUPT CODE AT        09800000 T  0000:0
                                                          START OF REL SEGMENT; DISK ADDRESS = 00683
BEGIN                   % INITIATE, INTERRUPTER PROCESSES ENABLED              09800100 T  0000:0
    REAL  ADDR=+1,      % INTERRUPTS IN SFINTQ, AN IP1 HAS JUST BEEN           09800200 T  0000:0
          I   =+2,      % EXECUTED, POINTING REG-F AND REG-S TO THE             09800300 T  0000:0
       NOTDONE=+3,      % STACK COPY OF THE OLD INCW,                           09800400 T  0000:0
          DONE=+4;                                                            09800500 T  0000:0
    REAL  PERFORMGEN=13;                                                      09800700 T  0000:0
    ARRAY TSKA      =22[*],           % TASK ARRAY                           09800750 T  0000:0
                  % CONTENTS OF TSKA[8]:                                      09800755 T  0000:0
                  %    [1:1]=1 IFF INTERRUPTER HAS JUST RUN AND               09800760 T  0000:0
                  %            SFINTQ IS NON-EMPTY                            09800765 T  0000:0
                  %    [2:1]=1 IFF SFINTQ IS NON-EMPTY                        09800770 T  0000:0
                  %    [3:1]=1 IFF INTERRUPTER IS RUNNING                     09800775 T  0000:0
                  %    [4:1]: SFINTQ INTERLOCK BIT                            09800780 T  0000:0
                  %    [FF] = ADDRESS OF OLD IRCW                             09800785 T  0000:0
                  %    [CF] = RELATIVE PRT ADDRESS OF FIRST IN LINKED         09800790 T  0000:0
```

```
                    %           LIST OF DECLARED INTERRUPTS                          09800795 T  0000:0
          SFINTQ    =27[*],               % SOFTWARE INTERRUPT QUEUE                09800800 T  0000:0
          PRTBASE   =10[*];                                                        09800900 T  0000:0
     LABEL AGAIN;                                                                  09800920 T  0000:0
     DEFINE IMASK = @200000000000000#;                                             09800940 T  0000:0
% * * * * * * * * * * * S T A R T   H E R E * * * * * * * * * * * * * * * * *     09800950 T  0000:0
          TSKA[8] ← ABS(*P(DUP)) & 1 [3:47:1];                                     09800955 T  0000:0
          IF NOT TSKA[8].[4:1] THEN P([TSKA[8]],IMASK,2,COM,DEL,DEL);              09800970 T  0002:3
          TSKA[8].[4:1] ← 0;                                                       09800980 T  0006:1
          P(0,0,0,0);                                                              09801000 T  0008:3
AGAIN:    WHILE I<SFINTQ.[8:10] DO                                                 09801100 T  0009:3
     BEGIN     IF SFINTQ[I]≠0 THEN              % VALID ENTRY                     09801150 T  0011:3
               IF M[SFINTQ[I]+1]<0              % LINK WORD                       09801200 T  0012:3
                    THEN NOTDONE ← 1            % SKIP DISABLED INTERRUPT         09801300 T  0015:1
                    ELSE                        % PERFORM ENABLED INTERRUPT       09801400 T  0016:1
               BEGIN     ADDR ← SFINTQ[I];                                       09801410 T  0016:3
                         TSKA[8] ← *P(DUP) OR IMASK;                             09801420 T  0020:0
                         P(MKS,ADDR-PRTBASE,[CF],0,PERFORMGEN);                  09801450 T  0022:0
                         IF NOT TSKA[8].[4:1] THEN                               09801460 T  0024:1
                              P([TSKA[8]],IMASK,2,COM,DEL,DEL);                  09801470 T  0025:2
                         TSKA[8].[4:1] ← 0;                                      09801480 T  0027:3
                         SFINTQ[I] ← 0;                                          09801500 T  0030:1
                         DONE ← 1;                                               09801600 T  0031:2
               END;                                                              09801700 T  0032:1
               I ← I+1;                                                          09801800 T  0032:1
     END;                                                                        09801900 T  0033:2
          IF DONE THEN BEGIN I ← DONE ← NOTDONE ← 0; GO AGAIN; END;              09801950 T  0035:0
          TSKA[8].[1:4] ← 12×NOTDONE + 1;                                        09802000 T  0038:0
          P(47,COM);                                                             09802100 T  0041:2
END INTERRUPTER;                                                                 09802200 T  0042:0
                                                                                        SIZE= 0043 WORDS
```

```
COMMENT DO NOT PUT ANY DECLARATIONS PAST THIS POINT OR THE CONTROL                99998000 T  0000:0
    STATE PROCEDURE WHATINTRNSIC WILL PROBABLY HANG THE SYSTEM;                   99998010 T  0000:0
PROCEDURE WHATINTRINSIC;                                                         99998020 T  0000:0
                                                  START OF REL SEGMENT; DISK ADDRESS = 00685
   BEGIN                                                                         99998030 T  0000:0
   LABEL L;                                                                      99998040 T  0000:0
   P(XIT); P(.L,DEL);                                                            99998050 T  0000:0
L:::                                                                             99998100 T  0000:3
   "INTRINS",                                                                    99998200 T  0001:0
   "ICS  @@",                                                                    99998300 T  0002:0
   "XVI.0,@",                                                                    99998400 T  0003:0
% PATCH LEVEL ON NEXT CARD PLEASE                                               99998500 T  0004:0
   "00@@@@@",                                                                    99999000 T  0004:0
 $ SET OMIT = NOT(TIMESHARING)                                                   99999100 T  0005:0
   " INCLUD",                                                                    99999840 T  0005:0
   "ES @@@@",                                                                    99999850 T  0006:0
   "TIMESHA",                                                                    99999860 T  0007:0
   "RING@@@",                                                                    99999870 T  0008:0
 $ POP OMIT                                                                      99999880 T  0009:0
   "←      ";                                                                    99999890 T  0009:0
   END WHATINTRINSIC;                                                            99999900 T  0010:0
                                                                                        SIZE= 0011 WORDS
```

```
                END.

NUMBER OF ERRORS DETECTED = 000.  COMPILATION TIME = 1513 SECONDS.
PRT SIZE=250 BASE ADDRESS=0000 CORE REQ=0000 DISK REQ=20580

 LABEL  000000000LINE   001771138?  EXECUTE ESPOL/DISK

                                                99999990 T  0000:0
                                                       SIZE= 0000 WORDS

                                                                ESPOL  /DISK
```