patches at lines
481-482
1

DAN ROSS

B5700 SNOBOL COMPILER

this listing is a copy of
tape CUBEA13
file SNOBOL/L2ØØ1Ø

---

when a copy of this file is
on the disk, it should be
named SYMBOL/SNOBOL
type=ALGOL

```
10000 %$RESET LIST
20000 % B5500 SNOBOL3 SYSTEM.
30000 %
40000 % WRITTEN BY:
50000 % JOHN M. CHAMBERS
60000 % DEPT. OF COMPUTER SCIENCES
70000 % UNIVERSITY OF WISCONSIN
80000 % 1210 W. DAYTON ST.
90000 % MADISON, WISCONSIN  53706
100000 %
110000 % USERS MANUAL AVAILABLE FROM ABOVE ADDRESS.
120000 %
130000 %
140000 %  DATA REPRESENTATION:
150000 %
160000 %  TO TAKE ADVANTAGE OF THE CHARACTER MODE STRING COMPARE AND MOVE
170000 %  INSTRUCTIONS, STRINGS ARE STORED AS CONSECUTIVE CHARS, PRECEDED BY A
180000 %  3-CHAR BACK-POINTER TO THE SYMBOL TABLE.  THIS POINTER CONSISTS OF AN
190000 %  "ILLEGAL CHARACTER" FOLLOWED BY 2 CHARS (12 BITS) GIVING A SYMBOL
200000 %  TABLE LOCATION (SEE DESCRIPTION OF SYMBTABL).  STRINGS ARE STORED IN
210000 %  ROWS OF THE ARRAY DATA[*,*].
220000 %
230000 %
240000 %     DUE TO THE SIZE LIMITATION ON ALGOL ARRAYS, AND THE FACT THAT IT
250000 %  WAS FELT UNDESIRABLE TO BREAK UP A STRING, THERE IS A LIMITATION TO
260000 %  THE SIZE OF A STRING--8181 CHARS, TO BE EXACT.  IF THIS LIMIT IS
270000 %  EXCEEDED, THE PROCEDURE STRING WILL PRINT A DIAGNOSTIC AND SET THE
280000 %  FLAG DEATH TO TRUE, TERMINATING THE PROGRAM.
290000 %     A STRING IS IDENTIFIED BY A 31-BIT "DESCRIPTOR"; IF D IS SUCH A
300000 %  DESCRIPTOR, THE FOLLOWING FIELDS LOCATE THE STRING (SEE DEFINE
310000 %  DECLARATIONS):
320000 %     D.S: SIZE OF STRING
330000 %     D.R: STRING IS IN DATA[D.R,*]
340000 %     D.CH: FIRST CHAR OF STRING, RELATIVE TO DATA[D.R,0]
350000 %  IN OTHER WORDS, THE STRING CONSISTS OF D.S CHARACTERS, STARTING D.C
360000 %  CHARS AFTER DATA[D.R,D.W].
370000 %     NOTE THAT, SINCE ALL STRINGS ARE PRECEDED BY A 3-CHAR POINTER
380000 %  TO SYMBTABL, THE VALUE OF THE CH FIELD MUST BE ≥ 3.  THIS FACT
390000 %  IS USED TO DISTINGUISH CELLS WHICH HAVE "MISSING" VALUES (SUCH
400000 %  AS THE VALUE PARTS FOR STRINGS WHICH HAVE NOT BEEN ASSIGNED A
410000 %  VALUE).  THUS, IF THE CH FIELD IS < 3, THERE IS NO STRING ATTACHED,
420000 %  AND THE VALUE IS CONSIDERED TO BE NULL.  THE S FIELD SHOULD BE 0
430000 %  IN SUCH CASES.
440000 %     SEVERAL "BUGS" HAVE BEEN FOUND TO BE CAUSED BY NOT RECOGNIZING
450000 %  THAT A STRING DESCRIPTOR MAY NOT POINT TO A STRING--THIS CAN BE
460000 %  AVOIDED BY TESTING TO SEE IF THE CH FIELD IS ≥ 3.  IF SO, THE
470000 %  STRING EXISTS; IF NOT, THERE IS NO STRING, AND THE VALUE IS NULL.
480000 %
490000 %
500000 %
510000 %
```

```
520000 %*************************************************************************
530000 BEGIN
540000 INTEGER COMMON;     % THESE BITS ARE CURRENTLY RELEVANT:
550000    % [47:1] = 1 TURNS ON TRACING FOR ALL VARIABLES.
560000    % [46:1] = 1 CAUSES A COMPLETE DUMP AT THE END OF RUN.
570000    % [45:1] = 1 SAYS THAT MODE("INFORM") OR =INFORM IS TO TURN ON
580000    %     SYSTEM DEBUGGING AIDS.
590000 DEFINE
600000    C=[45:3]#,      % CHAR WITH IN WORD OF STRING DESCRIPTOR
610000    C1=[6:6]#,
620000    C2=[12:6]#,
630000    C3=[18:6]#,
640000    C4=[24:6]#,
650000    C5=[30:6]#,
660000    C6=[36:6]#,
670000    C7=[42:6]#,
680000    CH=[35:13]#,    % CHAR OF DATA[*,*] ROW OF STRING DESCRIPTOR
690000    CQUOTE4=""""[24:42:6]#,    % MOVE QUOTE TO CHAR 4
700000    CQUOTE5=""""[30:42:6]#,    % MOVE QUOTE TO CHAR 5
710000    CONCR=[30:43:5]#,    % MOVE TO R FIELD
720000    CONCS=[17:35:13]#,   % MOVE TO S FIELD
730000    CONCW=[35:38:10]#,   % MOVE TO W FIELD
740000    CONTROLPOINT=INST[INSTRUCT],NEARPOINT#,
750000    DATASIZE=31#,        % UPPER BOUND ON FIRST SUBSCRIPT OF DATA[*,*]
760000    DISKRECORD=RECORD#,
770000    F= FALSE #,
780000    FIRSTCHAR(FIRSTCHAR1)=DATA[(FIRSTCHAR1),R,(FIRSTCHAR1),W],
790000       (FIRSTCHAR1),C#,
800000    INTRINSMAX=17#,      % MAX # INTRINSIC FCTS OF SAME SIZE
810000    MAXLABELLIMIT=15#,   % MAX NUMBER OF LABEL LIMITS FOR DEBUGGING
820000    MAXLEVEL=100#,  % MAXIMUM ALLOWED DEPTH OF FUNCTION CALLS
830000    MAXSUSPENDREASON=3#, % NUMBER DIFFERENT REASONS FOR SUSPENDING.
840000    MNEMMAX=20#,
850000    NUMCONTROLCARDS=21#, % NUMBER OF CONTROL CARDS IMPLEMENTED;
860000    %                      MUST BE CHANGED IF NEW CONTROL CARDS ARE
870000    %                      TO BE IMPLEMENTED
880000    R=[30:5]#,      % DATA[*,*] ROW OF STRING DESCRIPTOR
890000    RCH=[30:18]#,   % R, W, AND C FIELDS
900000    S=[17:13]#,     % SIZE OF STRING
910000    SEGMENT = LABEL DUMMY;#,  % FOR FORCING NEW SEGMENT.
920000    STACKSIZE=50#,  % SIZE OF INTERPRETER STACK ARRAYS
930000    STRINGLOC(STRINGLOC1)=(STRINGLOC1),S,FIRSTCHAR(STRINGLOC1)#,
940000    T= TRUE #,
950000    TEMP=AA#,       %
960000    TEMP1=AB#,      %
970000    TEMP2=AC#,      %
980000    TEMPCEL(TEMPCEL1) = TEMPCELL#, % PARAMETER NO LONGER IN USE
990000    VERSION = 3.0#, % VERSION NUMBER:
1000000   %  2.0  IS THE REWRITTEN COMPILER.
1010000   %  2.1  INCLUDES FATAL ERROR SUSPENSIONS WHEN DEBUGGING.
1020000   %  2.2  HAS MOST SYSTEM DEBUGGING AIDS CHANGED TO COMMENTS.
1030000   %  3.0 HAS THE ARRAY USEDCELL[*] DELETED (CHANGES LIBRARY FILES).
1040000   W=[35:10]#;     % WORD WITHIN DATA[*,*] ROW OF STRING DESCRIPTOR
1050000 DEFINE
1060000    ABORT = GO TO ABORTION;#,  % FATAL SYSTEM ERROR.
1070000    SETLIMITFLAG = ALIMITEXISTS +
1080000       CPULIMITEXISTS OR
```

```
1090000        IOLIMITEXISTS OR
1100000        RULELIMITEXISTS OR
1110000        (DEBUGGING AND
1120000           (DEBUGRULELIMITEXISTS OR
1130000           NLABELLIMIT ≥ 0
1140000           )
1150000        )
1160000        #;
1170000 BOOLEAN
1180000    ALIMITEXISTS,    % THERE IS A LIMIT TO CHECK AT START OF INSTS.
1190000    B1,              % TEMPORARY BOOLEAN "REGISTER"
1200000    CONVERTF,        % DECK IS 026 CHAR SET
1210000    CONVERTSTRINGS,  % CONVERT 3600 I/O STRINGS
1220000    CPULIMITEXISTS,  %
1230000    DATACOMF,        % DATA COM UNIT IN USE
1240000    DEATH,           % FATAL ERROR HAS OCCURRED
1250000    DEBUGGING,       % DEBUG MODE IN USE
1260000    DEBUGRULELIMITEXISTS,
1270000    DMPDATA,         % EXECUTE WRITEDATA AT END OF RUN
1280000    DMPST,           % EXECUTE WRITEST (DUMP SYMB TABLE) AT END OF RUN
1290000    DMPSTR,          % EXECUTE STRINGDUMP AT END OF RUN
1300000    DUMPALL,         % TRUE IF COMPLETE DUMP TO BE DONE AT EOJ
1310000    ERRDUMP,         % DUMP IF FATAL PROGRAM ERROR
1320000    EXECUTE,         % SET FALSE IF FATAL ERROR DURING COMPILATION
1330000    INFORM,          % PRINT SYSTEM DEBUGGING MESSAGES
1340000    IOLIMITEXISTS,   %
1350000    LOADF,           % LOADER IS BEING EXECUTED
1360000    PRINTMESSAGES,   % PRINT DIAGNOSTIC (WARNING) MESSAGES.
1370000    RESULT,          % SUCCESS/FAIL FLAG FOR INTERPRETER
1380000    RULELIMITEXISTS,%
1390000    SYSTEMDEBUGGING,% TRUE IF COMMON.[45:1] = 1
1400000    SYSTEMERROR,     % FATAL ERROR BY SNOBOL SYSTEM, NOT PROGRAM
1410000    TRACEALL;        % TRACE EVERY VARIABLE WITHOUT I/O USE
1420000 BOOLEAN ARRAY
1430000    SUSPENDREASON[0:MAXSUSPENDREASON], % REASONS FOR SUSPENDING PROGRAM:
1440000    %                        0: SUSPEND() WAS EXECUTED
1450000    %                        1: RULE LIMIT WAS HIT
1460000    %                        2: LABEL LIMIT WAS HIT
1470000    %                        3: FATAL ERROR IN PROGRAM
1480000    USEDROW[0:DATASIZE]; % DATA[I,*] IS IN USE
1490000 INTEGER
1500000    ANCHORMODE,      % 2: UNANCHORED, 3: ANCHORED
1510000    ANCHORSIZE,      % FOR ANCHORED MODE, SIZE OF INITIAL SEGMENT
1520000    DCSIZE,          % SIZE OF LAST INPUT FROM DCREAD
1530000    DEBUGRULELIMIT,
1540000    DIVIDEMODE,      % 0: ROUND; 1: TRUNCATION; 2: INTEGER
1550000    GCS,             % NUMBER OF GARBAGE COLLECTOR CALLS
1560000    I,
1570000    I1, I2,          % TEMPORARY INTEGER "REGISTERS".
1580000    J,
1590000    LEVEL,           % LEVEL (OR DEPTH) OF RECURSION OF THE INTERPRETER
1600000    LISTSPACES,      % NUMBER OF BLANK LINES BETWEEN LINES OF LISTING
1610000    LOADERLEVEL,     % LEVEL OF RECURSION IN LOADER
1620000    MAXINSTSIZE,     % SIZE OF LARGEST PROGRAM SEGMENT (IN CHARS)
1630000    NLABELLIMIT,     % NUMBER OF LABEL LIMITS SET AT LAST SUSPENSION
1640000    PARENCOUNT,      % DEPTH IN PARENTHESIS NESTING (COMPILE AND SCAN)
1650000    RW,              % NEW STRINGS COME FROM DATA[RW,*] (SEE STRING)
```

```
1660000     TEMPROW,          % ROW OF SYMB TABLE GIVING TEMP CELLS
1670000     UNIT;             % USED BY I/O ROUTINES--FILE NUMBER
1680000 INTEGER ARRAY
1690000     CONVERTVAL,       % FOR CHAR SET CONVERSIONS
1700000        DOTTYPE,       % PARTITIONS CHARS IN INTERNAL REP. OF ARITH,
1710000        OPLEVEL[0:63],    % PRECEDENCE LEVEL OF ARITH OPS
1720000     DPNTR,            % POINTER TO NEXT AVAILABLE CHAR IN DATA[I,*]
1730000        NOTMOVED[0:DATASIZE],  % POINTS TO FIRST CHAR MOVED BY LAST CALL
1740000        %                      OF GARBAGE COLLECTOR.
1750000     MONTHS[1:12],     % NUMBER OF DAYS IN EACH MONTH
1760000     RULES[0:MAXLEVEL];   % # RULES EXECUTED AT VARIOUS LEVELS
1770000 REAL
1780000     CPULIMIT,         % LIMIT TO CPU TIME IN SECONDS.
1790000     GCTIMECP,         % CPU TIME SPENT COLLECTING GARBAGE
1800000     GCTIMEIO,         % I/O TIME SPENT COLLECTING GARBAGE
1810000     IOLIMIT,          % LIMIT TO I/O TIME IN SECONDS.
1820000     RANDNO,           % NUMBER FOR .RANF FUNCTION
1830000     RULELIMIT,        % LIMIT TO NUMBER OF RULES EXECUTED.
1840000     TEENYNEG;         % = OCT3777777777777777
1850000 ALPHA
1860000     AA,AB,AC,         % TEMPORARY ALPHA "REGISTERS"
1870000     ARROW,            % = "←"
1880000     BLANK,            % = " "
1890000     BLANKS,           % = "        " (8 BLANKS)
1900000     COLON,            % = ":"
1910000     COMMA,            % = ","
1920000     CRLF,             % = "≤≠←"
1930000     EQSIGN,           % = "="
1940000     QMARK,            % ILLEGAL CHAR
1950000     QUOTE,            % = """
1960000     PRINTLOC,         % SYMBOL TABLE LOCATION OF PRINT (SEE CODE
1970000                       % FOR SYSPOT IN SNBLOUT.)
1980000     SLASH,            % = "/"
1990000     STAR,             % = "*"
2000000     STOPPER;          % = BLANK & QUOTE & QMARK (FOR ENDING SCANS.)
2010000 ALPHA ARRAY
2020000     BUFFER[0:14],     % USED BY COMPILER TO HOLD NEXT INPUT CARD
2030000     BUFOUT[0:16],     % FOR BUILDING PRINTER OUTPUT MESSAGES.
2040000     CODE[0:1022],
2050000     CONTROLCARD[0:NUMCONTROLCARDS],  % NAMES OF CONTROL CARDS
2060000     DATA[0:DATASIZE,0:1022],  % STRING AND PROGRAM STORAGE AREA
2070000     INTRINSFCT[3:7,0:INTRINSMAX],  % NAMES OF INTRINSIC FUNCTION.
2080000     INTRINSNDX[3:7,1:INTRINSMAX],  % INDICES OF INTRINSIC FUNCTIONS.
2090000     LABELLIMIT[0:MAXLABELLIMIT],   % SYMBOL TABLE LOCATIONS OF
2100000     %                        LABEL LIMITS DURING DEBUGGING
2110000     MNEMONIC[0:1,0:MNEMMAX],  % VARIOUS MNEMONIC COMMANDS AND WORDS
2120000     NEXTRECORD[0:15,0:17], % HOLDS RECORDS FOR LOOK-TYPE STRINGS
2130000     SCRATCH[0:1022],      % TEMPORARY STRING ARRAY
2140000     WORDS[0:30];          % VARIOUS ALPHANUMERIC STRINGS
2150000 MONITOR INTOVR, EXPOVR, INDEX, FLAG;
2160000 %********SYMBOL TABLE*****************************************************
2170000 %
2180000 % THE SYMBOL TABLE:
2190000 %
2200000 %
2210000 % EVERY DATA OBJECT (STRINGS, LITERAL, FUNCTION, LABEL) REQUIRES
2220000 % AN ENTRY IN THE SYMBOL TABLE.  THIS TABLE CONSISTS OF THE FOLLOWING
```

```
2800000 INTEGER ARRAY
2810000    NEXTCELL[0:STRMAX];  % HEADS OF AVAILABLE SPACE LISTS.
2820000 %
2830000 %
2840000 % A SYMBOL TABLE ENTRY CONSISTS OF THREE WORDS--ONE WORD
2850000 % FROM EACH OF THE THREE ARRAYS.  A "SYMBOL TABLE ADDRESS" IS A
2860000 % 12-BIT INTEGER CONSISTING OF TWO FIELDS CALLED STR AND STW.
2870000 % THE THREE WORDS ASSOCIATED WITH AN ADDRESS P CAN BE REFERRED
2880000 % TO IN THE FOLLOWING WAYS:
2890000 %
2900000 %     NAME[P] = NAMTABL[P.STR,P.STW] IS THE NAME OF THE OBJECT
2910000 %     VALU[P] = VALTABL[P.STR,P.STW] IS THE VALUE OF THE OBJECT.
2920000 %     IO[P] = IOTABL[P.STR,P.STW] IS THE I/O USE OF THE OBJECT
2930000 %
2940000 % NORMALLY, THE FIRST FORM GIVEN (SINGLY SUBSCRIPTED) IS USED; THE
2950000 % DOUBLY-SUBSCRIPTED FORMS ARE USED ONLY WHEN THE TWO SUBSCRIPTS
2960000 % HAVE BEEN CALCULATED SEPARATELY.
2970000 %
2980000 % THE SYMBOL TABLE IS A SCATTERED-ENTRY, LINKED-LIST TYPE OF TABLE.
2990000 % ENTRIES ARE SCATTERED INTO THE "SCATTER AREAS", WHICH ARE WORDS 0 TO
3000000 % SCATTERSIZE+15 OF ROWS 0 TO SCATTERNO.  THAT IS, THE I,J WORDS
3010000 % OF THE SYMBOL TABLE (NAMTABL[I,J], VALTABL[I,J], IOTABLE[I,J])
3020000 % ARE IN THE SCATTER AREA IF:
3030000 %     0 ≤ I ≤ SCATTERNO
3040000 %     0 ≤ J ≤ SCATTERSIZE+15
3050000 % WORD 0,0 IS NOT INCLUDED; IT IS A "NON-CELL" WHICH SHOULD ALWAYS
3060000 % BE FILLED WITH ZEROES.  THE REASON FOR THE "+15" IS THAT THERE
3070000 % ARE REALLY TWO SCATTER AREAS--ONE FOR LITERALS AND ONE FOR ALL
3080000 % OTHER TYPES OF DATA OBJECTS.  THIS IS TO KEEP THE LENGTH OF
3090000 % LISTS THAT MUST BE SCANNED AT RUN TIME (FOR INDIRECTION, CREATING
3100000 % NEW FUNCTIONS, ETC.) AS SMALL AS POSSIBLE.  LITERALS ARE SCATTERED
3110000 % INTO THE [SCATTERSIZE,SCATTERSIZE+15] PART OF EACH ROW, AND THE
3120000 % [0,SCATTERSIZE) PART IS FOR OTHER TYPES OF OBJECTS.  THE PART
3130000 % OF THE SYMBOL TABLE OUTSIDE OF THE SCATTER AREAS IS INITIALIZED
3140000 % AS AVAILABLE SPACE LISTS (ONE PER ROW, THE TOP ADDRESS GIVEN BY
3150000 % NEXTCELL[ROW]).  WHEN SEVERAL ENTRIES SCATTER INTO THE SAME
3160000 % LOCATION, A LINKED LIST IS FORMED INTO THE PART OF THE ROW THAT
3170000 % IS OUTSIDE THE SCATTER AREA.  LINKS THROUGH THESE LISTS ARE IN
3180000 % THE LINK FIELD OF THE NAME PART (NAME[P]).
3190000 %
3200000 %  THE FOLLOWING FIELDS ARE USED IN A SYMBOL TABLE ENTRY:
3210000 %  NAME[P].TYPE IS THE TYPE OF THE ENTRY:
3220000 %     0 = NAMED STRING OR TEMPORARY CELL.
3230000 %     1 = LITERAL
3240000 %     2 = FUNCTION
3250000 %     3 = LABEL (CODE SEGMENT)
3260000 %  NAME[P] AND GARBAGE COLLECTOR.[3:1] IS USED BY CHECKSYMBTABL.
3270000 %  NAME[P].[4:1] IS UNUSED.
3280000 %  NAME[P].LINK IS THE LINK TO THE NEXT ENTRY THAT HAD THE SAME
3290000 %     LOCATION IN THE SCATTER AREA.  A ZERO LINK IS END-OF-LIST.
3300000 %     FOR TEMPORARY CELLS, THE LINK SHOULD BE ZERO.
3310000 %  NAME[P].LOC POINTS TO THE NAME OF THE OBJECT (IN DATA[*,*]).
3320000 %     FOR LITERALS, THIS FIELD IS ZERO.
3330000 %  VALU[P].[1:1] = 1 FOR PROGRAM-DEFINED FUNCTIONS, OTHERWISE
3340000 %     IT SHOULD BE ZERO.
3350000 %  VALU[P].[2:1] IS UNUSED.
3360000 %  VALU[P].IOUSE INDICATES THE I/O USE, AND HAS TWO SUBFIELDS:
```

```
2230000 % THREE ARRAYS AND ASSOCIATED MACROS:
2240000 %
2250000 DEFINE
2260000    NAME[NAME1] = NAMTABL[(NAME1).STR,(NAME1).STW] #,
2270000    VALU[VALU1] = VALTABL[(VALU1).STR,(VALU1).STW] #,
2280000    IO[IO1] = IOTABL[(IO1).STR,(IO1).STW] #,
2290000    STRMAX = 15 #,      % MAX FIRST SUBSCRIPT TO SYMB TABLE.
2300000    STWMAX = 255 #;     % MAX SECOND SUBSCRIPT TO SYMB TABLE.
2310000 ALPHA ARRAY
2320000    NAMTABL,
2330000    VALTABL,
2340000    IOTABL[0:STRMAX,0:STWMAX];
2350000 %
2360000 % SOME MORE VARIABLES ASSOCIATED WITH THE SYMBOL TABLE ARE:
2370000 %
2380000 DEFINE
2390000    CCYCLE=[9:44:4]#,    % MOVE TO CYCLE FIELD
2400000    CFILNO=[5:44:4]#,    % MOVE TO FILNO FIELD
2410000    CINUSE=[3:47:1]#,    %
2420000    CIOTYPE=[14:45:3]#,  % MOVE TO IOTYPE FIELD
2430000    CIOUSE=[3:46:2]#,
2440000    CLINK=[5:36:12]#,    % MOVE TO LINK FIELD
2450000    CONCSTR = [36:44:4] #,
2460000    COUTUSE=[4:47:1]#,   %
2470000    CYCLE=[9:4]#,        % NUMBER TIMES FILE LOCATION USED
2480000    FILNO=[5:4]#,        % INDEX OF ASSOCIATED FILE
2490000    INUSE=[3:1]#,        % = 1 IF INPUT STRING
2500000    IOTYPE=[14:3]#,      % TYPE OF I/O USAGE:
2510000    %                        VALUE     INPUT      OUTPUT
2520000    %                          0       ILLEGAL    ILLEGAL
2530000    %                          1       I/O        I/O
2540000    %                          2       LOOK       TRACE
2550000    %                          3       TELETYPE   TELETYPE
2560000    %                          4       ILLEGAL    ILLEGAL
2570000    %                          5       ILLEGAL    SYSPOT
2580000    %
2590000    IOUSE=[3:2]#,    % INUSE AND OUTUSE FIELDS COMBINED.
2600000    LINK=[5:12]#,    % LIST LINKS IN SYMBOL TABLE.
2610000    LOC=[17:31]#,    % S, R, W, AND C FIELDS (STRING LOC IN DATA[*,*])
2620000    OUTUSE=[4:1]#,   % = 1 IF OUTPUT STRING
2630000    OVFL=[13:1]#,    % = 1 IF OVERFLOW ALLOWED ON OUTPUT
2640000    SCATTERSIZE=75#,% SIZE OF SCATTER AREAS IN SYMB TABLE.
2650000    STR = [36:4] #,
2660000    STW = [40:8] #,
2670000    TCYCLE=[9:9:4]#,% TRANSFER TO CYCLE FIELD
2680000    TFILNO=[5:5:4]#,% TRANSFER TO FILNO FIELD
2690000    TYPE=[1:2]#;     % TYPE OF ENTRY:
2700000    %                  0 = STRING       2 = FUNCTION
2710000    %                  1 = LITERAL      3 = LABEL
2720000 BOOLEAN
2730000    SYMBTABLSETUP;      % AVSLS SET UP IN SCATTER ROWS.
2740000 BOOLEAN ARRAY
2750000    USEDST[0:STRMAX];   % TRUE IF SYMB TABLE ROW IN USE.
2760000 INTEGER
2770000    SCATTERNO;          % MAX ROW INDEX OF SCATTER AREAS.
2780000 ALPHA ARRAY
2790000    TEMPLIST[0:STRMAX]; % TEMP S.T. LOCS FOR INTERPRETER.
```

```
3370000 %     VALU[P].INUSE = 1 FOR INPUT STRINGS AND FUNCTIONS WITH
3380000 %        THE CALLS BEING TRACED.
3390000 %     VALU[P].OUTUSE = 1 FOR OUTPUT STRINGS, LABELS BEING TRACED,
3400000 %        AND FUNCTIONS WITH RETURNS BEING TRACED.
3410000 %  VALU[P].LINK HAS SEVERAL USES, DEPENDING ON THE TYPE OF THE OBJECT:
3420000 %     STRINGS: THE LINK FIELD GIVES THE NEXT CELL ON THE PUSH-DOWN
3430000 %        STACK.  IF ZERO, THERE IS NO PUSH-DOWN STACK.
3440000 %     INTRINSIC FUNCTIONS:  THE INDEX FOR THE CASE STATEMENT IN
3450000 %        THE INTERPRETER THAT BRANCHES TO THE FUNCTION CODE.
3460000 %     PROGRAM-DEFINED FUNCTIONS:  THE SYMBOL TABLE LOCATION OF
3470000 %        THE ENTRY POINT.
3480000 %     LABELS: THE REFERENCE COUNT.  IT IS BUMPED BY ONE EACH TIME
3490000 %        THE LABEL IS ENCOUNTERED.
3500000 %  VALU[P].LOC IS THE LOCATIONS OF:
3510000 %     STRINGS: THE VALUE.
3520000 %     LITERALS: DITTO.
3530000 %     FUNCTIONS: (PROGRAM-DEFINED) A STRING OF 2-CHAR POINTERS TO
3540000 %        THE SYMB TABLE LOCATIONS OF THE FORMAL PARAMETERS AND
3550000 %        LOCAL VARIABLES.  FOR INTRINSIC FUNCTIONS, THIS FIELD
3560000 %        IS IGNORED.
3570000 %  IO[P].[1:2] IS UNUSED.
3580000 %  IO[P].IOUSE = VALU[P].IOUSE (REDUNDANT).
3590000 %  IO[P].FILNO IS THE INDEX TO THE FILE ARRAYS FOR FILE I/O STRINGS.
3600000 %  IO[P].CYCLE IS CURRENTLY NOT IN USE.
3610000 %  IO[P].OVFL = 1 FOR OUTPUT STRINGS WHEN THE TAILS OF OUTPUT STRINGS
3620000 %     TOO LONG FOR A RECORD ARE TO BE WRITTEN ON THE NEXT RECORD.
3630000 %     NORMALLY = 0, WHICH CAUSES TRUNCATION OF LONG OUTPUT STRINGS.
3640000 %  IO[P].IOTYPE IS THE TYPE OF I/O OPERATION, AS FOLLOWS:
3650000 %                  VALUE     INPUT     OUTPUT
3660000 %                    0       ILLEGAL   ILLEGAL
3670000 %                    1       I/O       I/O
3680000 %                    2       LOOK      TRACE
3690000 %                    3       TELETYPE  TELETYPE
3700000 %                    4       ILLEGAL   ILLEGAL
3710000 %                    5       ILLEGAL   SYSPOT
3720000 %  IO[P].[17:31] IS UNUSED.
3730000 %********LOCAL VARIABLES FOR COMPILER**************************************
3740000 BOOLEAN
3750000     BUFFERFULL, % BUFFER[*] CONTAINS A CARD.
3760000     DCLIST,     % LIST PROG ON TELETYPE.
3770000     GT,         % FALSE AT START OF COMPILE; TRUE WHEN GO-TO PART REACHED
3780000     GTLAST,     % LAST INST HAD A GO-TO PART
3790000     GTF,        % FAILURE EXIT HAS BEEN FOUND
3800000     GTS,        % SUCCESS EXIT HAS BEEN FOUND
3810000     LSTF,       % LIST PROGRAM ON LINE PRINTER FILE PRINT
3820000     PCC,        % LIST CONTROL CARDS, EVEN IF NOT LSTF
3830000     PROGRAMFROMREMOTE,   % PROGRAM FROM TELETYPE, NOT FILE PROGRAM
3840000     PUNCHF;     % PUNCH NEW PROGRAM DECK
3850000 INTEGER
3860000     ERRORS,     % NUMBER OF SYNTAX ERRORS FOUND
3870000     FIELDSIZE,  % NUMBER OF CHARS PER CARD THAT ARE INSTRUCTION
3880000     INSTNUM,    % NUMBER OF INST BEING COMPILED
3890000     INSTSIZE,   % SIZE OF INST BEING COMPILED
3900000     MARKER,     % FIRST CHAR (QMARK) OF INST IN CODE[*]
3910000     MESSAGES,   % NUMBER OF INFORMATIVE MESSAGES PRINTED
3920000     P,          % POINT IN INST[*] BEING COMPILED
3930000     PTR;        % NEXT CHAR IN CODE[*]
```

```
3940000 INTEGER ARRAY
3950000     BACKTRACK[0:50,0:5],
3960000     CHARTYPE[0:63];
3970000 ALPHA
3980000     NEXTSEGMENT,% SYMBTABL LOC OF NEXT LABELED INSTRUCTION
3990000     SLASTLABEL; % SYMBTABL LOC OF LAST LABELED INSTRUCTION
4000000 %********LOCAL VARIABLES FOR COMPILER************************************
4010000 %********STACK ARRAYS****************************************************
4020000 %  THE FOLLOWING ARRAYS CONSTITUTE THE "STACK" USED BY THE
4030000 %  COMPILER AND INTERPRETER.  THE INDEX OF THE NEXT WORD AVAILABLE
4040000 %  IS THE GLOBAL INTEGER SP.
4050000 BOOLEAN ARRAY
4060000     PNAME[0:STACKSIZE];  % TRUE FOR NAMED QUANTITIES
4070000 INTEGER ARRAY
4080000     PBACK,               % LINK FOR BACK REFERENCE
4090000     PMINLEFT,            % MIN SIZE OF REST OF PATTERN
4100000     PPOINT,              % POINTER TO MATCHED SUBSTRING
4110000     PSIZE,               % SIZE OF PATTERN ELEMENT
4120000     PTYPE[0:STACKSIZE];  % TYPE OF ELEMENT, AS FOLLOWS;
4130000     %                      0: UNDEFINED
4140000     %                      1: CONSTANT PATTERN ELEMENT
4150000     %                      2: ARBITRARY STRING VARIABLE
4160000     %                      3: FIXED-LENGTH VARIABLE
4170000     %                      4: BALANCED VARIABLE
4180000     %                      5: BACK REFERENCE
4190000     %                      6: UNDEFINED
4200000     %                      7: NUMERIC VALUE--IN PST[I]
4210000 ALPHA ARRAY
4220000     PLOC,                % SCANNER ONLY: LOC OF VALUE OF ELEMENT
4230000     PST[0:STACKSIZE];    % SYMBOL TABLE ADDRESS OF ELEMENT
4240000 %********STACK ARRAYS****************************************************
4250000 %********LOCAL VARIABLES FOR INTERPRETER*********************************
4260000 % WARNING: THE NUMBER OF VARIABLES IN THIS SECTION MUST BE THE SAME
4270000 % AS THE NUMBER OF VARIABLES COPIED IN THE DEFINED SECTION OF FUNCTION.
4280000 BOOLEAN              %
4290000     BACKREFLAG,      %
4300000     NOREPLACEMENT,   %
4310000     NOBACKORBAL,
4320000     SELFREFLAG,      %
4330000     SUCCESS,         %
4340000     VARFLAG;         %
4350000 INTEGER              %
4360000     COUNT,
4370000     ENTRY,           % ENTRY POINT & LAST LABEL TRANSFERED TO
4380000     FLOC,            % POINTER TO FAILURE EXIT.
4390000     FRONTEND,
4400000     INCREASE,
4410000     INSTNO,
4420000     INSTRUCT,
4430000     MKS,             % "MARK STACK" POINTER
4440000     NEARPOINT,
4450000     NEXTSOURCERECORD,    % DISK ADDR OF NEXT AVAILABLE RECORD OF SOURCE.
4460000     NOPATTERNS,
4470000     NOREPLACES,
4480000     REAREND,
4490000     REFI,
4500000     REFJ,
```

```
4510000     REFLOC,
4520000     REFPT,
4530000     REFSTEP,
4540000     RELATIVEPOINTER,
4550000     RPR,
4560000     RSIZE,
4570000     SIZE,
4580000     SLOC,            % POINTER TO SUCCESS EXIT.
4590000     SP;
4600000 ALPHA
4610000     TEMPREF;
4620000 %********LOCAL VARIABLES FOR INTERPRETER**********************************
4630000 %********FILE-HANDLING DECLARATIONS***************************************
4640000 DEFINE FILMAX = 5 #;     % MAX NUMBER OF FILES
4650000 BOOLEAN ARRAY
4660000     IOEOF,           % EOF ON LAST I/O OPERATION.
4670000     IOFILEOPEN,      % TRUE FOR OPEN FILES.
4680000     LOOKF[1:FILMAX];% NEXT RECORD IS IN NEXTRECORD[I,*].
4690000     %                  USED WITH LOOK-TYPE INPUT STRINGS.
4700000 INTEGER ARRAY
4710000     IOSIZE,
4720000     IOSPACE,
4730000     RECORD[1:FILMAX];
4740000 ALPHA ARRAY
4750000     IOUSAGE[1:FILMAX];
4760000 FILE
4770000     CARD (1,10,30);
4780000 FILE IN
4790000     PROGRAM (1,10,30);
4800000 FILE OUT
4810000     PRINT[18] (1,17),
4820000     PUNCH[25] (1,10);
4830000 SAVE FILE OUT
4840000     NEWDISK DISK SERIAL [20:600]  (1,10,30,SAVE 15);
4850000 SWITCH FILE IOFILE :=
4860000     CARD,        % 0--DUMMY ENTRY
4870000     CARD,        % CARD-IMAGE INPUT AND OUTPUT FILE
4880000     PROGRAM,     % PROGRAM FILE FOR BATCH JOBS; READ FILE
4890000     PRINT,       % LINE PRINTER FILE
4900000     PUNCH,       % CARD PUNCH FILE
4910000     NEWDISK;     % FILE FOR CREATING NEW DISK FILES
4920000 %********FILE-HANDLING DECLARATIONS***************************************
4930000 %********SWITCH FORMAT MESSAGE********************************************
4940000 SWITCH FORMAT  MESSAGE   := ("**MISSING QUOTE."),                    %00
4950000 (//"**IGNORE ANY OUTPUT AFTER THIS--IT MAY BE INCORRECT,"//),        %01
4960000 ("**UNRECOGNIZED CONSTRUCT IN STRING REFERENCE."),                   %02
4970000 ("**UNIDENTIFIABLE PUNCHED OBJECT."),                               %03
4980000 ("**DOUBLY DEFINED EXIT."),                                         %04
4990000 ("**CHARACTER AFTER S OR F IN GO-TO NOT (."),                       %05
5000000 ("**STRING REFERENCE MISSING."),                                    %06
5010000 ("**UNRECOGNIZED GO-TO CONSTRUCT."),                                %07
5020000 ("**ILLEGAL LABEL IN GO-TO PART."),                                 %08
5030000 ("**ILLEGAL COMMA."),                                               %09
5040000 ("**PARENTHESIS COUNT NON-ZERO AT START OF GO-TO PART."),           %10
5050000 ("**THE ONLY UNARY OPERATOR IS ","""","-","""","."),                %11
5060000 ("**EXTRA RIGHT PARENTHESIS."),                                     %12
5070000 ("**EXTRA ARROW OR EQUAL SIGN."),                                   %13
```

```
5080000 ("**STRANGE USE OF ","""","/","""",".",),                              %14
5090000 ("**ATTEMPTED REPLACEMENT IN VALUE EXPRESSION."),                      %15
5100000 ("**IMPROPER ARITHMETIC OPERAND."),                                    %16
5110000 ("**ERROR IN USE OF ARITHMETIC OPERATOR."),                            %17
5120000 ("**PARENTHESIS COUNT AT END OF INSTRUCTION NON-ZERO."),               %18
5130000 ("**CONTROL PARAMETER NON-NUMERIC OR OUTSIDE ALLOWED RANGE."),         %19
5140000 (/"**DUMMY FMT--MESSAGE[20]"),                                         %20
5150000 ("**TOO MANY ELEMENTS IN INSTRUCTION"),                                %21
5160000 ("**UNRECOGNIZED INSTRUCTION TYPE."),                                  %22
5170000 (/"**DUMMY FMT--MESSAGE[23]"),                                         %23
5180000 ("**THIS LABEL HAS ALREADY BEEN USED; FORMER VALUE LOST."),            %24
5190000 (/"**NO END CARD."/),                                                  %25
5200000 ("**ENTRY POINT UNDEFINED."),                                          %26
5210000 ("**MISSING RIGHT PARENTHESIS."),                                      %27
5220000 ("**ILLEGAL USE OF LITERAL."),                                         %28
5230000 ("**MISSING OPERAND TO ARITHMETIC"),                                   %29
5240000 (/"**DUMMY FMT--MESSAGE[30]"),                                         %30
5250000 (/"**DUMMY FMT--MESSAGE[31]"),                                         %31
5260000 ("**DEFINE FAILURE--UNKNOWN ERROR IN FIRST ARGUMENT."),                %32
5270000 ("**DEFINE FAILURE--UNKNOWN ERROR IN LOCAL VARIABLE LIST."),           %33
5280000 ("**INVALID ARGUMENT FOR INDIRECTION."),                               %34
5290000 ("USEDROW[",I2,"] IS ",L5,"; DPNTR[",I2,"] = ",I5),                    %35
5300000 (X16,"*",10(X4,"+",X4,"*")),                                          %36
5310000 ("DATA[",I2,"] =                ",                                    %37
5320000 ("**MISSING STRING VARIABLE ASTERISK."),                               %38
5330000 ("**MISSING PARAMETER."),                                              %39
5340000 (///"**COMPILER SCREWED UP."///),                                      %40
5350000 (/"**DATA COMMUNICATIONS FILE IN USE**"/),                             %41
5360000 ("**PROGRAM SEGMENT TOO LONG--INSERT EXTRA LABEL ON PRECEDING"
5370000    " INSTRUCTION."),                                                   %42
5380000 ("**END OF FILE."),                                                    %43
5390000 ("**PARITY ERROR IN "),                                                %44
5400000 (/"**OUT OF SPACE IN STRING STORAGE AREA."),                           %45
5410000 (/"**OUT OF SPACE IN SYMBOL TABLE."),                                  %46
5420000 ("**ILLEGAL USE OF ARROW OR EQUAL SIGN."),                             %47
5430000 ("**NON-NUMERIC LITERAL IN ARITHMETIC."),                              %48
5440000 ("**THIS STATEMENT CAN NOT BE REACHED."),                              %49
5450000 ("**UNDEFINED LABEL: "),                                               %50
5460000 (//"**RETURN FROM FUNCTION NOT PRECEDED BY CALL; STATEMENT ",
5470000    "NUMBER ",I6),                                                      %51
5480000 ("**ILLEGAL NAME FOR STRING VARIABLE."),                               %52
5490000 ("**END FORMAT--MESSAGE.");
5500000 %********SWITCH FORMAT MESSAGE*****************************************
5510000 %********SWITCH FORMAT FTIME******************************************
5520000 SWITCH FORMAT FTIME =
5530000 (/"CPU TIME = ",F10.1," SEC."/"I/O TIME = ",F10.1," SEC."),            %00
5540000 (X25,"B 5 5 0 0     S N O B O L 3     S Y S T E M"//
5550000    "VERSION ",F3.1/
5560000    "THE DATE IS ",I*," ",A3," 19",A2,/
5570000    "COMPILATION STARTED AT ",2(I2,":"),I2),                            %01
5580000 (/"COMPILATION COMPLETED AT ",2(I2,":"),I2),                           %02
5590000 (/"EXECUTION STARTED AT ",2(I2,":"),I2),                               %03
5600000 (/"NORMAL EXIT AT ",2(I2,":"),I2,", AT LEVEL ",I*," IN STATEMENT ",
5610000    I*),                                                                %04
5620000 (/"ABNORMAL EXIT AT ",2(I2,":"),I2,", AT LEVEL ",I*," IN STATEMENT ",
5630000    I*),                                                                %05
5640000 (//"**END FORMAT--FTIME,"//);
```

```
5650000 %********SWITCH FORMAT FTIME*************************************************
5660000 %*******************************************************************************
5670000 %********VARIOUS FORMATS*******************************************************
5680000 FORMAT % NOT GENERALLY USED BY MOST PROGRAMS:
5690000    F80A1 (80A1),
5700000    FCLOSEDR ("**ATTEMPT TO READ FROM CLOSED FILE:"),
5710000    FCLOSEDW ("**ATTEMPT TO WRITE ON CLOSED FILE:"),
5720000    FCRLF ("≤≠←"),
5730000    FENDPROG ("**END OF PROGRAM FILE "),
5740000    FFIXVARSIZE ("**ILLEGAL SIZE ","""",X*,"""," FOR FIXED-LENGTH ",
5750000       "VARIABLE IN STATEMENT ",I6),
5760000    FGC (//"**GARBAGE COLLECTOR ",10("********")),
5770000    FGCRES (/"**GARBAGE COLLECTOR RESULTS:"/
5780000       I5," ROWS COLLECTED"/
5790000       I5," CHARS IN LONGEST ROW"/
5800000       I5," CHARS TOTAL"/),
5810000    FGCS (/"GARBAGE COLLECTOR CALLED ",I*," TIMES; USED ",F*.1,
5820000       " SEC CPU TIME, ",F*.1," SEC I/O TIME."),
5830000    FNOFILE ("**FILE NOT AVAILABLE."),
5840000    FO (O),
5850000    FPARITY ("**IRRECOVERABLE PARITY ERROR ON"),
5860000    FRULES ("**NUMBER OF RULES EXECUTED = ",I8),
5870000    FSENDCOPY ("**SEND COPY OF PROGRAM TO SYSTEM AUTHORS."),
5880000    FTRACE ("*S",I5,X1,X*," = ",""""),
5890000    FTRACEFCTCALL ("*C",I5,X1,X*,"("),
5900000    FTRACEFCTRET ("*R",I5,X1,X*,"() = ",""""),
5910000    FTRACEL ("*L  ",X*,I*,*(" FROM ",I*)),
5920000    FUNDEFFCT ("**UNDEFINED FUNCTION CALLED:"),
5930000    FUNDEFLABEL ("**ATTEMPTED TRANSFER TO UNDEFINED LABEL ");
5940000 FORMAT   % FOR USE WITH CREATELIBRARY AND LOADLIBRARY.
5950000    FLIB0("SNOBOL LIBRARY FILE ",I1),
5960000    FLIB1 (I1,I4,3(L1,0)),
5970000    FLIB2 (A2,16L1,I6),
5980000    FLIB3 (16A2),
5990000    FLIB4 (60),
6000000    FLIB5 (32L1),
6010000    FLIB6 (12I4),
6020000    FLIBOLDLP("**LIBRARY FILE INCOMPATIBLE WITH THIS VERSION OF SNOBOL"),
6030000    FLIBOLDTT("≤≠LIBRARY FILE INCOMPATIBLE WITH THIS VERSION OF SNOBOL",
6040000       "≤≠←");
6050000 FORMAT % USED DURING MOST RUNS:
6060000    FASTERISKS (/1023("********")),
6070000    FBL ((X8)),
6080000    FDBL (*(/)),
6090000    FERRS (I*," SYNTAX ERRORS DETECTED"),
6100000    FI7 (I7),
6110000    FI16 (I16),
6120000    FINT (*D,X*,I*),
6130000    FNUM (X*,I*);
6140000 FORMAT % USED WITH TELETYPES
6150000    FAGAIN ("≤≠TRY AGAIN≤≠←"),
6160000    FDKSEARCH (X*,"/",X*,":",3I1,4(":",I*)),
6170000    FEH ("≤≠EH",A1,"≤≠←"),
6180000    FFROMREMOTE ("≤≠DO YOU WANT TO TYPE A PROGRAM",A1,"≤≠←"),
6190000    FNOINPUT ("≤≠≠WAITING TIME UP"-DO YOU WISH TO CONTINUE",A1,"≤≠←"),
6200000    FSTAT (*("(",2(I*,"/",I*,":"),8I1,")")),
6210000    FTTADDR (I*,"/",I*),
```

```
6220000     FTTHELLO ("≤≠SNOBOL VERSION ",F3,1,"≤≠←"),
6230000     FYESORNO ("≤≠TYPE YES OR NO.≤≠←"),
6240000     FVCRLF (X*,"≤≠←");
6250000 FORMAT    % USED WITH DUMP PROCEDURES
6260000     FSTACKHEAD (/"**THE STACK IS:"/
6270000         "  WORD TYPE ST NAME ....LOC.... SIZE POINT MINLEFT BACK"),
6280000     FSTACKENTRY (I6,I4,X2,A2,X3,L1,X1,I4,X1,I2,X1,I4,I5,I6,I8,I5);
6290000 %*************************************************************************
6300000 % ************************************************************************
6310000 %========== BEGIN DATA COM GLOBAL DECLARATIONS ========================
6320000 DEFINE
6330000     DCINCHAR =28#,
6340000     NBUFIN =4#,
6350000     BUFINSIZE =5#,
6360000     DCOUTCHAR =28#,
6370000     NBUFOUT =5#,
6380000     BUFOUTSIZE =5#,
6390000     MSGSIZE = 250 #,
6400000     ANSSIZE = 250 #,          % MAX # OF CHARS IN OUTPUT STRING.
6410000     TTMAX = 15 #;             % MAX # OF TELETYPES ALLOWED.
6420000 ALPHA ARRAY
6430000     DCREAD[0:(MSGSIZE-1),W],
6440000     DCWRITE[0:(ANSSIZE-1),W],
6450000     ID,                     % I. D. OF USER OF TELETYPE
6460000         STAT[0:TTMAX];      % STATUS OF TELETYPE
6470000 ALPHA FILE IN  DCIN  14 ( NBUFIN, BUFINSIZE);
6480000 ALPHA FILE OUT DCOUT 14 (NBUFOUT,BUFOUTSIZE);
6490000 INTEGER
6500000     MAINUSER,   % TELETYPE CURRENTLY IN CHARGE
6510000     NUMUSERS,   % NUMBER OF TELETYPES ATTACHED.
6520000     USER;       % TELETYPE CURRENTLY BEING TALKED TO
6530000 REAL
6540000     WAITTIME;
6550000 BOOLEAN BREAK;
6560000 %========== END DATA COM GLOBAL DECLARATIONS ==========================
6570000 BEGIN    % GLOBAL STREAM PROCEDURES
6580000 %********ABSADDR*****************************************************
6590000 INTEGER STREAM PROCEDURE ABSADDR(A);
6600000 BEGIN SI ← A; ABSADDR ← SI; END;
6610000 %********ABSADDR*****************************************************
6620000 %************************************************************************
6630000 COMMENT CHAR RETURNS THE PTH CHARACTER OF NAME, RIGHT JUSTIFIED;
6640000 INTEGER STREAM PROCEDURE CHAR(NAME,P);
6650000 VALUE P;
6660000 %
6670000 %
6680000 BEGIN SI ← NAME;
6690000 SI ← SI + P;
6700000 DI ← LOC CHAR; DI ← DI + 7;
6710000 DS ← 1 CHR;
6720000 END;
6730000 %************************************************************************
6740000 %********EQ*********************************************************
6750000 BOOLEAN STREAM PROCEDURE EQ(N,L1,I1,L2,I2);
6760000 VALUE N, I1, I2;
6770000 BEGIN SI ← L1; SI ← SI + I1;
6780000 DI ← L2; DI ← DI + I2;
```

```
6790000 N(IF 1 SC ≠ DC THEN JUMP OUT);
6800000 IF TOGGLE THEN TALLY ← 0 ELSE TALLY ← 1;
6810000 EQ ← TALLY;
6820000 END EQ;
6830000 %********EQ*************************************************************
6840000 %********EQUAL**********************************************************
6850000 % EQUAL RETURNS TRUE IF THE N CHARS STARTING AT THE I1TH CHAR AFTER L1
6860000 % ARE THE SAME AS THE N CHARS STARTING AT THE L2TH CHAR AFTER L2. I1
6870000 % AND I2 MUST BE ≥ 0 AND ≤ 63.
6880000 BOOLEAN STREAM PROCEDURE EQUAL(N,L1,I1,L2,I2);
6890000 VALUE N, I1, I2;
6900000 %
6910000 %
6920000 BEGIN LOCAL NA, NB;
6930000 LABEL L;
6940000 SI ← LOC N; SI ← SI + 5;
6950000 DI ← LOC NA; DI ← DI + 7; DS ← 1 CHR;
6960000 DI ← LOC NB; DI ← DI + 7; DS ← 1 CHR;
6970000 SI ← L1;
6980000 SI ← SI + I1;
6990000 DI ← L2;
7000000 DI ← DI + I2;
7010000 NA(16(32(IF 8 SC ≠ DC THEN JUMP OUT 3 TO L))));
7020000 NB(8(IF 8 SC ≠ DC THEN JUMP OUT 2 TO L));
7030000 N(IF 1 SC ≠ DC THEN JUMP OUT);
7040000 L:  IF TOGGLE THEN TALLY ← 0 ELSE TALLY ← 1;
7050000 EQUAL ← TALLY;
7060000 END;
7070000 %********EQUAL*************************************************************
7080000 BEGIN % MOVE STREAM PROCEDURES
7090000 %********MV***************************************************************
7100000 STREAM PROCEDURE MV(N,L1,I1,L2,I2);
7110000 VALUE N, I1, I2;
7120000 BEGIN SI ← L1; SI ← SI + I1;
7130000 DI ← L2; DI ← DI + I2;
7140000 DS ← N CHR;
7150000 END MV;
7160000 %********MV***************************************************************
7170000 %********MOVE*************************************************************
7180000 STREAM PROCEDURE MOVE(N,SOURCE,I1,DEST,I2);
7190000 %
7200000 VALUE N, I1, I2;
7210000 BEGIN LOCAL NA, NB;
7220000 SI ← LOC N; SI ← SI + 5;
7230000 DI ← LOC NA; DI ← DI + 7; DS ← 1 CHR;
7240000 DI ← LOC NB; DI ← DI + 7; DS ← 1 CHR;
7250000 SI ← SOURCE;
7260000 SI ← SI + I1;
7270000 DI ← DEST;
7280000 DI ← DI + I2;
7290000 NA(4(32(DS ← 32 CHR)));
7300000 NB(2(DS ← 32 CHR));
7310000 DS ← N CHR;
7320000 END MOVE;
7330000 %********MOVE*************************************************************
7340000 %********MOVEWDS**********************************************************
7350000 STREAM PROCEDURE MOVEWDS(N,L1,L2); VALUE N;
```

```
7360000 BEGIN LOCAL NA;
7370000 SI ← LOC N; SI ← SI + 6;
7380000 DI ← LOC NA; DI ← DI + 7; DS ← 1 CHR;
7390000 SI ← L1; DI ← L2;
7400000 NA(2(DS ← 32 WDS)); N(DS ← 1 WDS);
7410000 END MOVEWDS;
7420000 %********MOVEWDS*************************************************************
7430000 %********MOVEWORD************************************************************
7440000 STREAM PROCEDURE MOVEWORD(A,B); VALUE A, B;
7450000 BEGIN SI ← A; DI ← B; DS ← 1 WDS; END;
7460000 %********MOVEWORD************************************************************
7470000 BEGIN % SCAN STREAM PROCEDURES
7480000 %********SCANCHAR************************************************************
7490000 % SCANCHAR RETURNS THE NUMBER OF CHARS FROM THE ITH CHAR AFTER L
7500000 % (0 ≤ I ≤ 63) TO THE FIRST OCCURRENCE OF EITHER C1 OR C2.
7510000 INTEGER STREAM PROCEDURE SCANCHAR(C1,C2,L,I); VALUE C1,C2,I;
7520000 BEGIN LOCAL P1, P2, P3, TEMP;
7530000 LABEL CHERCHE,TROUVE;
7540000 SI ← L; SI ← SI + I;
7550000 DI ← LOC C1; DI ← DI + 7;
7560000 CHERCHE: IF 1 SC = DC THEN GO TO TROUVE;
7570000    TEMP ← DI; DI ← LOC C2;
7580000    DI ← DI + 7; SI ← SI - 1;
7590000    IF 1 SC = DC THEN GO TO TROUVE;
7600000    DI ← TEMP;
7610000    DI ← DI - 1;
7620000    TALLY ← TALLY + 1;
7630000 TEMP ← SI;
7640000 P3 ← TALLY; SI ← LOC P3; SI ← SI + 7;
7650000 IF SC ≠ "0" THEN
7660000    BEGIN SI ← TEMP; GO TO CHERCHE; END;
7670000 TALLY ← P2; TALLY ← TALLY + 1;
7680000 P2 ← TALLY; SI ← LOC P2; SI ← SI + 7;
7690000 IF SC ≠ "0" THEN
7700000    BEGIN TALLY ← 0;
7710000    SI ← TEMP; GO TO CHERCHE;
7720000    END;
7730000 TALLY ← P1; TALLY ← TALLY + 1;
7740000 P1 ← TALLY; TALLY ← 0;
7750000 SI ← TEMP;
7760000 GO TO CHERCHE;
7770000 TROUVE:
7780000    DI ← LOC SCANCHAR; DI ← DI + 5;
7790000    SI ← LOC P1; SI ← SI + 7; DS ← 1 CHR;
7800000    SI ← LOC P2; SI ← SI + 7; DS ← 1 CHR;
7810000    SI ← LOC P3; SI ← SI + 7; DS ← 1 CHR;
7820000 END SCANCHAR;
7830000 %********SCANCHAR************************************************************
7840000 %********SKIPCHAR************************************************************
7850000 INTEGER STREAM PROCEDURE SKIPCHAR(C,L,I); VALUE C,I;
7860000 %
7870000 BEGIN LOCAL P1, P2, P3, TEMP;
7880000 LABEL CHERCHE,TROUVE;
7890000 SI ← L; SI ← SI + I;
7900000 DI ← LOC C; DI ← DI + 7;
7910000 CHERCHE: IF 1 SC ≠ DC THEN GO TO TROUVE;
7920000    DI ← DI - 1;
```

```
7930000     TALLY ← TALLY + 1;
7940000 TEMP ← SI;
7950000 P3 ← TALLY; SI ← LOC P3; SI ← SI + 7;
7960000 IF SC ≠ "0" THEN
7970000     BEGIN SI ← TEMP; GO TO CHERCHE; END;
7980000 TALLY ← P2; TALLY ← TALLY + 1;
7990000 P2 ← TALLY; SI ← LOC P2; SI ← SI + 7;
8000000 IF SC ≠ "0" THEN
8010000     BEGIN TALLY ← 0;
8020000     SI ← TEMP; GO TO CHERCHE;
8030000     END;
8040000 TALLY ← P1; TALLY ← TALLY + 1;
8050000 P1 ← TALLY; TALLY ← 0;
8060000 SI ← TEMP;
8070000 GO TO CHERCHE;
8080000 TROUVE:
8090000     DI ← LOC SKIPCHAR; DI ← DI + 5;
8100000     SI ← LOC P1; SI ← SI + 7; DS ← 1 CHR;
8110000     SI ← LOC P2; SI ← SI + 7; DS ← 1 CHR;
8120000     SI ← LOC P3; SI ← SI + 7; DS ← 1 CHR;
8130000 END SKIPCHAR;
8140000 %********SKIPCHAR*************************************************************
8150000 %*******************************************************************************
8160000 %  CARDTYPE RETURNS A NUMBER IDENTIFYING THE TYPE OF INSTRUCTION
8170000 %  BEING COMPILED--THE TYPES ARE:
8180000 %       1  COMMENT
8190000 %       2  CONTINUATION
8200000 %       3  CONTROL CARD
8210000 %       4  END
8220000 %       5  UNLABELED
8230000 %       6  LABELED
8240000 INTEGER STREAM PROCEDURE CARDTYPE(CARD);
8250000 BEGIN SI ← CARD;
8260000 IF SC = "*" THEN TALLY ← 1 ELSE
8270000 IF SC = "%" THEN TALLY ← 1 ELSE
8280000 IF SC = "." THEN TALLY ← 2 ELSE
8290000 IF SC = "=" THEN TALLY ← 3 ELSE
8300000 IF SC = "E" THEN
8310000     BEGIN SI ← SI + 1;
8320000     IF SC = "N" THEN
8330000         BEGIN SI ← SI + 1;
8340000         IF SC = "D" THEN
8350000             BEGIN SI ← SI + 1;
8360000             IF SC = " " THEN TALLY ← 4 ELSE TALLY ← 6;
8370000             END ELSE TALLY ← 6;
8380000         END ELSE TALLY ← 6;
8390000     END ELSE
8400000 IF SC = " " THEN TALLY ← 5 ELSE TALLY ← 6;
8410000 CARDTYPE ← TALLY;
8420000 END TYPE;
8430000 %********CARDTYPE************************************************************
8440000 BEGIN     % ALGOL PROCEDURES
8450000 %********FORWARD DECLARATIONS*************************************************
8460000 PROCEDURE CLEAR(AR,N);
8470000     VALUE N; INTEGER N;
8480000     ALPHA ARRAY AR[0];
8490000     FORWARD;
```

```
8500000 BOOLEAN PROCEDURE COMBINEARITHMETIC(I);
8510000     VALUE I;
8520000     INTEGER I;
8530000     FORWARD;
8540000 PROCEDURE COMBINEFUNCTION(MKS);
8550000     VALUE MKS;
8560000     INTEGER MKS;
8570000     FORWARD;
8580000 BOOLEAN PROCEDURE COMBINEGOTOPART(MKS);
8590000     VALUE MKS;
8600000     INTEGER MKS;
8610000     FORWARD;
8620000 BOOLEAN PROCEDURE COMBINEGROUP(MKS,CON);
8630000     VALUE MKS, CON;
8640000     INTEGER MKS;
8650000     BOOLEAN CON;
8660000     FORWARD;
8670000 BOOLEAN PROCEDURE COMBINEINDIRECTION(MKS);
8680000     VALUE MKS;
8690000     INTEGER MKS;
8700000     FORWARD;
8710000 BOOLEAN PROCEDURE COMBINESTRVARNAME(MKS);
8720000     VALUE MKS;
8730000     INTEGER MKS;
8740000     FORWARD;
8750000 PROCEDURE COMPILE(INST);
8760000     ALPHA ARRAY INST[0];
8770000     FORWARD;
8780000 INTEGER PROCEDURE CONTROLPARAMETER(A,P);
8790000     ALPHA ARRAY A[0];
8800000     INTEGER P;
8810000     FORWARD;
8820000 PROCEDURE CONVERT(A);
8830000     ALPHA ARRAY A[0];
8840000     FORWARD;
8850000 ALPHA PROCEDURE DATE;
8860000     FORWARD;
8870000 PROCEDURE DEBUG(L);
8880000     VALUE L;
8890000     LABEL L;
8900000     FORWARD;
8910000 INTEGER PROCEDURE DIGITS(N);
8920000     VALUE N;
8930000     REAL N;
8940000     FORWARD;
8950000 ALPHA PROCEDURE ENTERST(N,L,P,X);
8960000     VALUE N, P, X;
8970000     INTEGER N, P;
8980000     ALPHA ARRAY L[0];
8990000     ALPHA X;
9000000     FORWARD;
9010000 PROCEDURE FINDUSERS;
9020000     FORWARD;
9030000 PROCEDURE GARBAGECOLLECTOR;
9040000     FORWARD;
9050000 PROCEDURE INDIRECT(SP);
9060000     VALUE SP;
```

```
9070000     INTEGER SP;
9080000     FORWARD;
9090000 PROCEDURE INFORMO(I);
9100000     VALUE I;
9110000     INTEGER I;
9120000     FORWARD;
9130000 PROCEDURE INFORMA(I,P);
9140000     VALUE I, P;
9150000     INTEGER I; ALPHA P;
9160000     FORWARD;
9170000 PROCEDURE INFORMI(I,P);
9180000     VALUE I, P;
9190000     INTEGER I, P;
9200000     FORWARD;
9210000 PROCEDURE INFORMIA(I,P1,P2);
9220000     VALUE I, P1, P2;
9230000     INTEGER I, P1;
9240000     ALPHA P2;
9250000     FORWARD;
9260000 PROCEDURE INFORMII(I,P1,P2);
9270000     VALUE I, P1, P2;
9280000     INTEGER I, P1, P2;
9290000     FORWARD;
9300000 PROCEDURE INITIALIZESYMBTABL;
9310000     FORWARD;
9320000 BOOLEAN PROCEDURE INPUT;
9330000     FORWARD;
9340000 PROCEDURE INSERTSTRINGCONVERT(SP);
9350000     VALUE SP;
9360000     INTEGER SP;
9370000     FORWARD;
9380000 PROCEDURE INTERPRETER;
9390000     FORWARD;
9400000 INTEGER PROCEDURE INTRINSIC(L,P,S);
9410000     VALUE P, S;
9420000     ALPHA ARRAY L[0];
9430000     INTEGER P, S;
9440000     FORWARD;
9450000 PROCEDURE LOADER(F);
9460000     FILE F;
9470000     FORWARD;
9480000 PROCEDURE LST(A);
9490000     ALPHA ARRAY A[0];
9500000     FORWARD;
9510000 PROCEDURE MESSAGEO(I);
9520000     VALUE I;
9530000     INTEGER I;
9540000     FORWARD;
9550000 PROCEDURE MESSAGEAI(I,P1,P2);
9560000     VALUE I, P1, P2;
9570000     INTEGER I, P2;
9580000     ALPHA P1;
9590000     FORWARD;
9600000 PROCEDURE MESSAGEI(I,P);
9610000     VALUE I, P;
9620000     INTEGER I, P;
9630000     FORWARD;
```

```
9640000 PROCEDURE MESSAGETTO(I);
9650000    VALUE I;
9660000    INTEGER I;
9670000    FORWARD;
9680000 PROCEDURE MESSAGETTA(I,P1);
9690000    VALUE I, P1;
9700000    INTEGER I;
9710000    ALPHA P1;
9720000    FORWARD;
9730000 PROCEDURE MESSAGETTAI(I,P1,P2);
9740000    VALUE I, P1, P2;
9750000    INTEGER I, P2;
9760000    ALPHA P1;
9770000    FORWARD;
9780000 PROCEDURE MESSAGETTI(I,P);
9790000    VALUE I, P;
9800000    INTEGER I, P;
9810000    FORWARD;
9820000 INTEGER PROCEDURE MNEMNO(N,A,P);
9830000    VALUE N;
9840000    INTEGER N, P;
9850000    ARRAY A[0];
9860000    FORWARD;
9870000 ALPHA PROCEDURE NEWCELL(I);
9880000    VALUE I;
9890000    INTEGER I;
9900000    FORWARD;
9910000 PROCEDURE NEWSTROW(I);
9920000    VALUE I;
9930000    INTEGER I;
9940000    FORWARD;
9950000 BOOLEAN PROCEDURE NULLARGS(N);
9960000    VALUE N;
9970000    INTEGER N;
9980000    FORWARD;
9990000 BOOLEAN PROCEDURE NUMVAL(ST,VAL);
10000000    VALUE ST;
10010000    ALPHA ST;
10020000    INTEGER VAL;
10030000    FORWARD;
10040000 BOOLEAN PROCEDURE OUTPUT;
10050000    FORWARD;
10060000 BOOLEAN PROCEDURE POP(STLOC);
10070000    VALUE STLOC;
10080000    ALPHA STLOC;
10090000    FORWARD;
10100000 PROCEDURE PROCESSCONTROLCARD(A);
10110000    ALPHA ARRAY A[0];
10120000    FORWARD;
10130000 BOOLEAN PROCEDURE PUSH(S,T);
10140000    VALUE S, T; ALPHA S, T;
10150000    FORWARD;
10160000 PROCEDURE RETURNCELL(L);
10170000    VALUE L; ALPHA L;
10180000    FORWARD;
10190000 PROCEDURE RETURNTEMPS;
10200000    FORWARD;
```

```
10210000 ALPHA PROCEDURE SCATTER(SIZE,LOC,P,X);
10220000    VALUE SIZE, P, X;
10230000    INTEGER SIZE, P;
10240000    ALPHA ARRAY LOC[0];
10250000    ALPHA X;
10260000    FORWARD;
10270000 ALPHA PROCEDURE SEARCHST(N,L,P,X);
10280000    VALUE N, P, X;
10290000    INTEGER N, P;
10300000    ALPHA ARRAY L[0];
10310000    ALPHA X;
10320000    FORWARD;
10330000 BOOLEAN PROCEDURE SNBLDEFINE(ST1,ST2,ST3);
10340000    VALUE ST1, ST2, ST3;
10350000    ALPHA ST1, ST2, ST3;
10360000    FORWARD;
10370000 BOOLEAN PROCEDURE SNBLIN(ST);
10380000    VALUE ST;
10390000    ALPHA ST;
10400000    FORWARD;
10410000 BOOLEAN PROCEDURE SNBLOUT(ST);
10420000    VALUE ST;
10430000    ALPHA ST;
10440000    FORWARD;
10450000 PROCEDURE STORECHARS(N,L,I);
10460000    VALUE N, I;
10470000    INTEGER N, I;
10480000    ALPHA L;
10490000    FORWARD;
10500000 ALPHA PROCEDURE STRING(N,STLOC);
10510000    VALUE N, STLOC;
10520000    INTEGER N; ALPHA STLOC;
10530000    FORWARD;
10540000 PROCEDURE STRINGDUMP(N);
10550000    VALUE N;
10560000    INTEGER N;
10570000    FORWARD;
10580000 PROCEDURE SYNTAXERR(N,P);
10590000    VALUE N, P;
10600000    INTEGER N, P;
10610000    FORWARD;
10620000 ALPHA PROCEDURE TEMPCELL;
10630000    FORWARD;
10640000 ALPHA PROCEDURE TEMPVAL(I);
10650000    VALUE I;
10660000    INTEGER I;
10670000    FORWARD;
10680000 BOOLEAN PROCEDURE TRACEFCTCALL(F);
10690000    VALUE F;
10700000    INTEGER F;
10710000    FORWARD;
10720000 BOOLEAN PROCEDURE TRACEFCTRETURN(F,SUC);
10730000    VALUE F, SUC;
10740000    INTEGER F;
10750000    BOOLEAN SUC;
10760000    FORWARD;
10770000 INTEGER PROCEDURE TTINDEX(ST);
```

```
10780000     VALUE ST;
10790000     ALPHA ST;
10800000     FORWARD;
10810000 BOOLEAN PROCEDURE WRITEBUFF(BUFOUT,P,SIZE);
10820000     VALUE SIZE;
10830000     INTEGER P, SIZE;
10840000     ARRAY BUFOUT[0];
10850000     FORWARD;
10860000 PROCEDURE WRITEDATA;
10870000     FORWARD;
10880000 PROCEDURE WRITEINST;
10890000     FORWARD;
10900000 PROCEDURE WRITEST;
10910000     FORWARD;
10920000 PROCEDURE WRITESTACK(N);
10930000     VALUE N;
10940000     INTEGER N;
10950000     FORWARD;
10960000 PROCEDURE WRITETIME(F,N);
10970000     VALUE F, N;
10980000     INTEGER F, N;
10990000     FORWARD;
11000000 LABEL
11010000     ABORTION,        % FATAL SYSTEM ERROR
11020000     ENDOFRUN,        % FINAL DUMPS, IF ANY
11030000     ENDTERPRET;      % TRANSFERRED TO AT END OR FATAL ERROR
11040000 %********CHECKOUTUSE*************************************************
11050000 BOOLEAN PROCEDURE CHECKOUTUSE(ST);
11060000 VALUE ST; ALPHA ST;
11070000 CHECKOUTUSE ← IF BOOLEAN(VALTABL[ST.STR,ST.STW].OUTUSE)
11080000     THEN SNBLOUT(ST) ELSE TRUE;
11090000 %********CHECKOUTUSE*************************************************
11100000 %********CHECKSYMBTABL***********************************************
11110000 %  CHECKSYMBTABL LOOKS FOR SYMBOL TABLE ENTRIES WHICH POINT TO STRINGS
11120000 %  NOT PRECEDED BY A POINTER BACK TO THE SAME SYMBTABL LOCATION.  IF
11130000 %  ANY SUCH ENTRIES ARE FOUND, AN ERROR MESSAGE IS PRINTED AND
11140000 %  THE JOB IS TERMINATED ABNORMALLY (WITH A COMPLETE DUMP).
11150000 PROCEDURE CHECKSYMBTABL;
11160000 BEGIN INTEGER I, J, K, RW, CHR, P;
11170000 BOOLEAN URK;
11180000 ALPHA AA;
11190000 FORMAT FF(/"**SYMBTABL LOCATION ",A2," IS NOT POINTED TO BY ITS STRING"
11200000     " AT DATA[",I2,",*] + ",I4," CHARS."),
11210000     FPUSH (//"**PUSH-DOWN STACK INTO AVAILABLE SPACE LIST AT ",A2),
11220000     FG (/"**SYMBTABL LOCATION ZERO HAS BEEN ALTERED.");
11230000 URK ← FALSE;
11240000 IF NAMTABL[0,0]≠0 OR VALTABL[0,0]≠0 THEN
11250000     BEGIN WRITE(PRINT,FG);
11260000     URK ← TRUE;
11270000     END;
11280000 FOR I ← 0 STEP 1 WHILE I ≤ STRMAX AND USEDST[I] DO
11290000     BEGIN J ← NEXTCELL[I],STW;
11300000     IF J ≠ 0 THEN DO
11310000         BEGIN NAMTABL [I,J].[3:1] ← 1;
11320000         J ← (NAMTABL[I,J].LINK),STW;
11330000         END UNTIL J = 0 OR J ≥ STWMAX;
11340000     FOR J ← 0 STEP 1 UNTIL STWMAX DO
```

```
11350000        BEGIN
11360000        FOR AA ← NAMTABL[I,J],VALTABL[I,J] DO IF AA.CH > 3 THEN
11370000           % CHECK STRINGS FOR BACK-POINTERS
11380000           BEGIN RW ← TEMP.R;
11390000           CHR ← TEMP.CH - 2;
11400000           MV(2,DATA[RW,CHR,W],CHR,C,P,6);
11410000           IF I ≠ P.STR OR J ≠ P.STW THEN
11420000              BEGIN WRITE(PRINT,FF,J&I CONCSTR,RW,CHR+2);
11430000              URK ← TRUE;
11440000              END;
11450000           END;
11460000        IF NAMTABL[I,J].[1:2]=0 THEN % STRING--CHECK PUSH-DOWN STACK
11470000           BEGIN
11480000        AA ← VALTABL[I,J].LINK;
11490000        WHILE AA ≠ 0 DO
11500000           BEGIN IF BOOLEAN(NAME[AA].[3:1]) THEN
11510000              BEGIN WRITE(PRINT,FPUSH,AA);
11520000              URK ← TRUE;
11530000              END;
11540000           AA ← VALU[AA].LINK;
11550000           END;
11560000           END;
11570000        END;
11580000     END;
11590000 IF URK THEN ABORT;
11600000 END CHECKSYMBTABL;
11610000 %********CHECKSYMBTABL*************************************************
11620000 %********CLEAR*********************************************************
11630000 PROCEDURE CLEAR(AR,N); VALUE N; ALPHA ARRAY AR[0]; INTEGER N;
11640000 FOR N ← N - 1 WHILE N ≥ 0 DO MOVE(8,BLANKS,0,AR[N],0);
11650000 %********CLEAR*********************************************************
11660000 %********COMBINEARITHMETIC*********************************************
11670000 BOOLEAN PROCEDURE COMBINEARITHMETIC(I);
11680000     VALUE I;
11690000     INTEGER I;
11700000 BEGIN
11710000 LABEL FAUT, FIN, UNARY;
11720000 INTEGER J, K, LEVOP, N;
11730000 % IF INFORM THEN INFORMO(20);
11740000 COMBINEARITHMETIC ← FALSE;
11750000 IF I+1 ≥ SP THEN
11760000     BEGIN SYNTAXERR(29,PLOC[I]);
11770000     GO TO FIN;
11780000     END;
11790000 IF I + 2 < SP THEN IF PTYPE[I+2] = 7 THEN
11800000     IF OPLEVEL[PST[I+2]] > LEVOP←OPLEVEL[PST[I]] THEN
11810000        IF NOT COMBINEARITHMETIC(I+2) THEN GO TO FIN;
11820000 J ← I;
11830000 DO J ← J-1 UNTIL IF J<PBACK[I] THEN TRUE ELSE PTYPE[J]≠0;
11840000 IF J < PBACK[I] THEN GO TO UNARY;
11850000 FOR K ← J, I+1 DO
11860000     CASE PTYPE[K] OF
11870000        BEGIN GO TO FAUT;
11880000     % 1: STRING NAME
11890000     ;
11900000     % 2: FUNCTION NAME:
11910000        GO TO IF K=J THEN UNARY ELSE FAUT;
```

```
11920000     % 3: LITERAL
11930000        ;
11940000     % 4: "(" OF GROUP
11950000        GO TO IF K=J THEN UNARY ELSE FAUT;
11960000     % 5: STRING VARIABLE
11970000        BEGIN SYNTAXERR(16,PLOC[K]);
11980000        GO TO FIN;
11990000        END;
12000000     % 6: INDIRECTION, UNCOMBINED
12010000        IF NOT COMBINEINDIRECTION(K) THEN GO TO FIN;
12020000     % 7: CONSECUTIVE ARITHMETIC OPERATORS
12030000        BEGIN SYNTAXERR(16,PLOC[K]);
12040000        GO TO FIN;
12050000        END;
12060000     % 8: "/" OF FIXED-LENGTH VAR
12070000        GO TO IF K=J THEN UNARY ELSE FAUT;
12080000     % 9: STR REF, PATTERN & REPLACEMENT
12090000        GO TO FAUT;
12100000     % 10: INDIRECTION COMBINED
12110000        ;
12120000     % 11: NOT IN USE
12130000        GO TO FAUT;
12140000     % 12: NOT IN USE
12150000        GO TO FAUT;
12160000     % 13: "*" OF STR VAR
12170000        BEGIN SYNTAXERR(16,PLOC[K]);
12180000        GO TO FIN;
12190000        END;
12200000     % 14: LABEL IN GO-TO PART
12210000        GO TO FAUT;
12220000     % 15: EXPRESSION
12230000        ;
12240000     % 16: ARITH EXPR
12250000        ;
12260000     % 17: "*(" OF BAL VAR
12270000        GO TO FAUT;
12280000     % 18: STR REF & PATTERN
12290000        GO TO FAUT;
12300000     % 19: COMPLETE ARITH EXPR--REMOVE STRING CONVERT
12310000        PSIZE[K] ← PSIZE[K] - 2;
12320000     % 20: "S(", "F(", OR "(" OF GO-TO PART
12330000        GO TO FAUT;
12340000     % 21: COMBINED GO-TO PART
12350000        GO TO FAUT;
12360000        END CASES;
12370000 N ← 0;
12380000 FOR K ← J, I+1, I DO
12390000    BEGIN MOVE(PSIZE[K],CODE[PPOINT[K],W],PPOINT[K].C,SCRATCH[N,W],N.C);
12400000    N ← N + PSIZE[K];
12410000    PTYPE[K] ← 0;
12420000    END;
12430000 MOVE(N,SCRATCH[*],0,CODE[PPOINT[J],W],PPOINT[J],C);
12440000 PSIZE[J] ← N;
12450000 PTYPE[J] ← 16; % INCOMPLETE ARITH EXPR
12460000 COMBINEARITHMETIC ← TRUE;
12470000 GO TO FIN;
12480000 FAUT:
```

```
12490000     WRITEINST;
12500000     SYNTAXERR(40,0);
12510000     WRITE(PRINT,17,CODE[*]);
12520000     WRITESTACK(SP);
12530000     SYSTEMERROR ← TRUE;
12540000     GO TO FIN;
12550000 UNARY:
12560000     IF PST[I] = "-" THEN
12570000        BEGIN MOVE(K←PSIZE[I+1],CODE[(J←PPOINT[I+1]),W],J,C,SCRATCH[*],0);
12580000        AA ← ",N";
12590000        MV(2,AA,6,SCRATCH[K,W],K,C);
12600000        MOVE(K+2,SCRATCH[*],0,CODE[(J←PPOINT[I]),W],J,C);
12610000        PTYPE[I+1] ← 0;
12620000        PTYPE[I] ← 16; % ARITH EXPR--NO STRING CONVERT
12630000        PSIZE[I] ← K+2;
12640000        COMBINEARITHMETIC ← TRUE;
12650000        GO TO FIN;
12660000        END ELSE
12670000     % CANT BE USED AS UNARY:
12680000        BEGIN SYNTAXERR(11,PLOC[I]);
12690000        GO TO FIN;
12700000        END;
12710000 FIN:
12720000     IF SYSTEMERROR THEN INFORM0(20);
12730000     % IF INFORM THEN WRITEINST;
12740000 END COMBINEARITHMETIC;
12750000 %********COMBINEARITHMETIC***************************************************
12760000 %********COMBINEFUNCTION*****************************************************
12770000 PROCEDURE COMBINEFUNCTION(MKS);
12780000     VALUE MKS;
12790000     INTEGER MKS;
12800000 BEGIN INTEGER I, J, N;
12810000 % IF INFORM THEN INFORM0(25);
12820000 N ← J ← 0;
12830000 FOR I ← MKS+1 STEP 1 UNTIL SP-1 DO IF PTYPE[I] ≠ 0 THEN
12840000     BEGIN MOVE(PSIZE[I],CODE[PPOINT[I],W],PPOINT[I],C,SCRATCH[J,W],J,C);
12850000     I ← I;
12860000     J ← J + PSIZE[I];
12870000     PTYPE[I] ← 0;    % WIPE OUT ELEMENT
12880000     N ← N + 1;
12890000     END;
12900000 MV(3,CODE[(I←PPOINT[MKS]),W],I,C,SCRATCH[J,W],J,C);
12910000 MV(1,N,7,SCRATCH[J,W],J,C+3);
12920000 MOVE(J+4,SCRATCH[*],0,CODE[I,W],I,C);
12930000 PTYPE[MKS] ← 15;    % EXPRESSION
12940000 PNAME[MKS] ← FALSE;
12950000 PTR ← I + (PSIZE[MKS]←J+4);
12960000 % IF INFORM THEN WRITEINST;
12970000 END COMBINEFUNCTION;
12980000 %********COMBINEFUNCTION*****************************************************
12990000 %********COMBINEGOTOPART*****************************************************
13000000 BOOLEAN PROCEDURE COMBINEGOTOPART(MKS);
13010000     VALUE MKS;
13020000     INTEGER MKS;
13030000 BEGIN
13040000 LABEL FAIL, FAUT, FIN;
13050000 INTEGER I, M;
```

```
13060000 % IF INFORM THEN INFORMO(24);
13070000 IF PTYPE[MKS] ≠ 20 THEN GO TO FAUT;
13080000 M ← MKS + 1;
13090000 IF PTYPE[M] = 6 THEN % INDIRECTION
13100000    BEGIN
13110000    % GET NUMBER OF INDIRECTIONS:
13120000    I ← CHAR(CODE[PPOINT[M],W],PPOINT[M],C+1);
13130000    IF I > 1 THEN
13140000       BEGIN I ← I - 1;
13150000       MV(1,I,7,CODE[PPOINT[M],W],PPOINT[M],C+1); % DECREASE BY 1
13160000       IF NOT COMBINEINDIRECTION(M) THEN GO TO FAIL;
13170000       END ELSE     % MOVE BACK TO M;
13180000       BEGIN PTYPE[M+1] ← 0;
13190000       PPOINT[M] ← PPOINT[M+1];
13200000       PLOC[M] ← PLOC[M+1];
13210000       PSIZE[M] ← PSIZE[M+1];
13220000       END;
13230000    PTR ← PPOINT[M] + PSIZE[M];
13240000    STORECHARS(1,"L",7); % LABEL INDIRECTION CODE
13250000    PSIZE[M] ← PSIZE[M] + 1;
13260000    PTYPE[M] ← 14;  % LABEL EXPRESSION
13270000    END;
13280000 IF PTYPE[M] ≠ 14 THEN
13290000    BEGIN SYNTAXERR(8,PLOC[M]); % ILLEGAL LABEL
13300000    GO TO FAIL;
13310000    END;
13320000 FOR I ← M+1 STEP 1 UNTIL SP-1 DO IF PTYPE[I] ≠ 0 THEN
13330000    BEGIN SYNTAXERR(8,PLOC[I]); % ILLEGAL LABEL
13340000    GO TO FAIL;
13350000    END;
13360000 PTR ← PPOINT[M] + PSIZE[M];
13370000    PSIZE[MKS] ← PSIZE[M];
13380000 PTYPE[MKS] ← 21;   % GO-TO PART COMBINED
13390000 PPOINT[MKS] ← PPOINT[M];
13400000 PLOC[MKS] ← PLOC[M];
13410000 COMBINEGOTOPART ← TRUE;
13420000 GO TO FIN;
13430000 %
13440000 FAUT:
13450000    WRITEINST;
13460000    SYNTAXERR(40,0);
13470000    WRITESTACK(SP);
13480000    SYSTEMERROR ← TRUE;
13490000 FAIL:
13500000    IF SYSTEMERROR THEN INFORMO(24);
13510000    COMBINEGOTOPART ← FALSE;
13520000 FIN:
13530000    % IF INFORM THEN WRITEINST;
13540000 END COMBINEGOTOPART;
13550000 %********COMBINEGOTOPART**************************************************
13560000 %********COMBINEGROUP*****************************************************
13570000 BOOLEAN PROCEDURE COMBINEGROUP(MKS,CON);
13580000    VALUE MKS, CON;
13590000    INTEGER MKS;
13600000    BOOLEAN CON;
13610000 BEGIN
13620000 LABEL FAUT, FAIL, FIN;
```

```
13630000 BOOLEAN ARITH;     % FOR DELETING PARENS AROUND ARITH EXPRS
13640000 INTEGER ERRNO, ERRPOINT, I, J, K, N;
13650000 %
13660000 % IF INFORM THEN INFORMO(21);
13670000 COMBINEGROUP ← ARITH ← FALSE;
13680000 FOR I ← MKS STEP 1 UNTIL SP-1 DO IF PTYPE[I] ≠ 0 THEN
13690000    CASE PTYPE[I] OF
13700000       BEGIN;
13710000    % 1: STRING NAME
13720000       ;
13730000    % 2: FUNCTION (INCOMPLETE)
13740000       GO TO FAUT;
13750000    % 3: LITERAL
13760000       ;
13770000    % 4: START OF GROUP
13780000       GO TO FAUT;
13790000    % 5: NOT USED
13800000       GO TO FAUT;
13810000    % 6: $N--INDIRECTION, UNCOMBINED
13820000       IF NOT COMBINEINDIRECTION(I) THEN GO TO FIN;
13830000    % 7: ARITH OPERATOR:
13840000       BEGIN
13850000       IF NOT COMBINEARITHMETIC(I) THEN GO TO FIN;
13860000       WHILE I > 0 AND PTYPE[I]≠16 DO I ← I - 1;  % FIND RESULT
13870000       IF PTYPE[I] ≠ 16 THEN GO TO FAUT;
13880000       INSERTSTRINGCONVERT(I);
13890000       ARITH ← TRUE;
13900000       END;
13910000    % 8: FIXED-LENGTH "/"
13920000       GO TO FAUT;
13930000    % 9: STR REF, PATTERN & REPLACEMENT
13940000       GO TO FAUT;
13950000    % 10: INDIRECTION COMBINED
13960000       ;
13970000    % 11: NOT USED
13980000       GO TO FAUT;
13990000    % 12: QMARK--ERROR
14000000       GO TO FAUT;
14010000    % 13: "*"  STR VAR--ERROR
14020000       GO TO FAUT;
14030000    % 14: LABEL IN GO-TO PART
14040000       GO TO FAUT;
14050000    % 15: EXPRESSION
14060000       ;
14070000    % 16: ARITH EXPR--NO STRING CONVERT
14080000       BEGIN
14090000       INSERTSTRINGCONVERT(I);
14100000       ARITH ← TRUE;     % MIGHT BE ARITH EXPR
14110000       END;
14120000    % 17: "*(" OF BAL VAR
14130000       GO TO FAUT;
14140000    % 18: STR REF & PATTERN
14150000       GO TO FAUT;
14160000    % 19: COMPLETE ARITH EXPR
14170000       ARITH ← TRUE;
14180000    % 20: "S(", "F(", OR "(" OF GO-TO PART
14190000       GO TO FAUT;
```

```
14200000     % 21: COMBINED GO-TO PART
14210000        GO TO FAUT;
14220000        END CASES;
14230000 N ← J ← 0;
14240000 FOR I ← MKS STEP 1 UNTIL SP-1 DO IF PTYPE[I] ≠ 0 THEN
14250000     BEGIN
14260000     N ← N + 1;
14270000     MOVE(PSIZE[I],CODE[PPOINT[I].W],PPOINT[I].C,SCRATCH[J.W],J.C);
14280000     IF N ≤ 1 AND NOT CON THEN
14290000        BEGIN PNAME[MKS] ← PNAME[I];
14300000        PTYPE[MKS] ← PTYPE[I];
14310000        END ELSE PTYPE[I] ← 0;
14320000     J ← J + PSIZE[I];
14330000     END;
14340000 IF J = 0 THEN
14350000     BEGIN AA ← "%00";
14360000     MV(J←3,AA,5,SCRATCH[*],0);
14370000     PTYPE[MKS] ← 1; % STRING NAME
14380000     PNAME[MKS] ← TRUE;
14390000     END;
14400000 IF N > 1 THEN ARITH ← FALSE; % NOT A SIMPLE ARITH EXPR.
14410000 IF ARITH THEN
14420000     BEGIN PTYPE[MKS] ← 19; % COMPLETE ARITH EXPR
14430000     PNAME[MKS] ← FALSE;
14440000     END ELSE
14450000 IF N > 1 OR CON THEN
14460000     BEGIN AA ← N & "("[36:42:6];
14470000     MV(2,AA,6,SCRATCH[J.W],J.C);
14480000     J ← J + 2;
14490000     PNAME[MKS] ← FALSE;
14500000     PTYPE[MKS] ← 15; % EXPRESSION
14510000     END;
14520000 MOVE(J,SCRATCH[*],0,CODE[(I←PPOINT[MKS]).W],I.C);
14530000 PTR ← I + (PSIZE[MKS]←J);
14540000 COMBINEGROUP ← TRUE;
14550000 GO TO FIN;
14560000 %
14570000 FAUT:
14580000     WRITEINST;
14590000     SYNTAXERR(40,0);
14600000     WRITE(PRINT,17,CODE[*]);
14610000     WRITESTACK(SP);
14620000     SYSTEMERROR ← TRUE;
14630000     GO TO FIN;
14640000 FAIL:
14650000     SYNTAXERR(ERRNO,ERRPOINT);
14660000 FIN:
14670000     IF SYSTEMERROR THEN INFORM0(21);
14680000     % IF INFORM THEN WRITEINST;
14690000 END COMBINEGROUP;
14700000 %********COMBINEGROUP***********************************************
14710000 %********COMBINEINDIRECTION*****************************************
14720000 BOOLEAN PROCEDURE COMBINEINDIRECTION(MKS);
14730000     VALUE MKS;
14740000     INTEGER MKS;
14750000 BEGIN INTEGER I, ERRNO, J, K;
14760000 LABEL FIN, FAIL, FAUT;
```

```
14770000 %
14780000 % IF INFORM THEN INFORMO(22);
14790000 FOR I ← MKS STEP 1 UNTIL SP-2 DO IF PTYPE[I] = 6 THEN
14800000     BEGIN
14810000     CASE PTYPE[I+1] OF  % CHECK FOR SYNTAX ERRORS
14820000        BEGIN GO TO FAUT;
14830000     % 1: STRING NAME
14840000        BEGIN J ← "%";    % TO MARK AS NON-INPUT STRING NAME
14850000        MV(1,J,7,CODE[PPOINT[I+1].W],PPOINT[I+1].C);
14860000        END;
14870000     % 2: FUNCTION--INCOMPLETE
14880000        ; % TO BECOME AN ERROR EVENTUALLY
14890000     % 3: LITERAL
14900000        ;
14910000     % 4: GROUPING--INCOMPLETE
14920000        BEGIN ERRNO ← 27;
14930000        GO TO FAIL;
14940000        END;
14950000     % 5: NOT USED
14960000        GO TO FAUT;
14970000     % 6: INDIRECTION--SHOULD NOT OCCUR
14980000        GO TO FAUT;
14990000     % 7: ARITHMETIC OPERATOR
15000000        BEGIN ERRNO ← 34;
15010000        GO TO FAIL;
15020000        END;
15030000     % 8: FIXED-LENGTH "/"
15040000        BEGIN ERRNO ← 34;
15050000        GO TO FAIL;
15060000        END;
15070000     % STR REF, PATTERN & REPLACEMENT
15080000        GO TO FAUT;
15090000     % 10: INDIRECTION COMBINED
15100000        GO TO FAUT;
15110000     % 11: ","
15120000        BEGIN ERRNO ← 34;
15130000        GO TO FAIL;
15140000        END;
15150000     % 12: QMARK--ERROR
15160000        GO TO FAUT;
15170000     % 13: "*"  STR VAR
15180000        BEGIN ERRNO ← 34;
15190000        GO TO FAIL;
15200000        END;
15210000     % 14: LABEL IN GO-TO PART
15220000        GO TO FAUT;
15230000     % 15: EXPRESSION
15240000        ;
15250000     % 16: INCOMPLETE ARITH EXPRESSION
15260000        INSERTSTRINGCONVERT(I);
15270000     % 17: "*(" OF BAL VAR
15280000        GO TO FAUT;
15290000     % 18: STR REF & PATTERN
15300000        GO TO FAUT;
15310000     % 19: COMPLETE ARITH EXPR
15320000        ;
15330000     % 20: "S(", "F(", OR "(" OF GO-TO PART
```

```
15340000        GO TO FAUT;
15350000     % 21: COMBINED GO-TO PART
15360000        GO TO FAUT;
15370000        END CASES;
15380000     % INTERCHANGE ELEMENTS:
15390000     MOVE(K+PSIZE[I+1],CODE[(J+PPOINT[I+1]).W],J.C,SCRATCH[*],0);
15400000     MV(3,CODE[PPOINT[I].W],PPOINT[I].C,SCRATCH[K.W],K.C);
15410000     MOVE(K+3,SCRATCH[*],0,CODE[PPOINT[I].W],PPOINT[I].C);
15420000     PSIZE[I] ← K + 3;
15430000     PTYPE[I] ← 10;  % INDIRECTION, COMBINED
15440000     PNAME[I] ← TRUE;
15450000     PTYPE[I+1] ← 0; % WIPE OUT NEXT ELEMENT
15460000     COMBINEINDIRECTION ← TRUE;
15470000     GO TO FIN;
15480000     END;
15490000 %   IF FALL THRU, NO ARGUMENT TO $:
15500000     ERRNO ← 34;
15510000     GO TO FAIL;
15520000 FAUT:
15530000     WRITEINST;
15540000     SYNTAXERR(40,0);
15550000     WRITE(PRINT,17,CODE[*]);
15560000     WRITESTACK(SP);
15570000     SYSTEMERROR ← TRUE;
15580000     GO TO FIN;
15590000 FAIL:
15600000     SYNTAXERR(ERRNO,PLOC[I+1]);
15610000 FIN:
15620000     IF SYSTEMERROR THEN INFORMO(22);
15630000     % IF INFORM THEN WRITEINST;
15640000 END COMBINEINDIRECTION;
15650000 %********COMBINEINDIRECTION***********************************************
15660000 %********COMBINESTRVARNAME************************************************
15670000 BOOLEAN PROCEDURE COMBINESTRVARNAME(MKS);
15680000     VALUE MKS;
15690000     INTEGER MKS;
15700000 BEGIN LABEL FIN, SUCCEED;
15710000 INTEGER I;
15720000 % IF INFORM THEN INFORMO(23);
15730000 COMBINESTRVARNAME ← FALSE;
15740000 IF I←PTYPE[MKS] ≠ 13 AND I ≠ 17 THEN
15750000     BEGIN WRITEINST;
15760000     SYNTAXERR(40,0);
15770000     WRITE(PRINT,17,CODE[*]);
15780000     WRITESTACK(SP);
15790000     SYSTEMERROR ← TRUE;
15800000     GO TO FIN;
15810000     END;
15820000 PTYPE[MKS] ← 0;
15830000 IF NOT COMBINEGROUP(MKS,FALSE) THEN GO TO FIN;
15840000 IF NOT PNAME[MKS] THEN
15850000     BEGIN SYNTAXERR(52,PLOC[MKS+1]); % INVALID NAME
15860000     GO TO SUCCEED;
15870000     END;
15880000 % MARK AS NON-INPUT
15890000 IF PTYPE[MKS] = 1 THEN  % IDENTIFIER--CHANGE TO "%" TYPE
15900000     BEGIN I ← "%";
```

```
15910000     MV(1,I,7,CODE[PPOINT[MKS],W],PPOINT[MKS],C);
15920000     END ELSE
15930000 IF PTYPE[MKS] = 10 THEN % INDIRECTION--REMOVE "I"
15940000     BEGIN PTYPE[MKS] ← 15; % EXPRESSION
15950000     PSIZE[MKS] ← PSIZE[MKS] - 1;
15960000     END;
15970000 SUCCEED:
15980000     COMBINESTRVARNAME ← TRUE;
15990000 FIN:
16000000     IF SYSTEMERROR THEN INFORMO(23);
16010000     % IF INFORM THEN WRITEINST;
16020000 END COMBINESTRVARNAME;
16030000 %********COMBINESTRVARNAME***************************************************
16040000 %********COMPILE*************************************************************
16050000 %
16060000 %       THE INTERNAL FORM OF A SNOBOL PROGRAM IS DESCRIBED HERE:
16070000 %       THE PROGRAM IS "SEGMENTED", WITH A LABELED INSTRUCTION STARTING A
16080000 %   SEGMENT, AND THE UNLABELED INSTRUCTIONS FOLLOWING IT ARE IN THE
16090000 %   SAME SEGMENT.  A SEGMENT IS A STRING, WHOSE "NAME" IS THE LABEL OF
16100000 %   ITS FIRST INSTRUCTION, AND WHOSE "VALUE" IS THE INTERNAL CODED
16110000 %   FORM OF THE INSTRUCTIONS IN THE SEGMENT.
16120000 %       EACH SEGMENT STARTS WITH 2 CHARS GIVING THE LOC OF THE NEXT
16130000 %   SEGMENT, FOR PATCHING PURPOSES.
16140000 %       EACH INSTRUCTION STARTS WITH A 7-CHAR "INTRODUCTION" GIVING:
16150000 %          1 CHAR: THE ILLEGAL CHAR
16160000 %          2 CHARS: THE NUMBER OF THE INSTRUCTION, AS GIVEN IN THE LISTING
16170000 %          2 CHARS: THE LOCATION OF THE SUCCESS EXIT, RELATIVE TO THE
16180000 %             FIRST CHAR OF THE SEGMENT.
16190000 %          2 CHARS: THE SAME FOR THE FAILURE EXIT.
16200000 %       IF EITHER EXIT ISNT SPECIFIED, AND THE NEXT INSTRUCTION IS
16210000 %   UNLABELED, THE POINTER IS TO THE ILLEGAL CHAR AT THE START OF THE
16220000 %   NEXT INSTRUCTION; IF THE NEXT INST IS LABELED, THE GO-TO PART IS
16230000 %   "FILLED OUT" WITH A TRANSFER TO THIS LABEL.
16240000 %
16250000 %   THE INTERNAL CODE IS:
16260000 %       STRING NAMES AND LABELS IN THE GO-TO PARTS ARE REPRESENTED BY
16270000 %   A QUOTE FOLLOWED BY A 12-BIT (2-CHARS) POINTER TO SYMBTABL.
16280000 %       LITERALS ARE REPRESENTED BY "@" FOLLOWED BY A 2-CHAR POINTER
16290000 %   TO SYMBTABL.
16300000 %       PATTERN MATCHES ARE INDICATED BY "=" FOLLOWING THE CODE FOR
16310000 %   THE PATTERN; REPLACEMENTS ARE INDICATED BY "←" FOLLOWING THE CODE
16320000 %   FOR THE REPLACEMENT.  (BOTH ARE POSTFIX OPERATORS.)
16330000 %       THE RESERVED LABELS ARE REPRESENTED AS FOLLOWS:
16340000 %          END: "-E"
16350000 %          RETURN: "-R"
16360000 %          FRETURN: "-F"
16370000 %       SYNTAX ERRORS CAUSE THE CHARACTER "X" WHERE THE STRING REFERENCE
16380000 %   SHOULD BE.
16390000 %       FUNCTIONS ARE POSTFIX OPERATORS, WHICH USE 4 CHARACTERS: "#"
16400000 %   TO INDICATE THE FUNCTION CALL, 2 CHARS FOR A POINTER TO THE
16410000 %   SYMBOL TABLE ENTRY FOR THE FUNCTION, AND 1 CHAR TO GIVE THE NUMBER
16420000 %   OF PARAMETERS.
16430000 %       INDIRECTION IS A POSTFIX OPERATOR CONSISTING OF A "$" AND ONE
16440000 %   CHARACTER GIVING THE NUMBER OF INDIRECTIONS.  NOTE THAT THIS MEANS
16450000 %   THAT THE SYSTEM CANT HANDLE MORE THAN 63 CONSECUTIVE DOLLAR SIGNS
16460000 %   WITHOUT USING PARENTHESES TO CAUSE A GROUPING.  THIS IS NOT
16470000 %   EXPECTED TO CREATE ANY HARDSHIPS WITH ANY USERS.
```

```
16480000 %
16490000 %  THE FOLLOWING STACK ARRAYS ARE USED BY THE COMPILER:
16500000 %
16510000 %  PLOC[*]  HOLDS POINTERS INTO INST[*], TO THE START OF THE ORIGINAL
16520000 %     CODE FOR THE ELEMENT, FOR DIAGNOSTIC PURPOSES
16530000 %  PPOINT[*] HOLDS POINTERS INTO CODE[*], TO THE START OF THE
16540000 %     COMPILED CODE FOR THE ELEMENTS
16550000 %  PSIZE[*] HOLDS THE SIZE OF THE COMPILED PIECE OF CODE
16560000 %  PBACK[*] LINKS BACK TO THE START OF NEXTINGS OF GROUPINGS OR
16570000 %     OF FUNCTION CALLS
16580000 %  PST[*] HOLDS THE OPERATOR FOR ARITHMETIC OPERATORS
16590000 %  PTYPE[*] IS THE TYPE OF ELEMENT, AS FOLLOWS:
16600000 %   0: IGNORE--COMBINED WITH EARLIER ELEMENTS
16610000 %   1: STRING NAME
16620000 %   2: START OF FUNCTION CALL (#---)
16630000 %   3: LITERAL
16640000 %   4: "(" AT START OF GROUPING
16650000 %   5: STRING VARIABLE--COMBINED INTO ONE ELEMENT
16660000 %   6: INDIRECTION, UNCOMBINED ($N)
16670000 %   7: ARITHMETIC OPERATOR
16680000 %   8: "/" OF FIXED-LENGTH VARIABLE
16690000 %   9: STR REF, PATTERN, & REPLACEMENT COMBINED
16700000 %  10: INDIRECTION, COMBINED (INCLUDES "I" FOR INPUT CHECK)
16710000 %  11: NOT IN USE
16720000 %  12: NOT IN USE
16730000 %  13: "*" AT START OF STRING VARIABLE
16740000 %  14: LABEL IN GO-TO PART
16750000 %  15: EXPRESSION (COMBINED)
16760000 %  16: ARITHMETIC EXPRESSION--NO STRING CONVERT ADDED
16770000 %  17: "*(" OF BALANCED STRING VARIABLE
16780000 %  18: STR REF & PATTERN TOGETHER,
16790000 %  19: COMPLETE ARITH EXPR (WITH STRING CONVERT)
16800000 %  20: "S(", "F(", OR "(" OF GO-TO PART
16810000 %  21: COMBINED GO-TO PART
16820000 %
16830000 %
16840000 PROCEDURE COMPILE(INST); ALPHA ARRAY INST[0];
16850000 BEGIN LABEL
16860000     NEXT, ILLEGAL,
16870000     BL, LIT, STR,
16880000     NAME, NAME1, CMA, GOTO, GTPART,
16890000     CONDENSEPATTERN,
16900000     CONDENSEREPLACEMENT,
16910000     DLR,
16920000     ENDSTRVAR,
16930000     ERROR,
16940000     NOP,
16950000     OP,
16960000     PARENERR,
16970000     QMRK, RETURN,
16980000     MESFIL,
16990000     SEMICOLON, START, L1
17000000     ;
17010000 %********
17020000 BOOLEAN
17030000     ARITH,
17040000     ARF,
```

```
17050000    FIXEDLENGTH,
17060000    NF,
17070000    PATTERN,
17080000    REPLACEMENT,
17090000    SVF;
17100000 BOOLEAN ARRAY COMPILINGFCT[0:50];
17110000 INTEGER
17120000    FIRSTCHAR,      % FLAGS FOR FIRST OF INST.
17130000    I,
17140000    J,
17150000    T,
17160000    TEMP;
17170000 %
17180000 P ← SCANCHAR(" ",INST[*],0);
17190000 START:
17200000 FOR I ← 0 STEP 1 UNTIL 5 DO BACKTRACK[0,I] ← 0;
17210000 FIRSTCHAR ← MKS ← PARENCOUNT ← 0;
17220000 GTLAST ← GT;
17230000 PATTERN ← REPLACEMENT ← GT ← GTS ← GTF ← SYSTEMERROR ← SVF ← ARF ←
17240000    FIXEDLENGTH ← FALSE;
17250000 P ← P + 1;
17260000 P ← P + SKIPCHAR(" ",INST[P,W],P,C);
17270000 IF P ≥ INSTSIZE THEN GO TO RETURN;
17280000    SP ← -1;
17290000    IF DATACOMF THEN IF INSTNUM MOD 10 = 0 THEN
17300000       IF INSTNUM = 0 THEN  % RETURN, LINE FEED.
17310000          BEGIN MV(3,CRLF,5,DCWRITE[*],0);
17320000          IF OUTPUT THEN;
17330000          END ELSE
17340000       % WRITE INST #:
17350000          MESSAGETTI(15,INSTNUM);  % TYPE # ON TELETYPE.
17360000 NEXT:
17370000    IF (SP←SP+1) ≥ STACKSIZE THEN
17380000       BEGIN SYNTAXERR(21,P);
17390000       GO TO RETURN;
17400000       END;
17410000 %  IF INFORM THEN
17420000 %     BEGIN MV(6,INST[P,W],P,C,AA,2);
17430000 %     INFORMIA(2,SP,AA);
17440000 %     END;
17450000    PPOINT[SP] ← PTR;
17460000    PLOC[SP] ← P;   % SAVE FOR SYNTAX ERRORS
17470000    IF P ≥ INSTSIZE THEN GO TO QMRK;
17480000    CASE CHARTYPE[CHAR(INST[P,W],P,C)] OF
17490000    BEGIN
17500000 % 0: ERROR
17510000    BEGIN SYNTAXERR(3,P);
17520000    GO TO RETURN;
17530000    END;
17540000 % 1: BLANK
17550000    BEGIN P ← P + SKIPCHAR(" ",INST[P,W],P,C);
17560000    SP ← SP - 1;
17570000    GO TO NEXT;
17580000    END;
17590000 % 2: "←" OR "="
17600000    BEGIN
17610000    IF SP < 1 THEN
```

```
17620000        BEGIN SYNTAXERR(6,P);
17630000        GO TO RETURN;
17640000        END;
17650000    IF NOT (PNAME[0] OR PTYPE[0]=6) THEN
17660000        BEGIN SYNTAXERR(15,P); % REPLACEMENT IN VALUE EXPR
17670000        GO TO RETURN;
17680000        END;
17690000    IF ARF THEN
17700000        BEGIN SYNTAXERR(13,P); % EXTRA "←" OR "="
17710000        GO TO RETURN;
17720000        END;
17730000    IF PARENCOUNT ≠ 0 THEN
17740000        BEGIN SYNTAXERR(27,P); % MISSING ")"
17750000        GO TO RETURN;
17760000        END;
17770000    IF SVF THEN
17780000        BEGIN SYNTAXERR(38,P); % MISSING "*"
17790000        GO TO RETURN;
17800000        END;
17810000    IF GT THEN
17820000        BEGIN SYNTAXERR(7,P);  % UNRECOGNIZED GO-TO CONSTRUCT
17830000        GO TO RETURN;
17840000        END;
17850000    ARF ← TRUE;
17860000    P ← P + 1;
17870000 CONDENSEPATTERN:
17880000    % IF INFORM THEN INFORM0(27);
17890000    FOR I ← 0 STEP 1 UNTIL SP-1 DO
17900000        CASE PTYPE[I] OF
17910000            BEGIN;  % IF 0, IGNORE
17920000        %1: STRING NAME:
17930000            ;
17940000        % 2: START OF FUNCTION CALL
17950000            GO TO ERROR;
17960000        % 3: LITERAL
17970000            ;
17980000        % 4: "(" OF GROUPING
17990000            GO TO ERROR;
18000000        % 5: STR VAR
18010000            ;
18020000        % 6: INDIRECTION
18030000            IF NOT COMBINEINDIRECTION(I) THEN GO TO RETURN;
18040000        % 7: ARITH OPERATOR
18050000            BEGIN IF NOT COMBINEARITHMETIC(I) THEN GO TO RETURN;
18060000            WHILE I > 0 AND PTYPE[I]≠16 DO I ← I - 1;  % FIND RESULT
18070000            IF PTYPE[I] ≠ 16 THEN GO TO ERROR;
18080000            INSERTSTRINGCONVERT(I);
18090000            END;
18100000        % 8: "/" OF FIXED-LENGTH VAR
18110000            GO TO ERROR;
18120000        % 9: STR REF, PATTERN & REPL
18130000            GO TO ERROR;
18140000        % 10: INDIRECTION COMBINED
18150000            ;
18160000        % 11: UNUSED
18170000            GO TO ERROR;
18180000        % 12: UNUSED
```

```
18190000            GO TO ERROR;
18200000         % 13: "*" OF STR VAR
18210000            GO TO ERROR;
18220000         % 14: LABEL IN GO-TO PART
18230000            GO TO ERROR;
18240000         % 15: EXPRESSION
18250000            ;
18260000         % 16: ARITH EXPR--NO STRING CONVERT
18270000            INSERTSTRINGCONVERT(I);
18280000         % 17: "*(" OF BAL VAR
18290000            GO TO ERROR;
18300000         % 18: STR REF & PATTERN
18310000            GO TO ERROR;
18320000         % 19: COMPLETE ARITH EXPR
18330000            ;
18340000         % 20: "S(", "F(", OR "(" OF GO-TO PART
18350000            GO TO ERROR;
18360000         % 21: COMBINED GO-TO PART
18370000            GO TO ERROR;
18380000            END CASES;
18390000      I1 ← I2 ← 0;
18400000      FOR I ← 1 STEP 1 UNTIL SP-1 DO IF PTYPE[I] ≠ 0 THEN
18410000         BEGIN
18420000         MOVE(PSIZE[I],CODE[PPOINT[I],W],PPOINT[I].C,SCRATCH[I2,W],I2.C);
18430000         I2 ← I2 + PSIZE[I];
18440000         I1 ← I1 + 1;
18450000         PTYPE[I] ← 0;
18460000         END;
18470000      IF I1 ≥ 1 THEN
18480000         BEGIN
18490000         IF PTYPE[0] = 1 THEN
18500000            BEGIN AA ← "%";
18510000            MV(1,AA,7,CODE[PPOINT[0],W],PPOINT[0],C);
18520000            END ELSE
18530000         IF PTYPE[0] = 10 THEN
18540000            BEGIN PTYPE[0] ← 15; % EXPRESSION
18550000            PSIZE[0] ← PSIZE[0] - 1; % DROP "I"
18560000            END;
18570000      PTR ← PPOINT[0] + PSIZE[0];
18580000         STORECHARS(1,"S",7);
18590000         STORECHARS(I2,SCRATCH[0],0);
18600000         STORECHARS(1,"=",7);
18610000         PSIZE[0] ← PSIZE[0] + I2 + 2;
18620000         FIRSTCHAR.[43:2] ← 3;  % STR REF, PATTERN EXIST
18630000         PTYPE[0] ← 18;     % STR REF & PATTERN
18640000         END ELSE
18650000         FIRSTCHAR.[43:2] ← 2;  % STR REF, NO PATTERN
18660000      IF GT THEN       % PUT IN "/" AS GO-TO DELIMITER
18670000         BEGIN STORECHARS(1,"/",7);
18680000         PSIZE[0] ← PSIZE[0] + 1;
18690000         END;
18700000      PATTERN ← TRUE;
18710000      SP ← 0;
18720000      PBACK[MKS+1] ← 0;
18730000      % IF INFORM THEN WRITEINST;
18740000      GO TO NEXT;
18750000      END;  % CASE FOR "=" OR "←",
```

```
18760000 % 3: LETTER, DIGIT, OR PERIOD
18770000    GO TO NAME;
18780000 % 4: """
18790000    GO TO LIT;
18800000 % 5: "("
18810000    BEGIN
18820000    IF GT THEN IF PARENCOUNT = 0 THEN GO TO GTPART;
18830000    COMPILINGFCT[PARENCOUNT←PARENCOUNT+1] ← FALSE;
18840000    P ← P + 1;
18850000    PBACK[SP] ← MKS;
18860000    MKS ← SP;
18870000    PTYPE[SP] ← 4;  % "("=BEGIN GROUPING
18880000    PSIZE[SP] ← 0;  % NOTHING TO STORE IN CODE HERE
18890000    GO TO NEXT;
18900000    END;
18910000 % 6: ")"
18920000    BEGIN
18930000    IF PARENCOUNT ≤ 0 THEN
18940000       BEGIN SYNTAXERR(12,P); % EXTRA RIGHT PAREN
18950000       GO TO RETURN;
18960000       END;
18970000    IF SVF AND NOT FIXEDLENGTH THEN IF PARENCOUNT = 1 THEN
18980000       IF PTYPE[MKS] = 17 THEN
18990000       BEGIN
19000000       IF CHAR(INST[P,W],P,C+1) ≠ "*" THEN
19010000          BEGIN SYNTAXERR(38,P+1);
19020000          GO TO RETURN;
19030000          END;
19040000       P ← P + 1; % SKIP ")"
19050000       PARENCOUNT ← 0;
19060000       GO TO ENDSTRVAR;
19070000       END BALANCED VARIABLE ELEMENT;
19080000    IF GT AND PARENCOUNT=1 THEN  % END OF A PIECE OF THE GO-TO PART
19090000       BEGIN
19100000       IF NOT COMBINEGOTOPART(MKS) THEN GO TO RETURN;
19110000       END ELSE
19120000    IF COMPILINGFCT[PARENCOUNT] THEN
19130000       BEGIN
19140000       IF NOT COMBINEGROUP(MKS+1,FALSE) THEN GO TO RETURN;
19150000       COMBINEFUNCTION(IF PTYPE[MKS]=2 THEN MKS ELSE (MKS←PBACK[MKS]));
19160000       END ELSE
19170000       BEGIN PTYPE[MKS] ← 0;  % WIPE OUT "(" ENTRY
19180000       IF NOT COMBINEGROUP(MKS,TRUE) THEN GO TO RETURN;
19190000       END;
19200000    PARENCOUNT ← PARENCOUNT - 1;
19210000    P ← P + 1;
19220000    IF PTYPE[SP←MKS]≠2 THEN MKS ← PBACK[MKS];
19230000    GO TO NEXT;
19240000    END;
19250000 % 7: "*"
19260000    GO TO STR;
19270000 % 8: "$"
19280000    GO TO DLR;
19290000 % 9: "!"  GO-TO PART
19300000 GOTO:
19310000    BEGIN
19320000    IF SVF THEN  % UNMATCHED STR VAR * BEFORE GO-TO.
```

```
19330000          BEGIN SYNTAXERR(38,PPOINT[SP-1]);
19340000          GO TO RETURN;
19350000          END;
19360000       IF GT THEN
19370000          BEGIN SYNTAXERR(7,P);  % UNRECOGNIZED CONSTRUCT
19380000          GO TO RETURN;
19390000          END;
19400000       IF PARENCOUNT ≠ 0 THEN
19410000          BEGIN SYNTAXERR(10,P); % PAREN COUNT NON-ZERO
19420000          GO TO RETURN;
19430000          END;
19440000       GT ← TRUE;
19450000       P ← P + 1;
19460000       PSIZE[SP] ← PTYPE[SP] ← 0;
19470000       IF NOT ARF THEN
19480000          BEGIN FIRSTCHAR.[45:1] ← 0; % NO REPLACEMENT
19490000          GO TO CONDENSEPATTERN;
19500000          END;
19510000       % REPLACEMENT TO BE CONDENSED
19520000    CONDENSEREPLACEMENT:
19530000       % IF INFORM THEN INFORMO(28);
19540000       FOR I ← 0 STEP 1 UNTIL SP-1 DO
19550000          CASE PTYPE[I] OF
19560000             BEGIN
19570000          % 0: IGNORE
19580000             ;
19590000          % 1: STRING NAME
19600000             ;
19610000          % 2: FUNCTION CALL
19620000             GO TO ERROR;
19630000          % 3: LITERAL
19640000             ;
19650000          % 4: "(" OF GROUPING
19660000             GO TO ERROR;
19670000          % 5: STR VAR COMBINED
19680000             GO TO ERROR;
19690000          % 6: INDIRECTION
19700000             IF NOT COMBINEINDIRECTION(I) THEN GO TO RETURN;
19710000          % 7: ARITH OPERATOR
19720000             BEGIN
19730000             IF NOT COMBINEARITHMETIC(I) THEN GO TO RETURN;
19740000             WHILE I > 0 AND PTYPE[I]≠16 DO I ← I-1; % FIND RESULT
19750000             IF PTYPE[I] ≠ 16 THEN GO TO ERROR;
19760000             INSERTSTRINGCONVERT(I);
19770000             END;
19780000          % 8: "/" OF FIXED-LENGTH VAR
19790000             GO TO ERROR;
19800000          % 9: STR REF, PATTERN & REPLACEMENT
19810000             GO TO ERROR;
19820000          % 10: INDIRECTION COMBINED
19830000             ;
19840000          % 11: UNUSED
19850000             GO TO ERROR;
19860000          % 12: UNUSED
19870000             GO TO ERROR;
19880000          % 13: "*" OF STR VAR
19890000             GO TO ERROR;
```

```
19900000        % 14: GO-TO LABEL
19910000           GO TO ERROR;
19920000        % 15: EXPRESSION
19930000           ;
19940000        % 16: ARITH EXPR--NO STR CONVERT
19950000           INSERTSTRINGCONVERT(I);
19960000        % 17: "*(" OF BAL VAR
19970000           GO TO ERROR;
19980000        % 18: STR REF & PATTERN
19990000           IF I ≠ 0 THEN GO TO ERROR;
20000000        % 19: COMPLETE ARITH EXPR
20010000           ;
20020000        % 20: "S(", "F(", OR "(" OF GO-TO PART
20030000           GO TO ERROR;
20040000        % 21: COMBINED GO-TO PART
20050000           GO TO ERROR;
20060000           END CASES;
20070000     IF PTYPE[0] = 1 THEN % MARK AS NON-INPUT
20080000        BEGIN AA ← "%";
20090000        MV(1,AA,7,CODE[PPOINT[0],W],PPOINT[0],C);
20100000        END ELSE
20110000     IF PTYPE[0] = 10 THEN
20120000        BEGIN PTYPE[0] ← 15;    % EXPRESSION
20130000        PSIZE[0] ← PSIZE[0] - 1; % DROP "I"
20140000        END;
20150000     I1 ← I2 ← 0;
20160000     FOR I ← 1 STEP 1 UNTIL SP-1 DO IF PTYPE[I]≠0 THEN
20170000        BEGIN
20180000        MOVE(PSIZE[I],CODE[PPOINT[I],W],PPOINT[I],C,SCRATCH[I2,W],I2,C);
20190000        I2 ← I2 + PSIZE[I];
20200000        I1 ← I1 + 1;
20210000        PTYPE[I] ← 0;
20220000        END;
20230000     PTR ← PPOINT[0] + PSIZE[0];
20240000     FIRSTCHAR.[45:1] ← 1; % REPLACEMENT EXISTS
20250000     STORECHARS(I2,SCRATCH[0],0);
20260000     STORECHARS(1,"←",7);
20270000     PSIZE[0] ← PSIZE[0] + I2 + 1;
20280000     PTYPE[0] ← 9;    % STR REF, PATTERN & REPLACEMENT
20290000     REPLACEMENT ← TRUE;
20300000     SP ← 0;
20310000     PBACK[MKS+1] ← 0;
20320000     % IF INFORM THEN WRITEINST;
20330000     GO TO NEXT;
20340000     END;    % CASE FOR GO-TO ":" OR " /"
20350000  % 10: "=","+",OR "x"
20360000     GO TO OP;
20370000  % 11: ","
20380000     BEGIN
20390000     IF NOT COMPILINGFCT[PARENCOUNT] THEN
20400000        BEGIN SYNTAXERR(9,P);  % ILLEGAL COMMA
20410000        GO TO RETURN;
20420000        END;
20430000     IF NOT COMBINEGROUP(MKS+1,FALSE) THEN GO TO RETURN;
20440000     PBACK[MKS+1] ← (IF PTYPE[MKS]=2 THEN MKS ELSE PBACK[MKS]);
20450000     MKS ← SP ← MKS+1;
20460000     P ← P + 1;
```

```
20470000     GO TO NEXT;
20480000     END;
20490000 % 12: "/"
20500000     BEGIN IF SVF AND PARENCOUNT = 0 AND NOT FIXEDLENGTH THEN
20510000        BEGIN % FIXED-LENGTH VARIABLE SLASH
20520000        IF PTYPE[MKS] ≠ 13 THEN
20530000           BEGIN SYNTAXERR(14,P);
20540000           GO TO RETURN;
20550000           END;
20560000        IF NOT COMBINESTRVARNAME(MKS) THEN GO TO RETURN;
20570000        PTYPE[SP-MKS+1] ← 8; % FIXED-LENGTH "/"
20580000        PPOINT[SP] ← PTR;
20590000        PSIZE[SP] ← 0;
20600000        P ← P + 1;
20610000        FIXEDLENGTH ← TRUE;
20620000        PBACK[SP] ← PBACK[MKS];
20630000        MKS ← SP;
20640000        GO TO NEXT;
20650000        END;
20660000     IF PARENCOUNT = 0 THEN
20670000        IF CHAR(INST[(P-1),W],(P-1),C) = " " THEN
20680000           IF I←CHAR(INST[P,W],P,C+1) = "S" OR I = "F" OR I = "("
20690000              THEN GO TO GOTO;
20700000     GO TO OP;
20710000     END; % CASE FOR "/"
20720000 % 13: QMARK
20730000     IF P ≥ INSTSIZE THEN GO TO QMRK;
20740000 % 14: ";"
20750000     GO TO SEMICOLON;
20760000     END CASES;
20770000 ILLEGAL:
20780000     SYNTAXERR(3,P);
20790000     GO TO RETURN;
20800000 LIT:
20810000     IF GT THEN IF PARENCOUNT ≤ 1 THEN
20820000        BEGIN SYNTAXERR(28,P); GO TO RETURN; END;
20830000     BEGIN
20840000     PSIZE[SP] ← 3; PTYPE[SP] ← 3;  % LITERAL
20850000     T ← P;
20860000     P ← P + SCANCHAR("'","'",INST[P,W],P,C+1) + 1;
20870000     IF P ≥ INSTSIZE THEN
20880000        BEGIN SYNTAXERR(0,T);
20890000        GO TO RETURN;
20900000        END;
20910000     T ← ENTERST(P-T-1,INST[*],T+1,"LIT") & "@"[30:42:6];
20920000     STORECHARS(3,T,5);
20930000     P ← P + 1;
20940000     GO TO NEXT;
20950000     END;
20960000 NAME:
20970000     IF GT THEN IF PARENCOUNT = 0 OR PTYPE[SP-1] = 20 THEN GO TO GTPART;
20980000     PSIZE[SP] ← 3;
20990000     T ← P;
21000000     DO P ← P + 1 UNTIL CHARTYPE[CHAR(INST[P,W],P,C)] ≠ 3;
21010000     T ← IF CHAR(INST[P,W],P,C) = "("
21020000        THEN ENTERST(P-T,INST[*],T,"FCT") & "#"[30:42:6]
21030000        ELSE ENTERST(P-T,INST[*],T,"SYMB") & "'"[30:42:6];
```

```
21040000     STORECHARS(3,T,5);
21050000     IF CHAR(INST[P,W],P,C) = "(" THEN
21060000        BEGIN
21070000        COMPILINGFCT[PARENCOUNT←PARENCOUNT+1] ← TRUE;
21080000        PTR ← PTR + 1; % MAKE ROOM FOR CHAR GIVING # OF PARAMS
21090000        P ← P + 1;
21100000        PBACK[SP] ← MKS;
21110000        MKS ← SP;
21120000        PTYPE[SP] ← 2;     % FUNCTION CALL
21130000        PSIZE[SP] ← 3;     % #--
21140000        GO TO NEXT;
21150000        END ELSE
21160000        BEGIN PTYPE[SP] ← 1; % STRING NAME
21170000        PNAME[SP] ← TRUE;
21180000        END;
21190000     GO TO NEXT;
21200000 GTPART;
21210000     IF PARENCOUNT = 0 THEN
21220000        BEGIN
21230000        IF (TEMP ← CHAR(INST[P,W],P,C)) = "S" OR TEMP = "F" THEN
21240000        IF CHAR(INST[P,W],P,C+1) = "(" THEN
21250000           BEGIN
21260000           IF (IF TEMP="S" THEN GTS ELSE GTF) THEN
21270000              BEGIN SYNTAXERR(4,P);
21280000              GO TO RETURN;
21290000              END;
21300000           P ← P + 2;
21310000           PARENCOUNT ← 1; COMPILINGFCT[1] ← FALSE;
21320000           IF TEMP = "S" THEN GTS ← TRUE ELSE GTF ← TRUE;
21330000           PTYPE[SP] ← 20; % "S(", "F(", OR "(" OF GO-TO PART
21340000           PBACK[SP] ← MKS; MKS ← SP;
21350000           PST[SP] ← TEMP;
21360000           GO TO NEXT;
21370000           END ELSE
21380000           BEGIN SYNTAXERR(5,P+1);
21390000           COMMENT CHAR AFTER S OR F NOT "(";
21400000           GO TO RETURN;
21410000           END ELSE
21420000        IF TEMP = "(" THEN
21430000           BEGIN
21440000           IF GTS AND GTF THEN
21450000              BEGIN SYNTAXERR(4,P);
21460000              GO TO RETURN;
21470000              END;
21480000           P ← P + 1;
21490000           PARENCOUNT ← 1; COMPILINGFCT[1] ← FALSE;
21500000           GTS ← GTF ← TRUE;
21510000           PTYPE[SP] ← 20; % "S(", "F(", OR "(" OF GO-TO PART
21520000           PBACK[SP] ← MKS; MKS ← SP;
21530000           PST[SP] ← "(";
21540000           GO TO NEXT;
21550000           END ELSE
21560000           BEGIN SYNTAXERR(7,P);
21570000           GO TO RETURN;
21580000           END;
21590000        END ELSE
21600000     IF PARENCOUNT = 1 THEN
```

```
21610000        BEGIN
21620000        T ← P; P ← P + 1;
21630000        P ← P + MIN(SCANCHAR(" ",")",INST[P,W],P,C),
21640000                   SCANCHAR(" ",")",INST[P,W],P,C));
21650000        IF CHAR(INST[P,W],P,C) = "(" THEN
21660000           BEGIN SYNTAXERR(8,P);
21670000           GO TO RETURN;
21680000           END;
21690000        IF CHAR(INST[P,W],P,C) = " " THEN
21700000           BEGIN TEMP ← P;
21710000           WHILE CHAR(INST[P,W],P,C) = " " DO P ← P + 1;
21720000           IF P ≥ INSTSIZE THEN
21730000              BEGIN SYNTAXERR(18,0);
21740000              GO TO RETURN;
21750000              END;
21760000           IF CHAR(INST[P,W],P,C) ≠ ")" THEN
21770000              BEGIN SYNTAXERR(8,P);
21780000              GO TO RETURN;
21790000              END;
21800000           END ELSE TEMP ← P;
21810000           IF EQ(3,WORDS[6],2,INST[T,W],T,C) THEN
21820000              IF T+3 = TEMP THEN
21830000                 BEGIN STORECHARS(2,"-E",6);
21840000              PSIZE[SP] ← 2;
21850000                 GO TO L1;
21860000                 END ELSE
21870000              ELSE
21880000           IF EQ(6,WORDS[6],6,INST[T,W],T,C) THEN
21890000              IF T+6 = TEMP THEN
21900000                 BEGIN STORECHARS(2,"-R",6);
21910000              PSIZE[SP] ← 2;
21920000                 GO TO L1;
21930000                 END ELSE
21940000              ELSE
21950000           IF EQ(7,WORDS[6],5,INST[T,W],T,C) THEN
21960000              IF T+7 = TEMP THEN
21970000                 BEGIN STORECHARS(2,"-F",6);
21980000              PSIZE[SP] ← 2;
21990000                 GO TO L1;
22000000                 END;
22010000           T ← ENTERST(TEMP-T,INST[*],T,"INST") & ":"[30:42:6];
22020000           STORECHARS(3,T,5);
22030000           PSIZE[SP] ← 3;
22040000 L1:       PTYPE[SP] ← 14; % IDENTIFIER--LABEL
22050000           GO TO NEXT;
22060000           END;
22070000        GO TO ERROR;
22080000 STR:      % "*" ENCOUNTERED--TEST FOR STRING VARIABLE OR MULTIPLY.
22090000        % AN ASTERISK IS A STRING VARIABLE ASTERISK IFF IT IS IN THE
22100000        % PATTERN PART AND OUTSIDE OF PARENTHESES.
22110000        BEGIN
22120000        DEFINE DUMMY=#;
22130000        IF SP < 1 THEN
22140000           BEGIN SYNTAXERR(2,P);  % UNRECOGNIZED CONSTRUCT IN STR REF
22150000           GO TO RETURN;
22160000           END;
22170000        IF PARENCOUNT > 0 THEN GO TO OP;
```

```
22180000     IF ARF AND NOT GT THEN GO TO OP;
22190000     IF SVF THEN GO TO ENDSTRVAR;
22200000     IF GT THEN
22210000        BEGIN SYNTAXERR(3,P);
22220000        GO TO RETURN;
22230000        END;
22240000     IF CHAR(INST[P,W],P,C+1) = "(" THEN
22250000        BEGIN PTYPE[SP] ← 17;  % "*(" OF BAL STR VAR
22260000        PARENCOUNT ← 1;
22270000        P ← P + 2;
22280000        END ELSE
22290000        BEGIN PTYPE[SP] ← 13;  % "*" OF ARB OR F/L STR VAR
22300000        P ← P + 1;
22310000        END;
22320000     PSIZE[SP] ← 0;
22330000     PTR ← PTR + 2;  % FOR *N
22340000     PBACK[SP] ← MKS;
22350000     MKS ← SP;
22360000     SVF ← TRUE;
22370000     GO TO NEXT;
22380000     END;
22390000 ENDSTRVAR;
22400000     BEGIN
22410000     IF I←PTYPE[MKS] = 13 THEN % SIMPLE STR VAR
22420000        BEGIN
22430000        IF NOT COMBINESTRVARNAME(MKS) THEN GO TO RETURN;
22440000        AA ← "*1";
22450000        END ELSE
22460000     IF I = 17 THEN  % BALANCED STR VAR
22470000        BEGIN
22480000        IF NOT COMBINESTRVARNAME(MKS) THEN GO TO RETURN;
22490000        AA ← "*2";
22500000        END ELSE
22510000     IF I = 8 THEN   % FIXED-LENGTH VAR
22520000        BEGIN PTYPE[MKS] ← 0;
22530000        IF NOT COMBINEGROUP(MKS,FALSE) THEN GO TO RETURN;
22540000        AA ← "*3";
22550000        END ELSE
22560000     % ERROR--MKS DOESNT POINT TO VALID ELEMENT
22570000        GO TO ERROR;
22580000     PTR ← PPOINT[MKS] + PSIZE[MKS];
22590000     STORECHARS(2,AA,6);
22600000     PSIZE[SP←MKS] ← PSIZE[MKS] + 2;
22610000     PTYPE[SP] ← 5;  % STR VAR--COMBINED
22620000     P ← P + 1;  % SKIP "*"
22630000     SVF ← FIXEDLENGTH ← FALSE;
22640000     GO TO NEXT;
22650000     END;
22660000 OP;
22670000     I ← CHAR(INST[P,W],P,C);
22680000     IF I = "*" THEN
22690000        IF CHAR(INST[P,W],P,C+1) = "*"
22700000           THEN P ← P + 1 % TWO START--EXPONENTIATION
22710000           ELSE I ← "×";  % ONE STAR--MULTIPLICATION
22720000     PST[SP] ← I;    % FOR COMBINEARITHMETIC
22730000     I.C6 ← ".";
22740000     PTYPE[SP] ← 7;
```

```
22750000     PBACK[SP] ← MKS;
22760000     PSIZE[SP] ← 2;
22770000     STORECHARS(2,I,6);
22780000     PTR ← PTR + 2;  % MAKE ROOM FOR STRING CONVERT (2 CHARS).
22790000     P ← P + 1;
22800000     GO TO NEXT;
22810000 PARENERR:
22820000     SYNTAXERR(12,P);
22830000     GO TO RETURN;
22840000 ERROR:
22850000     WRITEINST;
22860000     SYNTAXERR(40,0);
22870000     WRITESTACK(SP);
22880000     GO TO RETURN;
22890000 DLR:
22900000     I ← 1;
22910000     DO BEGIN P ← P + 1;
22920000        IF J ← CHAR(INST[P,W],P,C) = "$" THEN I ← I + 1;
22930000        END UNTIL J ≠ "$" AND J ≠ " ";
22940000     AA ← "$-I" & I[36:42:6];
22950000     STORECHARS(3,AA,5);  % "$", NO. OF INDIRECTIONS, "I"
22960000     PTYPE[SP] ← 6;       % INDIRECTION
22970000     PSIZE[SP] ← 2;
22980000     GO TO NEXT;
22990000 MESFIL:
23000000     SYNTAXERR(38,0);
23010000     COMMENT MISSING STR VAR ASTERISK;
23020000     GO TO RETURN;
23030000 SEMICOLON:
23040000     BEGIN
23050000     % IF INFORM THEN INFORM0(29);
23060000     IF SP > 1 THEN GO TO QMRK;
23070000     IF ARF THEN IF NOT GT THEN IF SP=1 THEN IF PTYPE[0]≠9
23080000        THEN STORECHARS(1,"←",7);   % STMT OF FORM <STRREF> ←
23090000     IF NOT GTS THEN SLOC ← PTR;
23100000     IF NOT GTF THEN FLOC ← PTR;
23110000     TEMP ← FLOC & SLOC[24:36:12];
23120000     MOVE(4,TEMP,4,CODE[MARKER,W],MARKER,C+3);
23130000     INSTNUM ← INSTNUM + 1;
23140000     MARKER ← PTR;
23150000     AA ← 0 & INSTNUM[12:36:12] & QMARK[6:42:6];
23160000     STORECHARS(7,AA,1);
23170000     PTR ← PTR + 1;
23180000     GO TO START;
23190000     END;
23200000 QMRK:
23210000     IF PARENCOUNT ≠ 0 THEN
23220000        BEGIN SYNTAXERR(18,0);
23230000        COMMENT PARENCOUNT NON-ZERO AT END OF INSTRUCTION;
23240000        GO TO RETURN;
23250000        END;
23260000     IF SVF THEN GO TO MESFIL;
23270000     PTYPE[SP] ← PSIZE[SP] ← 0;
23280000     IF NOT GT THEN
23290000        BEGIN
23300000        IF NOT ARF THEN IF PTYPE[0]≠18 THEN IF SP>1
23310000           THEN GO TO CONDENSEPATTERN;
```

```
23320000        IF ARF THEN IF NOT GT THEN IF PTYPE[0] ≠ 9
23330000           THEN GO TO CONDENSEREPLACEMENT;
23340000        END;
23350000     I1 ← 0;
23360000     PTR ← PPOINT[0];
23370000     SLOC ← FLOC ← 0;
23380000     FOR I ← 0 STEP 1 UNTIL SP DO IF PTYPE[I] ≠ 0 THEN
23390000        BEGIN
23400000        MOVE(PSIZE[I],CODE[PPOINT[I],W],PPOINT[I],C,SCRATCH[I1,W],I1,C);
23410000        IF PTYPE[I] = 21 THEN  % GO-TO PART--CHECK LOCS
23420000           IF I2←PST[I] = "S" THEN SLOC ← PTR+I1 ELSE
23430000           IF I2 = "F" THEN FLOC ← PTR+I1 ELSE
23440000           IF I2 = "(" THEN
23450000              BEGIN
23460000              IF SLOC = 0 THEN SLOC ← PTR + I1;
23470000              IF FLOC = 0 THEN FLOC ← PTR + I1;
23480000              END ELSE GO TO ERROR;
23490000        I1 ← I1 + PSIZE[I];
23500000        PTYPE[I] ← 0;
23510000        END;
23520000     STORECHARS(I1,SCRATCH[0],0);
23530000     PSIZE[0] ← I1;
23540000     IF P < INSTSIZE THEN
23550000        BEGIN SP ← 1;
23560000        GO TO SEMICOLON;
23570000        END;
23580000     IF NOT GTS THEN SLOC ← PTR;
23590000     IF NOT GTF THEN FLOC ← PTR;
23600000 RETURN;
23610000     MV(2,SLOC,6,CODE[MARKER,W],MARKER,C+3);
23620000     MV(2,FLOC,6,CODE[MARKER,W],MARKER,C+5);
23630000     MV(1,FIRSTCHAR,7,CODE[MARKER,W],MARKER,C+7);
23640000     IF INFORM THEN WRITEINST;
23650000 END COMPILE;
23660000 %********COMPILE*************************************************************
23670000 %********CONTROLPARAMETER****************************************************
23680000 INTEGER PROCEDURE CONTROLPARAMETER(A,P);
23690000     ALPHA ARRAY A[*];
23700000     INTEGER P;
23710000 BEGIN DEFINE DUMMY =#;
23720000 I1 ← P;
23730000 WHILE CHAR(A[*],I1) > 9 AND I1 < 64 DO I1 ← I1 + 1;
23740000 P ← I1;
23750000 WHILE CHAR(A[*],P) ≤ 9 AND P < 64 DO P ← P + 1;
23760000 IF I1 ≥ P THEN I1 ← -1 ELSE
23770000     READ(A[*],FNUM,I1,P-I1,I1);
23780000 CONTROLPARAMETER ← I1;
23790000 END CONTROLPARAMETER;
23800000 %********CONTROLPARAMETER****************************************************
23810000 %********CONVERT*************************************************************
23820000 % THIS ROUTINE DOES A CHARACTER-SET CONVERT, AND (IF CONVERTSTRINGS
23830000 % IS TRUE) A CONVERSION OF I/O STRING NAMES.  IT IS CURRENTLY SET
23840000 % TO CONVERT FROM CDC 3600 SNOBOL (AS IMPLEMENTED AT THE UNIV. OF
23850000 % WISCONSIN) TO B5500 SNOBOL.  USERS AT OTHER INSTALLATIONS WILL
23860000 % PROBABLY WANT TO WRITE THEIR OWN ROUTINE, TO CONVERT FROM A
23870000 % LOCAL VERSION OF SNOBOL TO B5500 SNOBOL.
23880000 PROCEDURE CONVERT(A);
```

```
23890000     ALPHA ARRAY A[0];
23900000 BEGIN INTEGER I;
23910000 ALPHA ARRAY C[0:79];
23920000 DEFINE D = C[I]#, D1 = C[I+1]#, D2 = C[I+2]#,
23930000   D3 = C[I+3]#, D4 = C[I+4]#, D5 = C[I+5]#;
23940000 READ(A[*],F80A1,FOR I ← 0 STEP 1 UNTIL 79 DO C[I]);
23950000 IF CONVERTSTRINGS THEN
23960000 FOR I ← 1 STEP 1 UNTIL FIELDSIZE-6 DO
23970000   IF D = "S" AND D1 = "Y" AND D2 = "S" THEN
23980000       IF D3 = "L" AND D4 = "O" AND D5 = "K" THEN        % SYSLOK : LOOK
23990000     BEGIN D ← "L"; D1 ← "O"; D2 ← "O"; D3 ← "K"; D4 ← D5 ← " ";
24000000   END ELSE IF D3 = "P" AND D5 = "T" THEN
24010000    IF D4 = "I" THEN                          % SYSPIT : READ
24020000     BEGIN D ← "R"; D1 ← "E"; D2 ← "A"; D3 ← "D"; D4 ← D5 ← " ";
24030000    END ELSE IF D4 = "P" THEN                  % SYSPPT : PUNCH
24040000     BEGIN D ← "P"; D1 ← "U"; D2 ← "N"; D3 ← "C"; D4 ← "H"; D5 ← " ";
24050000     END;
24060000 FOR I ← 0 STEP 1 UNTIL 79 DO
24070000    C[I] ← CONVERTVAL[C[I]];
24080000 WRITE(A[*],F80A1,FOR I←0 STEP 1 UNTIL 79 DO C[I]);
24090000 END CONVERT;
24100000 %********CONVERT*************************************************************
24110000 %********CREATELIBRARY*******************************************************
24120000 PROCEDURE CREATELIBRARY(MFID,FID);
24130000 VALUE MFID, FID;
24140000 ALPHA MFID, FID;
24150000 BEGIN INTEGER I, J;
24160000 ALPHA ARRAY X[0:5];
24170000 SAVE FILE LIBE DISK SERIAL [20:200] (15,6,60,SAVE 15);
24180000 %
24190000 FILL LIBE WITH MFID, FID;
24200000 WRITE(LIBE,FLIB0,ENTIER(VERSION));
24210000 WRITE(LIBE,FLIB1,SCATTERNO,MAXINSTSIZE,
24220000       CPULIMITEXISTS,CPULIMIT,
24230000       IOLIMITEXISTS,IOLIMIT,
24240000       RULELIMITEXISTS,RULELIMIT);
24250000 WRITE(LIBE,FLIB2,ENTRY,FOR I←0 STEP 1 UNTIL 15 DO USEDST[I],INSTNUM);
24260000 WRITE(LIBE,FLIB3,FOR I←0 STEP 1 UNTIL 15 DO NEXTCELL[I]);
24270000 FOR I ← 0 STEP 1 UNTIL 15 DO
24280000    IF USEDST[I] THEN
24290000       FOR J ← 0 STEP 2 UNTIL 254 DO
24300000          WRITE(LIBE,FLIB4,NAMTABL[I,J],VALTABL[I,J],IOTABL[I,J],
24310000             NAMTABL[I,J+1],VALTABL[I,J+1],IOTABL[I,J+1]);
24320000 WRITE(LIBE,FLIB5,FOR I←0 STEP 1 UNTIL 31 DO USEDROW[I]);
24330000 WRITE(LIBE,FLIB6,FOR I←0 STEP 1 UNTIL 31 DO DPNTR[I]);
24340000 FOR I ← 0 STEP 1 WHILE USEDROW[I] DO
24350000    FOR J ← 0 STEP 6 UNTIL DPNTR[I].W DO
24360000       BEGIN MOVEWDS(6,DATA[I,J],X[*]);
24370000       WRITE(LIBE,6,X[*]);
24380000       END;
24390000 LOCK(LIBE,SAVE);
24400000 END CREATELIBRARY;
24410000 %********CREATELIBRARY*******************************************************
24420000 %********DATE****************************************************************
24430000 ALPHA PROCEDURE DATE;
24440000 BEGIN % PRODUCES DATE IN FORM:
24450000 %              MM/DD/YY
```

```
24460000 ALPHA X;
24470000 INTEGER D, M, Y;
24480000 STREAM PROCEDURE ALPHADATE(D,M,Y,W);
24490000    VALUE D, M, Y;
24500000    BEGIN DI ← W;
24510000    SI ← LOC M; DS ← 2 DEC; DS ← 1 LIT "/";
24520000    SI ← LOC D; DS ← 2 DEC; DS ← 1 LIT "/";
24530000    SI ← LOC Y; DS ← 2 DEC;
24540000    END ALPHADATE;
24550000 %
24560000 X ← TIME(0);
24570000 Y ← 10 × X.[18:6] + X.[24:6];
24580000 X ← 100 × X.[30:6] + 10 × X.[36:6] + X.[42:6];
24590000 M ← 0;
24600000 WHILE X > 0 DO
24610000    BEGIN M ← M + 1;
24620000    X ← X - MONTHS[M];
24630000    END;
24640000 D ← X + MONTHS[M];
24650000 ALPHADATE(D,M,Y,X);
24660000 DATE ← X;
24670000 END DATE;
24680000 %********DATE*************************************************************
24690000 %********DEBUG************************************************************
24700000 % THIS IS THE CENTRAL CONTROL ROUTINE FOR THE INTERACTIVE
24710000 % DEBUGGING TOOL.
24720000 PROCEDURE DEBUG(RETURNLABEL);
24730000 VALUE RETURNLABEL;
24740000 LABEL RETURNLABEL;
24750000 BEGIN INTEGER I, J, K, L, SAVEUSER;
24760000 BOOLEAN NEWINST;
24770000 DEFINE NUMCOMMANDS= 7 #;
24780000 ALPHA ARRAY COMMAND[0:NUMCOMMANDS];
24790000 SWITCH FORMAT FDBUG ←
24800000    ("≤≠STATEMENT ",I*,"←"),                                    %00
24810000    ("≤≠",I*," STATEMENTS EXECUTED←"),                          %01
24820000    ("≤≠LAST LABEL: ",X*,"←"),                                  %02
24830000    ("≤≠AT LABEL: ",X*,"←"),                                    %03
24840000    ("≤≠TYPE REQUESTS.,.≤≠←"),                                  %04
24850000    ("≤≠LEVEL = ",I*,"≤≠←"),                                    %05
24860000    ("≤≠",X*," NOT DEFINED≤≠←"),                                %06
24870000    ("≤≠",X*," NOT IN USE≤≠←"),                                 %07
24880000    ("≤≠UNRECOGNIZED COMMAND≤≠←"),                              %08
24890000    ("≤≠OK≤≠←"),                                                %09
24900000    ("≤≠",X*," EXECUTED ",I*," TIMES≤≠←"),                      %10
24910000    ("≤≠",X*," = ",""""),                                       %11
24920000    ("≤≠TOO MANY LABELS.≤≠←"),                                  %12
24930000    ("≤≠SUSPENDED≤≠←"),                                         %13
24940000    ("≤≠≠END FORMAT--FDBUG≤≠≠←",(0));
24950000 SWITCH FORMAT FSUSPENDREASON ←
24960000    ("≤≠SUSPEND CALLED≤≠←"),                                    %00
24970000    ("≤≠RULE LIMIT HIT≤≠←"),                                    %01
24980000    ("≤≠LABEL LIMIT HIT≤≠←"),                                   %02
24990000    ("≤≠FATAL ERROR≤≠←"),                                       %03
25000000    ("≤≠END FMT--FSUSRSN≤≠←");
25010000 LABEL
25020000    GETCOMMAND,
```

```
25030000     GET1,
25040000     FAIL,
25050000     ILLEGAL,
25060000     RESUME,
25070000     RUNLOOP;
25080000 %
25090000 SAVEUSER ← USER;
25100000 USER ← MAINUSER;
25110000 NEWINST ← FALSE;
25120000 DEBUGGING ← TRUE;
25130000 FILL COMMAND[*] WITH
25140000     "END",
25150000     "TYPE",
25160000     "SET",
25170000     "RUN",
25180000     "LABEL",
25190000     "ABORT",
25200000     "WHERE",
25210000     "WHY",
25220000     0;
25230000 WRITE(DCWRITE[*],FDBUG[13]);
25240000 IF OUTPUT THEN;
25250000 GETCOMMAND:
25260000     IF NOT INPUT THEN GO TO FAIL;
25270000 GET1:
25280000     MV(1,ARROW,7,DCREAD[DCSIZE,W],DCSIZE,C);
25290000     % IF INFORM THEN WRITE(PRINT,10,DCREAD[*]);
25300000     IF CHAR(DCREAD[*],0) = "=" THEN % CONTROL CARD:
25310000         BEGIN INSTSIZE ← DCSIZE;
25320000         MV(3,STOPPER,5,DCREAD[DCSIZE,W],DCSIZE,C);
25330000         PROCESSCONTROLCARD(DCREAD[*]);
25340000         MV(3,CRLF,5,DCWRITE[*],0);
25350000         IF OUTPUT THEN;
25360000         GO TO GETCOMMAND;
25370000         END;
25380000     I ← SKIPCHAR(" ",DCREAD[*],0);
25390000 % EDIT ROUTINES GO HERE SOMETIME...
25400000     J ← I + SCANCHAR(" ","←",DCREAD[*],I);
25410000     AA ← 0;
25420000     IF J=I > 5 THEN GO TO ILLEGAL;
25430000     MV(J-I,DCREAD[*],I,AA,8-J+I);
25440000     FOR I ← NUMCOMMANDS STEP -1 UNTIL 0 DO
25450000         IF AA = COMMAND[I] THEN CASE I OF
25460000             BEGIN     % CODE FOR VARIOUS COMMANDS:
25470000 %*****************************************
25480000 % 0: END←
25490000     BEGIN RESULT ← TRUE;
25500000     GO TO ENDTERPRET;
25510000     END;
25520000 % 1: TYPE <NAME>←
25530000     BEGIN
25540000     IF (AA←SEARCHST(I←(DCSIZE-J-1),DCREAD[*],J+1,"SYMB")) ≤ 0 THEN
25550000         BEGIN WRITE(DCWRITE[*],FDBUG[7],I←MIN(I,63));
25560000         MV(I,DCREAD[*],J+1,DCWRITE[*],2);
25570000         IF OUTPUT THEN;
25580000         GO TO GETCOMMAND;
25590000         END;
```

```
25600000     AB ← NAME[AA];
25610000     AC ← VALU[AA];
25620000     WRITE(DCWRITE[*],FDBUG[11],I←MIN(I,63));
25630000     MV(I,DCREAD[*],J+1,DCWRITE[*],2);
25640000     I ← I + 6; K ← AC.CH;
25650000     MV(J←MIN(70-I,L←AC.S),FIRSTCHAR(AC),DCWRITE[I.W],I.C);
25660000     MV(1,ARROW,7,DCWRITE[(I+J).W],(I+J).C);
25670000     WHILE I + J ≥ 70 DO
25680000        BEGIN IF NOT OUTPUT THEN GO TO GETCOMMAND;
25690000        K ← K + J;
25700000        L ← L - J;
25710000        MV(I←2,CRLF,5,DCWRITE[*],0);
25720000        MOVE(J←MIN(70-I,L),DATA[AC.R,K,W],K.C,DCWRITE[*],2);
25730000        END;
25740000     MV(1,QUOTE,7,DCWRITE[(I←I+J).W],I.C);
25750000     MV(3,CRLF,5,DCWRITE[I.W],I.C+1);
25760000     IF OUTPUT THEN;
25770000     GO TO GETCOMMAND;
25780000     END TYPE;
25790000 % 2: SET <NAME>←  <VALUE>←
25800000     BEGIN
25810000     AA ← ENTERST(DCSIZE-J-1,DCREAD[*],J+1,"SYMB");
25820000     MV(3,CRLF,5,DCWRITE[*],0); IF OUTPUT THEN;
25830000     IF NOT INPUT THEN GO TO FAIL;
25840000     AB ← STRING(DCSIZE,AA);
25850000     IF DEATH THEN GO TO ENDTERPRET;
25860000     MOVE(DCSIZE,DCREAD[*],0,FIRSTCHAR(AB));
25870000     VALU[AA].LOC ← AB;
25880000     MV(3,CRLF,5,DCWRITE[*],0); IF OUTPUT THEN;
25890000     GO TO GETCOMMAND;
25900000     END SET;
25910000 % 3: RUN <LIMIT> FROM <LABEL> TO <LABEL> <LABEL> <LABEL> ...←
25920000     BEGIN
25930000     I1 ← NLABELLIMIT;  % SAVE IN CASE OF  "TO←".
25940000     NLABELLIMIT ← -1;  % -1 MEANS NO TO-PART FOUND YET.
25950000     DEBUGRULELIMITEXISTS ← NEWINST ← FALSE;
25960000 RUNLOOP:
25970000     J ← J + SKIPCHAR(" ",DCREAD[J.W],J.C);
25980000     IF I2 ← CHAR(DCREAD[J.W],J.C) = "←" THEN GO TO RESUME;
25990000     IF I2 ≤ 9 THEN  % <INTEGER>
26000000        BEGIN
26010000        IF DEBUGRULELIMITEXISTS THEN % TWO OF THEM--ERROR
26020000           BEGIN WRITE(DCWRITE[*],FEH,QMARK);
26030000           IF OUTPUT THEN;
26040000           GO TO GETCOMMAND;
26050000           END;
26060000        I ← 0;    % FOR BUILDING LIMIT
26070000        DO BEGIN I ← I×10 + I2;
26080000           J ← J + 1;
26090000           END UNTIL I2←CHAR(DCREAD[J.W],J.C) > 9;
26100000        DEBUGRULELIMITEXISTS ← TRUE;
26110000        DEBUGRULELIMIT ← RULES[0] + I;
26120000        GO TO RUNLOOP;
26130000        END;
26140000     IF (I←MNEMNO(1,DCREAD[*],J)) = 1 THEN
26150000        BEGIN    % FROM <LABEL>
26160000        IF NEWINST THEN   % TWO OF THEM--ERROR
```

```
26170000           BEGIN WRITE(DCWRITE[*],FEH,QMARK);
26180000           IF OUTPUT THEN;
26190000           GO TO GETCOMMAND;
26200000           END;
26210000        J ← J + SKIPCHAR(" ",DCREAD[J,W],J,C);
26220000        I ← SCANCHAR(" ","←",DCREAD[J,W],J,C);
26230000        IF I ≤ 0 THEN GO TO ILLEGAL;
26240000        AA ← SEARCHST(I,DCREAD[*],J,"INST");
26250000        IF (AB←VALU[AA]).S ≤ 9 THEN  % NO SUCH LABEL:
26260000           BEGIN WRITE(DCWRITE[*],FDBUG[6],K←MIN(63,I));
26270000           MV(K,DCREAD[J,W],J,C,DCWRITE[*],2);
26280000           IF OUTPUT THEN;
26290000           GO TO GETCOMMAND;
26300000           END;
26310000        NEWINST ← TRUE;
26320000        PST[0] ← AA;
26330000        J ← J + I;
26340000        GO TO RUNLOOP;
26350000        END;
26360000     IF I = 2 THEN
26370000        BEGIN    % TO <LABEL> <LABEL> <LABEL> ...←
26380000        J ← J + SKIPCHAR(" ",DCREAD[J,W],J,C);
26390000        IF CHAR(DCREAD[J,W],J,C) = "←" THEN
26400000           BEGIN     % TO←  (SAME LIMITS AS LAST TIME)
26410000           NLABELLIMIT ← I1;
26420000           GO TO RESUME;
26430000        END;
26440000        DO BEGIN     % FIND LABELS:
26450000           I ← SCANCHAR(" ","←",DCREAD[J,W],J,C);
26460000           IF I > 0 THEN  % LABEL FOUND
26470000              BEGIN
26480000              IF (NLABELLIMIT←NLABELLIMIT+1) > MAXLABELLIMIT THEN
26490000                 BEGIN    % TOO MANY LABELS:
26500000                 WRITE(DCWRITE[*],FDBUG[12]);
26510000                 IF OUTPUT THEN;
26520000                 GO TO GETCOMMAND;
26530000                 END;
26540000              IF LABELLIMIT[NLABELLIMIT]←SEARCHST(I,DCREAD[*],J,"INST")
26550000                 ≤ 0 THEN    % NO SUCH LABEL:
26560000                 BEGIN WRITE(DCWRITE[*],FDBUG[6],K←MIN(63,I));
26570000                 MV(K,DCREAD[J,W],J,C,DCWRITE[*],2);
26580000                 IF OUTPUT THEN;
26590000                 GO TO GETCOMMAND;
26600000                 END;
26610000              J ← J + I + SKIPCHAR(" ",DCREAD[(J+I),W],(J+I),C);
26620000              END;
26630000        END UNTIL CHAR(DCREAD[J,W],J,C) = "←";
26640000        GO TO RESUME;
26650000        END;
26660000     % FALL THROUGH: INVALID DELIMITER:
26670000     GO TO ILLEGAL;
26680000     END RUN;
26690000  % 4: LABEL <LABEL>←
26700000     BEGIN
26710000     IF (AA←SEARCHST(I←(DCSIZE-J-1),DCREAD[*],J+1,"INST")) ≤ 0
26720000        THEN WRITE(DCWRITE[*],FDBUG[6],I←MIN(I,63))
26730000        ELSE WRITE(DCWRITE[*],FDBUG[10],I←MIN(I,63),
```

```
26740000           DIGITS(K←VALU[AA],LINK),K);
26750000     MV(I,DCREAD[*],J+1,DCWRITE[*],2);
26760000     IF OUTPUT THEN;
26770000     GO TO GETCOMMAND;
26780000     END LABEL;
26790000 % 5: ABORT ←
26800000     BEGIN RESULT ← FALSE;
26810000     GO TO ENDTERPRET;
26820000     END ABORT;
26830000 % 6: WHERE←  GIVES PLACE OF SUSPENSION.
26840000     BEGIN
26850000     % TYPE STATEMENT NUMBER:
26860000     WRITE(DCWRITE[*],FDBUG[0],DIGITS(INSTNO),INSTNO);
26870000     IF NOT OUTPUT THEN GO TO GETCOMMAND;
26880000     % TYPE NUMBER OF RULES EXECUTED.
26890000     WRITE(DCWRITE[*],FDBUG[1],DIGITS(RULES[0]),RULES[0]);
26900000     IF NOT OUTPUT THEN GO TO GETCOMMAND;
26910000     % LAST LABEL:
26920000     AA ← NAME[ENTRY];
26930000     J ← MIN(AA,S,63);
26940000     IF INSTRUCT > 0 OR NEARPOINT > 9 THEN
26950000        BEGIN WRITE(DCWRITE[*],FDBUG[2],J);
26960000        I ← 14;
26970000        END ELSE
26980000        BEGIN WRITE(DCWRITE[*],FDBUG[3],J);
26990000        I ← 12;
27000000        END;
27010000     MV(J,FIRSTCHAR(AA),DCWRITE[*],I);
27020000     IF NOT OUTPUT THEN GO TO GETCOMMAND;
27030000     % TYPE LEVEL:
27040000     WRITE(DCWRITE[*],FDBUG[5],DIGITS(LEVEL),LEVEL);
27050000     IF OUTPUT THEN;
27060000     GO TO GETCOMMAND;
27070000     END;
27080000 % 7: WHY←    GIVES REASONS FOR SUSPENSION.
27090000     BEGIN
27100000     FOR I ← 0 STEP 1 UNTIL MAXSUSPENDREASON DO
27110000        IF SUSPENDREASON[I] THEN
27120000           BEGIN WRITE(DCWRITE[*],FSUSPENDREASON[I]);
27130000           IF NOT OUTPUT THEN GO TO GETCOMMAND;
27140000           END;
27150000     GO TO GETCOMMAND;
27160000     END;
27170000 %*****************************************
27180000        END COMMAND CASES;
27190000 ILLEGAL:
27200000     WRITE(DCWRITE[*],FDBUG[8]);
27210000     IF OUTPUT THEN;
27220000     GO TO GETCOMMAND;
27230000 FAIL:
27240000     IF INPUT THEN GO TO GET1;
27250000     DEBUGRULELIMITEXISTS ← FALSE;
27260000 RESUME:
27270000     WRITE(DCWRITE[*],FDBUG[9]);
27280000     IF OUTPUT THEN;
27290000     SETLIMITFLAG;
27300000     USER ← SAVEUSER;
```

```
27310000     FOR I ← 0 STEP 1 UNTIL MAXSUSPENDREASON DO SUSPENDREASON[I] ← FALSE;
27320000     IF NEWINST THEN GO TO RETURNLABEL;
27330000     AA ← PST[0];
27340000     % AA CONTAINS THE SYMBOL TABLE ADDRESS OF THE NEXT INST.
27350000 END DEBUG;
27360000 %********DEBUG***********************************************************
27370000 %********DIGITS**********************************************************
27380000 INTEGER PROCEDURE DIGITS(N);
27390000     VALUE N;
27400000     REAL N;
27410000 BEGIN INTEGER D;
27420000 D ← IF N > 0 THEN 0 ELSE 1;
27430000 N ← ENTIER(ABS(N));
27440000 WHILE N > 0 DO
27450000     BEGIN D ← D + 1;
27460000     N ← ENTIER(N DIV 10);
27470000     END;
27480000 DIGITS ← D;
27490000 END DIGITS;
27500000 %********DIGITS**********************************************************
27510000 %********ENTERST*********************************************************
27520000 % ENTERST(N,L,P,X) SEARCHES THE SYMBOL TABLE FOR THE OBJECT OF
27530000 % TYPE X (SEE BELOW); IF THE OBJECT IS NOT PRESENT, A NEW ENTRY
27540000 % IS CREATED FOR IT.  THE RETURN VALUE IS THE SYMB TABLE ADDRESS
27550000 % OF THE OBJECT.  THE NAME CONSISTS OF THE N CHARS STARTING FROM
27560000 % L[P.W],P.C--N CAN BE ANY (NON-NEGATIVE) NUMBER.
27570000 % IF THE SYMBOL TABLE IS FULL AND A NEW ENTRY MUST BE CREATED,
27580000 % MESSAGES ARE PRINTED (LP AND TT), AND -1 IS RETURNED.  ALSO,
27590000 % THE PROCEDURE NEWCELL WILL HAVE SET THE DEATH FLAG TRUE.
27600000 ALPHA PROCEDURE ENTERST(N,L,P,X);
27610000     VALUE N, P, X;
27620000     INTEGER N, P;
27630000     ALPHA ARRAY L[0];
27640000     ALPHA X;
27650000 BEGIN ALPHA J, K;
27660000 INTEGER I, Y, M, XTEST;
27670000 LABEL FOUND, FAIL, RETURN, CREATE;
27680000 DEFINE STWORD = (IF BOOLEAN(Y) THEN VALU[I] ELSE NAME[I])#;
27690000 %
27700000 XTEST ← IF X = "SYMB" THEN 0 ELSE
27710000          IF X = "LIT"  THEN 1 ELSE
27720000          IF X = "FCT"  THEN 2 ELSE
27730000          IF X = "INST" THEN 3 ELSE 0;
27740000 Y ← IF X = "LIT" THEN 1 ELSE 0;
27750000 I ← SCATTER(N,L,P,X);
27760000 WHILE TRUE DO
27770000     % IF YOU DONT UNDERSTAND THIS, YOU ARENT NECESSARILY STUPID;
27780000     BEGIN IF (IF N ≠ (K ← STWORD).S
27790000                 THEN FALSE
27800000                 ELSE EQUAL(N,L[P,W],P,C,FIRSTCHAR(K)))
27810000           THEN IF (IF N = 0 THEN (K.LOC ≠ 0) ELSE TRUE)
27820000                THEN IF (IF X = "LIT"
27830000                         THEN NAME[I]
27840000                         ELSE K).TYPE = XTEST
27850000                     THEN GO TO FOUND;
27860000     IF X = "LIT" THEN K ← NAME[I];
27870000     IF K.LINK = 0 THEN GO TO CREATE;
```

```
27880000     I ← K.LINK;
27890000     END;
27900000 CREATE:
27910000     BEGIN DEFINE DUMMY=#;
27920000     J ← STRING(N,0);
27930000     MOVE(N,L[P,W],P,C,FIRSTCHAR(J));
27940000     IF STWORD.LOC ≠ 0 THEN
27950000        BEGIN M ← NEWCELL(I,STR);
27960000        IF DEATH THEN GO TO FAIL;
27970000        NAME[I].LINK ← I ← M;
27980000        END;
27990000     NAME[I] ← IF BOOLEAN(Y) THEN 1 ELSE J;
28000000     VALU[I] ← IF BOOLEAN(Y) THEN J ELSE 1;
28010000     M ← J.CH = 2;
28020000     MV(2,I,6,DATA[J,R,M,W],M,C);
28030000     NAME[I].TYPE ← XTEST;
28040000     IF X = "FCT" THEN
28050000        BEGIN
28060000        K ← INTRINSIC(L,P,N);
28070000        IF K ≠ 0 THEN VALU[I] ← 1 & K CLINK;
28080000        END;
28090000     IF TRACEALL THEN
28100000        BEGIN VALU[I].IOUSE ← (IF X="FCT" THEN 3 ELSE 1);
28110000        IO[I] ← 0 & 2 CIOTYPE & 3 CFILNO
28120000           & (IF X="FCT" THEN 3 ELSE 1) CIOUSE;
28130000        END;
28140000     GO TO FOUND;
28150000     END;
28160000 FAIL:
28170000     MESSAGEO(14);
28180000     MESSAGETTO(14);
28190000     I ← -1;
28200000 FOUND: ENTERST ← I;
28210000 RETURN:
28220000 END ENTERST;
28230000 %********ENTERST*****************************************************
28240000 %********FINDUSERS***************************************************
28250000 PROCEDURE FINDUSERS;
28260000 BEGIN INTEGER I, J;
28270000 ALPHA ARRAY X[0:2×TTMAX];
28280000 ALPHA ST1, ST2;
28290000 %
28300000 ST1 ← STAT[USER].[9:9];
28310000 ST2 ← STAT[MAINUSER].[9:9];
28320000 NUMUSERS ← STATUS(X[*]) / 2;
28330000 FOR I ← NUMUSERS-1 STEP -1 UNTIL 0 DO
28340000     BEGIN J ← I × 2;
28350000     STAT[I] ← X[J];
28360000     ID[I] ← X[J+1];
28370000     END;
28380000 IF ST2 ≠ STAT[MAINUSER].[9:9] THEN
28390000     BEGIN MAINUSER ← 0;
28400000     FOR I ← NUMUSERS-1 STEP -1 UNTIL 0 DO
28410000        IF STAT[I].[9:9] = ST2 THEN MAINUSER ← I;
28420000     END;
28430000 IF ST1 ≠ STAT[USER].[9:9] THEN
28440000     BEGIN USER ← -1;
```

```
28450000     FOR I ← 0 STEP 1 WHILE USER < 0 AND I < NUMUSERS DO
28460000        IF ST1 = STAT[I].[9:9] THEN USER ← I;
28470000     IF USER < 0 THEN USER ← MAINUSER;
28480000     END;
28490000 DATACOMF ← NUMUSERS > 0;
28500000 END FINDUSERS;
28510000 %********FINDUSERS************************************************************
28520000 %********GARBAGE COLLECTOR****************************************************
28530000 PROCEDURE GARBAGECOLLECTOR;
28540000 %
28550000 BEGIN INTEGER I,K,M,MAX,P,ROWS,ST,TOTAL;
28560000 REAL TIMECP, TIMEIO;
28570000 ALPHA J;
28580000 INTEGER ARRAY NEWPNTR[0:DATASIZE];
28590000 LABEL L;
28600000 %
28610000 TIMECP ← TIME(2); TIMEIO ← TIME(3);
28620000 IF INFORM THEN
28630000     BEGIN
28640000     WRITE(PRINT,FGC);
28650000     WRITETIME(0,0);
28660000     CHECKSYMBTABL;
28670000     END;
28680000 %  FREE STRINGS ATTACHED TO CELLS ON AVAILABLE SPACE LISTS:
28690000 FOR I ← 0 STEP 1 UNTIL STRMAX DO IF USEDST[I] THEN
28700000     BEGIN K ← NEXTCELL[I];
28710000     WHILE K ≠ 0 DO
28720000        BEGIN NAME[K].LOC ← VALU[K] ← 0;
28730000        K ← NAME[K].LINK;
28740000        END;
28750000     END;
28760000 % CHECK FOR CELLS HELD A LONG TIME, THEN DISCARDED:
28770000 IF GCS.C = 0 THEN
28780000     FILL NOTMOVED[*] WITH 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
28790000                           0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0;
28800000 MAX ← TOTAL ← ROWS ← 0;
28810000 IF NOT DEATH THEN
28820000 FOR I ← 0 STEP 1 UNTIL DATASIZE DO
28830000     IF USEDROW[I] THEN
28840000        BEGIN
28850000        ROWS ← ROWS + 1;
28860000        MV(1,QMARK,7,DATA[I,DPNTR[I],W],DPNTR[I],C);
28870000        P ← NEWPNTR[I] ← NOTMOVED[I];
28880000        L: P ← P + SCANCHAR(QMARK,QMARK,DATA[I,P,W],P,C);
28890000        IF P < DPNTR[I] THEN
28900000           BEGIN MV(2,DATA[I,P,W],P,C+1,ST,6);
28910000           FOR K ← 1,0 DO
28920000              BEGIN
28930000              J ← IF BOOLEAN(K) THEN VALU[ST] ELSE NAME[ST];
28940000              IF J.CH = P+3 THEN
28950000              IF J.R = I THEN IF J.CH-3 > NEWPNTR[I] THEN
28960000                 BEGIN M ← J.CH - 3;
28970000                 MOVE(J.S+3,DATA[I,M,W],M,C,
28980000                    DATA[I,NEWPNTR[I],W],NEWPNTR[I],C);
28990000                 IF BOOLEAN(K)
29000000                    THEN VALU[ST].CH ← NEWPNTR[I] + 3
29010000                    ELSE NAME[ST].CH ← NEWPNTR[I] + 3;
```

```
29020000                    NEWPNTR[I] ← NEWPNTR[I] + J.S + 3;
29030000                    P ← P + J.S + 3;
29040000                    GO TO L;
29050000                    END ELSE
29060000                    BEGIN P ← P + J.S + 3;
29070000                    NOTMOVED[I] ← NEWPNTR[I] ← P;
29080000                    GO TO L;
29090000                    END;
29100000                 END;
29110000              P ← P + 1; GO TO L;
29120000              END;
29130000           IF 8184 - (DPNTR[I] ← NEWPNTR[I]) > MAX THEN
29140000              MAX ← 8184 - DPNTR[I];
29150000           TOTAL ← TOTAL + 8184 - DPNTR[I];
29160000           END;
29170000 IF MAX < 1000 OR TOTAL < 2000 THEN
29180000    BEGIN I ← 0;
29190000    WHILE USEDROW[I] AND I < DATASIZE DO I ← I + 1;
29200000    USEDROW[I] ← TRUE;
29210000    END;
29220000 CHECKSYMBTABL;
29230000 IF INFORM THEN
29240000    BEGIN
29250000    WRITETIME(0,0);
29260000    WRITE(PRINT,FGCRES,ROWS,MAX,TOTAL);
29270000    END;
29280000 GCS ← GCS + 1;
29290000 GCTIMECP ← GCTIMECP + (TIME(2)-TIMECP)/60;
29300000 GCTIMEIO ← GCTIMEIO + (TIME(3)-TIMEIO)/60;
29310000 END GARBAGECOLLECTOR;
29320000 %********GARBAGE COLLECTOR***********************************************
29330000 %********INDIRECT*******************************************************
29340000 % INDIRECT(SP) INDIRECTS ON THE STRING NAMED BY STACK[I], AND
29350000 % LEAVES THE RESULT IN THE SAME STACK LOCATION.  THIS ROUTINE IS
29360000 % CALLED BY VARIOUS INTRINSIC FUNCTIONS THAT NEED TO INTERPRET
29370000 % THEIR PARAMETERS AS STRING NAMES.
29380000 PROCEDURE INDIRECT(SP);
29390000 VALUE SP; INTEGER SP;
29400000 BEGIN ALPHA AA, AB;
29410000 % IF INFORM THEN INFORMO(19);
29420000 AA ← PST[SP];
29430000 AB ← VALU[AA];
29440000 PST[SP] ← ENTERST(AB.S,DATA[AB.R,*],AB.CH,"SYMB");
29450000 PNAME[SP] ← TRUE;
29460000 END INDIRECT;
29470000 %********INDIRECT*******************************************************
29480000 %********INFORM*********************************************************
29490000 %  THE FOLLOWING PROCEDURES ARE CALLED ONLY TO OUTPUT CERTAIN
29500000 %  INFORMATIVE MESSAGES.  THE PROCEDURE NAMES CONSIST OF "INFORM"
29510000 %  FOLLOWED BY A STRING OF "I"-S OR "A"-S, TELLING WHAT SORT OF
29520000 %  FORMAT THE PARAMETERS WILL BE WRITTEN OUT AS.  THE FIRST PARAMETER
29530000 %  GIVES THE SUBSCRIPT TO A SWITCH FORMAT.
29540000 %
29550000 PROCEDURE INFORMO(I);
29560000 VALUE I; INTEGER I;
29570000 BEGIN SWITCH FORMAT NFRMO ←
29580000    (X80,"SCANSUCCESS"),                                          %00
```

```
29590000     (X80,"ASSIGNMENT OF STRING VARIABLES"),                          %01
29600000     (X80,"DELETE PATTERN"),                                          %02
29610000     (X80,"RRETURN"),                                                 %03
29620000     (X80,"INSTRUCTION FAILED"),                                      %04
29630000     (X80,"START OF DEFINE SEGMENT"),                                 %05
29640000     (X80,"END OF DEFINE SEGMENT"),                                   %06
29650000     (X80,"RETURNTEMPS"),                                             %07
29660000     (X80,"SCAN FAILED"),                                             %08
29670000     (/"**ILLEGAL ARGUMENT TO .S"),                                   %09
29680000     (//"**DUMMY FORMAT--NFRMO[10]",I10),                             %10
29690000     (///"**SEND COPY OF PROGRAM TO SYSTEM AUTHORS."//),               %11
29700000     (X90,"SCAN--CONSTANT"),                                          %12
29710000     (X90,"SCAN--DROPBACK"),                                          %13
29720000     (X80,"FUNCTION"),                                                %14
29730000     (X80,"GROUP"),                                                   %15
29740000     (X80,"INDIRECTION"),                                             %16
29750000     (X80,"DOTFCT"),                                                  %17
29760000     (X80,"ELEMENT"),                                                 %18
29770000     (X80,"INDIRECT"),                                                %19
29780000     (X80,"**COMBINEARITHMETIC."),                                    %20
29790000     (X80,"**COMBINEGROUP."),                                         %21
29800000     (X80,"**COMBINEINDIRECTION."),                                   %22
29810000     (X80,"**COMBINESTRVARNAME."),                                    %23
29820000     (X80,"**COMBINEGOTOPART."),                                      %24
29830000     (X80,"**COMBINEFUNCTION."),                                      %25
29840000     (X80,"GO-TO PART"),                                              %26
29850000     (X80,"**CONDENSEPATTERN"),                                       %27
29860000     (X80,"**CONDENSEREPLACEMENT"),                                   %28
29870000     (X80,"**SEMICOLON"),                                             %29
29880000     (X80,"**INSERT STRING CONVERT"),                                 %30
29890000     (/"**END FORMAT--NFRMO");
29900000 WRITE(PRINT,NFRMO[I]);
29910000 END INFORMO;
29920000 PROCEDURE INFORMA(I,P);
29930000 VALUE I, P; INTEGER I; ALPHA P;
29940000 BEGIN SWITCH FORMAT NFRMA ←
29950000     (/"**ENTRY = ",A2),                                              %00
29960000     (X80,"PUSH(",A2,")"),                                            %01
29970000     (X80,"POP(",A2,")"),                                             %02
29980000     (X80,"RETURNCELL(",A2,")"),                                      %03
29990000     (X80,"PROGRAM SEGMENT = ",A6),                                   %04
30000000     (//"**ILLEGAL I/O DESCRIPTOR AT WORD ",A2),                      %05
30010000     (/"**END FORMAT--NFRMA",A6);
30020000 IF I > 5 OR I < 0 THEN I ← 5;
30030000 WRITE(PRINT,NFRMA[I],P);
30040000 END INFORMA;
30050000 PROCEDURE INFORMI(I,P);
30060000 VALUE I, P; INTEGER I, P;
30070000 BEGIN SWITCH FORMAT NFRMI ←
30080000     (X80,"FIXED LENGTH = ",I4),                                      %00
30090000     (//"INVALID TYPE ",I5," IN GROUPING."),                          %01
30100000     (//"ERROR IN DOTFCT IN STATEMENT ",I6/),                         %02
30110000     (//"ERROR IN GROUPING IN STATEMENT ",I6/),                       %03
30120000     (//"INTERPRETER STACK OVERFLOW IN STATEMENT ",I6//),             %04
30130000     (//"**PATTERN ERROR IN STATEMENT ",I6),                          %05
30140000     (X80,"INPUT FROM FILE ",I2),                                     %06
30150000     (X80,"TEMPCELL(",I2,") CALLED."),                                %07
```

```
30160000     (X80,"INSTNO = ",I6),                                          %08
30170000     (X80,"INTERPRETER CALLED--LEVEL = ",I3),                       %09
30180000     (X80,"END INTERPRETER--LEVEL ",I3),                            %10
30190000     (X80,"INTRINSIC ",I2),                                         %11
30200000     (X80,"NUMVAL = ",I12),                                         %12
30210000     (X80,"OUTPUT TO FILE ",I2),                                    %13
30220000     (/"**END FORMAT--NFRMI",I10);
30230000 WRITE(PRINT,NFRMI[I],P);
30240000 END INFORMI;
30250000 PROCEDURE INFORMIA(I,P1,P2);
30260000 VALUE I, P1, P2; INTEGER I, P1; ALPHA P2;
30270000 BEGIN SWITCH FORMAT NFRMIA ←
30280000     (X80,"TEMPCEL(",I2,") = ",A2),                                 %00
30290000     (X80,"NEWCELL(",I2,") = ",A2),                                 %01
30300000     (X80,"SP = ",I3,"; CODE = ",A6),                               %02
30310000     (/"**END FORMAT--NFRMIA",I10,A6);
30320000 WRITE(PRINT,NFRMIA[I],P1,P2);
30330000 END INFORMIA;
30340000 PROCEDURE INFORMII(I,P1,P2);
30350000 VALUE I, P1, P2;
30360000 INTEGER I, P1, P2;
30370000 BEGIN SWITCH FORMAT NFRMII ←
30380000     ("DUMMY FORMAT--NFRMII[0]",2I10),                              %00
30390000     ("DUMMY FORMAT--NFRMII[1]",2I10),                              %01
30400000     (X80,"**DUMMY FORMAT--NFRMII[2]"),                             %02
30410000     (//"**FATAL ERROR IN SCANNER IN STATEMENT ",I6,"; SP = ",I4),  %03
30420000     (X80,"TT IN FROM ",I2,"/",I2),                                 %04
30430000     (X80,"TT OUT TO  ",I2,"/",I2),                                 %05
30440000     (X85,"SCAN--SP = ",I3,", POINT = ",I4),                        %06
30450000     (/"**END FORMAT--NFRMII",2I10);
30460000 WRITE(PRINT,NFRMII[I],P1,P2);
30470000 END INFORMII;
30480000 %********INFORM*************************************************
30490000 %********INITIALIZESYMBTABL*************************************
30500000 PROCEDURE INITIALIZESYMBTABL;
30510000 BEGIN DEFINE DUMMY=#;
30520000 FOR I ← 0 STEP 1 UNTIL SCATTERNO DO
30530000     BEGIN
30540000     FOR J ← SCATTERSIZE+16 STEP 1 WHILE J < STWMAX DO
30550000         NAMTABL[I,J].LINK ← (J+1) & I CONCSTR;
30560000     NEXTCELL[I] ← (SCATTERSIZE+16) & I CONCSTR;
30570000     USEDST[I] ← TRUE;
30580000     END;
30590000 SYMBTABLSETUP ← TRUE;
30600000 END INITIALIZESYMBTABL;
30610000 %********INITIALIZESYMBTABL*************************************
30620000 %********INPUT**************************************************
30630000 % TELETYPE INPUT PROCEDURE:
30640000 BOOLEAN PROCEDURE INPUT;
30650000 BEGIN
30660000 SWITCH FORMAT FLAGS ← ("≤≠FLAGS[0]≤≠←"),
30670000             ("≠≤  MESSAGE WAS TOO LONG...PLEASE REENTER≠≤←"),
30680000             ("≠≤  MESSAGE WAS PARTLY LOST...PLEASE REENTER≠≤←");
30690000 ARRAY SINK[0:BUFINSIZE-1];
30700000 INTEGER SIZ, TIMEX, FL, Q, R;
30710000 LABEL LOOP, NOGO, ABNORM, CHECKIT, EXIT, FAIL;
30720000 ALPHA S;
```

```
30730000 INTEGER STREAM PROCEDURE SIZE(A);
30740000           BEGIN SI ← A; TALLY ← 0;
30750000             DCINCHAR(IF SC="←" THEN JUMP OUT; SI←SI+1; TALLY←TALLY+1);
30760000             SIZE ← TALLY
30770000           END SIZE;
30780000 STREAM PROCEDURE MOVE(A,B,C,D); VALUE C, D;
30790000           BEGIN SI←A; DI←B; DI←DI+D; DS←C CHR  END MOVE;
30800000 %
30810000 % IF INFORM THEN INFORMII(4,STAT[USER],[9:4],STAT[USER],[14:4]);
30820000 IF NOT DATACOMF THEN GO TO FAIL;
30830000 INPUT ← TRUE;
30840000 BREAK ← FALSE;
30850000 FL ← DCSIZE ← 0;
30860000 TIMEX ← TIME(1);
30870000 LOOP: Q ← DCSIZE DIV 8; R ← DCSIZE MOD 8;
30880000    IF DATACOMF
30890000       THEN READ(DCIN(STAT[USER],0),BUFINSIZE,SINK[*]) [NOGO:ABNORM]
30900000       ELSE GO TO FAIL;
30910000    IF BOOLEAN((S←STATUS(STAT[USER],0)),[28:1]) THEN
30920000       BEGIN % BUFFER OVERFLOW
30930000       FL ← 2;
30940000       GO TO LOOP;
30950000       END;
30960000       TIMEX ← TIME(1);
30970000    IF BOOLEAN(SINK[0],[25:1])
30980000       THEN SIZ ← DCINCHAR
30990000       ELSE SIZ ← SIZE(SINK[1]);
31000000    IF DCSIZE ← DCSIZE + SIZ > MSGSIZE
31010000       THEN FL ← 1
31020000       ELSE MOVE(SINK[1],DCREAD[Q],SIZ,R);
31030000    IF BOOLEAN(SINK[0],[25:1])
31040000       THEN GO TO LOOP;
31050000 CHECKIT: % CHECK FOR OVERFLOW AND TRANSMISSION ERROR
31060000    IF FL > 0 THEN
31070000       BEGIN WRITE(DCWRITE[*],FLAGS[FL]);
31080000       IF NOT OUTPUT THEN GO TO FAIL;
31090000       IF INPUT THEN GO TO EXIT;
31100000       END ELSE GO TO EXIT;
31110000 ABNORM:  S ← STATUS(STAT[USER],0);
31120000    % [28:1] = 1 IF BUFFER OVERFLOW
31130000    IF BOOLEAN(S,[28:1]) THEN FL ← 2;
31140000    % [23:1]  = 1 IF ABNORMAL CONDITION
31150000    % [24:1]  = 1 IF BUFFER IS READ READY
31160000    % [30:1]  = 1 IF UNIT NOT READY
31170000    IF S.[23:2]=3 OR BOOLEAN(S,[30:1]) THEN GO TO FAIL;
31180000    READ(DCIN(STAT[USER],0),BUFINSIZE,SINK[*]) [NOGO:NOGO];
31190000 NOGO:
31200000    IF TIME(1) - TIMEX < WAITTIME THEN
31210000       BEGIN WHEN(1); GO TO LOOP; END;
31220000    S ← STATUS(STAT[USER],1);
31230000    IF BOOLEAN(S,[30:1]) THEN GO TO FAIL;
31240000    IF BOOLEAN(S,[28:1]) THEN
31250000       BEGIN FL ← 2; GO TO LOOP; END;
31260000 FAIL:
31270000    INPUT ← FALSE;
31280000    FINDUSERS;
31290000 EXIT:   END INPUT;
```

```
31300000 %========== END DATA COM INPUT PROCEDURE ==============================
31310000 %********INSERTSTRINGCONVERT*********************************************
31320000 PROCEDURE INSERTSTRINGCONVERT(SP);
31330000    VALUE SP;
31340000    INTEGER SP;
31350000 BEGIN INTEGER I;
31360000 % IF INFORM THEN INFORMO(30);
31370000 I ← PPOINT[SP] + PSIZE[SP];
31380000 AA ← ".S";
31390000 MV(2,AA,6,CODE[I,W],I,C);
31400000 PSIZE[SP] ← PSIZE[SP] + 2;
31410000 PNAME[SP] ← FALSE;
31420000 PTYPE[SP] ← 19;  % COMBINED ARITHMETIC EXPRESSION
31430000 END INSERTSTRINGCONVERT;
31440000 %********INSERTSTRINGCONVERT*********************************************
31450000 %********INTERPRETER*****************************************************
31460000 % THIS IS THE CENTRAL CONTROL ROUTINE FOR THE INTERPRETER.  IT IS
31470000 % SEVERAL THOUSAND CARDS LONG.
31480000 PROCEDURE INTERPRETER;
31490000 BEGIN
31500000 %******************
31510000 INTEGER STREAM PROCEDURE CHAR(L,I);
31520000 VALUE I;
31530000 BEGIN SI ← L; SI ← SI + I;
31540000 DI ← LOC CHAR; DI ← DI + 7;
31550000 DS ← 1 CHR;
31560000 END CHAR;
31570000 %******************
31580000 BOOLEAN STREAM PROCEDURE EQ(N,L1,I1,L2,I2);
31590000 VALUE N, I1, I2;
31600000 BEGIN SI ← L1; SI ← SI + I1;
31610000 DI ← L2; DI ← DI + I2;
31620000 N(IF SC ≠ DC THEN JUMP OUT);
31630000 IF TOGGLE THEN TALLY ← 0 ELSE TALLY ← 1;
31640000 EQ ← TALLY;
31650000 END EQ;
31660000 %******************
31670000 STREAM PROCEDURE MV(N,L1,I1,L2,I2);
31680000 VALUE N, I1, I2;
31690000 BEGIN SI ← L1; SI ← SI + I1;
31700000 DI ← L2; DI ← DI + I2;
31710000 DS ← N CHR;
31720000 END MOVE;
31730000 %******************
31740000 ALPHA
31750000    FCT,
31760000    FCTN,
31770000    FCTV,
31780000    RETURNVAL;
31790000 ALPHA ARRAY
31800000    INST[0:MAXINSTSIZE,W];
31810000 DEFINE
31820000    DIE = GO TO DEAD #,
31830000    FAIL=GO TO FAILED#;
31840000 BOOLEAN INGOTOPART;
31850000 ALPHA FIRSTCH;
31860000 LABEL
```

```
31870000     ARITHOVFL,        % INTEGER OVERFLOW IN ARITHMETIC
31880000     BRANCH,           % INTRINSIC FCT CASE STMT
31890000     DEAD,             % FATAL ERROR LABEL
31900000     DEFINEDFCT,       % PROGRAM-DEFINED FUNCTION
31910000     FCTFAIL,          % FAILURE OF FUNCTION CALLS
31920000     DVDZERO,          % DIVIDE-BY-ZERO IN ARITHMETIC
31930000     FAILED,           % STATEMENT FAILURE LABEL
31940000     FIXERR,
31950000     GOTO,
31960000     INTERPRET,
31970000     LIMITHIT,
31980000     MINLEF,
31990000     NONNUMERIC,       % NON-NUM ARITH OPERAND
32000000     PERROR,
32010000     PFIX,
32020000     PSCAN,
32030000     REFSET,
32040000     RETURN,
32050000     SCANFAILURE,
32060000     SETGO,
32070000     SUCCEED,          % FUNCTION SUCCESS LABEL
32080000     UNDEFFCT,         % CALL OF UNDEFINED FUNCTION
32090000     UNDEFINED;        % UNDEFINED LABEL
32100000 LABEL    % LOCAL TO SCANNER:
32110000     BALANCE,
32120000     DROPBACK,
32130000     NEWPOINT,
32140000     NEXTPATTERN,
32150000     SCANERR,
32160000     SCANSUCCESS,
32170000     SIZEFAILURE;
32180000 %
32190000 NEARPOINT ← 9;
32200000 LEVEL ← LEVEL + 1;
32210000 AA ← ENTRY;
32220000 ENTRY ← RULES[LEVEL] ← 0;
32230000 % IF INFORM THEN INFORMI(9,LEVEL);
32240000 GO TO SETGO;
32250000 %
32260000 INTERPRET:
32270000     % START OF A SINGLE INSTRUCTION
32280000     IF DEATH THEN DIE;
32290000 %   IF INFORM THEN
32300000 %      BEGIN WRITE(PRINT,FBL);
32310000 %      WRITE(PRINT,17,INST[*]);
32320000 %      INFORMI(8,INSTNO);
32330000 %      WRITETIME(0,0);
32340000 %      END;
32350000     IF ALIMITEXISTS THEN
32360000        BEGIN
32370000        IF RULELIMITEXISTS THEN
32380000           IF RULES[0] > RULELIMIT THEN GO TO LIMITHIT;
32390000        IF CPULIMITEXISTS THEN
32400000           IF TIME(2) > CPULIMIT THEN GO TO LIMITHIT;
32410000        IF IOLIMITEXISTS THEN
32420000           IF TIME(3) > IOLIMIT THEN GO TO LIMITHIT;
32430000        IF DEBUGGING THEN
```

(LISTING CONT'D)

B5700
SNOBOL
COMPILER
(LISTING CONT'D)

```
32440000            IF DEBUGRULELIMITEXISTS THEN
32450000               IF DEBUGRULELIMIT ≤ RULES[0] THEN
32460000                  BEGIN SUSPENDREASON[1] ← TRUE;
32470000                  DEBUG(SETGO);
32480000                  END;
32490000         END;
32500000      RULES[0] ← RULES[0] + 1;          % INCR GLOBAL RULE COUNT.
32510000      RULES[LEVEL] ← RULES[LEVEL] + 1; % INCR LOCAL RULE COUNT.
32520000      SP ← RSIZE ← FRONTEND ← REAREND ← 0;
32530000      INGOTOPART ← SELFREFLAG ← NOREPLACEMENT ← VARFLAG ← FALSE;
32540000      SUCCESS ← TRUE;
32550000      RELATIVEPOINTER ← 1;
32560000      % CONTROLPOINT SHOULD BE AT THE 1ST CHAR OF THE INSTRUCTION.
32570000      % FIRST CHAR NO LONGER IN USE BY INTERPRETER.
32580000 PSCAN:    % NEXT PATTERN ELEMENT
32590000      IF (NEARPOINT←NEARPOINT+RELATIVEPOINTER) ≥ 63 THEN
32600000         BEGIN INSTRUCT ← INSTRUCT + NEARPOINT.W;
32610000         NEARPOINT ← NEARPOINT.C;
32620000         END;
32630000      RELATIVEPOINTER ← 0;
32640000      % CONTROLPOINT SHOULD NOW POINT TO THE NEXT PATTERN ELEMENT.
32650000      MV(3,CONTROLPOINT,AA,2);  % GET NEXT PIECE OF CODE.
32660000      % IF INFORM THEN INFORMA(4,AA);
32670000      CASE AA.C2 OF
32680000         BEGIN
32690000 %*****************************************
32700000 %********OP CODES*************************
32710000 % 0-9: ERRORS
32720000      ;;;;;;;;;;
32730000 %*****************************************
32740000 %********FUNCTION CALL********************
32750000 % 10: "#"  FUNCTION CALL
32760000      % 2-CHAR POINTER TO SYMB TABLE LOC OF FCT
32770000      % 1 CHAR GIVING THE NUMBER OF PARAMETERS.
32780000      BEGIN
32790000      % IF INFORM THEN INFORMO(14);
32800000      MKS ← SP - CHAR(CONTROLPOINT+3); % FIND FIRST PARAM.
32810000      RETURNVAL ← 0; % HOLDS ST LOC OF RETURN VALUE.
32820000      FCT ← AA.[18:12];
32830000      PTYPE[MKS] ← 1;
32840000      FCTV ← VALU[FCT];
32850000 BRANCH:
32860000      IF BOOLEAN(FCTV.INUSE) THEN
32870000         IF NOT TRACEFCTCALL(FCT) THEN GO TO FCTFAIL;
32880000      IF BOOLEAN(FCTV.[1:1]) THEN GO TO DEFINEDFCT;
32890000      % IF INFORM THEN INFORMI(11,FCTV.LINK);
32900000      CASE FCTV.LINK OF
32910000 %*****************************************
32920000 %********INTRINSIC FUNCTIONS**************
32930000 % THE INTRINSIC FUNCTIONS ARE ARRANGED IN THE FOLLOWING ORDER:
32940000 %    1 ANCHOR
32950000 %    2 CALLF
32960000 %    3 CLOSE
32970000 %    4 DATE
32980000 %    5 DEFINE
32990000 %    6 DETACH
33000000 %    7 DUMP
```

```
33010000 %    8 EOF
33020000 %    9 .EQ
33030000 %   10 EQUALS
33040000 %   11 EXECUTE
33050000 %   12 FILE
33060000 %   13 FILL
33070000 %   14 .GE
33080000 %   15 .GT
33090000 %   16 .LE
33100000 %   17 LEVEL
33110000 %   18 LOOK
33120000 %   19 .LT
33130000 %   20 MODE
33140000 %   21 .NE
33150000 %   22 .NUM
33160000 %   23 OPSYN
33170000 %   24 PAGE
33180000 %   25 POP
33190000 %   26 PUSH
33200000 %   27 .REMDR
33210000 %   28 REWIND
33220000 %   29 RULES
33230000 %   30 SIZE
33240000 %   31 SPACE
33250000 %   32 TIME
33260000 %   33 TRACES
33270000 %   34 TRACEF
33280000 %   35 TRACEL
33290000 %   36 TRIM
33300000 %   37 UNANCH
33310000 %   38 UNEQL
33320000 %   39 WAIT
33330000 %   40 SUSPEND
33340000 %   41 STATUS
33350000 %   42 USER
33360000 %   43 SEEK
33370000 %   44 RELEASE
33380000 %   45 RECORD
33390000 %   46 TRACE
33400000 %   47 ASSIGN
33410000 %   48 COPY
33420000 %   49 SEARCH
33430000 %   50 .RANF
33440000 %
33450000     BEGIN % CODE FOR INTRINSIC FUNCTIONS FOLLOWS;
33460000 % 0: UNDEFINED
33470000     GO TO UNDEFFCT;
33480000 % 1: ANCHOR
33490000     BEGIN PTYPE[0] ← 3;
33500000     IF SP > MKS THEN
33510000        BEGIN IF NOT NUMVAL(PST[MKS],I1) OR I1 < 0
33520000           THEN GO TO FCTFAIL;
33530000        PSIZE[0] ← I1;
33540000        END ELSE PSIZE[0] ← ANCHORSIZE;
33550000     GO TO SUCCEED;
33560000     END ANCHOR;
33570000 % 2: CALLF(F,P1,P2,...,PN)  CALLS THE FUNCTION NAMED BY F, WITH THE
```

```
33580000     % PARAMETERS P1, P2, ..., PN.
33590000     BEGIN SEGMENT
33600000     AA ← PST[MKS];
33610000     AB ← VALU[AA];
33620000     FCT ← ENTERST(AB.S,DATA[AB.R,*],AB.CH,"FCT");
33630000     FCTV ← VALU[FCT];
33640000     SP ← SP - 1;
33650000     FOR I ← MKS STEP 1 UNTIL SP DO
33660000        BEGIN PST[I] ← PST[I+1];
33670000        PNAME[I] ← PNAME[I+1];
33680000        END;
33690000     GO TO BRANCH;
33700000     END CALLF;
33710000  % 3: CLOSE(F,P)
33720000     % CLOSE(F,P) CLOSES THE FILE ASSOCIATED WITH $F, IN THE MANNER
33730000     % SPECIFIED BY P. THE VALUES OF P ARE:
33740000     %     "*"  ONLY FOR TAPE FILES.  THE TAPE IS POSITIONED JUST BEYOND
33750000     %        THE END-OF-FILE FOR THE CURRENT FILE.  IF LAST I/O
33760000     %        OPERATION HAD EOF, NO ACTION IS TAKEN.
33770000     %     "LOCK"  CLOSES THE FILE AND SAVES THE UNIT FOR THE PROGRAM.
33780000     %        NEW DISK FILES ARE SAVED FOR THE TIME GIVEN IN THEIR DECLAR-
33790000     %        ATIONS; TAPE FILES ARE REWOUND AND THE OPERATOR IS TOLD
33800000     %        TO REMOVE THE TAPE AND SAVE IT.
33810000     %     "RELEASE"  RELEASES I/O UNIT TO THE SYSTEM.  TAPE FILES ARE
33820000     %        REWOUND AND DISK FILES ARE DESTROYED IF CREATED BY
33830000     %        THE PROGRAM.
33840000     %     "PURGE"  THE FILE IS CLOSED, PURGED, AND RELEASED TO THE
33850000     %        SYSTEM.
33860000     BEGIN SEGMENT
33870000     IF SP ≤ MKS THEN GO TO FCTFAIL;
33880000     IF SP-MKS = 1 THEN IF NOT NULLARGS(1) THEN GO TO FCTFAIL;
33890000     INDIRECT(MKS);
33900000     AA ← PST[MKS];
33910000     AB ← PST[MKS+1];
33920000     AA ← IO[AA];
33930000     IF AA.IOUSE=0 OR I1←AA.FILNO=0 THEN GO TO FCTFAIL;
33940000     AB ← VALU[AB];
33950000     IF (I2←AB.S) > 7 THEN GO TO FCTFAIL;
33960000     AC ← 0;
33970000     IF I2 = 0
33980000        THEN AC ← 0
33990000        ELSE MV(I2,FIRSTCHAR(AB),AC,8-I2);
34000000     AC ← AC;
34010000     IF AC = 0 THEN CLOSE(IOFILE[I1]) ELSE
34020000     IF AC = "LOCK" THEN LOCK(IOFILE[I1],SAVE) ELSE
34030000     IF AC = "SAVE" THEN CLOSE(IOFILE[I1],SAVE) ELSE
34040000     IF AC = "PURGE" THEN CLOSE(IOFILE[I1],PURGE) ELSE
34050000     IF AC = "RELEASE" THEN
34060000        BEGIN IOFILEOPEN[I1] ← FALSE;
34070000        CLOSE(IOFILE[I1],RELEASE);
34080000        END ELSE
34090000     IF AC = "*" THEN CLOSE(IOFILE[I1],*) ELSE
34100000        BEGIN IF PRINTMESSAGES THEN
34110000           BEGIN MESSAGEI(8,INSTNO);
34120000           MESSAGETTI(8,INSTNO);
34130000           END;
34140000        GO TO FCTFAIL;
```

```
34150000       END;
34160000     GO TO SUCCEED;
34170000     END INTRINSIC FUNCTION CLOSE;
34180000  % 4: DATE() RETURNS THE CURRENT DATE IN THE FORM:
34190000     %            MM/DD/YY
34200000     BEGIN SEGMENT
34210000     AA ← DATE;
34220000     AB ← TEMPCELL;
34230000     AC ← STRING(8,AB);
34240000     MV(8,AA,0,FIRSTCHAR(AC));
34250000     VALU[AB] ← AC;
34260000     RETURNVAL ← AB;
34270000     GO TO SUCCEED;
34280000     END LDATE;
34290000  % 5: DEFINE(A,B,C) CREATES A SNOBOL FUNCTION.  THE VALUES OF THE
34300000     % PARAMETERS ARE OF THE FORMS:
34310000     %      A ::= <FCT NAME>  (  <PARAMS>  )
34320000     %      B ::= <LABEL>
34330000     %      C ::= <LOC VARS>
34340000     % <FCT NAME> IS A LEGAL SNOBOL IDENTIFIER,
34350000     % <LABEL> IS A LABEL THAT OCCURS IN THE PROGRAM, OR IS NULL.  IF
34360000     %      B IS NULL, THE ENTRY POINT IS TAKEN TO BE SPELLED THE SAME
34370000     %      AS <FCT NAME>.
34380000     % <PARAMS> AND <LOC VARS> ARE THE FORMAL PARAMETERS AND LOCAL
34390000     %      VARIABLES.  THEY CAN BE NULL, OR CAN CONSIST OF A LIST OF
34400000     %      IDENTIFIERS SEPARATED BY COMMAS.  SEE ALSO THE PROCEDURE
34410000     %      SNBLDEFINE AND THE DEFINE PART OF PROCESSCONTROLCARD.
34420000     BEGIN SEGMENT
34430000     IF SP - MKS < 3 THEN
34440000        IF NOT NULLARGS(3-SP+MKS) THEN GO TO FCTFAIL;
34450000     IF SNBLDEFINE(PST[MKS],PST[MKS+1],PST[MKS+2])
34460000        THEN GO TO SUCCEED;
34470000     MESSAGEI(9,INSTNO);
34480000     MESSAGETTI(9,INSTNO);
34490000     GO TO FCTFAIL;
34500000     END LDEFINE;
34510000  % 6: DETACH(F,V) TERMINATES ANY I/O USE OF THE QUANTITY NAMED BY F.
34520000     % THE TYPE OF THIS QUANTITY IS DETERMINED BY THE FIRST CHARACTER
34530000     % OF V, AS FOLLOWS:
34540000     %      "S" OR NULL = STRING NAME
34550000     %      "F"  = FUNCTION NAME
34560000     %      "L"  = LABEL
34570000     BEGIN SEGMENT
34580000     IF SP ≤ MKS THEN GO TO FCTFAIL;
34590000     IF SP - MKS > 1 THEN
34600000        BEGIN AA ← PST[MKS+1];
34610000        AB ← VALU[AA];
34620000        AC ← CHAR(FIRSTCHAR(AB));
34630000        IF AB.S ¬= 0 OR AC = "S" THEN AC ← "SYMB" ELSE
34640000        IF AC = "F" THEN AC ← "FCT" ELSE
34650000        IF AC = "L" THEN AC ← "INST" ELSE GO TO FCTFAIL;
34660000        END ELSE AC ← "SYMB";
34670000     AA ← PST[MKS];
34680000     AB ← VALU[AA];
34690000     AA ← ENTERST(AB.S,DATA[AB.R,*],AB.CH,AC);
34700000     VALU[AA].IOUSE ← IO[AA] ← 0;
34710000     GO TO SUCCEED;
```

```
34720000     END LDETACH;
34730000 % 7: DUMP()
34740000     BEGIN SEGMENT
34750000     MESSAGEI(10,INSTNO); % DUMP REQUESTED...
34760000     IF INFORM THEN BEGIN WRITEST; WRITEDATA; END;
34770000     STRINGDUMP(INSTNO);
34780000     GO TO SUCCEED;
34790000     END DUMP;
34800000 % 8: EOF(F) SUCCEEDS IF THE LAST I/O OPERATION ON THE FILE ASSOCIATED
34810000     % WITH $F FAILED DUE TO END-OF-FILE.  IF NOT, OR IF $F IS NOT AN
34820000     % I/O STRING, THEN EOF(F) FAILS.
34830000     BEGIN SEGMENT
34840000     IF SP ≤ MKS THEN GO TO FCTFAIL;
34850000     INDIRECT(MKS);
34860000     AA ← VALU[PST[MKS]];
34870000     IF AA.IOUSE = 0 THEN GO TO FCTFAIL;
34880000     GO TO IF IOEOF[AA,FILNO] THEN SUCCEED ELSE FCTFAIL;
34890000     END EOF;
34900000 % 9: .EQ(A,B) SUCCEEDS IFF BOTH A AND B ARE NUMERIC, AND A = B.
34910000     BEGIN SEGMENT
34920000     IF SP - MKS < 2 THEN
34930000        IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
34940000     IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
34950000        THEN GO TO IF AA = AB THEN SUCCEED ELSE FCTFAIL;
34960000     IF PRINTMESSAGES THEN
34970000        BEGIN MESSAGEAI(0,"EQ",INSTNO);
34980000        MESSAGETTAI(0,"EQ",INSTNO);
34990000        END;
35000000     GO TO FCTFAIL;
35010000     END;  % .EQ(A,B)
35020000 % 10: EQUALS(A,B) SUCCEEDS IFF A AND B HAVE THE SAME STRINGS AS VALUES.
35030000     BEGIN
35040000     IF SP - MKS < 2 THEN
35050000        IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
35060000     AB ← PST[MKS];
35070000     AC ← PST[MKS+1];
35080000     AB ← VALU[AB];
35090000     AC ← VALU[AC];
35100000     GO TO IF AB.S ≠ AC.S
35110000        THEN FCTFAIL
35120000        ELSE IF EQUAL(AB.S,FIRSTCHAR(AB),FIRSTCHAR(AC))
35130000           THEN SUCCEED
35140000           ELSE FCTFAIL;
35150000     END LEQUALS;
35160000 % 11: EXECUTE(S) CURRENTLY WORKS IF THE FIRST CHAR OF S IS QMARK, IN
35170000     % WHICH CASE THE ALGOL "ZIP WITH <ARRAY ROW>" IS EXECUTED ON S;
35180000     % OR IF S IS AN INPUT STRING, IN WHICH CASE THE ALGOL "ZIP WITH
35190000     % <FILE PART>" IS EXECUTED ON THE FILE BELONGING TO S, OR IF S
35200000     % STARTS WITH "-", WHICH IS A SNOBOL CONTROL CARD.
35210000     BEGIN SEGMENT
35220000     IF MKS ≥ SP THEN GO TO FCTFAIL;
35230000     AA ← PST[MKS];
35240000     AB ← VALU[AA];
35250000     IF BOOLEAN(AB,INUSE) THEN % ZIP WITH FILE
35260000        BEGIN
35270000        IF BOOLEAN(AB,INUSE) AND AB.FILNO > 0 THEN
35280000           BEGIN
```

```
35290000            ZIP WITH IOFILE[AB,FILNO];
35300000            GO TO SUCCEED;
35310000            END;
35320000         GO TO FCTFAIL;
35330000         END;
35340000      IF (I1←CHAR(FIRSTCHAR(AB))) = QMARK THEN  % MCP CONTROL CARD
35350000         BEGIN ALPHA ARRAY Z[0:AB.[17:10]];
35360000         MOVE(STRINGLOC(AB),Z[*],0);
35370000         ZIP WITH Z[*];
35380000         GO TO SUCCEED;
35390000         END;
35400000      IF I1 = "=" THEN      % SNOBOL CONTROL CARD
35410000         BEGIN ALPHA ARRAY SCRATCH[0:(AB.S),W];
35420000         INSTSIZE ← AB.S;
35430000         MOVE(INSTSIZE,FIRSTCHAR(AB),SCRATCH[0],0);
35440000         SCRATCH[INSTSIZE,W] ← STOPPER;
35450000         PROCESSCONTROLCARD(SCRATCH);
35460000         GO TO SUCCEED;
35470000         END;
35480000      GO TO FCTFAIL;
35490000      END LEXECUTE;
35500000 %12: FILE(NAME,I.O,BUFFERS,REC.SIZE,BUFF.SIZE,SAVE.FACT,DSK.AREAS,SIZE)
35510000      % FILE(...) OPENS A FILE, AS DESCRIBED BY THE PARAMETERS:
35520000      % NAME CONTAINS THE NAME OF THE PRIMARY I/O STRING
35530000      % I.O STARTS WITH "I" FOR INPUT FILES, "O" FOR OUTPUT FILES, AND
35540000      %     IS NULL FOR FILES WITH BOTH INPUT AND OUTPUT USE.
35550000      % BUFFERS IS THE NUMBER OF BUFFER AREAS TO BE USED.
35560000      % REC.SIZE IS THE SIZE (IN CHARACTERS) OF AN I/O RECORD
35570000      % BUFF.SIZE IS THE SIZE (IN CHARACTERS) OF A BUFFER (PHYSICAL RECORD)
35580000      % SAVE.FACT IS THE SAVE FACTOR FOR FILES CREATED BY THE PROGRAM
35590000      % DSK.AREAS IS THE NUMBER OF DISK AREAS FOR NEW DISK FILES (≤20)
35600000      % SIZE IS THE SIZE OF ONE OF THESE DISK AREAS, IN LOGICAL RECORDS.
35610000      BEGIN
35620000      INTEGER
35630000         AREAS,
35640000         ASIZE,
35650000         BLOCKCOUNT,
35660000         BUFFERS,
35670000         BUFSIZE,
35680000         CYC,
35690000         RECSIZE,
35700000         SAVEFACTOR,
35710000         TYPEA,        % "ALGOL" TYPE:  9, 10, OR 11
35720000         TYPES,        % "SNOBOL" TYPE:  1, 2, OR 3
35730000         UNIT;
35740000      MONITOR PRINT (AREAS,ASIZE,BLOCKCOUNT,BUFFERS,BUFSIZE,CYC,RECSIZE,
35750000         IOSIZE,VALTABL,IOUSAGE,
35760000         SAVEFACTOR,TYPEA,TYPES,UNIT,AA,AB,AC,I1,I2);
35770000      %
35780000      GO TO UNDEFFCT; % UNTIL A WAY IS FOUND TO IMPLEMENT FILE DECLARATIONS
35790000      UNIT ← 0;
35800000      CYC ← 16;
35810000      FOR I ← 1 STEP 1 UNTIL FILMAX DO
35820000         IF NOT IOFILEOPEN[I] THEN
35830000            IF IOUSAGE[I].CYCLE < CYC THEN
35840000               BEGIN UNIT ← I; CYC ← IOUSAGE[I].CYCLE; END;
35850000      IF UNIT = 0 THEN
```

```
35860000          BEGIN MESSAGEI(14,INSTNO);  % NO FILE AVAILABLE
35870000          GO TO FCTFAIL;
35880000          END;
35890000       IF SP - MKS < 3 THEN GO TO FCTFAIL;
35900000       %
35910000       AA ← PST[MKS+1];
35920000       AB ← VALU[AA];
35930000       IF AB.S = 0 THEN
35940000          BEGIN TYPEA ← 11;
35950000          TYPES ← 3;
35960000          END ELSE
35970000       IF (I1←CHAR(DATA[AB.R,AB.W],AB.C)) = "I" THEN
35980000          BEGIN TYPEA ← 9;
35990000          TYPES ← 2;
36000000          END ELSE
36010000       IF I1 = "O" THEN
36020000          BEGIN TYPEA ← 10;
36030000          TYPES ← 1;
36040000          END ELSE
36050000          BEGIN  % ERROR: INVALID I/O TYPE
36060000          MESSAGEI(15,INSTNO);
36070000          GO TO FCTFAIL;
36080000          END;
36090000       IF NOT NUMVAL(PST[MKS+2],BUFFERS) OR BUFFERS < 1 THEN
36100000          BEGIN  % ERROR: INVALID BUFFER NUMBER
36110000          MESSAGEI(16,INSTNO);
36120000          GO TO FCTFAIL;
36130000          END;
36140000       IF SP - MKS ≤ 3 THEN RECSIZE ← 0 ELSE
36150000          IF NOT NUMVAL(PST[MKS+3],RECSIZE) OR RECSIZE < 0
36160000                OR RECSIZE > 8182 THEN
36170000             BEGIN  % ERROR: INVALID RECORD SIZE
36180000             GO TO FCTFAIL;
36190000             END;
36200000       IOSIZE[UNIT] ← RECSIZE;
36210000       RECSIZE ← (RECSIZE+7).W;
36220000       IF SP - MKS ≤ 4 THEN BUFSIZE ← 0 ELSE
36230000          IF NOT NUMVAL(PST[MKS+4],BUFSIZE) OR BUFSIZE < 0
36240000                OR BUFSIZE > 8182 THEN
36250000             BEGIN  % ERROR: INVALID BUFFER SIZE
36260000             GO TO FCTFAIL;
36270000             END;
36280000       BUFSIZE ← (BUFSIZE+7).W;
36290000       IF SP - MKS ≤ 5 THEN SAVEFACTOR ← 0 ELSE
36300000          IF NOT NUMVAL(PST[MKS+5],SAVEFACTOR) OR SAVEFACTOR ≤ 0 THEN
36310000             BEGIN  % ERROR: INVALID SAVEFACTOR
36320000             GO TO FCTFAIL;
36330000             END;
36340000       IF SP - MKS ≤ 6 THEN AREAS ← 0 ELSE
36350000          IF NOT NUMVAL(PST[MKS+6],AREAS) OR AREAS < 0
36360000                OR AREAS > 20 THEN
36370000             BEGIN  % ERROR: INVALID NUMBER OF DISK AREAS
36380000             GO TO FCTFAIL;
36390000             END;
36400000       IF SP - MKS ≤ 7 THEN ASIZE ← 0 ELSE
36410000          IF NOT NUMVAL(PST[MKS+7],ASIZE) OR ASIZE < 0 THEN
36420000             BEGIN  % ERROR: INVALID DISK AREA SIZE
```

```
36430000            GO TO FCTFAIL;
36440000            END;
36450000       CYC ← IOUSAGE[UNIT].CYCLE;
36460000       CYC ← CYC + 1;
36470000       % CREATE NEW FILE HERE.
36480000       GO TO SUCCEED;
36490000       END INTRINSIC FILE;
36500000  % 13: FILL(NAME,MFID,FID,MEDIA,DATE,REEL,CYCLE)  FILLS THE FILE
36510000       % $NAME WITH A DESCRIPTION OF AN ACTUAL FILE, AS DESCRIBED BY
36520000       % THE PARAMETERS AS FOLLOWS:
36530000       %      MFID/FID  IS THE EXTERNAL NAME OF THE FILE.  IF EITHER
36540000       %         MFID OR FID CONTAIN MORE THAN 7 CHARS, ONLY THE FIRST 7
36550000       %         WILL BE USED.  IF EITHER IS NULL, "0000000" WILL BE USED.
36560000       %      MEDIA  IS ONE OF THE MEDIA DIGITS LISTED IN THE ALGOL MANUAL.
36570000       %      DATE IS THE DATE THAT SHOULD BE USED WITH OUTPUT FILES,
36580000       %         USUALLY THE CURRENT DATE.  IT CAN BE IN THE FORM MM/DD/YY
36590000       %         OR THE FORM YYDDD (WHICH IS HOW IT IS STORED IN THE
36600000       %         STANDARD FILE LABEL).
36610000       %      REEL IS THE REEL NUMBER, FOR MULTI-REEL TAPES
36620000       %      CYCLE IS THE CYCLE NUMBER, ALSO FOR TAPES MAINLY.
36630000       % IF ANY OF THE PARAMETERS IS MISSING, IT WILL NOT BE ASSIGNED;
36640000       % THUS, MULTIPLE CALLS OF FILL() CAN FILL IN DIFFERENT PIECES OF
36650000       % INFORMATION.  IF FILE() IS TO BE EFFECTIVE, IT MUST BE USED
36660000       % BEFORE ANY I/O IS DONE ON THE FILE.
36670000       BEGIN ALPHA ST, SIO, MFID, FID, D, M, Y;
36680000       INTEGER MED, REEL, CYC;
36690000       IF SP ≤ MKS THEN GO TO FCTFAIL;
36700000       IF SP - MKS < 3 THEN
36710000          IF NOT NULLARGS(3-SP+MKS) THEN GO TO FCTFAIL;
36720000       INDIRECT(MKS);
36730000       ST ← PST[MKS];
36740000       IF VALU[ST].IOUSE = 0 THEN GO TO FCTFAIL;
36750000       SIO ← IO[ST];
36760000       % GET MFID:
36770000       AA ← PST[MKS+1];
36780000       AB ← VALU[AA];
36790000       IF AB.S = 0 THEN MFID ← -1 ELSE
36800000          BEGIN MV(7,BLANKS,1,MFID,1);
36810000          MV(MIN(7,AB.S),FIRSTCHAR(AB),MFID,1);
36820000          END;
36830000       MFID ← MFID;
36840000       % GET FID:
36850000       AA ← PST[MKS+2];
36860000       AB ← VALU[AA];
36870000       IF AB.S = 0 THEN FID ← -1 ELSE
36880000          BEGIN MV(7,BLANKS,1,FID,1);
36890000          MV(MIN(7,AB.S),FIRSTCHAR(AB),FID,1);
36900000          END;
36910000       FID ← FID;
36920000       % GET MEDIA DIGIT:
36930000       IF SP-MKS≤3 OR VALTABL[(AA←PST[MKS+3]).STR,AA.STW].S = 0
36940000          THEN MED ← -1 ELSE
36950000          IF NOT NUMVAL(AA,MED) THEN GO TO FCTFAIL;
36960000       MED ← MED;
36970000       % GET DATE:
36980000       IF SP-MKS ≤ 4 OR VALTABL[(AA←PST[MKS+4]).STR,AA.STW].S = 0
36990000          THEN D ← -1 ELSE
```

```
37000000        BEGIN AA ← PST[MKS+4];
37010000        AB ← VALU[AA];
37020000        IF AB.S = 0 THEN D ← -1 ELSE
37030000        IF AB.S = 5 THEN
37040000           MV(5,FIRSTCHAR(AB),D,3) ELSE
37050000        IF AB.S = 8 THEN
37060000           BEGIN MV(8,FIRSTCHAR(AB),D,0);
37070000           IF CHAR(D,0) > 1 THEN GO TO FCTFAIL;
37080000           IF D.[12:8] ≠ "/" OR D.[30:6] ≠ "/" THEN GO TO FCTFAIL;
37090000           Y ← D.[36:12];
37100000           M ← D.[1:5] × 10 + D.[6:6];
37110000           IF M > 12 THEN GO TO FCTFAIL;
37120000           D ← D.[18:6] × 10 + D.[24:6];
37130000           IF D > 31 THEN GO TO FCTFAIL;
37140000           FOR I ← 1 STEP 1 UNTIL M DO D ← D + MONTHS[I];
37150000           D ← ENTIER(D MOD 10) & Y[18:36:12] &
37160000              ENTIER(D DIV 100)[30:42:6] &
37170000              ENTIER((D MOD 100) DIV 10)[36:42:6];
37180000           END;
37190000        END;
37200000     % FIND REEL NUMBER:
37210000     IF (IF SP=MKS≤5 THEN T ELSE VALTABL[(AA←PST[MKS+5]),STR,AA,STW].S=0)
37220000        THEN REEL ← -1 ELSE
37230000        IF NOT NUMVAL(AA,REEL) THEN GO TO FCTFAIL;
37240000     % FIND CYCLE NUMBER:
37250000     IF (IF SP=MKS≤6 THEN T ELSE VALTABL[(AA←PST[MKS+6]),STR,AA,STW].S=0)
37260000        THEN CYC ← -1
37270000        ELSE IF NOT NUMVAL(AA,CYC) THEN GO TO FCTFAIL;
37280000     % FOUND PARAMETERS; FILL FILE:
37290000     % NOTE THAT A FILL ELEMENT OF -1 MEANS "NO CHANGE".
37300000     FILL IOFILE[SIO,FILNO] WITH MFID, FID, REEL, D, CYC, MED;
37310000     GO TO SUCCEED;
37320000     END FILL;
37330000  % 14: .GE(A,B) SUCCEEDS IFF A AND B ARE NUMERIC AND A ≥ B.
37340000     BEGIN SEGMENT
37350000     IF SP - MKS < 2 THEN
37360000        IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
37370000     IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
37380000        THEN GO TO IF AA ≥ AB THEN SUCCEED ELSE FCTFAIL;
37390000     IF PRINTMESSAGES THEN
37400000        BEGIN MESSAGEAI(0,"GE",INSTNO);
37410000        MESSAGETTAI(0,"GE",INSTNO);
37420000        END;
37430000     GO TO FCTFAIL;
37440000     END LGE;
37450000  % 15: .GT(A,B)
37460000     % .GT(A,B) SUCCEEDS IFF A AND B ARE BOTH NUMERIC AND A > B.
37470000     BEGIN SEGMENT
37480000     IF SP - MKS < 2 THEN
37490000        IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
37500000     IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
37510000        THEN GO TO IF AA > AB THEN SUCCEED ELSE FCTFAIL;
37520000     IF PRINTMESSAGES THEN
37530000        BEGIN MESSAGEAI(0,"GT",INSTNO);
37540000        MESSAGETTAI(0,"GT",INSTNO);
37550000        END;
37560000     GO TO FCTFAIL;
```

```
37570000     END LGT;
37580000 % 16: .LE(A,B)
37590000     % .LE(A,B) SUCCEEDS IFF A AND B ARE BOTH NUMERIC AND A ≤ B.
37600000     BEGIN SEGMENT
37610000     IF SP - MKS < 2 THEN IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
37620000     IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
37630000        THEN GO TO IF AA ≤ AB THEN SUCCEED ELSE FCTFAIL;
37640000     IF PRINTMESSAGES THEN
37650000        BEGIN MESSAGEAI(0,"LE",INSTNO);
37660000        MESSAGETTAI(0,"LE",INSTNO);
37670000        END;
37680000     GO TO FCTFAIL;
37690000     END LLE;
37700000 % 17: LEVEL()
37710000     % LEVEL() RETURNS THE LEVEL AT WHICH THE PROGRAM IS CURRENTLY
37720000     % OPERATING.  THE LEVEL IS DEFINED AS FOLLOWS:  IT IS 1 AT THE
37730000     % START OF THE PROGRAM; IT INCREASES BY 1 WITH EACH CALL ON A
37740000     % PROGRAMMER-DEFINED FUNCTION; AND IT DECREASES BY 1 AT EACH
37750000     % RETURN (OR FRETURN).
37760000     BEGIN SEGMENT
37770000     RETURNVAL ← TEMPVAL(LEVEL);
37780000     GO TO SUCCEED;
37790000     END;  % LEVEL()
37800000 % 18: LOOK(S,F)
37810000     % LOOK(S,F) TURNS THE STRING $S INTO A NON-READINT I/O STRING FOR
37820000     % THE INPUT FILE BELONGING TO THE STRING $F.  IF $F IS NOT AN INPUT
37830000     % STRING (NOT DATACOMM), LOOK FCTFAILS.
37840000     BEGIN SEGMENT
37850000     IF SP - MKS < 2 THEN GO TO FCTFAIL;
37860000     INDIRECT(MKS);
37870000     AA ← PST[MKS];
37880000     INDIRECT(MKS+1);
37890000     AB ← PST[MKS+1];
37900000     AC ← IOTABL[AB.STR,AB.STW];
37910000     IF NOT BOOLEAN(AC.INUSE) OR AC.FILNO=0 THEN GO TO FCTFAIL;
37920000     VALU[AA].IOUSE ← 2;
37930000     IO[AA] ← 0 & 2 CIOUSE & 2 CIOTYPE & AC TFILNO;
37940000     GO TO SUCCEED;
37950000     END;
37960000 % 19: .LT(A,B)
37970000     % .LT(A,B) SUCCEEDS IFF A AND B ARE BOTH NUMERIC AND A < B.
37980000     BEGIN SEGMENT
37990000     IF SP-MKS < 2 THEN IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
38000000     IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
38010000        THEN GO TO IF AA < AB THEN SUCCEED ELSE FCTFAIL;
38020000     IF PRINTMESSAGES THEN
38030000        BEGIN MESSAGEAI(0,"LT",INSTNO);
38040000        MESSAGETTAI(0,"LT",INSTNO);
38050000        END;
38060000     GO TO FCTFAIL;
38070000     END LT;
38080000 % 20: MODE(V)
38090000     % MODE(V) SETS SYSTEM VARIABLE, DEPENDING ON THE CONTENTS OF V.
38100000     % THE VALUES CURRENTLY RECOGNIZED FOR V ARE:
38110000     %      "DUMP"  EXECUTE DUMP() AT END-OF-JOB.
38120000     %      "ROUND"  ROUND QUOTIENTS OF DIVISIONS (STANDARD).
38130000     %      "INTEGER"  DIVISION FCTFAILS IF REMAINDER ≠ 0.
```

```
38140000    %     "TRUNCATION"  TRUNCATE QUOTIENTS.
38150000    %     "ANCHOR"  ALL SCANS ARE ANCHORED UNLESS UNANCH() IS USED.
38160000    %     "UNANCH"  ALL SCANS ARE UNANCHORED, UNLESS ANCHOR() IS USED
38170000    %     "UNANCHOR"  SAME AS "UNANCH" (STANDARD).
38180000    %     "INFORM"  TURN ON SYSTEM DEBUGGING OUTPUT.
38190000    %     "SILENCE"  TURN OFF DEBUGGING OUTPUT (STANDARD).
38200000    BEGIN ALPHA ARRAY WORD[0:1];
38210000    IF SP ≤ MKS THEN GO TO FCTFAIL;
38220000    AA ← PST[MKS];
38230000    AB ← VALU[AA];
38240000    IF (I1←AB.S) < 3 OR I1 > 10 THEN GO TO FCTFAIL;
38250000    MV(I1,FIRSTCHAR(AB),WORD[*],0);
38260000    CASE I1 OF
38270000       BEGIN ;;; % NONE OF SIZE 0, 1, OR 2.
38280000       IF EQ(3,WORD[*],0,WORDS[13],5) THEN
38290000          BEGIN DEBUGGING ← FALSE;       % RUN
38300000          SETLIMITFLAG;
38310000          GO TO SUCCEED;
38320000          END;
38330000       IF EQUAL(4,WORD[*],0,WORDS[1],3) THEN
38340000          BEGIN DMPSTR ← TRUE;           % DUMP
38350000          GO TO SUCCEED;
38360000          END;
38370000       IF EQUAL(5,WORD[*],0,WORDS[0],7) THEN
38380000          BEGIN DIVIDEMODE ← 0;          % ROUND
38390000          GO TO SUCCEED;
38400000          END ELSE
38410000       IF EQ(5,WORD[*],0,WORDS[13],0) THEN
38420000          BEGIN DEBUGGING ← TRUE;        % DEBUG
38430000          SETLIMITFLAG;
38440000          GO TO SUCCEED;
38450000          END;
38460000       IF EQUAL(6,WORD[*],0,WORDS[0],2) THEN
38470000          BEGIN IF SP - MKS > 1 THEN  % ANCHOR
38480000             BEGIN IF NOT NUMVAL(PST[MKS+1],I2) OR I2 < 0 OR I2 > 8191
38490000                THEN GO TO FCTFAIL;
38500000             ANCHORSIZE ← I2;
38510000             END;
38520000          ANCHORMODE ← 3;
38530000          GO TO SUCCEED;
38540000          END ELSE
38550000       IF EQUAL(6,WORD[*],0,WORDS[0],0) THEN
38560000          BEGIN ANCHORMODE ← 2;          % UNANCH
38570000          ANCHORSIZE ← 0;
38580000          GO TO SUCCEED;
38590000          END ELSE
38600000       IF EQUAL(6,WORD[*],0,WORDS[4],6) THEN
38610000          BEGIN PRINTMESSAGES ← TRUE;    % INFORM
38620000          INFORM ← SYSTEMDEBUGGING;
38630000          GO TO SUCCEED;
38640000          END;
38650000       IF EQUAL(7,WORD[*],0,WORDS[2],0) THEN
38660000          BEGIN DIVIDEMODE ← 2;          % INTEGER
38670000          GO TO SUCCEED;
38680000          END ELSE
38690000       IF EQUAL(7,WORD[*],0,WORDS[5],4) THEN
38700000          BEGIN                          % SILENCE
```

```
38710000            PRINTMESSAGES ← INFORM ← FALSE;
38720000            GO TO SUCCEED;
38730000            END ELSE
38740000         IF EQ(7,WORD[*],0,WORDS[14],0) THEN
38750000            BEGIN ERRDUMP ← TRUE;          % ERRDUMP
38760000            GO TO SUCCEED;
38770000            END;
38780000         IF EQUAL(8,WORD[*],0,WORDS[0],0) THEN
38790000            BEGIN ANCHORMODE ← 2;          % UNANCHOR
38800000            ANCHORSIZE ← 0;
38810000            GO TO SUCCEED;
38820000            END ELSE
38830000         IF EQ(8,WORD[*],0,WORDS[14],7) THEN   % TRUNCATE, FILE
38840000            BEGIN IF SP < MKS+2 THEN GO TO FCTFAIL;
38850000            INDIRECT(MKS+2);
38860000            AA ← PST[MKS+1];
38870000            AB ← IO[AA];
38880000            IF BOOLEAN(AB.OUTUSE) THEN IF AB.FILNO ≠ 0 THEN
38890000               BEGIN IO[AA].OVFL ← 0;
38900000               GO TO SUCCEED;
38910000               END;
38920000            GO TO FCTFAIL;
38930000            END ELSE
38940000         IF EQ(8,WORD[*],0,WORDS[15],7) THEN   % OVERFLOW, FILE
38950000            BEGIN IF SP < MKS+2 THEN GO TO FCTFAIL;
38960000            INDIRECT(MKS+1);
38970000            AA ← PST[MKS+1];
38980000            AB ← IO[AA];
38990000            IF BOOLEAN(AB.OUTUSE) THEN IF AB.FILNO ≠ 0 THEN
39000000               BEGIN IO[AA].OVFL ← 1;
39010000               GO TO SUCCEED;
39020000               END;
39030000            GO TO FCTFAIL;
39040000            END;
39050000         ;  % NONE WITH 9 CHARS
39060000         IF EQUAL(10,WORD[*],0,WORDS[2],7) THEN
39070000            BEGIN DIVIDEMODE ← 1;          % TRUNCATION
39080000            GO TO SUCCEED;
39090000            END;
39100000         END;
39110000      GO TO FCTFAIL;
39120000      END;  % MODE(V)
39130000 % 21: .NE(A,B)
39140000      % .NE(A,B) SUCCEEDS IFF A AND B ARE BOTH NUMERIC AND A ≠ B.
39150000      BEGIN SEGMENT
39160000      IF SP-MKS < 2 THEN IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
39170000      IF NUMVAL(PST[MKS],AA) AND NUMVAL(PST[MKS+1],AB)
39180000         THEN GO TO IF AA ≠ AB THEN SUCCEED ELSE FCTFAIL;
39190000      IF PRINTMESSAGES THEN
39200000         BEGIN MESSAGEAI(0,"NE",INSTNO);
39210000         MESSAGETTAI(0,"NE",INSTNO);
39220000         END;
39230000      GO TO FCTFAIL;
39240000      END LNE;
39250000 % 22: .NUM(A)
39260000      % .NUM(A) SUCCEEDS IF A IS NUMERIC; THAT IS, IF A IS NULL, OR
39270000      % A IS NOT NULL AND THE FIRST CHAR OF A IS "-" OR A DIGIT AND
```

```
39280000    % THE OTHER CHARS OF A ARE ALL DIGITS.
39290000    BEGIN SEGMENT
39300000    IF MKS = SP THEN GO TO SUCCEED;
39310000    GO TO IF NUMVAL(PST[MKS],AA) THEN SUCCEED ELSE FCTFAIL;
39320000    END LNUM;
39330000 % 23: OPSYN(S1,S2,T)
39340000    % OPSYN(S1,S2,T) CAUSES THE ALTERING OF THE SYMBOL TABLE ENTRY
39350000    % DESCRIBED BY S1 AND T; THE QUANTITY NAMED BY S1 IS SET TO BE
39360000    % IDENTICAL TO THE QUANTITY NAMED BY S2.  THE TYPES OF THE QUANTITIES
39370000    % ARE DETERMINED BY T AS FOLLOWS:
39380000    %     T IS NULL:  S1 AND S2 NAME FUNCTIONS.
39390000    %     T STARTS WITH "F":  S1 AND S2 NAME FUNCTIONS.
39400000    %     T STARTS WITH "S":  S1 AND S2 NAME STRINGS.
39410000    %     T STARTS WITH "L":  S1 AND S2 NAME LABELS.
39420000    % NOT ONLY THE VALUE OF THE QUANTITY NAMED BY S2 IS COPIED, BUT
39430000    % ALSO ANY I/O USE IT MAY HAVE.  THUS, OPSYN("A","B","S") CREATES
39440000    % A STRING NAMED "A" (IF ONE DOES NOT ALREADY EXIST), AND ASSIGNS
39450000    % TO A THE VALUE AND I/O USE OF B.  OPSYN("F","ARGH","F") CREATES
39460000    % A FUNCTION NAMED "F", WHICH IS IDENTICAL TO THE FUNCTION ARGH.
39470000    % ANY FORMER FUNCTION NAMED "F" IS LOST.  SIMILARLY, OPSYN("L1",
39480000    % ,"L2","L") CREATES A LABEL NAMED L1, WHICH IS THE SAME STATEMENT
39490000    % AS L2.  ANY TRANSFER TO L1 WILL THEN RESULT IN A TRANSFER TO
39500000    % A STATEMENT THAT IS THE SAME AS L2.
39510000    BEGIN ALPHA I, TYPE;
39520000    IF SP-MKS < 3 THEN IF NOT NULLARGS(3-SP+MKS) THEN GO TO FCTFAIL;
39530000    AA ← PST[MKS+2];
39540000    AA  ← VALU[AA];
39550000    IF AA.S = 0 THEN TYPE ← "FCT" ELSE
39560000       BEGIN AA ← CHAR(FIRSTCHAR(AA));
39570000       TYPE ← IF AA = "S" THEN "SYMB" ELSE      % "S"  STRING NAME
39580000              IF AA = "F" THEN "FCT"  ELSE      % "F"  FUNCTION
39590000              IF AA = "L" THEN "INST" ELSE 0;   % "L"  LABEL
39600000       IF TYPE = 0 THEN GO TO FCTFAIL;
39610000       END;
39620000    AA ← PST[MKS];
39630000    AB ← VALU[AA];
39640000    AB ← ENTERST(AB,S,DATA[AB,R,*],AB,CH,TYPE);
39650000    AA ← PST[MKS + 1];
39660000    AC ← VALU[AA];
39670000    AC ← ENTERST(AC,S,DATA[AC,R,*],AC,CH,TYPE);
39680000    AA ← VALU[AC];
39690000    I ← STRING(AA,S,AB);
39700000    MOVE(I.S,FIRSTCHAR(AA),FIRSTCHAR(I));
39710000    VALU[AB] ← I & AA[1:1:16];
39720000    IO[AB] ← IO[AC];
39730000    GO TO SUCCEED;
39740000    END LOPSYN;
39750000 % 24: PAGE(N)
39760000    % PAGE(N) EJECTS THE LINE PRINTER FILE PRINT TO CHANNEL N.  IF N < 0
39770000    % OR N > 11, PAGE FAILS.  IF N = 0, CHANNEL 1 ( THE TOP OF THE NEXT
39780000    % PAGE) IS USED.
39790000    BEGIN SEGMENT
39800000    IF SP > MKS+1 THEN   % OUTPUT FILE AS 2ND PARAMETER
39810000       BEGIN INDIRECT(MKS+1);
39820000       AB ← PST[MKS+1];
39830000       IF VALU[AB].IOUSE = 0 THEN GO TO FCTFAIL ELSE
39840000          IF I2←IO[AB].FILNO=0 THEN GO TO FCTFAIL;
```

```
39850000        END ELSE I2 ← 3;  % ASSUME PRINT.
39860000     IF SP = MKS THEN I1 ← 1 ELSE
39870000        BEGIN AA ← PST[MKS];
39880000        IF NOT NUMVAL(AA,I1) OR I1<0 OR I1>11 THEN GO TO FCTFAIL;
39890000        IF I1 = 0 THEN I1 ← 1;
39900000        END;
39910000     RECORD[I2] ← I1;
39920000     GO TO SUCCEED;
39930000     END LPAGE;
39940000 % 25: POP(S)
39950000     % POP(S) POPS THE VARIABLE S--I. E., THE TOP VALUE ON THE PUSH-DOWN
39960000     % STACK OF S IS REMOVED.  IF S IN NOT A NAMED STRING OR IF S CANT
39970000     % BE POPPED (ONLY ONE VALUE IN ITS STACK), POP FAILS.  IF S IS AN
39980000     % OUTPUT STRING, OUTPUT OCCURS.
39990000     BEGIN SEGMENT
40000000     IF SP ≤ MKS THEN GO TO FCTFAIL;
40010000     IF NOT PNAME[MKS] THEN GO TO FCTFAIL;
40020000     GO TO IF POP(AA←PST[MKS])
40030000        THEN IF BOOLEAN(VALU[AA].OUTUSE)
40040000           THEN IF SNBLOUT(AA)
40050000              THEN SUCCEED
40060000              ELSE FCTFAIL
40070000           ELSE SUCCEED
40080000        ELSE FCTFAIL;
40090000     END LPOP;
40100000 % 26: PUSH(S,V)
40110000     % PUSH(S,V) PUSHES THE VALUE OF V ONTO THE TOP OF THE PUSH-DOWN
40120000     % STACK OF THE VARIABLE S.  IF POP(S) IS THEN EXECUTED, S RETURNS
40130000     % TO ITS FORMER VALUE.  PUSH FAILS IF S IS NOT A NAMED STRING, OR
40140000     % IF THE SYMBOL TABLE IS FULL (A FATAL ERROR).  IF S IS AN OUTPUT
40150000     % STRING, OUTPUT OCCURS.
40160000     BEGIN SEGMENT
40170000     IF SP ≤ MKS THEN GO TO FCTFAIL;
40180000     IF NOT PNAME[MKS] THEN GO TO FCTFAIL;
40190000     GO TO IF PUSH(AA←PST[MKS],IF SP-MKS ≤ 1 THEN 0 ELSE PST[MKS+1])
40200000        THEN IF BOOLEAN(VALU[AA].OUTUSE)
40210000           THEN IF SNBLOUT(AA)
40220000              THEN SUCCEED
40230000              ELSE FCTFAIL
40240000           ELSE SUCCEED
40250000        ELSE FCTFAIL;
40260000     END LPUSH;
40270000 % 27: .REMDR(A,B)
40280000     % .REMDR(A,B) RETURNS THE QUOTIENT OF A / B.  IF EITHER A OR B
40290000     % IS NONNUMERIC, OR B = 0, .REMDR FAILS.
40300000     BEGIN SEGMENT
40310000     IF SP-MKS < 2 THEN GO TO FCTFAIL;
40320000     AA ← PST[MKS];
40330000     AB ← PST[MKS+1];
40340000     IF NOT(NUMVAL(AA,AA) AND NUMVAL(AB,AB)) THEN GO TO FCTFAIL;
40350000     IF AB = 0 THEN GO TO FCTFAIL;
40360000     RETURNVAL ← TEMPVAL(AA - (AA DIV AB) × AB);
40370000     GO TO SUCCEED;
40380000     END LREMDR;
40390000 % 28: REWIND(F)
40400000     % REWIND(F) CAUSES THE FILE ASSOCIATED WITH $F TO BE REWOUND.
40410000     % IF $F ISNT AN I/O STRING, REWIND FAILS.
```

```
40420000     BEGIN SEGMENT
40430000     IF SP ≤ MKS THEN GO TO FCTFAIL;
40440000     INDIRECT(MKS);
40450000     AB ← IO[PST[MKS]];
40460000     IF AB.IOUSE = 0 OR AB.IOTYPE ≠ 1 THEN GO TO FCTFAIL;
40470000     REWIND(IOFILE[AB.FILNO]);
40480000     GO TO SUCCEED;
40490000     END INTRINSIC FUNCTION REWIND;
40500000 %29: RULES(L)
40510000     % RULES(L) RETURNS THE NUMBER OF RULES EXECUTED AT LEVEL L SINCE
40520000     % THE LAST TIME LEVEL L WAS REACHED.  IF L = 0 (OR IS NULL), THE
40530000     % TOTAL FOR THE PROGRAM IS RETURNED.  IF L ≤ LEVEL(), THE CURRENT
40540000     % RULE BEING EXECUTED IS INCLUDED IN THE TOTAL.  RULES WILL FAIL IF
40550000     % L < 0 OR L > MAXLEVEL.  NOTE THAT IF RULES(L) = "0", THEN THE
40560000     % LEVEL L HAS NEVER BEEN REACHED.
40570000     BEGIN SEGMENT
40580000     AA ← IF SP > MKS THEN PST[MKS] ELSE 0;
40590000     IF AA = 0
40600000        THEN I1 ← 0
40610000        ELSE IF NOT NUMVAL(AA,I1) THEN GO TO FCTFAIL;
40620000     IF I1 < 0 THEN GO TO FCTFAIL;
40630000     RETURNVAL ← TEMPVAL(IF I1 > MAXLEVEL THEN 0 ELSE RULES[I]);
40640000     GO TO SUCCEED;
40650000     END LRULES;
40660000 % 30: SIZE(S)
40670000     % SIZE(S) RETURNS THE NUMBER OF CHARACTERS IN S.
40680000     BEGIN
40690000     RETURNVAL ← TEMPVAL(IF SP ≤ MKS THEN 0 ELSE VALU[PST[MKS]],S);
40700000     GO TO SUCCEED;
40710000     END LSIZE;
40720000 % 31: SPACE(F,N)  SETS A COUNTER FOR THE I/O FILE BELONGING TO $F,
40730000     % SO THAT ANY FURTHER I/O OPERATIONS ON THIS FILE WILL BE
40740000     % PRECEDED BY A SKIPPING OF N RECORDS.  SPACE FAILS IN THE
40750000     % FOLLOWING SITUATIONS:
40760000     %     N NON-NUMERIC.
40770000     %     $F NOT AN I/O STRING.
40780000     %     $F A DATACOMM I/O STRING.
40790000     % NOTE THAT FOR MOST FILES, N < 0 IS MEANINGLESS.  ALSO, N = 0
40800000     % FOR LINE PRINTER FILES CAUSES OVERPRINTING.
40810000     BEGIN SEGMENT
40820000     IF SP ≤ MKS+1 THEN IF NOT NULLARGS(2) THEN GO TO FCTFAIL;
40830000     INDIRECT(MKS);
40840000     AB ← PST[MKS];
40850000     AB ← IO[AB];
40860000     IF AB.IOUSE = 0 OR AB.FILNO = 0 THEN GO TO FCTFAIL;
40870000     IF NOT NUMVAL(PST[MKS+1],I1) THEN GO TO FCTFAIL;
40880000     IOSPACE[AB.FILNO] ← I1;
40890000     GO TO SUCCEED;
40900000     END LSPACE;
40910000 % 32: TIME(N)
40920000     BEGIN ALPHA ARRAY NUM[0:1];
40930000     STREAM PROCEDURE TEMPS(TH,TM,TS,W);
40940000        VALUE TH,TM,TS;
40950000        BEGIN DI ← W; DS ← 8 LIT " ";
40960000        SI ← LOC TH; DS ← 2 DEC; DS ← 1 LIT ":";
40970000        SI ← LOC TM; DS ← 2 DEC; DS ← 1 LIT ":";
40980000        SI ← LOC TS; DS ← 2 DEC;
```

```
40990000        END TEMPS;
41000000    INTEGER TH,TM,TS;
41010000    IF SP = MKS THEN
41020000        IF NOT NULLARGS(1) THEN GO TO FCTFAIL;
41030000    AB ← PST[MKS];
41040000    IF NOT NUMVAL(AB,AA) THEN GO TO FCTFAIL;
41050000    IF AA < 0 OR AA > 4 THEN GO TO FCTFAIL;
41060000    IF AA = 0 THEN
41070000        BEGIN
41080000        AA ← TIME(1) / 60;
41090000        TH ← ENTIER(TEMP/3600);
41100000        TM ← ENTIER((TEMP-TH×3600)/60);
41110000        TS ← AA - TH×3600 - TM×60;
41120000        TEMPS(TH,TM,TS,NUM[*]);
41130000        END ELSE
41140000    IF AA = 4 THEN
41150000        BEGIN CLEAR(NUM,2);
41160000        I1 ← TIME(4);
41170000        MV(1,I1,7,NUM[*],15);
41180000        END ELSE
41190000        BEGIN RETURNVAL ← TEMPVAL(TIME(AA));
41200000        GO TO SUCCEED;
41210000        END;
41220000    AA ← 16 - SKIPCHAR(" ",NUM[*],0);
41230000    RETURNVAL ← TEMPCELL;
41240000    AC ← STRING(AA,RETURNVAL);
41250000    MOVE(AA,NUM[*],16-AA,FIRSTCHAR(AC));
41260000    VALU[RETURNVAL] ← AC;
41270000    GO TO SUCCEED;
41280000    END LTIME;
41290000 % 33: TRACES(S1,S2,S3,...)  TURNS ON TRACING FOR THE STRINGS
41300000    % NAMED BY S1, S2, S3, ETC.  ALL OUTPUT IS TO THE STANDARD
41310000    % OUTPUT FILE, PRINT.  WHENEVER ANY OF THE STRINGS IS ALTERED,
41320000    % OUTPUT OCCURS GIVING THE STATEMENT NUMBER, THE STRING NAME, AND
41330000    % THE NEW VALUE.
41340000    BEGIN SEGMENT
41350000    FOR I ← MKS STEP 1 UNTIL SP-1 DO
41360000        BEGIN INDIRECT(I);
41370000        AA ← PST[I];
41380000        VALU[AA].IOUSE ← 1;
41390000        IO[AA] ← 0 & 3 CFILNO & 2 CIOTYPE & 1 CIOUSE;
41400000        END;
41410000    GO TO SUCCEED;
41420000    END TRACES;
41430000 % 34: TRACEF(F1,F2,F3,...)  TURNS ON TRACING FOR THE FUNCTIONS
41440000    % WHOSE NAMES ARE CONTAINED IN F1, F2, F3, ETC.  AFTER THIS,
41450000    % UNLESS DETACH() IS USED, EACH CALL OF ANY OF THESE FUNCTIONS
41460000    % WILL PRODUCE OUTPUT ON THE STANDARD OUTPUT FILE, PRINT, GIVING
41470000    % THE INSTRUCTION NUMBER, THE FUNCTION NAME, AND THE VALUES OF
41480000    % ALL THE PARAMETERS.
41490000    BEGIN SEGMENT
41500000    FOR I ← MKS STEP 1 UNTIL SP-1 DO
41510000        BEGIN AA ← PST[I];
41520000        AB ← VALU[AA];
41530000        AC ← ENTERST(AB.S,DATA[AB.R,*],AB.CH,"FCT");
41540000        VALU[AC].IOUSE ← 3;
41550000        IO[AC] ← 0 & 2 CIOTYPE & 3 CFILNO & 3 CIOUSE;
```

```
41560000        END;
41570000     GO TO IF DEATH THEN FCTFAIL ELSE SUCCEED;
41580000     END TRACEF;
41590000  % 35: TRACEL(L1,L2,L3,...)  TURNS ON TRACING FOR THE LABELS WHOSE
41600000     % NAMES ARE CONTAINED IN L1, L2, L3, ETC.  AFTER THIS, UNLESS
41610000     % DETACH() IS USED, EACH TIME ANY OF THESE LABELS IS ENCOUNTERED,
41620000     % OUTPUT ON THE STANDARD OUTPUT FILE, PRINT, WILL OCCUR, GIVING
41630000     % THE LABEL AND THE NUMBER OF TIMES IS HAS BEEN ENCOUNTERED SO
41640000     % FAR DURING THE PROGRAM"S EXECUTION.
41650000     BEGIN SEGMENT
41660000     FOR I ← MKS STEP 1 UNTIL SP-1 DO
41670000        BEGIN AA ← PST[I];
41680000        AB ← VALU[AA];
41690000        AC ← ENTERST(AB.S,DATA[AB.R,*],AB.CH,"INST");
41700000        VALU[AC].OUTUSE ← 1;
41710000        IO[AC] ← 0 & 2 CIOTYPE & 3 CFILNO & 1 CIOUSE;
41720000        END;
41730000     GO TO IF DEATH THEN FCTFAIL ELSE SUCCEED;
41740000     END TRACEL;
41750000  % 36: TRIM(S)  RETURNS THE VALUE OF S WITH ALL TRAILING BLANKS DELETED.
41760000     BEGIN STREAM PROCEDURE TRIM(L,INC,SIZE); VALUE INC;
41770000        BEGIN LOCAL S5, S6, S7, TEMP;
41780000        LABEL TEST, RETURN;
41790000        SI ← SIZE; SI ← SI + 5;
41800000        DI ← LOC S5; DI ← DI + 7; DS ← 1 CHR;
41810000        DI ← LOC S6; DI ← DI + 7; DS ← 1 CHR;
41820000        DI ← LOC S7; DI ← DI + 7; DS ← 1 CHR;
41830000        SI ← L; SI ← SI + INC;
41840000        S5(4(32(SI ← SI + 32)));
41850000        S6(2(SI ← SI + 32));
41860000        SI ← SI + S7;
41870000        TEST: SI ← SI - 1;
41880000           IF SC ≠ " " THEN GO TO RETURN;
41890000           TEMP ← SI;
41900000           SI ← LOC S7; SI ← SI + 7;
41910000           IF SC = "0" THEN
41920000              BEGIN
41930000              SI ← LOC S6; SI ← SI + 7;
41940000              IF SC = "0" THEN
41950000                 BEGIN
41960000                 SI ← LOC S5; SI ← SI + 7;
41970000                 IF SC = "0" THEN GO TO RETURN;
41980000                 TALLY ← S5; TALLY ← TALLY + 63; S5 ← TALLY;
41990000                 END;
42000000              TALLY ← S6; TALLY ← TALLY + 63; S6 ← TALLY;
42010000              END;
42020000           TALLY ← S7; TALLY ← TALLY + 63; S7 ← TALLY;
42030000           SI ← TEMP; GO TO TEST;
42040000        RETURN: DI ← SIZE; DI ← DI + 5;
42050000           SI ← LOC S5; SI ← SI + 7; DS ← 1 CHR;
42060000           SI ← LOC S6; SI ← SI + 7; DS ← 1 CHR;
42070000           SI ← LOC S7; SI ← SI + 7; DS ← 1 CHR;
42080000        END TRIM;
42090000     IF SP ≤ MKS THEN GO TO SUCCEED;
42100000     AC ← PST[MKS];
42110000     AA ← VALU[AC];
42120000     AB ← AA.S;
```

```
42130000     TRIM(FIRSTCHAR(AA),AB);
42140000     AC ← TEMPCELL;
42150000     AB ← STRING(AB,AC);
42160000     VALU[AC] ← AB;
42170000     MOVE(AB.S,FIRSTCHAR(AA),FIRSTCHAR(AB));
42180000     RETURNVAL ← AC;
42190000     GO TO SUCCEED;
42200000     END LTRIM;
42210000 % 37: UNANCH()
42220000     BEGIN PTYPE[0] ← 2;
42230000     PSIZE[0] ← 0;
42240000     GO TO SUCCEED;
42250000     END UNANCH;
42260000 % 38: UNEQL(A,B)
42270000     BEGIN SEGMENT
42280000     IF SP-MKS < 2 THEN IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
42290000     AB ← PST[MKS];
42300000     AC ← PST[MKS+1];
42310000     AB ← VALU[AB];
42320000     AC ← VALU[AC];
42330000     GO TO IF AB.S ≠ AC.S
42340000        THEN SUCCEED
42350000        ELSE IF EQUAL(AB.S,FIRSTCHAR(AB),FIRSTCHAR(AC))
42360000           THEN FCTFAIL
42370000           ELSE SUCCEED;
42380000     END LUNEQL;
42390000 % 39: WAIT(T)  SETS THE WAITING TIME FOR TELETYPE I/O TO T SECONDS.
42400000     % IF T IS NON-NUMERIC OR < 0, WAIT(T) FAILS.
42410000     BEGIN SEGMENT
42420000     RETURNVAL ← TEMPVAL(WAITTIME/60);  % RETURN PREVIOUS WAITTIME.
42430000     IF SP > MKS THEN IF VALU[PST[MKS]].S > 0 THEN % GET NEW WAITTIME
42440000        BEGIN IF NOT NUMVAL(PST[MKS],AB) OR AB < 0 THEN GO TO FCTFAIL;
42450000        WAITTIME ← AB × 60;
42460000        END;
42470000     GO TO SUCCEED;
42480000     END WAIT;
42490000 % 40: SUSPEND(T)  SUSPENDS THE PROGRAM FOR T SECONDS.  IF T IS MISSING,
42500000     % AND THERE IS A TELETYPE ATTACHED, DEBUG IS CALLED.
42510000     BEGIN LABEL NEWINST;
42520000     IF SP > MKS THEN
42530000        BEGIN IF NOT NUMVAL(PST[MKS],AA) THEN GO TO FCTFAIL;
42540000        IF AA < 0 THEN GO TO FCTFAIL;
42550000        IF AA > 0 THEN
42560000              BEGIN WHEN(AA);
42570000              GO TO SUCCEED;
42580000              END;
42590000        END;
42600000     IF NOT DATACOMF THEN FINDUSERS;
42610000     IF DATACOMF THEN
42620000        BEGIN SUSPENDREASON[0] ← TRUE;
42630000        DEBUG(NEWINST);
42640000        END;
42650000     GO TO IF DEATH THEN FCTFAIL ELSE SUCCEED;
42660000 NEWINST:
42670000     INST[0] ← AA & "!"[30:42:6];   % BUILD GO-TO PART
42680000     FLOC ← 5;
42690000     ENTRY ← 0;  % SINCE INST[*] HAS BEEN ALTERED.
```

```
42700000     GO TO FCTFAIL; % TERMINATE CURRENT INSTRUCTION
42710000     END SUSPEND;
42720000  % 41: STATUS(A,I)  RETURNS THE STATUS OF ATTACHED TELETYPES, AS
42730000     % GIVEN BY A, WHICH IS EITHER NULL OR A TELETYPE ADDRESS.  I IS THE
42740000     % INTEGER IN THE ALGOL STATUS(S,I) FUNCTION.  IF A IS NULL, THE
42750000     % STATUS OF ALL ATTACHED TELETYPES IS RETURNED.
42760000     % THE FORM OF THE STATUS FOR ONE TELETYPE IS:
42770000     %     <STATUS> ::= ( <PROJ#> / <USER#> : <TU> / <BA> : <CONDITIONS> )
42780000     % WHERE <PROJ#>/<USER#> IS THE USER I. D., <TU>/<BA> IS THE ADDRESS
42790000     % OF THE TELETYPE, AND <CONDITIONS> IS A STRING OF 8 0-S OR 1-S
42800000     % WITH THE FOLLOWING MEANINGS:
42810000     %     1ST = 1 IF UNIT IS READY
42820000     %     2ND = 1 IF UNIT IS READ READY
42830000     %     3RD = 1 IF UNIT IS WRITE READY
42840000     %     4TH = 1 IF WRITE IS IN PROGRESS
42850000     %     5TH = 1 IF BREAK KEY USED DURING LAST OUTPUT
42860000     %     6TH = 1 IF ABNORMAL CONDITION SENSED
42870000     %     7TH = 1 IF UNIT IS BUSY
42880000     %     8TH = 1 IF SEEK IN EFFECT ON STATION
42890000     % WHEN THE STATUS OF ALL TELETYPES IS RETURNED, THE FORM OF
42900000     % THE VALUE IS:
42910000     %     <STATUS> <STATUS> <STATUS> ... <STATUS>
42920000     % I. E., THERE IS ONE <STATUS> FOR EACH ATTACHED TELETYPE.
42930000     % STATUS() FAILS IF:
42940000     %     A IS NOT NULL, AND IS NOT THE ADDRESS OF AN ATTACHED TELETYPE
42950000     %     A IS NULL, AND TO TELETYPES ARE ATTACHED
42960000     BEGIN
42970000     ALPHA ARRAY STTS[0:4×TTMAX]; % FOR BUILDING STATUS STRING
42980000     INTEGER ARRAY SIZE[0:TTMAX,0:3]; % HOLDS SIZES OF STATUS NUMBERS
42990000     INTEGER HIGH, LOW, N, STYPE, U;
43000000     BOOLEAN ALL;
43010000     %
43020000     FINDUSERS;
43030000     IF SP ≤ MKS THEN ALL ← TRUE ELSE
43040000        BEGIN % FIND UNIT OR UNITS:
43050000        AA ← PST[MKS];
43060000        IF VALU[AA].S = 0 THEN ALL ← TRUE ELSE
43070000           BEGIN ALL ← FALSE;  % ONLY ONE UNIT.
43080000           IF U←TTINDEX(AA) < 0 THEN GO TO FCTFAIL;
43090000           END;
43100000        END FINDING UNITS;
43110000     IF SP ≤ MKS+1 THEN STYPE ← 0 ELSE
43120000        IF NOT NUMVAL(PST[MKS+1],STYPE) OR STYPE < 0 OR STYPE > 1
43130000           THEN GO TO FCTFAIL;
43140000     IF N←NUMUSERS-1 < 0 THEN GO TO FCTFAIL;
43150000     LOW ← -1;
43160000     IF ALL THEN
43170000        BEGIN LOW ← 0;
43180000        HIGH ← N;
43190000        END ELSE HIGH ← LOW ← U;
43200000     % UPDATE STATUS WORDS IF NECESSARY:
43210000     IF STYPE > 0 THEN
43220000        FOR I ← LOW STEP 1 UNTIL HIGH DO
43230000           STAT[I] ← STATUS(STAT[I],STYPE);
43240000     %
43250000     % OTHER INSTALLATIONS WILL WANT TO RE-DO THE I.D. PART OF THIS FCT
43260000     %
```

```
43270000    % WRITE STATUS STRING:
43280000    WRITE(STTS[*],FSTAT,HIGH-LOW+1,FOR I ← LOW STEP 1 UNTIL HIGH DO
43290000      [SIZE[I,0]←DIGITS(ID[I].[12:18]),     % I.D. PART
43300000       ID[I].[12:18],    % PROJ #           % I.D. PART
43310000       SIZE[I,1]←DIGITS(ID[I].[30:18]),     % I.D. PART
43320000       ID[I].[30:18],    % USER #           % I.D. PART
43330000       SIZE[I,2]←DIGITS(STAT[I].[9:4]),
43340000       STAT[I].[9:4],    % TERMINAL UNIT
43350000       SIZE[I,3]←DIGITS(STAT[I].[14:4]),
43360000       STAT[I].[14:4],   % BUFFER ADDRESS
43370000       1-STAT[I].[30:1], % READY
43380000       STAT[I].[24:1],   % READ READY
43390000       STAT[I].[27:1],   % WRITE READY
43400000       STAT[I].[29:1],   % WRITE IN PROGRESS
43410000       STAT[I].[26:1],   % BREAK KEY ON LAST INPUT
43420000       STAT[I].[23:1],   % ABNORMAL CONDITION SENSED
43430000       STAT[I].[22:1],   % STATION BUSY
43440000       STAT[I].[38:1]]); % SEEK IN EFFECT
43450000    % CALCULATE SIZE OF STRING:
43460000    I1 ← 0;
43470000    FOR I ← LOW STEP 1 UNTIL HIGH DO  % CALCULATE SIZE OF STRING
43480000       BEGIN I1 ← I1 + 14;  % FOR "(/:/:)" & CONDITIONS
43490000       FOR J ← 0 STEP 1 UNTIL 3 DO
43500000          I1 ← I1 + SIZE[I,J];
43510000       END;
43520000    % SET UP STRING FOR RETURN VALUE:
43530000       RETURNVAL ← TEMPCELL;
43540000       AA ← STRING(I1,RETURNVAL);
43550000       IF DEATH THEN GO TO FCTFAIL;
43560000       MOVE(I1,STTS[*],0,FIRSTCHAR(AA));
43570000       VALU[RETURNVAL] ← AA;
43580000       GO TO SUCCEED;
43590000    END STATUS;
43600000 % 42: USER(A) OR USER()  HAS TWO USES.  IF USED WITHOUT PARAMETER,
43610000    % IT RETURNS THE ADDRESS OF THE TELETYPE ON WHICH THE LAST TELETYPE
43620000    % I/O TOOK PLACE.  IF IT HAS A PARAMETER, IT MUST BE THE ADDRESS OF
43630000    % AN ATTACHED TELETYPE; FURTHER OUTPUT WILL BE TO THIS TELETYPE,
43640000    % AND THE VALUE RETURNED WILL BE A.
43650000    BEGIN INTEGER TU, BA;
43660000    ALPHA ARRAY ADDR[0:1];
43670000    FINDUSERS;
43680000    IF NOT DATACOMF THEN GO TO FCTFAIL;
43690000    IF SP > MKS THEN
43700000       BEGIN I1 ← PST[MKS];
43710000       IF VALU[I1].S ≠ 0 THEN
43720000          BEGIN IF I1 ← TTINDEX(I1) < 0 THEN GO TO FCTFAIL;
43730000          USER ← I1;
43740000          END;
43750000       END;
43760000    I2 ← STAT[USER].[9:9];
43770000    WRITE(ADDR[*],FTTADDR,I←DIGITS(I2.[39:4]),I2.[39:4],
43780000       J←DIGITS(I2.[44:4]),I2.[44:4]);
43790000    I ← I + J + 1;
43800000    RETURNVAL ← TEMPCELL;
43810000    AB ← STRING(I,RETURNVAL);
43820000    MV(I,ADDR[*],0,FIRSTCHAR(AB));
43830000    VALU[RETURNVAL] ← AB;
```

```
43840000    GO TO IF DEATH THEN FCTFAIL ELSE SUCCEED;
43850000    END USER;
43860000 % 43: SEEK(U)  DOES A READ SEEK ON TERMINAL UNIT U.
43870000    BEGIN SEGMENT
43880000    IF SP ≤ MKS THEN GO TO FCTFAIL;
43890000    FINDUSERS;
43900000    IF I←TTINDEX(PST[MKS]) < 0 THEN GO TO FCTFAIL;
43910000    AA ← STAT[I];
43920000    SEEK(DCIN(AA));
43930000    GO TO SUCCEED;
43940000    END SEEK;
43950000 % 44: RELEASE(A)  ELIMINATES TELETYPE A FROM THE LIST OF USERS.
43960000    BEGIN SEGMENT
43970000    IF SP ≤ MKS THEN GO TO FCTFAIL;
43980000    IF I←TTINDEX(PST[MKS]) < 0 THEN GO TO FCTFAIL;
43990000    RELEASE(STAT[I]);
44000000    FINDUSERS;
44010000    GO TO SUCCEED;
44020000    END RELEASE;
44030000 % 45: RECORD(FILE,N)  SETS THE RECORD POINTER FOR THE NEXT I/O
44040000    % OPERATION ON THE FILE $FILE.  FAILURE OCCURS IF:
44050000    %     N NON-NUMERIC OR < 0
44060000    %     $FILE NOT AN I/O STRING (NOT TELETYPE)
44070000    BEGIN SEGMENT
44080000    IF SP < MKS+2 THEN IF NOT NULLARGS(2) THEN GO TO FCTFAIL;
44090000    IF NOT NUMVAL(PST[MKS+1],I1) OR I1 < 0 THEN GO TO FCTFAIL;
44100000    INDIRECT(MKS);
44110000    AA ← PST[MKS];
44120000    AB ← IO[AA];
44130000    IF AB.IOUSE = 0 OR AB.FILNO = 0 THEN GO TO FCTFAIL;
44140000    RECORD[AB.FILNO] ← I1;
44150000    GO TO SUCCEED;
44160000    END RECORD;
44170000 % 46: TRACE(NAME,TYPE,FILE,FCT)
44180000    GO TO UNDEFFCT;
44190000 % 47: ASSIGN(S,V)  ASSIGNS THE VALUE OF V TO S.  THE VALUE OF V IS
44200000    % WIPED OUT, SIE A STRICT TRANSFER OF POINTERS IS THE METHOD
44210000    % USED.  LITERALS SHOULD NEVER BE USED AS PARAMETERS TO ASSIGN().
44220000    BEGIN
44230000    IF SP≤MKS OR AA←PST[MKS]=0 THEN GO TO FCTFAIL;
44240000    IF SP = MKS+1 THEN
44250000       VALU[AA].LOC ← 1 ELSE
44260000       BEGIN
44270000       AB ← PST[MKS+1];
44280000       AC ← VALU[AB];
44290000       I ← AC.CH - 2;
44300000       MV(2,AA,6,DATA[AC.R,I.W],I.C);
44310000       VALU[AA].LOC ← AC;
44320000       VALU[AB].LOC ← 1; % NECESSARY DUE TO CHECKSYMBTABL
44330000       END;
44340000    GO TO SUCCEED;
44350000    END ASSIGN;
44360000 % 48: COPY(S,N)  PRODUCES N COPIES OF S. IF N IS NOT ≥ 0, COPY FAILS.
44370000    BEGIN SEGMENT
44380000    IF SP-MKS < 2 THEN IF NOT NULLARGS(2-SP+MKS) THEN GO TO FCTFAIL;
44390000    AA ← VALU[PST[MKS]];
44400000    IF NOT NUMVAL(PST[MKS+1],I1) OR I1<0 THEN GO TO FCTFAIL;
```

```
44410000     RETURNVAL ← TEMPCELL;
44420000     AC ← STRING(I1×(J+AA,S),RETURNVAL);
44430000     IF DEATH THEN GO TO FCTFAIL;
44440000     I2 ← AC.CH;
44450000     FOR I ← 1 STEP 1 UNTIL I1 DO
44460000        BEGIN MOVE(J,FIRSTCHAR(AA),DATA[AC,R,I2,W],I2,C);
44470000        I2 ← I2 + J;
44480000        END;
44490000     VALU[RETURNVAL] ← AC;
44500000     GO TO SUCCEED;
44510000     END COPY;
44520000 % 49: SEARCH(FILE)  DOES A DISK SEARCH FOR $FILE, IF $FILE IS AN I/O
44530000     % STRING TO A FILE; OTHERWISE SEARCH FAILS.  FAILURE ALSO OCCURS
44540000     % IF THE FILE IS NOT PRESENT.  IF IT IS PRESENT, THE VALUE IS OF
44550000     % THE FOLLOWING FORM:
44560000     %      <MFID> / <FID> : <SECURITY> : <REC SIZE> : <BLCK SIZE> :
44570000     %            <EOF ADDR> : <OPEN CNT>
44580000     % WHERE THE SECURITY STATUS, <SECURITY>, IS DEFINED BY:
44590000     %      <SECURITY> ::= <S> <R> <W>
44600000     %      <S>    ::= "1" IF USER CAN ALTER SECURITY STATUS
44610000     %      <R>    ::= "1" IF CAN READ
44620000     %      <W>    ::= "1" IF CAN WRITE
44630000     BEGIN
44640000     ALPHA ARRAY ST[0:6];
44650000     INTEGER ARRAY SIZE[1:6];
44660000     IF SP ≤ MKS THEN GO TO FCTFAIL;
44670000     INDIRECT(MKS);
44680000     AA ← PST[MKS];
44690000     IF VALU[AA].IOUSE=0 OR I←IO[AA].FILNO=0 THEN GO TO FCTFAIL;
44700000     SEARCH(IOFILE[I],ST[*]);
44710000     IF ST[0] < 0 THEN GO TO FCTFAIL;
44720000     FOR J ← 1,2 DO
44730000        BEGIN I1 ← SCANCHAR(" ",0,ST[J],1);
44740000        SIZE[J] ← IF I1 > 7 THEN 7 ELSE I1;
44750000        END;
44760000     FOR J ← 3,4 DO ST[J] ← ST[J] × 8; % CONVERT TO CHARACTERS
44770000     FOR J ← 3,4,5,6 DO SIZE[J] ← DIGITS(ST[J]);
44780000     WRITE(BUFOUT[*],FOKSEARCH,
44790000        SIZE[1], SIZE[2],
44800000        ST[0].[45:1],ST[0].[46:1],ST[0].[47:1],
44810000        FOR J ← 3,4,5,6 DO [SIZE[J],ST[J]]);
44820000     MV(SIZE[1],ST[1],1,BUFOUT[*],0);     % <MFID>
44830000     MV(SIZE[2],ST[2],1,BUFOUT[*],SIZE[1]+1); % <FID>
44840000     I1 ← 9;
44850000     FOR J ← 1 STEP 1 UNTIL 6 DO I1 ← I1+SIZE[J];
44860000     RETURNVAL ← TEMPCELL;
44870000     AA ← STRING(I1,RETURNVAL);
44880000     MOVE(I1,BUFOUT[*],0,FIRSTCHAR(AA));
44890000     VALU[RETURNVAL] ← AA;
44900000     GO TO SUCCEED;
44910000     END SEARCH;
44920000 % 50: .RANF(N)  PRODUCES A "RANDOM" NUMBER; 1 ≤ .RANF(N) ≤ N.
44930000     BEGIN DEFINE K=7557#, C=1#;
44940000     IF SP ≤ MKS THEN GO TO FCTFAIL;
44950000     IF NOT NUMVAL(PST[MKS],I1) OR I1 ≤ 0 THEN GO TO FCTFAIL;
44960000     RANDNO ← REAL(BOOLEAN(RANDNO←RANDNO×K+C) AND BOOLEAN(33554431));
44970000     RETURNVAL ← TEMPVAL(I1×(RANDNO/33554432)+0.5);
```

```
44980000     GO TO SUCCEED;
44990000     END RANDOM FUNCTION;
45000000 % 51: CLEAR() SETS ALL VARIABLES TO NULL, AT THEIR TOP LEVEL ONLY.
45010000     BEGIN ALPHA K;
45020000     INTEGER I, J;
45030000     FOR I ← 0 STEP 1 UNTIL SCATTERNO DO
45040000        FOR J ← 0 STEP 1 WHILE J < SCATTERSIZE DO
45050000           BEGIN
45060000           K ← J & I CONCSTR;
45070000           DO IF NAME[K].TYPE = 0 THEN VALU[K].LOC ← 0
45080000              UNTIL K←NAME[K].LINK = 0;
45090000           END;
45100000     GO TO SUCCEED;
45110000     END CLEAR;
45120000     END FUNCTION CASES;
45130000 %********INTRINSIC FUNCTIONS***********
45140000 %*****************************************
45150000     GO TO UNDEFFCT;
45160000 DEFINEDFCT:    % PROGRAM-DEFINED FUNCTION
45170000     BEGIN
45180000     LABEL TOODEEP, RET;
45190000     BOOLEAN FCTFAILURE;
45200000     ALPHA ARRAY
45210000        PRESERVE[0:5,0:STACKSIZE],
45220000        LVS[0:7];
45230000     ALPHA LV, K;
45240000     % IF INFORM THEN INFORMO(5);
45250000     FCTFAILURE ← FALSE;
45260000     MOVEWDS(30,BACKREFLAG,PRESERVE[0,16]); % INTERPRETER VARIABLES
45270000     MOVEWDS(SP,PNAME[*],PRESERVE[1,*]);
45280000     MOVEWDS(SP,PPOINT[*],PRESERVE[2,*]);
45290000     MOVEWDS(SP,PSIZE[*],PRESERVE[3,*]);
45300000     MOVEWDS(SP,PTYPE[*],PRESERVE[4,*]);
45310000     MOVEWDS(SP,PST[*],PRESERVE[5,*]);
45320000     MV(J←FCTV.S,DATA[FCTV.R,FCTV.W],FCTV.C,LVS[*],0);
45330000     % ASSIGN TO TEMPORARY LOCS:
45340000     FOR I ← SP = 1 STEP -1 UNTIL MKS DO IF PNAME[I] THEN
45350000        BEGIN
45360000        K ← PST[I];
45370000        AA ← TEMPCELL;
45380000        AB ← STRING(VALU[K].S,AA);
45390000        K ← VALU[K];
45400000        VALU[AA] ← AB;
45410000        MOVE(STRINGLOC(K),FIRSTCHAR(AB));
45420000        PST[I] ← AA;
45430000        END;
45440000     % PUSH PARAMS & LOC VARS AND ASSIGN VALUES:
45450000     MV(2,LVS[*],0,LV,6);
45460000     IF NOT PUSH(LV,0) THEN GO TO FCTFAIL;
45470000     IF NOT CHECKOUTUSE(LV) THEN FCTFAILURE ← TRUE;
45480000     K ← 2; I ← MKS ;
45490000     J ← J - 2;
45500000     FOR K ← 2 STEP 2 UNTIL J DO
45510000        BEGIN
45520000        MV(2,LVS[*],K,LV,6);
45530000        IF NOT PUSH(LV,IF I<SP THEN PST[I] ELSE 0) THEN GO TO FCTFAIL;
45540000        IF NOT CHECKOUTUSE(LV) THEN FCTFAILURE ← TRUE;
```

```
45550000        I ← I + 1;
45560000        END;
45570000     IF FCTFAILURE THEN GO TO RET;
45580000     IF DEATH THEN DIE;
45590000     FOR I ← 0 STEP 1 UNTIL 15 DO IF USEDST[I] THEN
45600000        BEGIN PRESERVE[0,I] ← TEMPLIST[I];
45610000        TEMPLIST[I] ← 0;
45620000        END;
45630000     ENTRY ← FCTV.LINK;
45640000     IF ENTRY = 0 THEN GO TO UNDEFFCT;
45650000     %
45660000     INTERPRETER;          % CALL FUNCTION
45670000     %
45680000     MOVEWDS(30,PRESERVE[0,16],BACKREFLAG); % INTERPRETER VARIABLES
45690000     MOVEWDS(SP,PRESERVE[1,*],PNAME[*]);
45700000     MOVEWDS(SP,PRESERVE[2,*],PPOINT[*]);
45710000     MOVEWDS(SP,PRESERVE[3,*],PSIZE[*]);
45720000     MOVEWDS(SP,PRESERVE[4,*],PTYPE[*]);
45730000     MOVEWDS(SP,PRESERVE[5,*],PST[*]);
45740000     FOR I ← 0 STEP 1 WHILE I ≤ 15 AND USEDST[I] DO
45750000     TEMPLIST[I] ← PRESERVE[0,I];
45760000     MV(2,LVS[*],0,LV,6);
45770000     RETURNVAL ← TEMPCELL;     % TO SAVE VALUE RETURNED.
45780000     IF DEATH THEN GO TO FCTFAIL;
45790000     AB ← VALTABL[LV.STR,LV.STW];
45800000     IF AB.CH ≥ 3 THEN     % MOVE POINTER
45810000        BEGIN I ← AB.CH - 2;
45820000        MV(2,RETURNVAL,6,DATA[AB.R,I.W],I.C);
45830000        END;
45840000     VALU[RETURNVAL] ← AB.LOC;
45850000     % THE VALUE IS NOW IN THE TEMPORARY LOCATION; THE SYMBTABL LOC
45860000     % LV (WITH SAME NAME AS FCT) IS INVALID, BUT WILL SOON BE WIPED
45870000     % OUT BY THE CALL OF POP(LV).
45880000 RET: J ← FCTV.S - 2;
45890000     FOR K ← 0 STEP 2 UNTIL J DO
45900000        BEGIN MV(2,LVS[*],K,LV,6);
45910000        IF NOT POP(LV) THEN
45920000           IF PRINTMESSAGES THEN
45930000              BEGIN MESSAGEI(20,INSTNO);
45940000              MESSAGETTI(20,INSTNO);
45950000              END;
45960000        IF NOT CHECKOUTUSE(LV) THEN FCTFAILURE ← TRUE;
45970000        END;
45980000     % IF INFORM THEN INFORMO(6);
45990000     PTYPE[MKS] ← 1;
46000000     GO TO IF FCTFAILURE THEN FCTFAIL ELSE
46010000        IF RESULT THEN SUCCEED ELSE FCTFAIL;
46020000     TOODEEP:    % FUNCTION DEPTH > MAXLEVEL
46030000        MESSAGEI(11,INSTNO);
46040000        MESSAGETTI(11,INSTNO);
46050000        DIE;
46060000     END DEFINED;
46070000 UNDEFFCT:       % UNDEFINED FUNCTION CALLED
46080000     BEGIN SEGMENT
46090000     WRITE(PRINT[DBL],FBL);
46100000     WRITE(BUFOUT[*],FUNDEFFCT);
46110000     AA ← NAME[FCT];
```

```
46120000     MOVE(I←MIN(AA.S,100),FIRSTCHAR(AA),BUFOUT[*],29);
46130000     WRITE(PRINT[DBL],17,BUFOUT[*]);
46140000     IF DATACOMF THEN % WRITE ON TELETYPE:
46150000        BEGIN I ← I + 29;
46160000        MV(2,CRLF,5,DCWRITE[*],0);  % "≤≠"
46170000        MOVE(I,BUFOUT[*],2,DCWRITE[*],2); % MESSAGE
46180000        MV(3,CRLF,5,DCWRITE[I.W],I.C+2); % "≤≠←"
46190000        IF OUTPUT THEN;
46200000        IF DEBUGGING THEN % SUSPEND
46210000           BEGIN SUSPENDREASON[3] ← TRUE;
46220000           DEBUG(SETGO);
46230000           FCTV ← VALU[FCT]; % RETREIVE NEW DEF., IF ANY.
46240000           GO TO BRANCH;      % TRY TO EXECUTE.
46250000           END;
46260000        END;
46270000     DIE;
46280000     END;
46290000 FCTFAIL:       % FAILURE EXIT FOR FUNCTIONS
46300000     IF BOOLEAN(VALU[FCT].OUTUSE) THEN
46310000        IF TRACEFCTRETURN(FCT,FALSE) THEN;
46320000     FAIL;
46330000 SUCCEED:
46340000     PST[MKS] ← RETURNVAL;
46350000     IF BOOLEAN(VALU[FCT].OUTUSE) THEN
46360000        IF NOT TRACEFCTRETURN(FCT,TRUE) THEN FAIL;
46370000     PNAME[MKS] ← FALSE;
46380000     SP ← MKS + 1;
46390000     RELATIVEPOINTER ← 4;
46400000     GO TO PSCAN;
46410000 END FUNCTION;
46420000 %********FUNCTION*************************************************************
46430000 %********LITERAL************************
46440000 % 11: "#"  LITERAL:
46450000     % 2=CHAR POINTER TO SYMB TABLE LOC.
46460000     BEGIN PTYPE[SP] ← 1;
46470000     PST[SP] ← TEMP.[18:12];
46480000     PNAME[SP] ← FALSE;
46490000     SP ← SP + 1;
46500000     RELATIVEPOINTER ← 3;
46510000     GO TO PSCAN;
46520000     END;
46530000 %********LITERAL************************
46540000 %********QMARK**************************
46550000 % 12: QMARK  END INST, NO GO-TO PART
46560000     BEGIN
46570000     MV(6,CONTROLPOINT+1,AA,2);
46580000     INSTNO ← AA.[12:12];
46590000     SLOC ← AA.[24:12];
46600000     FLOC ← AA.[36:12];
46610000     INSTRUCT ← (NEARPOINT←NEARPOINT+7).W + INSTRUCT;
46620000     NEARPOINT ← NEARPOINT.C;
46630000     % IF INFORM THEN CHECKSYMBTABL;
46640000     IF RULES[LEVEL].C = 7 THEN RETURNTEMPS;
46650000     GO TO INTERPRET;
46660000     END;
46670000 %********QMARK**************************
46680000 %********GO-TO**************************
```

```
46690000 % 13: ":"  GO-TO
46700000    BEGIN
46710000    AA ← AA.[18:12];
46720000 SETGO:
46730000    % AA MUST CONTAIN THE SYMBOL TABLE ADDRESS OF THE NEXT INST.
46740000    VALU[AA].LINK ← (AB←VALU[AA]).LINK + 1;  % BUMP REFERENCE COUNT.
46750000    IF BOOLEAN(AB.OUTUSE) THEN % LABEL IS BEING TRACED:
46760000       BEGIN SEGMENT
46770000       AC ← NAME[AA];
46780000       I1 ← MIN(AC.S,63);
46790000       I2 ← I1 + 3 - I1.[46:2];
46800000       I ← AB.LINK + 1;
46810000       WRITE(BUFOUT[*],FTRACEL,I2,DIGITS(I),I,1,DIGITS(INSTNO),INSTNO);
46820000       MV(I1,FIRSTCHAR(AC),BUFOUT[*],3);
46830000       WRITE(PRINT,FBL);
46840000       WRITE(PRINT[NO],17,BUFOUT[*]);
46850000       END LABEL TRACING OUTPUT;
46860000    IF ENTRY ≠ AA THEN
46870000       BEGIN
46880000       ENTRY ← AA;
46890000       IF I ← AB.S < 7 THEN GO TO UNDEFINED;
46900000       IF I ≤ 63 THEN
46910000          MV(I,FIRSTCHAR(AB),INST[*],0) ELSE
46920000          MOVE(I,FIRSTCHAR(AB),INST[*],0);
46930000       END;
46940000    MV(6,INST[*],3,AA,2);
46950000    INSTNO ← AA.[12:12];
46960000    SLOC ← AA.[24:12];
46970000    FLOC ← AA.[36:12];
46980000    INSTRUCT ← RELATIVEPOINTER ← 0;
46990000    NEARPOINT ← 9;
47000000    % IF INFORM THEN CHECKSYMBTABL;
47010000    IF RULES[LEVEL].C = 7 THEN RETURNTEMPS;
47020000    IF DEBUGGING THEN % CHECK LABEL LIMITS
47030000       FOR I ← 0 STEP 1 UNTIL NLABELLIMIT DO
47040000         IF ENTRY = LABELLIMIT[I] THEN
47050000              BEGIN SUSPENDREASON[2] ← TRUE;
47060000              DEBUG(SETGO);
47070000              GO TO INTERPRET;
47080000              END;
47090000    GO TO INTERPRET;
47100000    END;
47110000 %********GO-TO**************************
47120000    ;;;;;;;;;;;;
47130000 %********INPUT CHECK*********************
47140000 % 25: "I"  CHECK INPUT USE OF LAST ELEMENT
47150000    BEGIN AA ← PST[SP-1];
47160000    IF BOOLEAN(VALU[AA].INUSE) THEN IF NOT SNBLIN(AA) THEN FAIL;
47170000    RELATIVEPOINTER ← 1;
47180000    GO TO PSCAN;
47190000    END;
47200000 %********INPUT CHECK*********************
47210000 %********ARITHMETIC**********************
47220000 % 26: "."  ARITHMETIC OPERATOR
47230000    BEGIN
47240000    INTOVR ← ARITHOVFL;
47250000    EXPOVR ← ARITHOVFL;
```

```
47260000    CASE DOTTYPE[AA,C3] OF
47270000       BEGIN
47280000    % 0: ERROR
47290000       GO TO RERROR;
47300000    % 1: ".N"  NEGATION:
47310000       BEGIN
47320000       IF (MKS←SP-1) < 0 THEN GO TO PERROR;
47330000       IF PTYPE[MKS] = -1
47340000          THEN AB ← PST[MKS]
47350000          ELSE IF NOT NUMVAL(PST[MKS],AB) THEN GO TO NONNUMERIC;
47360000       PST[MKS] ← -AB;
47370000       PTYPE[MKS] ← -1;
47380000       END;
47390000    % 2: ARITHMETIC OPERATOR:
47400000       BEGIN
47410000       IF (MKS←SP-2) < 0 THEN GO TO PERROR;
47420000       IF PTYPE[MKS] = -1
47430000          THEN AB ← PST[MKS]
47440000          ELSE IF NOT NUMVAL(PST[MKS],AB) THEN GO TO NONNUMERIC;
47450000       IF PTYPE[MKS+1] = -1
47460000          THEN AC ← PST[MKS+1]
47470000          ELSE IF NOT NUMVAL(PST[MKS+1],AC) THEN GO TO NONNUMERIC;
47480000       IF I←AA.C3 = "+" THEN I1 ← AB + AC ELSE
47490000       IF I = "-" THEN I1 ← AB - AC ELSE
47500000       IF I = "x" THEN I1 ← AB x AC ELSE
47510000       IF I = "/" THEN IF AC=0 THEN GO TO DVDZERO ELSE
47520000          CASE DIVIDEMODE OF
47530000             BEGIN
47540000          % 0: ROUND
47550000             I1 ← AB/AC;
47560000          % 1: TRUNCATION
47570000             I1 ← ENTIER(AB/AC);
47580000          % 2: INTEGER
47590000             IF (I1←AA←AB/AC) ≠ ENTIER(AA) THEN FAIL
47600000             END DIVIDE CASES ELSE
47610000       IF I = "*" THEN
47620000          IF AC = 0 THEN I1 ← 1 ELSE
47630000             BEGIN
47640000             IF AC < 0 THEN
47650000                BEGIN AC ← -AC;
47660000                AA ← 1/AB;
47670000                END ELSE AA ← AB;
47680000             FOR I ← 1 STEP 1 UNTIL AC DO AA ← AA x AB;
47690000             I1 ← AA;
47700000             END ELSE
47710000          GO TO PERROR;  % INVALID CHAR AFTER "."
47720000       PST[MKS] ← I1;
47730000       PTYPE[MKS] ← -1;
47740000       END;
47750000    % 3: ".S"  CONVERT TO STRING
47760000       BEGIN
47770000       IF MKS←SP-1 < 0 OR PTYPE[MKS] ≠ -1 THEN
47780000          BEGIN INFORMO(9);
47790000          GO TO PERROR;
47800000          END;
47810000       PST[MKS] ← TEMPVAL(PST[MKS]);
47820000       PTYPE[MKS] ← 1;
```

```
47830000          PNAME[MKS] ← FALSE;
47840000          END
47850000          END ARITH CASES;
47860000       %
47870000       SP ← MKS + 1;
47880000       RELATIVEPOINTER ← 2;
47890000       INTOVR ← 0;
47900000       GO TO PSCAN;
47910000 ARITHOVFL: % INTEGER OVERFLOW IN ARITHMETIC:
47920000       IF PRINTMESSAGES THEN
47930000          BEGIN MESSAGEI(0,INSTNO);
47940000          MESSAGETTI(0,INSTNO);
47950000          END;
47960000       FAIL;
47970000 NONNUMERIC: % NON-NUMERIC ARGUMENT TO ARITHMETIC:
47980000       IF PRINTMESSAGES THEN
47990000          BEGIN MESSAGEI(17,INSTNO);
48000000          MESSAGETTI(17,INSTNO);
48010000          END;
48020000       FAIL;
48030000 DVDZERO: % DIVIDE-BY-ZERO IN ARITHMETIC
48040000       IF PRINTMESSAGES THEN
48050000          BEGIN MESSAGEI(18,INSTNO);
48060000          MESSAGETTI(18,INSTNO);
48070000          END;
48080000       FAIL;
48090000       END ARITHMETIC;
48100000 %********ARITHMETIC**********************
48110000       };
48120000 %********GROUPING************************
48130000 % 29: "("  GROUPING
48140000       BEGIN
48150000       MKS ← SP - AA.C3;
48160000       I1 ← 0;      % TO KEEP SIZE IN
48170000       FOR I ← MKS STEP 1 UNTIL SP-1 DO
48180000          BEGIN
48190000          I ← I;
48200000          IF PTYPE[I] ≠ 1 THEN
48210000             BEGIN INFORMI(1,PTYPE[I]);
48220000             SYSTEMERROR ← TRUE;
48230000             GO TO DEAD;
48240000             END;
48250000          AA ← PST[I];
48260000          I1 ← I1 + VALU[AA].S;
48270000          END;
48280000       AC ← TEMPCELL;
48290000       VALU[AC] ← AB ← STRING(I1,AC);
48300000       I1 ← AB.CH;
48310000       I2 ← AB.R;
48320000       FOR I ← MKS STEP 1 WHILE I<SP DO
48330000          BEGIN
48340000          I ← I;
48350000          AA ← PST[I];
48360000          AA ← VALU[AA];
48370000          IF J←AA.S > 0 THEN
48380000             BEGIN
48390000             IF J ≤ 63 THEN
```

```
48400000              MV(J,FIRSTCHAR(AA),DATA[I2,I1,W],I1,C) ELSE
48410000              MOVE(J,FIRSTCHAR(AA),DATA[I2,I1,W],I1,C);
48420000           I1 ← I1 + J;
48430000           END;
48440000        END;
48450000     PST[MKS] ← AC;
48460000     SP ← MKS + 1;
48470000     PNAME[MKS] ← FALSE;
48480000     RELATIVEPOINTER ← 2;
48490000     GO TO PSCAN;
48500000     END;  % GROUPING
48510000 %********GROUPING************************
48520000     ;
48530000 %********REPLACEMENT*********************
48540000 % 31: "←"  REPLACEMENT OR ASSIGNMENT
48550000     BEGIN RELATIVEPOINTER ← 1;
48560000     IF SELFREFLAG THEN   % STR REF USED AS STR VAR
48570000        BEGIN
48580000        IF PRINTMESSAGES THEN
48590000           BEGIN MESSAGEI(3,INSTNO);
48600000           MESSAGETTI(3,INSTNO);
48610000           END;
48620000        GO TO GOTO;
48630000        END;
48640000     % IF INFORM THEN INFORMO(3);
48650000     SIZE ← (AB←VALTABL[REFI←PST[0].STR,REFJ←PST[0].STW]).S;
48660000     NOREPLACES ← SP - 1;
48670000     % CALCULATE SIZE OF NEW STR REF;
48680000     FOR SP ← 1 STEP 1 UNTIL NOREPLACES DO
48690000        BEGIN AA ← PST[SP];
48700000        I1 ← VALU[AA].S;
48710000        RSIZE ← RSIZE + I1;
48720000        END;
48730000     % IF NEW STR REF IS NOT LARGER THAN THE OLD, THE SAME STRING
48740000     % CAN BE USED.  AB STILL POINTS TO THE STR REF.
48750000     IF RSIZE > SIZE THEN
48760000        BEGIN B1 ← TRUE;  % STR REF IN NEW STRING
48770000        AA ← STRING(RSIZE,PST[0]);
48780000        AB ← VALTABL[REFI,REFJ];
48790000        END ELSE
48800000        BEGIN B1 ← FALSE; % USE SAME STRING FOR STR REF.
48810000        IF RSIZE < SIZE THEN VALTABL[REFI,REFJ].S ← RSIZE;
48820000        AA ← AB;
48830000        END;
48840000     % AA POINTS TO NEW VALUE.
48850000     % AB POINTS TO OLD VALUE.
48860000     I ← AA.R;
48870000     J ← AB.R;
48880000     IF B1 THEN IF FRONTEND > 0 THEN
48890000        IF FRONTEND ≤ 63 THEN
48900000           MV(FRONTEND,DATA[J,AB.W],AB.C,DATA[I,AA.W],AA.C) ELSE
48910000           MOVE(FRONTEND,DATA[J,AB.W],AB.C,DATA[I,AA.W],AA.C);
48920000     I1 ← FRONTEND + AA.CH;     % POINTER TO NEXT CHAR TO BE OVERWRITTEN.
48930000     % REWRITE MATCHED PORTION:
48940000     FOR SP ← 1 STEP 1 UNTIL NOREPLACES DO
48950000        BEGIN AC ← PST[SP];
48960000        AC ← VALU[AC];
```

```
48970000        IF (I2←AC.S) > 0 THEN
48980000           BEGIN IF I2 ≤ 63 THEN
48990000              MV(I2,DATA[AC.R,AC.W],AC.C,DATA[I,I1.W],I1.C) ELSE
49000000              MOVE(I2,DATA[AC.R,AC.W],AC.C,DATA[I,I1.W],I1.C);
49010000           I1 ← I1 + I2;
49020000           END;
49030000        END;
49040000     IF REAREND > 0 THEN IF B1 OR RSIZE ≠ SIZE THEN
49050000        BEGIN I2 ← AB.CH + SIZE - REAREND;
49060000        IF REAREND > 0 THEN IF REAREND ≤ 63 THEN
49070000           MV(REAREND,DATA[J,I2.W],I2.C,DATA[I,I1.W],I1.C) ELSE
49080000           MOVE(REAREND,DATA[J,I2.W],I2.C,DATA[I,I1.W],I1.C);
49090000        END;
49100000     IF B1 THEN VALTABL[REFI,REFJ].LOC ← AA;
49110000     IF BOOLEAN(AB.OUTUSE) THEN
49120000        IF NOT SNBLOUT(PST[0]) THEN GO TO FAILED;
49130000     GO TO GOTO;
49140000     END;
49150000 %********REPLACEMENT*********************
49160000     ;;;
49170000 %********LABEL INDIRECTION**************
49180000 % 35: "L"  INDIRECT TO PRODUCE LABEL
49190000     BEGIN
49200000     IF SP ≠ 1 THEN GO TO PERROR;
49210000     AA ← VALU[PST[0]];
49220000     AA ← ENTERST (AA.S,DATA[AA.R,*],AA.CH,"INST");
49230000     GO TO SETGO;
49240000     END;
49250000 %********LABEL INDIRECTION**************
49260000     ;;;;;;
49270000 %********INDIRECTION********************
49280000 % 42: "$"  INDIRECTION
49290000     BEGIN
49300000     % IF INFORM THEN INFORMO(16);
49310000     IF (SP←SP-1) < 0 THEN
49320000        BEGIN INFORMI(23,INSTNO);
49330000        SYSTEMERROR ← INFORM ← TRUE;
49340000        GO TO DEAD;
49350000        END;
49360000     AB ← PST[SP];
49370000     FOR I ← AA.C3 STEP -1 UNTIL 1 DO
49380000        BEGIN
49390000        AC ← VALU[AB];
49400000        IF BOOLEAN(AC.INUSE) THEN
49410000           BEGIN IF NOT SNBLIN(AB) THEN GO TO FAILED;
49420000           AC ← VALU[AB];  % INPUT MAY MOVE VALUE
49430000           END;
49440000        IF PRINTMESSAGES THEN IF AC.S = 0 THEN
49450000           BEGIN MESSAGEI(19,INSTNO);
49460000           MESSAGETTI(19,INSTNO);
49470000           END;
49480000        AB ← ENTERST(AC.S,DATA[AC.R,*],AC.CH,"SYMB");
49490000        IF DEATH THEN GO TO DEAD;
49500000        END;
49510000     PST[SP] ← AB;
49520000     PNAME[SP] ← TRUE;
49530000     SP ← SP + 1;
```

```
49540000     RELATIVEPOINTER ← 2;
49550000     GO TO PSCAN;
49560000     END;
49570000 %********INDIRECTION*********************
49580000 %********STR VAR************************
49590000 % 43: "*"  STRING VARIABLE
49600000     BEGIN
49610000     RELATIVEPOINTER ← 2;
49620000     VARFLAG ← TRUE;
49630000     SP ← SP - 1;
49640000     CASE AA.C3 OF
49650000        BEGIN
49660000     % ERROR
49670000        GO TO PERROR;
49680000     % 1: ARB VAR
49690000        BEGIN PTYPE[SP] ← 2;
49700000        PSIZE[SP] ← 0;
49710000        END;
49720000     % 2: BAL VAR
49730000        BEGIN PTYPE[SP] ← 4;
49740000        PSIZE[SP] ← 1;
49750000        END;
49760000     % 3: FIXED-LENGTH
49770000        BEGIN
49780000        % IF SIZE < 0 THEN RULE FAILS.
49790000        IF (IF PTYPE[SP] = 7
49800000              THEN I1←PST[SP] < 0
49810000              ELSE NOT NUMVAL(PST[SP],I1) OR I1<0)
49820000           THEN GO TO FIXERR;
49830000        PSIZE[SP←SP-1] ← I1;
49840000        PTYPE[SP] ← 3;
49850000        END
49860000        END VAR CASES;
49870000     SP ← SP + 1;
49880000     GO TO PSCAN;
49890000     END;
49900000 %********STR VAR************************
49910000 %********RESERVED LABEL*****************
49920000 % 44: "="  RESERVED LABEL
49930000     IF I←AA.C3 = "R" THEN
49940000        BEGIN RESULT ← TRUE;
49950000        GO TO RETURN;
49960000        END ELSE
49970000     IF I = "F" THEN
49980000        BEGIN RESULT ← FALSE;
49990000        GO TO RETURN;
50000000        END ELSE
50010000     IF I = "E" THEN
50020000        BEGIN RESULT ← TRUE;
50030000        GO TO ENDTERPRET;
50040000        END;
50050000 %********RESERVED LABEL*****************
50060000     ;;;;
50070000 %********GO-TO PART*********************
50080000 % 49: "/" GO-TO PART DELIMITER AFTER PATTERN
50090000 GOTO:
50100000     BEGIN
```

```
50110000     % IF INFORM THEN INFORMO(26);
50120000     NEARPOINT ← IF SUCCESS THEN SLOC ELSE FLOC;
50130000     INSTRUCT ← NEARPOINT.W;
50140000     NEARPOINT ← NEARPOINT.C;
50150000     RELATIVEPOINTER ← SP ← 0;
50160000     INGOTOPART ← TRUE;
50170000     GO TO PSCAN;
50180000     END;
50190000 %********GO-TO PART**********************
50200000 %********END STR REF*********************
50210000 % 50: "S"  END OF STRING REFERENCE, BEFORE PATTERN
50220000     BEGIN
50230000     IF BOOLEAN(VALU[PST[0]],INUSE) THEN
50240000        IF NOT SNBLIN(PST[0]) THEN FAIL;
50250000     PTYPE[0] ← ANCHORMODE;
50260000     PSIZE[0] ← ANCHORSIZE;
50270000     PPOINT[0] ← 0;
50280000     RELATIVEPOINTER ← 1;
50290000     GO TO PSCAN;
50300000     END;
50310000 %********END STR REF*********************
50320000     ;;;;
50330000 %********SYNTAX ERROR*******************
50340000 % 55: "X"  SYNTAX ERROR IN ORIGINAL CODE
50350000     BEGIN MESSAGEI(1,INSTNO);
50360000     MESSAGETTI(1,INSTNO);
50370000     GO TO DEAD;
50380000     END;
50390000 %********SYNTAX ERROR*******************
50400000     ;;;;
50410000 %********NON-INPUT STRING***************
50420000 % 59: "%"  NON-INPUT STRING NAME
50430000     BEGIN PTYPE[SP] ← 1;
50440000     PST[SP] ← AA.[18:12];
50450000     PNAME[SP] ← TRUE;
50460000     SP ← SP + 1;
50470000     RELATIVEPOINTER ← 3;
50480000     GO TO PSCAN;
50490000     END;
50500000 %********NON-INPUT STRING***************
50510000     ;
50520000 %********PATTERN MATCH******************
50530000 % 61: "="  PATERN MATCH
50540000     BEGIN
50550000     % IF SP ≤ 1 THEN GO TO PERROR;
50560000     RELATIVEPOINTER ← 1;
50570000     PMINLEFT[SP] ← PTYPE[SP] ← 0;  % TO END SCAN.
50580000     NOPATTERNS ← SP - 1;
50590000     BACKREFLAG ← FALSE;
50600000     FOR I ← NOPATTERNS STEP -1 UNTIL 1 DO
50610000        BEGIN
50620000        IF I1 ← PTYPE[I] = 1 THEN
50630000        FOR J ← I-1 STEP -1 UNTIL 0 DO
50640000           BEGIN
50650000           IF PST[I] = PST[J]
50660000              THEN IF PTYPE[J] ≠ 1
50670000                 THEN BEGIN
```

```
50680000                  PTYPE[I] ← I1 ← 5;
50690000                  PBACK[I] ← J;
50700000                  PSIZE[I] ← PSIZE[J];
50710000                  BACKREFLAG ← TRUE;
50720000                  GO TO MINLEF;
50730000                  END;
50740000            END ELSE IF I1 ≤ 4 THEN
50750000               IF PST[I] = PST[0]
50760000               THEN SELFREFLAG ← TRUE;
50770000         IF I1 = 1 THEN    % SET PLENGTH & LLOC FIELDS
50780000            BEGIN AB ← PST[I];
50790000            PSIZE[I] ← (PLOC[I] ← VALU[AB]),S;
50800000            END;
50810000 MINLEF: % THE MINLEFT WORD HOLDS THE MINIMUM SIZE OF THE
50820000         % REMAINDER OF THE PATTERN.
50830000         PMINLEFT[I] ← PMINLEFT[I+1] + PSIZE[I];
50840000         END;
50850000      REFLOC ← (AA←VALTABL[REFI←PST[0],STR,REFJ←PST[0],STW]),CH;
50860000      SIZE ← AA,S;
50870000      RPR ← AA,R;
50880000      SP ← 0;
50890000      NOBACKORBAL ← TRUE;
50900000 NEWPOINT:
50910000      PPOINT[SP+1] ← PPOINT[SP] + PSIZE[SP];
50920000 NEXTPATTERN:
50930000      SP ← SP + 1;
50940000      IF PMINLEFT[SP] + PPOINT[SP] > SIZE THEN GO TO SIZEFAILURE;
50950000      CASE PTYPE[SP] OF
50960000         BEGIN
50970000      % 0: END OF SCAN, IF AFTER LAST PATTERN ELEMENT:
50980000         GO TO IF SP > NOPATTERNS THEN SCANSUCCESS ELSE SCANERR;
50990000      % 1: CONSTANT PATTERN ELEMENT
51000000         BEGIN
51010000         AA ← PLOC[SP];
51020000         REFPT ← REFLOC + PPOINT[SP];
51030000         GO TO IF (IF (I←AA,S) ≤ 63 THEN
51040000                  EQ(I,FIRSTCHAR(AA),DATA[RPR,REFPT,W],REFPT,C) ELSE
51050000                  EQUAL(I,FIRSTCHAR(AA),DATA[RPR,REFPT,W],REFPT,C))
51060000            THEN NEWPOINT
51070000            ELSE DROPBACK;
51080000         END;
51090000      % 2: ARBITRARY STRING VARIABLE
51100000         BEGIN PPOINT[SP+1] ← PPOINT[SP];
51110000         GO TO NEXTPATTERN;
51120000         END;
51130000      % 3: FIXED-LENGTH STRING VARIABLE
51140000         GO TO NEWPOINT;
51150000      % 4: BALANCED STRING VARIABLE
51160000         BEGIN NOBACKORBAL ← FALSE;
51170000         INCREASE ← 0;
51180000         REFPT ← REFLOC + PPOINT[SP];
51190000 BALANCE:
51200000         COUNT ← 0;
51210000         DO BEGIN
51220000            IF PPOINT[SP] + INCREASE←INCREASE+1 > SIZE THEN GO TO DROPBACK;
51230000            IF AA←CHAR(DATA[RPR,REFPT,W],REFPT,C) = "(" THEN
51240000               COUNT ← COUNT + 1 ELSE
```

```
51250000            IF AA = ")" THEN
51260000               COUNT ← COUNT - 1;
51270000            REFPT ← REFPT + 1;
51280000            END UNTIL COUNT ≤ 0;
51290000         IF COUNT < 0 THEN GO TO DROPBACK;
51300000         PPOINT[SP+1] ← PPOINT[SP] + PSIZE[SP]←INCREASE;
51310000         GO TO NEXTPATTERN;
51320000         END;
51330000      % 5: BACK REFERENCE:
51340000         BEGIN NOBACKORBAL ← FALSE;
51350000         INCREASE ← PPOINT[PBACK[SP]+1] - I1←PPOINT[PBACK[SP]];
51360000         IF PPOINT[SP] + INCREASE > SIZE THEN GO TO SIZEFAILURE;
51370000         I1 ← REFLOC + I1;
51380000         REFPT ← REFLOC + PPOINT[SP];
51390000         IF NOT EQUAL(INCREASE,DATA[RPR,I1,W],I1,C,
51400000            DATA[RPR,REFPT,W],REFPT,C) THEN GO TO DROPBACK;
51410000         PPOINT[SP+1] ← PPOINT[SP] + INCREASE;
51420000         GO TO NEXTPATTERN;
51430000         END
51440000         END CASES;
51450000      GO TO SCANERR;
51460000 SIZEFAILURE:
51470000      IF NOBACKORBAL THEN GO TO SCANFAILURE;
51480000 DROPBACK:
51490000      DO IF (SP←SP-1) < 0 THEN GO TO SCANFAILURE
51500000         UNTIL NOT BOOLEAN(PTYPE[SP]);
51510000      % NON-EXTENDABLE IF CONSTANT, FIXED-LENGTH OR BACK-REFERENCE
51520000      % CAN ONLY BE EXTENDED FOR SIMPLE & BALANCED VARIABLES.
51530000      IF PTYPE[SP] = 2 THEN     % ARBITRARY VARIABLE:
51540000         BEGIN
51550000         GO TO IF PMINLEFT[SP+1] + PPOINT[SP+1]←PPOINT[SP+1]+1 ≤ SIZE
51560000            THEN NEXTPATTERN
51570000            ELSE SIZEFAILURE;
51580000         END ELSE % BALANCED VARIABLE:
51590000         BEGIN
51600000         REFPT ← REFLOC + PPOINT[SP] + INCREASE←PSIZE[SP];
51610000         GO TO BALANCE;
51620000         END;
51630000 SCANERR:
51640000      BEGIN SEGMENT
51650000      INFORMII(3,INSTNO,SP);
51660000      WRITESTACK(NOPATTERNS+1);
51670000      ABORT;
51680000      END;
51690000 SCANSUCCESS:
51700000      % IF INFORM THEN WRITESTACK(NOPATTERNS+1);
51710000      IF PTYPE[SP-1] = 2 THEN PPOINT[SP] ← SIZE;
51720000 % ASSIGNMENT OF STRING VARIABLES:
51730000      % IF INFORM THEN INFORMO(0);
51740000      FRONTEND ← PPOINT[1];
51750000      REAREND ← SIZE - PPOINT[NOPATTERNS+1];
51760000      RSIZE ← FRONTEND + REAREND;
51770000      IF VARFLAG THEN
51780000      BEGIN
51790000      IF SELFREFLAG THEN
51800000         BEGIN % MAKE COPY OF STR REF VALUE:
51810000         TEMPREF ← TEMPCELL;
```

```
51820000       AA ← STRING(SIZE,TEMPREF);
51830000       IF DEATH THEN DIE;
51840000       AB ← VALTABL[REFI,REFJ];
51850000       MOVE(SIZE,FIRSTCHAR(AB),FIRSTCHAR(AA));
51860000       VALU[TEMPREF] ← AB ← AA;
51870000       END ELSE
51880000       BEGIN TEMPREF ← PST[0];
51890000       AB ← VALU[TEMPREF];
51900000       END;
51910000    FOR SP ← 1 STEP 1 UNTIL NOPATTERNS DO
51920000       IF PTYPE[SP]>1 THEN IF PTYPE[SP]<5 THEN IF AC←PST[SP]≠0 THEN
51930000       BEGIN
51940000       % STRING VARIABLE ASSIGNMENT:
51950000       IF (J←PPOINT[SP+1]-PPOINT[SP]) = 0 THEN
51960000          VALU[AC].S ← 0 ELSE
51970000          BEGIN
51980000          IF (AA←VALU[AC]).S < J THEN
51990000             % NEED NEW STRING--NEW VALUE LARGER THAN OLD.
52000000             BEGIN AA ← STRING(J,AC);
52010000             AB ← VALU[TEMPREF];
52020000             VALU[AC].LOC ← AA;
52030000             END ELSE
52040000             % CAN USE OLD VALUE--NEW VALUE SHORT ENOUGH
52050000             VALU[AC].S ← J;
52060000          I1 ← AB.CH + PPOINT[SP];
52070000          IF J ≤ 63 THEN
52080000             MV(J,DATA[AB,R,I1,W],I1,C,FIRSTCHAR(AA)) ELSE
52090000             MOVE(J,DATA[AB,R,I1,W],I1,C,FIRSTCHAR(AA));
52100000          END;
52110000       IF BOOLEAN(VALU[AC],OUTUSE) THEN IF NOT SNBLOUT(AC) THEN FAIL;
52120000       END;
52130000    END ASSIGNMENT OF STR VARS;
52140000    SP ← 1;
52150000    GO TO PSCAN;
52160000    END;
52170000 %********PATTERN MATCH*****************
52180000    ;
52190000 %********STRING NAME*******************
52200000 % 63: """  STRING NAME, WITH INPUT CHECK
52210000    BEGIN PTYPE[SP] ← 1;
52220000    PST[SP] ← I ← AA.[18:12];
52230000    IF BOOLEAN(VALU[I],INUSE) THEN IF NOT SNBLIN(I) THEN FAIL;
52240000    PNAME[SP] ← TRUE;
52250000    SP ← SP + 1;
52260000    RELATIVEPOINTER ← 3;
52270000    GO TO PSCAN;
52280000    END
52290000 %********STRING NAME*******************
52300000    END PSCAN CASES;
52310000 PERROR:
52320000    BEGIN SEGMENT
52330000    INFORMI(5,INSTNO);
52340000    WRITE(PRINT,17,INST[*]);
52350000    CLEAR(BUFOUT,17);
52360000    IF INSTRUCT × 8 + NEARPOINT < 132 THEN
52370000       MV(1,QMARK,7,BUFOUT[INSTRUCT],NEARPOINT);
52380000    WRITE(PRINT,17,BUFOUT[*]);
```

```
52390000     RESULT ← FALSE;
52400000     ABORT;
52410000     END;
52420000 FIXERR:
52430000     IF PRINTMESSAGES THEN
52440000     BEGIN SEGMENT
52450000     WRITE(PRINT,FBL);
52460000     AA ← VALU[PST[SP]];
52470000     WRITE(BUFOUT[*],FFIXVARSIZE,AA,S,INSTNO);
52480000     MOVE(MIN(AA.S,116),FIRSTCHAR(AA),BUFOUT[*],16);
52490000     WRITE(PRINT,17,BUFOUT[*]);
52500000     IF INFORM THEN WRITESTACK(SP);
52510000     FAIL;
52520000     END;
52530000 SCANFAILURE:
52540000 %   IF INFORM THEN
52550000 %      BEGIN INFORMO(8);
52560000 %      WRITESTACK(NOPATTERNS+1);
52570000 %      END;
52580000 FAILED:
52590000     % IF INFORM THEN INFORMO(4);
52600000     IF INGOTOPART THEN   % GO-TO PART HAS FAILED--FATAL ERROR
52610000        BEGIN
52620000        MESSAGEI(2,INSTNO);
52630000        MESSAGETTI(2,INSTNO);
52640000        DIE;
52650000        END;
52660000     IF DEATH THEN DIE;
52670000     SUCCESS ← FALSE;
52680000     GO TO GOTO;
52690000 DEAD:    % FATAL ERROR HAS OCCURRED--TERMINATE PROGRAM.
52700000     % ERROR MESSAGE SHOULD ALREADY HAVE BEEN PRINTED.
52710000     BEGIN SEGMENT
52720000     IF SYSTEMERROR THEN ABORT;
52730000     IF DEBUGGING THEN  % SUSPEND:
52740000        BEGIN SUSPENDREASON[3] ← TRUE;
52750000        DEATH ← FALSE;
52760000        DEBUG(SETGO);
52770000        END;
52780000     MESSAGETTI(13,INSTNO);
52790000     RESULT ← FALSE;
52800000     GO TO ENDTERPRET;
52810000     END;
52820000 LIMITHIT:     % SYSTEM LIMIT OF SOME SORT PASSED--PRINT ALL LIMITS.
52830000     BEGIN SEGMENT
52840000     DMPSTR ← TRUE;
52850000     MESSAGEI(4,INSTNO);
52860000     MESSAGETTI(4,INSTNO);
52870000     IF CPULIMITEXISTS THEN
52880000        BEGIN MESSAGEI(5,CPULIMIT/60);
52890000        MESSAGETTI(5,CPULIMIT/60);
52900000        END;
52910000     IF IOLIMITEXISTS THEN
52920000        BEGIN MESSAGEI(6,IOLIMIT/60);
52930000        MESSAGETTI(6,IOLIMIT/60);
52940000        END;
52950000     IF RULELIMITEXISTS THEN
```

```
52960000       BEGIN MESSAGEI(7,RULELIMIT);
52970000       MESSAGETTI(7,RULELIMIT);
52980000       END;
52990000    RESULT ← FALSE;
53000000    DIE;
53010000    END LIMITHIT;
53020000 UNDEFINED:    % ATTEMPTED TRANSFER TO UNDEFINED LABEL.
53030000    % AA MUST CONTAIN SYMB TABLE ADDRESS OF LABEL.
53040000    BEGIN ALPHA AC, AB;
53050000    DMPSTR ← TRUE;
53060000    WRITE(PRINT,FBL);
53070000    AB ← NAME[AA];
53080000    WRITE(BUFOUT[*],FUNDEFLABEL);
53090000    MOVE(I←MIN(90,AB.S),FIRSTCHAR(AB),BUFOUT[*],40);
53100000    WRITE(PRINT,17,BUFOUT[*]);
53110000    IF DATACOMF THEN % WRITE MESSAGE ON TELETYPE:
53120000       BEGIN MV(2,CRLF,5,DCWRITE[*],0);
53130000       MOVE(I←I+38,BUFOUT[*],2,DCWRITE[*],2);
53140000       MV(3,CRLF,5,DCWRITE[I,W],I.C+2);
53150000       IF OUTPUT THEN;
53160000       IF DEBUGGING THEN % SUSPEND
53170000          BEGIN SUSPENDREASON[3] ← TRUE;
53180000          AC ← AA;
53190000          DEBUG(SETGO);
53200000          % SEE IF LABEL DEFINED NOW:
53210000          GO TO IF VALU[AA←AC].S ≥ 10 THEN SETGO ELSE UNDEFINED;
53220000          END;
53230000       END;
53240000    RESULT ← FALSE;
53250000    DIE;
53260000    END;
53270000 RETURN:
53280000    % IF INFORM THEN INFORMI(10,LEVEL);   % END OF THIS LEVEL.
53290000    RETURNTEMPS;
53300000    IF LEVEL←LEVEL-1 ≤ 0 THEN % RETURN, NO FUNCTION CALLED
53310000       BEGIN MESSAGE0(22);
53320000       MESSAGETTI(14,INSTNO);
53330000       DIE;
53340000       END;
53350000 END INTERPRETER;
53360000 %********INTERPRETER*************************************************
53370000 %********INTRINSIC**************************************************
53380000 % GIVEN A FUNCTION NAME (S CHARS"-STARTING P CHARS PAST L[0]),
53390000 % INTRINSIC RETURNS THE INDEX FOR THE BRANCH TO INTRINSIC FCTS IN THE
53400000 % INTERPRETER.  IF THE FUNCTION IS NOT AN INTRINSIC, 0 IS RETURNED.
53410000 INTEGER PROCEDURE INTRINSIC(L,P,S);
53420000 VALUE P, S;
53430000 ALPHA ARRAY L[0];
53440000 INTEGER P, S;
53450000 BEGIN LABEL FIN, NON;
53460000 ALPHA F;
53470000 INTEGER I;
53480000 BOOLEAN FX;
53490000 %
53500000 IF S > 7 OR S < 3 THEN GO TO NON;
53510000 F ← 0;
53520000 MV(S,L[P,W],P.C,F,8-S);
```

```
53530000 FX ← NOT BOOLEAN(F);
53540000 FOR I ← INTRINSFCT[S,0] STEP -1 UNTIL 1 DO
53550000     IF REAL(NOT(BOOLEAN(INTRINSFCT[S,I]) EQV FX)) = TEENYNEG THEN
53560000         BEGIN INTRINSIC ← INTRINSNDX[S,I];
53570000         I ← INTRINSNDX[S,I];    % FOR TRACING
53580000         GO TO FIN;
53590000         END;
53600000 NON: INTRINSIC ← 0;
53610000 FIN:
53620000 END INTRINSIC;
53630000 %********INTRINSIC*************************************************
53640000 %********LOADER****************************************************
53650000 % LOADER READS THE PROGRAM MATERIAL IN THE FILE PROGRAM, AND
53660000 % CAUSES THE PROGRAM TO BE COMPILED.  INSTRUCTIONS ARE PUT INTO A
53670000 % SINGLE ARRAY ROW AND PASSED TO COMPILE.  CONTROL CARDS ARE
53680000 % SIMILARLY PASSED TO PROCESSCONTROLCARD.  WHEN AN END CARD IS FOUND,
53690000 % THE WORK ON THE FILE IS TERMINATED.  IF THIS WAS THE GLOBAL FILE,
53700000 % COMPILATION IS COMPLETED BY INITIALIZING VARIOUS GLOBAL ITEMS,
53710000 % AND THE LOADER RETURNS TO THE CONTROL CODE AT THE END OF THE
53720000 % DECK, CAUSING THE INTERPRETER TO BE CALLED.  NOTE THAT THE HIGHEST
53730000 % LEVEL OF RECURSIVE CALLS OF LOADER MAY BE USING THE TELETYPE (IF
53740000 % PROGRAMFROMREMOTE IS TRUE), IN WHICH CASE THE PROGRAM FILE IS NOT
53750000 % USED.
53760000 PROCEDURE LOADER(PROGRAM); FILE PROGRAM;
53770000 %
53780000 %
53790000 BEGIN
53800000 INTEGER
53810000     I,
53820000     J;
53830000 ALPHA ARRAY
53840000     INST[0:1022];
53850000 LABEL
53860000     A,
53870000     CONT,
53880000     CONTROL,
53890000     EN,
53900000     EOF,
53910000     FIN,
53920000     GOTINST,
53930000     INITIALIZE,
53940000     LABELED,
53950000     LOAD,
53960000     RD,
53970000     TTLOST,
53980000     UNLABELED;
53990000 SWITCH INSTBR ←
54000000     RD,
54010000     CONT,
54020000     CONTROL,
54030000     EN,
54040000     UNLABELED,
54050000     LABELED;
54060000 %
54070000 RD: IF DEATH THEN GO TO FIN;
54080000     INSTSIZE ← 0;
54090000     IF PROGRAMFROMREMOTE THEN
```

```
54100000     BEGIN WRITE(DCWRITE[*],FCRLF); IF OUTPUT THEN;
54110000     IF INPUT THEN
54120000        BEGIN CLEAR(BUFOUT,10);
54130000        WHILE CHAR(DCREAD[(I1+DCSIZE-1),W],I1,C) = "#" DO
54140000           BEGIN
54150000           WRITE(DCWRITE[*],FAGAIN);
54160000           IF NOT OUTPUT THEN GO TO EOF;
54170000           IF NOT INPUT THEN GO TO EOF;
54180000           END;
54190000        MOVE(DCSIZE,DCREAD[*],0,INST[*],0);
54200000        MOVE(80-DCSIZE,BUFOUT[*],0,INST[DCSIZE,W],DCSIZE,C);
54210000        INSTSIZE ← DCSIZE;
54220000        I ← MAX(80,INSTSIZE);
54230000        MV(3,STOPPER,5,INST[I,W],I,C);
54240000        LST(INST);
54250000        END ELSE GO TO EOF
54260000     END DATACOMM INPUT ELSE
54270000        BEGIN
54280000        IF NOT BUFFERFULL THEN
54290000           BEGIN READ(PROGRAM,10,BUFFER[*]) [EOF];
54300000           BUFFERFULL ← TRUE;
54310000           END;
54320000        IF CONVERTF THEN CONVERT(BUFFER);
54330000        LST(BUFFER);
54340000        MOVE(FIELDSIZE,BUFFER[*],0,INST[*],0);
54350000        BUFFERFULL ← FALSE;
54360000        INSTSIZE ← FIELDSIZE;
54370000        READ(PROGRAM,10,BUFFER[*]) [EOF];
54380000        BUFFERFULL ← TRUE;
54390000        WHILE CHAR(BUFFER[*],0) = "." DO
54400000           BEGIN IF CONVERTF THEN CONVERT(BUFFER);
54410000           LST(BUFFER);
54420000           MV(1,BLANK,7,BUFFER[*],0);
54430000           MOVE(FIELDSIZE,BUFFER[*],0,INST[INSTSIZE,W],INSTSIZE,C);
54440000           BUFFERFULL ← FALSE;
54450000           INSTSIZE ← INSTSIZE + FIELDSIZE;
54460000           READ(PROGRAM,10,BUFFER[*]) [EOF];
54470000           BUFFERFULL ← TRUE;
54480000           END;
54490000        MV(3,STOPPER,5,INST[INSTSIZE,W],INSTSIZE,C);
54500000        END;
54510000  GOTINST:
54520000     IF SLASTLABEL = 0 THEN IF CARDTYPE(INST[*]) ≥ 5 THEN
54530000        BEGIN
54540000        IF NOT SYMBTABLSETUP THEN INITIALIZESYMBTABL;
54550000        SLASTLABEL ← ENTERST(SCANCHAR(" "," ",INST[*],0),
54560000           INST[*],0,"INST");
54570000        IF INSTNUM = 0 THEN ENTRY ← SLASTLABEL;
54580000        PTR ← 2;
54590000        GO TO LOAD;
54600000        END;
54610000     GO TO INSTBR[CARDTYPE(INST[*])];
54620000     SYNTAXERR(22,0);
54630000     COMMENT UNRECOGNIZED INSTRUCTION TYPE;
54640000     GO TO RD;
54650000  CONTROL:
54660000     PROCESSCONTROLCARD(INST);
```

```
54670000     GO TO RD;
54680000 CONT:
54690000     MESSAGEO(15);
54700000     MESSAGETT@(15);
54710000     COMMENT ILLEGAL CONTINUATION--IGNORED;
54720000     GO TO RD;
54730000 LABELED:
54740000     I ← ENTERST(SCANCHAR(" "," ",INST[*],0),INST[*],0,"INST");
54750000     IF NEXTSEGMENT = 0 THEN NEXTSEGMENT ← I;
54760000     IF GTF AND GTS THEN GO TO A;
54770000     AA ← NEXTSEGMENT & "/:"[24:36:12];
54780000     J ← IF GT THEN 3 ELSE 4;
54790000     STORECHARS(J,AA,8-J);
54800000     IF NOT GTS THEN
54810000        BEGIN SLOC ← PTR - 3;
54820000        MV(2,SLOC,6,CODE[MARKER.W],MARKER.C+3);
54830000        END;
54840000     IF NOT GTF THEN
54850000        BEGIN FLOC ← PTR - 3;
54860000        MV(2,FLOC,6,CODE[MARKER.W],MARKER.C+5);
54870000        END;
54880000     GT ← GTS ← GTF ← TRUE;
54890000 A: AB ← STRING(PTR,SLASTLABEL);
54900000     MV(2,NEXTSEGMENT,6,CODE[0],0); % FILL IN LOC OF NEXT SEGMENT.
54910000     NEXTSEGMENT ← 0;      % RESET FOR NEXT SEGMENT
54920000     MOVE(PTR,CODE[*],0,FIRSTCHAR(AB));
54930000     VALU[SLASTLABEL].LOC ← AB;
54940000     IF PTR > MAXINSTSIZE THEN MAXINSTSIZE ← PTR;
54950000     % CHECK WHETHER LABEL HAS OCCURRED BEFORE:
54960000     IF (AA←VALU[I]).CH ≥ 3 THEN
54970000        BEGIN MESSAGEO(16);
54980000        MESSAGETTI(22,INSTNUM);
54990000        MV(2,FIRSTCHAR(AA),NEXTSEGMENT,6);
55000000        MV(2,NEXTSEGMENT,6,CODE[0],0);
55010000        END ELSE CODE[0] ← NEXTSEGMENT ← 0;
55020000     PTR ← 2;
55030000     SLASTLABEL ← I;
55040000     GO TO LOAD;
55050000 UNLABELED:
55060000     IF GTS AND GTF THEN
55070000        BEGIN MESSAGEO(17);
55080000        MESSAGETTI(16,INSTNUM);
55090000        MESSAGES ← MESSAGES + 1;
55100000        END;
55110000 LOAD:
55120000     MARKER ← PTR;
55130000     AA ← INSTNUM & QMARK[30:42:6];
55140000     STORECHARS(3,AA,5);
55150000     SLOC ← FLOC ← 0;
55160000     STORECHARS(5,0,3);
55170000     COMPILE(INST);
55180000     INSTNUM ← INSTNUM + 1;
55190000     GO TO RD;
55200000 EOF:
55210000     % EITHER EOF ON PROGRAM FILE OR NO INPUT FROM TELETYPE:
55220000     IF PROGRAMFROMREMOTE THEN % NO INPUT FROM TELETYPE:
55230000        BEGIN
```

```
55240000         WRITE(DCWRITE[*],FNOINPUT,QMARK);
55250000         IF NOT OUTPUT THEN GO TO TTLOST;
55260000         IF NOT INPUT THEN GO TO TTLOST;
55270000         WHILE J←CHAR(DCREAD[*],SKIPCHAR(" ",DCREAD[*],0)) = "Y"
55280000               AND J ≠ "N" DO
55290000            BEGIN WRITE(DCWRITE[*],FYESORNO);
55300000            IF NOT OUTPUT THEN GO TO TTLOST;
55310000            IF NOT INPUT THEN GO TO TTLOST;
55320000            END;
55330000         GO TO IF J = "Y" THEN RD ELSE TTLOST;
55340000         END;
55350000      IF LOADERLEVEL > 0 THEN  % EOF FOR SUB-PROGRAM FILE
55360000         BEGIN
55370000         IF INSTSIZE > 0 THEN GO TO GOTINST;
55380000         % LAST CARD HAS BEEN COMPILED:
55390000         GO TO FIN;
55400000         END;
55410000      % EOF AT LEVEL 0:
55420000      IF INSTSIZE > 0 THEN GO TO GOTINST;
55430000      % NO CARD IN INST[*]--END CARD MISSING.
55440000      WRITE(PRINT,MESSAGE[25]);
55450000      EXECUTE ← FALSE;
55460000      GO TO FIN;
55470000 TTLOST:
55480000      MESSAGE0(9);
55490000      EXECUTE ← FALSE;
55500000      GO TO FIN;
55510000 EN:       % END CARD ENCOUNTERED:
55520000      IF SLASTLABEL ≠ 0 THEN    % FINISH UP LAST INSTRUCTION:
55530000      BEGIN DEFINE DUMMY=#;
55540000      IF NOT (GTF AND GTS) THEN
55550000         BEGIN I ← IF GT THEN 3 ELSE 4;
55560000         IF NEXTSEGMENT = 0 THEN
55570000            BEGIN AA ← "/-E";
55580000            I ← IF GT THEN 2 ELSE 3;
55590000            J ← 2;
55600000            END ELSE
55610000         % PATCH--TIE TO NEXT SEGMENT
55620000            BEGIN I ← IF GT THEN 3 ELSE 4;
55630000            AA ← NEXTSEGMENT & "/:"[24:36:12];
55640000            J ← 3;
55650000            END;
55660000         STORECHARS(I,AA,8-I);
55670000      IF NOT GTS THEN
55680000         BEGIN SLOC ← PTR - J;
55690000         MOVE(2,SLOC,6,CODE[MARKER.W],MARKER.C+3);
55700000         END;
55710000      IF NOT GTF THEN
55720000         BEGIN FLOC ← PTR - J;
55730000         MOVE(2,FLOC,6,CODE[MARKER.W],MARKER.C+5);
55740000         END;
55750000         END;
55760000      TEMP ← STRING(PTR,SLASTLABEL);
55770000      MOVE(PTR,CODE[*],0,FIRSTCHAR(AA));
55780000      VALU[SLASTLABEL],LOC ← AA;
55790000      SLASTLABEL ← 0;
55800000      IF PTR > MAXINSTSIZE THEN MAXINSTSIZE ← PTR;
```

```
55810000     END;
55820000 % FIND ENTRY POINT, IF ANY:
55830000     P ← 4 + SKIPCHAR(" ",INST[*],4);
55840000     IF P < INSTSIZE THEN
55850000        BEGIN TEMP ← P;
55860000        P ← P + SCANCHAR(" ",QMARK,INST[P,W],P,C);
55870000        IF P > INSTSIZE THEN P ← INSTSIZE;
55880000        ENTRY ← ENTERST(P-TEMP,INST[*],TEMP,"INST");
55890000        END;
55900000 INITIALIZE: IF LOADERLEVEL = 0 THEN     % FINISH UP LOADING:
55910000     BEGIN ALPHA ARRAY SPECIALINST[0:1];
55920000     IF NOT(IOEOF[2] OR PROGRAMFROMREMOTE) THEN
55930000        BEGIN MOVE(80,BUFFER[*],0,NEXTRECORD[2,*],0);
55940000        LOOKF[2] ← TRUE;
55950000        END;
55960000     % CHECK FOR NON-EXISTENT LABELS:
55970000     IF PRINTMESSAGES THEN
55980000        FOR I ← 0 STEP 1 UNTIL STRMAX DO
55990000           FOR J ← 0 STEP 1 UNTIL STWMAX DO
56000000              IF (AA←NAMTABL[I,J]).[1:2] = 3 THEN
56010000                 IF VALTABL[I,J].S < 9 THEN
56020000                    BEGIN WRITE(BUFOUT[*],MESSAGE[50]);
56030000                    MOVE(MIN(AA.S,115),FIRSTCHAR(AA),BUFOUT[*],19);
56040000                    WRITE(PRINT,17,BUFOUT[*]);
56050000                    END;
56060000     IF INFORM THEN INFORMA(0,ENTRY);
56070000     IF VALU[ENTRY].S ≤ 10 THEN
56080000        BEGIN MESSAGEO(18); MESSAGETTO(18);
56090000        EXECUTE ← FALSE;
56100000        GO TO FIN;
56110000        END CHECKING ENTRY POINT;
56120000     %
56130000     % INITIALIZATION OF SPECIAL SNOBOL IDENTIFIERS:
56140000     %
56150000     % RETURN INSTRUCTION:
56160000     I ← ENTERST(6,WORDS[*],54,"INST");
56170000     IF VALU[I].S = 0 THEN
56180000        BEGIN FILL SPECIALINST[*] WITH OCT0000140000000000,"0,R00000";
56190000        SPECIALINST[0].[18:12] ← INSTNUM;
56200000        AA ← STRING(11,I);
56210000        MV(11,SPECIALINST[*],0,FIRSTCHAR(AA));
56220000        VALU[I].LOC ← AA;
56230000        END;
56240000     %
56250000     % FRETURN INSTRUCTION:
56260000     %
56270000     I ← ENTERST(7,WORDS[*],53,"INST");
56280000     IF VALU[I].S = 0 THEN
56290000        BEGIN FILL SPECIALINST[*] WITH OCT0000140000000000,"0,F00000";
56300000        SPECIALINST[0].[18:12] ← INSTNUM;
56310000        AA ← STRING(11,I);
56320000        MV(11,SPECIALINST[*],0,FIRSTCHAR(AA));
56330000        VALU[I].LOC ← AA;
56340000        END;
56350000     %
56360000     % INITIALIZE QUOTE TO """
56370000     %
```

```
56380000    I ← ENTERST(5,WORDS[*],135,"SYMB");
56390000    VALU[I].LOC ← AA ← STRING(1,I);
56400000    MOVE(1,QUOTE,7,FIRSTCHAR(AA));
56410000    %
56420000    % INITIALIZE QMARK TO ILLEGAL CHAR
56430000    %
56440000    I ← ENTERST(5,WORDS[*],140,"SYMB");
56450000    VALU[I].LOC ← AA ← STRING(1,I);
56460000    MOVE(1,QMARK,7,FIRSTCHAR(AA));
56470000    %
56480000    % INITIALIZE ARROW TO "←"
56490000    %
56500000    I ← ENTERST(5,WORDS[*],145,"SYMB");
56510000    VALU[I].LOC ← AA ← STRING(1,I);
56520000    MOVE(1,ARROW,7,FIRSTCHAR(AA));
56530000    %
56540000    % THE INPUT STRING READ GIVES THE REST OF THE FILE PROGRAM.
56550000    %
56560000    I ← ENTERST(4,WORDS[*],150,"SYMB");
56570000    VALU[I].IOUSE ← 2;
56580000    IO[I] ← 0 & 2 CFILNO & 1 CIOTYPE & 2 CIOUSE;
56590000    IOUSAGE[2] ← I & 2 CIOUSE & 1 CIOTYPE;
56600000    %
56610000    % INPUT STRING CARD IS FROM THE CARD-IMAGE FILE NAMED CARD.
56620000    %
56630000    I ← ENTERST(4,WORDS[*],174,"SYMB");
56640000    VALU[I].IOUSE ← 3;
56650000    IO[I] ← 0 & 1 CFILNO & 1 CIOTYPE & 3 CIOUSE;
56660000    IOUSAGE[1] ← I & 3 CIOUSE & 1 CIOTYPE;
56670000    %
56680000    % PRINT IS OUTPUT TO THE LINE PRINTER FILE PRINT
56690000    %
56700000    I ← ENTERST(5,WORDS[*],154,"SYMB");
56710000    PRINTLOC ← I;    % SAVE VALUE FOR TRACE FUNCTION
56720000    IOSPACE[3] ← 1;
56730000    VALU[I].IOUSE ← 1;
56740000    IO[I] ← 0 & 3 CFILNO & 1 CIOTYPE & 1 CIOUSE;
56750000    IOUSAGE[3] ← I & 1 CIOUSE & 1 CIOTYPE;
56760000    %
56770000    % PUNCH IS OUTPUT TO THE CARD PUNCH
56780000    %
56790000    I ← ENTERST(5,WORDS[*],159,"SYMB");
56800000    VALU[I].IOUSE ← 1;
56810000    IO[I] ← 0 & 4 CFILNO & 1 CIOTYPE & 1 CIOUSE;
56820000    IOUSAGE[4] ← I & 1 CIOUSE & 1 CIOTYPE;
56830000    %
56840000    % LOOK IS NON-READING INPUT FROM THE REST OF PROGRAM.
56850000    %
56860000    I ← ENTERST(4,WORDS[*],164,"SYMB");
56870000    VALU[I].IOUSE ← 2;
56880000    IO[I] ← 0 & 2 CFILNO & 2 CIOTYPE & 2 CIOUSE;
56890000    %
56900000    % SYSPOT--SEE CDC 3600 SNOBOL
56910000    %
56920000    I ← ENTERST(6,WORDS[*],168,"SYMB");
56930000    VALU[I].IOUSE ← 1;
56940000    IO[I] ← 0 & 3 CFILNO & 5 CIOTYPE & 1 CIOUSE;
```

```
56950000     %
56960000     % NEWDISK IS OUTPUT TO THE NEW DISK FILE NEWDISK
56970000     I ← ENTERST(7,WORDS[*],178,"SYMB");
56980000     VALU[I].IOUSE ← 3;
56990000     IO[I] ← 0 & 5 CFILNO & 1 CIOTYPE & 3 CIOUSE;
57000000     IOUSAGE[5] ← I & 3 CIOUSE & 1 CIOTYPE;
57010000     %
57020000     % INITIALIZE TELETYPE I/O STRING:
57030000     % I/O ON TELETYPES IS THROUGH THE STRING TELETYPE, WHICH
57040000     % FAILS ON INPUT IF THE WAITING TIME IS EXCEEDED, AND FAILS
57050000     % ON OUTPUT IF THE WAITING TIME IS EXCEEDED OR IF THE
57060000     % BREAK KEY IS USED.
57070000     I ← ENTERST(8,WORDS[*],96,"SYMB");
57080000     VALU[I].IOUSE ← 3;
57090000     IO[I] ← 0 & 0 CFILNO & 3 CIOTYPE & 3 CIOUSE;
57100000     %
57110000     FILL IOSIZE[*] WITH 80,80,132,80,80;
57120000     FILL RECORD[*] WITH -1,-1,-1,-1,-1,-1,-1,-1,-1;
57130000     END;
57140000 FIN:
57150000     IF DEATH THEN EXECUTE ← FALSE;
57160000 END LOADER;
57170000 %********LOADER*************************************************************
57180000 %********LOADLIBRARY********************************************************
57190000 % THIS PROCEDURE LOADS A LIBRARY ("OBJECT") FILE CREATED BY A =LIBRARY
57200000 % CARD DURING AN EARLIER RUN.
57210000 PROCEDURE LOADLIBRARY(MFID,FID);
57220000 VALUE MFID, FID;
57230000 ALPHA MFID, FID;
57240000 BEGIN INTEGER I, J;
57250000 REAL L1, L2, L3;
57260000 BOOLEAN B1, B2, B3;
57270000 ALPHA ARRAY X[0:7];
57280000 MONITOR INDEX;
57290000 LABEL FAIL, FIN;
57300000 FILE IN LIB DISK SERIAL (15,6,60);
57310000 %
57320000 INDEX ← FAIL;
57330000 FILL LIB WITH MFID, FID;
57340000 SEARCH(LIB,X[*]);
57350000 IF X[0] ≤ 0 THEN
57360000    BEGIN MESSAGE0(11);
57370000    MESSAGETTO(11);
57380000    GO TO FIN;
57390000    END;
57400000 READ(LIB,4,BUFOUT[*])  [FAIL:FAIL];
57410000 WRITE(X[*],FLIBO,ENTIER(VERSION));
57420000 % TEST VALIDITY OF FILE--SHOULD MATCH FLIBO:
57430000 IF NOT EQ(32,BUFOUT[*],0,X[*],0) THEN  % NOT A VALID LIBE FILE
57440000    BEGIN IF EQ(20,BUFOUT[*],0,X[*],0) THEN  % CREATED BY EARLIER VERSION
57450000       BEGIN WRITE(PRINT,FLIBOLDLP);
57460000       WRITE(DCWRITE[*],FLIBOLDTT);
57470000       IF DATACOMF THEN IF OUTPUT THEN;
57480000       END;
57490000    GO TO FIN;
57500000    END;
57510000 READ(LIB,FLIB1,SCATTERNO,MAXINSTSIZE,B1,L1,B2,L2,B3,L3) [FAIL:FAIL];
```

```
57520000 IF SCATTERNO > 15 OR MAXINSTSIZE > 8181 THEN GO TO FAIL;
57530000 IF B1 THEN BEGIN CPULIMITEXISTS←TRUE; CPULIMIT←L1; END;
57540000 IF B2 THEN BEGIN IOLIMITEXISTS←TRUE; IOLIMIT←L2; END;
57550000 IF B3 THEN BEGIN RULELIMITEXISTS←TRUE; RULELIMIT←L3; END;
57560000 READ(LIB,FLIB2,ENTRY,FOR I←0 STEP 1 UNTIL 15 DO USEDST[I],INSTNUM)
57570000       [FAIL:FAIL];
57580000 IF ENTRY > 4095 THEN GO TO FAIL;
57590000 READ(LIB,FLIB3,FOR I←0 STEP 1 UNTIL 15 DO NEXTCELL[I])[FAIL:FAIL];
57600000 FOR I ← 0 STEP 1 UNTIL 15 DO
57610000    IF USEDST[I] THEN
57620000       FOR J ← 0 STEP 2 UNTIL 254 DO
57630000          READ(LIB,FLIB4,NAMTABL[I,J],VALTABL[I,J],IOTABL[I,J],
57640000             NAMTABL[I,J+1],VALTABL[I,J+1],IOTABL[I,J+1])[FAIL:FAIL];
57650000 READ(LIB,FLIB5,FOR I←0 STEP 1 UNTIL 31 DO USEDROW[I])[FAIL:FAIL];
57660000 READ(LIB,FLIB6,FOR I←0 STEP 1 UNTIL 31 DO DPNTR[I])[FAIL:FAIL];
57670000 FOR I ← 0 STEP 1 UNTIL 31 DO IF DPNTR[I] > 8183 THEN GO TO FAIL;
57680000 FOR I ← 0 STEP 1 WHILE USEDROW[I] DO
57690000    FOR J ← 0 STEP 6 UNTIL DPNTR[I].W DO
57700000       BEGIN READ(LIB,6,X[*])[FAIL:FAIL];
57710000       IF J < 1020
57720000          THEN MOVEWDS(6,X[*],DATA[I,J])
57730000          ELSE MOVEWDS(3,X[*],DATA[I,J]);
57740000       END;
57750000 SYMBTABLSETUP ← TRUE;
57760000 CHECKSYMBTABL;
57770000 GO TO FIN;
57780000 %
57790000 FAIL: MESSAGE0(5);
57800000    MESSAGETTO(5);
57810000    DEATH ← TRUE;
57820000 FIN:
57830000 END LOADLIBRARY;
57840000 %********LOADLIBRARY*************************************************
57850000 %********LST*********************************************************
57860000 % LST LISTS THE CARD-IMAGE IN A[*].
57870000 PROCEDURE LST(A);
57880000    ALPHA ARRAY A[0];
57890000 BEGIN
57900000 INTEGER I;
57910000 %
57920000 IF CARDTYPE(A[*]) ≥ 4
57930000       THEN WRITE(BUFOUT[*],FI7,INSTNUM)
57940000       ELSE WRITE(BUFOUT[*],FBL);
57950000 FOR I ← LISTSPACES STEP -1 UNTIL 1 DO WRITE(PRINT,FBL);
57960000 MOVE(80,A[*],0,BUFOUT[*],8);
57970000 IF LSTF THEN WRITE(PRINT,12,BUFOUT[*]);
57980000 IF PUNCHF THEN WRITE(PUNCH,10,A[*]);
57990000 IF DCLIST THEN IF DATACOMF THEN
58000000    BEGIN FOR I ← 8 STEP -1 WHILE I ≥ 0 AND
58010000       EQUAL(8,BLANKS,0,BUFOUT[I],0) DO;
58020000    WRITE(DCWRITE[*],FVCRLF,8×I+1);
58030000    MOVE(8×I,BUFOUT[*],0,DCWRITE[*],0);
58040000    IF OUTPUT THEN;
58050000    END;
58060000 END LST;
58070000 %********LST*********************************************************
58080000 %********MESSAGE PROCEDURES******************************************
```

```
58090000 PROCEDURE MESSAGEO(I);
58100000 VALUE I; INTEGER I;
58110000 BEGIN SWITCH FORMAT MSGO ←
58120000    ("**PROGRAM SEGMENT TOO LONG--INSERT DUMMY LABEL AFTER LAST ",
58130000    "LABEL AND BEFORE THIS INSTRUCTION."),                          %00
58140000    (//"**DEFINE FAILURE--ERROR IN LOCAL VARIABLE LIST"),           %01
58150000    (//"**DEFINE FAILURE--ERROR IN FIRST ARGUMENT"),                %02
58160000    (//"**DEFINE FAILURE--NO ( IN FIRST ARGUMENT"),                 %03
58170000    (//"**DEFINE FAILURE--NO FIRST ARGUMENT"),                      %04
58180000    (//"**UNABLE TO READ LIBRARY FILE"),                            %05
58190000    (/"THE NAMES IN USE ARE:"/),                                    %06
58200000    (/"THE LABELS ARE:"/),                                          %07
58210000    (/"**DUMMY FMT--MSGO[8]"),
58220000    (//"**TELETYPE LOST--QUITTING."),                               %09
58230000    (//"**INVALID PARAMETER."),                                     %10
58240000    ("**FILE NOT AVAILABLE"),                                       %11
58250000    (/"**DUMMY FORMAT MSGO[12]"//),                                 %12
58260000    ("**FUNCTION ALREADY DEFINED--FORMER VALUE LOST"),              %13
58270000    (//"**SYMBOL TABLE FULL"),                                      %14
58280000    ("**ILLEGAL CONTINUATION CARD"),                                %15
58290000    ("**LABEL ALREADY DEFINED--FORMER VALUE LOST"),                 %16
58300000    ("**THIS STATEMENT CAN NOT BE REACHED"),                        %17
58310000    (///"**ENTRY POINT UNDEFINED."///),                             %18
58320000    ("**MISSING QUOTE"),                                            %19
58330000    ("**UNRECOGNIZED CONTROL CARD"),                                %20
58340000    ("**MISSING PARAMETER"),                                        %21
58350000    ("**ATTEMPTED RETURN WITH NO FUNCTION CALLED"),                 %22
58360000    (/"**END FORMAT--MSGO");
58370000 WRITE(PRINT,MSGO[I]);
58380000 END MESSAGEO;
58390000 %*************************************
58400000 PROCEDURE MESSAGEAI(I,P1,P2);
58410000 VALUE I, P1, P2;
58420000 INTEGER I, P2;
58430000 ALPHA P1;
58440000 BEGIN SWITCH FORMAT MSGAI ←
58450000    (//"**NON-NUMERIC ARGUMENT TO ,",A2,"() IN STATEMENT ",I4),
58460000    (//"**END FORMAT--MSGAI",A6,I10);
58470000 WRITE(PRINT,MSGAI[I],P1,P2);
58480000 END MESSAGEAI;
58490000 %*************************************
58500000 PROCEDURE MESSAGEI(I,P);
58510000 VALUE I, P;
58520000 INTEGER I, P;
58530000 BEGIN SWITCH FORMAT MSGI ←
58540000    (//"**INTEGER OVERFLOW IN STATEMENT ",I6),                       %00
58550000    (//"**ATTEMPTED EXECUTION OF INSTRUCTION WITH SYNTAX ERROR--",
58560000       "STATEMENT ",I6),                                            %01
58570000    (//"**FAILURE OF GO-TO PART IN STATEMENT ",I6),                  %02
58580000    (//"**ATTEMPTED REPLACEMENT WITH STRING REFERENCE USED IN STRING"
58590000       " VARIABLE--STATEMENT ",I6/X10,"REPLACEMENT NOT ATTEMPTED."//),%03
58600000    (//"**PROGRAM LIMIT REACHED IN STATEMENT ",I6),                  %04
58610000    ("**CPU TIME LIMIT = ",I12," SECONDS."),                        %05
58620000    ("**I/O TIME LIMIT = ",I12," SECONDS."),                        %06
58630000    ("**RULE LIMIT = ",I12),                                        %07
58640000    (//"**ILLEGAL SECOND PARAMETER TO CLOSE IN STATEMENT ",I6),      %08
58650000    (//"**DEFINE FAILURE IN STATEMENT ",I6),                        %09
```

```
58660000     (//"**DUMP REQUESTED IN STATEMENT ",I6," !"),                          %10
58670000     (//"**FUNCTION DEPTH BEYOND SYSTEM CAPACITY IN STATEMENT ",I6),        %11
58680000     (//"**OUT OF SPACE IN STRING STORAGE AREA--UNABLE TO FIND ROOM FOR"
58690000        " A STRING OF ",I6," CHARACTERS.",//),                              %12
58700000     (//"**PROGRAM DIED IN STATEMENT ",I6/),                                %13
58710000     (//"**UNABLE TO OPEN FILE IN STATEMENT ",I6,"--TOO MANY FILES ",
58720000        "ALREADY IN USE."/),                                                %14
58730000     (//"**FILE() ERROR IN STATEMENT ",I6,"--I/O TYPE (2ND PARAMETER)",
58740000        " MUST BE NULL OR START WITH ","""","I",""","," OR ",""","O",""""),%15
58750000     (//"**FILE() ERROR IN STATEMENT ",I6,"--NUMBER OF BUFFERS (3RD ",
58760000        "PARAMETER) MUST BE NUMERIC AND ≥ 1"),                              %16
58770000     (//"**NON-NUMERIC ARITHMETIC OPERAND IN STATEMENT ",I6),               %17
58780000     (//"**DIVIDE BY ZERO IN STATEMENT ",I6),                               %18
58790000     (//"**INDIRECT ON NULL STRING IN STATEMENT ",I6),                      %19
58800000     (//"**PUSH-DOWN STACK OVERPOPPED AT FUNCTION RETURN IN STATEMENT ",
58810000        I6),                                                                %20
58820000     (/"**FUNCTION REDEFINED IN STATEMENT ",I6),                            %21
58830000     (/"**MAX NUMBER OF PARAMS & LOCAL VARIABLES IS ",I3),                  %22
58840000     (//"**END FORMAT--MSGI",I10);
58850000 WRITE(PRINT,MSGI[I],P);
58860000 END MESSAGEI;
58870000 %*************************************
58880000 PROCEDURE MESSAGETTO(I);
58890000     VALUE I;
58900000     INTEGER I;
58910000 BEGIN SWITCH FORMAT MSGTO ←
58920000     ("≤≠**PROGRAM SEGMENT TOO LONG←"),                                     %00
58930000     ("≤≠**DEFINE FAILURE--ERROR IN LOCAL VARIABLE LIST←"),                 %01
58940000     ("≤≠**DEFINE FAILURE--ERROR IN FIRST ARGUMENT←"),                      %02
58950000     ("≤≠**DEFINE FAILURE--NO ( IN FIRST ARG←"),                            %03
58960000     ("≤≠**DEFINE FAILURE--NO FIRST ARGUMENT←"),                            %04
58970000     ("≤≠BAD LIBRARY FILE≤≠←"),                                             %05
58980000     ("≤≠DUMP() CALLED≤≠←"),                                                %06
58990000     ("≤≠≠SNOBOL WIPED OUT--SORRY≤≠≠←"),                                    %07
59000000     ("≤≠INTERPRETER STACK OVERFLOW--STATEMENT TOO COMPLEX≤≠←"),             %08
59010000     ("≤≠DUMMY FMT--MSGTO[9]≤≠≠←"),                                         %09
59020000     ("≤≠**INVALID PARAMETER←"),                                            %10
59030000     ("≤≠FILE NOT AVAILABLE≤≠←"),                                           %11
59040000     ("≤≠DUMMY FMT MSGTO[12]--SHOW TO SYSTEM AUTHORS.≤≠≠"),                 %12
59050000     ("≤≠**FUNCTION ALREADY DEFINED--FORMER VALUE LOST←"),                  %13
59060000     ("≤≠SYMBOL TABLE FULL≤≠←"),                                            %14
59070000     ("≤≠ILLEGAL CONTINUATION CARD≤≠←"),                                    %15
59080000     ("≤≠DUMMY FMT--MSGTO[16]≤≠≠←"),                                        %16
59090000     ("≤≠DUMMY FMT--MSGTO[17]≤≠≠←"),                                        %17
59100000     ("≤≠≠ENTRY POINT UNDEFINED≤≠←"),                                       %18
59110000     ("≤≠MISSING QUOTE≤≠←"),                                                %19
59120000     ("≤≠ILLEGAL CONTROL CARD≤≠←"),                                         %20
59130000     ("≤≠MISSING PARAMETER≤≠←"),                                            %21
59140000     ("≤≠≠END FMT MSGTO≤≠≠←");
59150000 IF DATACOMF THEN
59160000     BEGIN WRITE(DCWRITE[*],MSGTO[I]);
59170000     IF OUTPUT THEN;
59180000     END;
59190000 END MESSAGETTO;
59200000 %*************************************
59210000 PROCEDURE MESSAGETTA(I,P);
59220000     VALUE I, P;
```

```
59230000     INTEGER I;
59240000     ALPHA P;
59250000 BEGIN SWITCH FORMAT MSGTA ←
59260000     ("≤≠EH",A1,"≤≠←"),                                          %00
59270000     ("≤≠END FMT MSGTA ",A6,"≤≠≠←");
59280000 IF DATACOMF THEN
59290000     BEGIN WRITE(DCWRITE[*],MSGTA[I]);
59300000     IF OUTPUT THEN;
59310000     END;
59320000 END;
59330000 %*****************************************
59340000 PROCEDURE MESSAGETTAI(I,P1,P2);
59350000     VALUE I, P1, P2;
59360000     INTEGER I, P2;
59370000     ALPHA P1;
59380000 BEGIN SWITCH FORMAT MSGTAI ←
59390000     ("≤≠NON-NUMERIC ARGUMENT TO .",A2,"() IN STMT ",I*,"≤≠←"),      %00
59400000     ("≤≠≠END FMT MSGTAI ",I*,A6,"≤≠≠←");
59410000 IF DATACOMF THEN
59420000     BEGIN WRITE(DCWRITE[*],MSGTAI[I],P1,DIGITS(P2),P2);
59430000     IF OUTPUT THEN;
59440000     END;
59450000 END;
59460000 %*****************************************
59470000 PROCEDURE MESSAGETTI(I,P);
59480000     VALUE I, P;
59490000     INTEGER I, P;
59500000 BEGIN SWITCH FORMAT MSGTI ←
59510000     ("≤≠**INTEGER OVERFLOW IN STMT ",I*,"≤≠←"),                  %00
59520000     ("≤≠≠**ATTEMPTED EXECUTION OF STMT WITH SYNTAX ERROR--",
59530000       "STMT ",I*,"≤≠←"),                                        %01
59540000     ("≤≠≠**GO-TO PART IN STMT ",I*," FAILED≤≠←"),                %02
59550000     ("≤≠**ATTEMPTED REPLACEMENT WITH STRING REFERENCE USED IN STRING ",
59560000       "VARIABLE, IN STMT ",I*,"≤≠**REPLACEMENT NOT ATTEMPTED≤≠←"), %03
59570000     ("≤≠**PROGRAM LIMIT REACHED IN STATEMENT ",I*,"≤≠←"),        %04
59580000     ("CPU TIME LIMIT = ",I*," SECONDS≤≠←"),                      %05
59590000     ("I/O TIME LIMIT = ",I*," SECONDS≤≠←"),                      %06
59600000     ("RULE LIMIT = ",I*,"≤≠←"),                                  %07
59610000     ("≤≠ILLEGAL 2ND PARAM TO CLOSE IN STMT ",I*,"≤≠←"),          %08
59620000     ("≤≠DEFINE FAILURE IN STATEMENT ",I*,"≤≠←"),                 %09
59630000     ("≤≠≠FATAL ERROR IN STATEMENT ",I*,"≤≠←"),                   %10
59640000     ("≤≠≠**FUNCTION DEPTH BEYOND SYSTEM CAPACITY IN STMT ",I*,"≤≠←"), %11
59650000     ("≤≠**OUT OF SPACE--UNABLE TO FIND ROOM FOR STRING OF ",I*,
59660000       " CHARACTERS≤≠←"),                                        %12
59670000     ("≤≠**PROGRAM DIED IN STATEMENT ",I*,"≤≠≠←"),                %13
59680000     ("≤≠**ATTEMPTED RETURN WITH NO FUNCTION CALLED--STMT ",I*,"≤≠←"), %14
59690000     ("# ",I*,"≤≠←"),                                             %15
59700000     ("≤≠STMT. ",I*," CANT BE REACHED.≤≠←"),                      %16
59710000     ("≤≠NON-NUMERIC ARITHMETIC OPERAND IN STATEMENT ",I*,"≤≠←"),  %17
59720000     ("≤≠DIVIDE BY ZERO IN STATEMENT ",I*,"≤≠←"),                 %18
59730000     ("≤≠INDIRECT ON NULL STRING IN STMT ",I*,"≤≠←"),             %19
59740000     ("≤≠PUSH-DOWN STACK OVERPOPPED AT FCT RETURN IN STMT ",I*,"≤≠←"), %20
59750000     ("≤≠**FUNCTION RE-DEFINED IN STATEMENT ",I*,"≤≠←"),          %21
59760000     ("≤≠**LABEL RE-DEFINED IN STATEMENT ",I*,"≤≠←"),             %22
59770000     ("≤≠≠**NORMAL TERMINATION IN STATEMENT ",I*,".≤≠≠←"),        %23
59780000     ("≤≠MAX # OF PARAMS & LOC VARS IS ",I*,"≤≠←"),               %24
59790000     ("≤≠≠END FMT MSGTI ",I*,"≤≠≠←");
```

```
59800000 IF DATACOMF THEN
59810000    BEGIN WRITE(DCWRITE[*],MSGTI[I],DIGITS(P),P);
59820000    IF OUTPUT THEN;
59830000    END;
59840000 END;
59850000 %********MESSAGE PROCEDURES*************************************************
59860000 %********MIN***************************************************************
59870000 INTEGER PROCEDURE MIN(A,B); VALUE A,B; INTEGER A,B;
59880000 MIN ← IF A ≤ B THEN A ELSE B;
59890000 %********MIN***************************************************************
59900000 %********MNEMNO************************************************************
59910000 % THE STRING BETWEEN A[P,W],P,C AND THE NEXT BLANK IS SEARCHED FOR
59920000 % IN THE N-TH LIST OF MNEMONICS, AND THE APPROPRIATE INDEX IS RETURNED.
59930000 INTEGER PROCEDURE MNEMNO(N,A,P);
59940000    VALUE N;
59950000    INTEGER N, P;
59960000    ARRAY A[0];
59970000 BEGIN LABEL FIN;
59980000 INTEGER I, J;
59990000 ALPHA AA;
60000000 IF I←SCANCHAR(" ","←",A[P,W],P,C) > 6
60010000    THEN MNEMNO ← 0 ELSE
60020000    BEGIN AA ← 0;
60030000    MV(I,A[P,W],P,C,AA,8-I);
60040000    AA ← AA;
60050000    FOR J←MNEMONIC[N,0] STEP -1 UNTIL 1 DO
60060000       IF AA = MNEMONIC[N,J] THEN
60070000          BEGIN MNEMNO ← J;
60080000          P ← P + I;
60090000          GO TO FIN;
60100000          END;
60110000    MNEMNO ← 0;
60120000    END;
60130000 FIN:
60140000 END MNEMNO;
60150000 %********MNEMNO************************************************************
60160000 %********NEWCELL***********************************************************
60170000 % NEWCELL RETURNS THE ADDRESS OF AN AVAILABLE SYMBOL TABLE CELL--USING
60180000 % ROW I IF POSSIBLE.  CELLS GIVEN BY NEWCELL ARE "PERMANENT" IN THE
60190000 % SENSE THAT RETURNCELL IS NEEDED TO RETURN THEM TO AVAILABLE SPACE.
60200000 % COMPARE TEMPCELL, WHICH RETURNS A CELL THAT WILL ONLY LAST UNTIL
60210000 % THE END OF THE INSTRUCTION
60220000 ALPHA PROCEDURE NEWCELL(I);
60230000 VALUE I; INTEGER I;
60240000 BEGIN LABEL ROWFULL, RETURN;
60250000 INTEGER C;
60260000 IF NEXTCELL[I] = 0 THEN GO TO ROWFULL;
60270000 C ← NEXTCELL[I];
60280000    NEXTCELL[I] ← NAME[C].LINK;
60290000 GO TO RETURN;
60300000 ROWFULL: FOR I ← 0 STEP 1 WHILE I ≤ 15 AND USEDST[I] DO
60310000    IF NEXTCELL[I] ≠ 0 THEN
60320000       BEGIN C ← NEXTCELL[I];
60330000       NEXTCELL[I] ← NAME[C].LINK;
60340000       GO TO RETURN;
60350000       END;
60360000    IF I ≤ 15 THEN
```

```
60370000        BEGIN NEWSTROW(I);
60380000        C ← 0 & I CONCSTR;
60390000        GO TO RETURN;
60400000        END;
60410000 MESSAGE0(14);
60420000 MESSAGETTO(14);
60430000 DEATH ← TRUE;
60440000 C ← 0;
60450000 RETURN:
60460000 % IF NEXTCELL[I] > USEDCELL[I] THEN USEDCELL[I] ← NEXTCELL[I];
60470000 NAME[C] ← 0;
60480000 VALU[C] ← 0;
60490000 NEWCELL ← C;
60500000 % IF INFORM THEN INFORMIA(1,I,C);
60510000 END NEWCELL;
60520000 %********NEWCELL*************************************************************
60530000 %********NEWSTROW************************************************************
60540000 % NEWSTROW OPENS UP ROW I OF THE SYMBOL TABLE, AND INITIALIZES AN
60550000 % AVAILABLE SPACE LIST IN THIS ROW.
60560000 PROCEDURE NEWSTROW(I); VALUE I; INTEGER I;
60570000 BEGIN INTEGER J;
60580000 NAMTABL[I,0].LINK ← 1 & I CONCSTR;
60590000 FOR J ← 1 STEP 1 WHILE J < STWMAX DO
60600000    NAMTABL[I,J].LINK ← (J+1) & I CONCSTR;
60610000 USEDST[I] ← TRUE;
60620000 NEXTCELL[I] ← 1 & I CONCSTR;
60630000 END NEWSTROW;
60640000 %********NEWSTROW************************************************************
60650000 %********NULLARGS************************************************************
60660000 % INTERPRETER PROCEDURE:
60670000 % NULLARGS(N) PUTS N NULL STRINGS ON TOP OF THE STACK.
60680000 BOOLEAN PROCEDURE NULLARGS(N); VALUE N; INTEGER N;
60690000 BEGIN LABEL RETURN;
60700000 INTEGER I;
60710000 IF SP + N ≥ STACKSIZE THEN
60720000    BEGIN
60730000    INFORMI(4,INSTNO);
60740000    NULLARGS ← FALSE;
60750000    GO TO RETURN;
60760000    END;
60770000 FOR I ← SP+N-1 STEP -1 UNTIL SP DO
60780000    PST[I] ← PTYPE[I] ← PSIZE[I] ← 0;
60790000 SP ← SP + N;
60800000 NULLARGS ← TRUE;
60810000 RETURN:
60820000 END NULLARGS;
60830000 %********NULLARGS************************************************************
60840000 %********NUMVAL**************************************************************
60850000 % INTERPRETER PROCEDURE:
60860000 % IF THE STRING VALUE OF SYMB TABLE ENTRY ST IS NON-NUMERIC,
60870000 % FALSE IS RETURNED AND VAL IS UNALTERED.  IF THE STRING IS NUMERIC,
60880000 % THE VALUE OF THE STRING IS ASSIGNED TO VAL, AND TRUE IS RETURNED.
60890000 BOOLEAN PROCEDURE NUMVAL(ST,VAL);
60900000 VALUE ST; ALPHA ST; INTEGER VAL;
60910000 BEGIN BOOLEAN STREAM PROCEDURE NUMERIC(S,L,I); VALUE S,I;
60920000    BEGIN LABEL RETURN;
60930000    TALLY ← 0;
```

```
60940000        SI ← LOC S; SI ← SI + 7;
60950000     IF SC = "0" THEN
60960000        BEGIN TALLY ← 1; GO TO RETURN; END;
60970000     SI ← L; SI ← SI + I;
60980000     IF SC = "-" THEN
60990000        BEGIN SI ← SI + 1; TALLY ← S;
61000000        TALLY ← TALLY + 63; S ← TALLY; TALLY ← 0;
61010000        END;
61020000     S(IF SC < "0" THEN JUMP OUT TO RETURN;
61030000        IF SC > "9" THEN JUMP OUT TO RETURN;
61040000        SI ← SI + 1);
61050000     TALLY ← 1;
61060000     RETURN: NUMERIC ← TALLY;
61070000     END NUMERIC;
61080000 %
61090000 ST ← VALU[ST];
61100000 IF NUMERIC(STRINGLOC(ST)) THEN
61110000     BEGIN IF ST.CH < 4096
61120000        THEN READ(DATA[ST,R,*],FNUM,ST,CH,ST,S,VAL)
61130000        ELSE READ(DATA[ST,R,*],FINT,ST,W,ST,C,ST,S,VAL);
61140000        % THIS DISTINCTION IS NECESSARY BECAUSE WHEN AN "X*" FORMAT
61150000        % (AS IN FNUM) IS USED, ONLY THE LAST 12 BITS OF THE NUMBER
61160000        % ARE USED, AND A CH FIELD CONTAINS 13 BITS.  ON THE OTHER
61170000        % HAND, IF "*D" IS USED (AS IN FINT), AND THE NUMBER GIVEN
61180000        % IS ZERO, AN INFINITE LOOP OCCURS, SINCE 0 IS TAKEN TO MEAN
61190000        % "REPEAT INDEFINITELY", AND THE REST OF THE FORMAT WILL NEVER
61200000        % BE REACHED.  THUS, TWO CASES ARE NECESSARY.
61210000     NUMVAL ← TRUE;
61220000        END ELSE NUMVAL ← FALSE;
61230000 % IF INFORM THEN INFORMI(12,VAL);
61240000 END NUMVAL;
61250000 %********NUMVAL*****************************************************
61260000 %********OUTPUT*****************************************************
61270000 % TELETYPE OUTPUT PROCEDURE:
61280000 BOOLEAN PROCEDURE OUTPUT;
61290000 BEGIN
61300000        LABEL LOOP, NOGO, BROKE, BADNEWS, EXIT;
61310000        INTEGER  CNT, Q, R, TIMEX;
61320000        BOOLEAN GPMK;
61330000        REAL S;
61340000        ARRAY TANK[0:BUFOUTSIZE-1];
61350000        STREAM PROCEDURE STOPPER(A);
61360000             BEGIN DI←A; DI←DI+7; DS←LIT "←"  END STOPPER;
61370000        BOOLEAN STREAM PROCEDURE MOVE(A,B,C);  VALUE C;
61380000             BEGIN  SI←A; SI←SI+C; DI←B; TALLY←0;
61390000                  DCOUTCHAR(IF SC="←" THEN TALLY←1; DS←CHR);
61400000                  MOVE←TALLY
61410000             END MOVE;
61420000     % IF INFORM THEN INFORMII(5,STAT[USER].[9:4],STAT[USER].[14:4]);
61430000     IF NOT DATACOMF THEN GO TO EXIT;
61440000     OUTPUT ← TRUE;
61450000     IF BREAK THEN
61460000        BEGIN IF (BREAK←BOOLEAN(STATUS(STAT[USER],1).[26:1])) THEN
61470000           BEGIN WHEN(5);
61480000           BREAK ← BOOLEAN(STATUS(STAT[USER],1).[26:1]);
61490000           END;
61500000        IF BREAK THEN GO TO BADNEWS;
```

```
61510000       END;
61520000    STOPPER(DCWRITE[(ANSSIZE-1),W]);
61530000    TIMEX ← TIME(1);
61540000 LOOP:   Q←CNT DIV 8;  R←CNT MOD 8;
61550000      GPMK ← MOVE(DCWRITE[Q],TANK[1],R);
61560000    MV(1,ARROW,7,TANK[4],4);
61570000    WRITE(DCOUT(STAT[USER],0),BUFOUTSIZE,TANK[*]) [NOGO:BROKE];
61580000    IF BOOLEAN(STATUS(STAT[USER],1).[26:1]) THEN GO TO BROKE;
61590000    IF GPMK THEN GO TO EXIT;
61600000    CNT ← CNT + DCOUTCHAR;
61610000    GO TO LOOP;
61620000 BROKE:
61630000    BREAK ← TRUE;
61640000    OUTPUT ← FALSE;
61650000    GO TO EXIT;
61660000 NOGO:
61670000    % SOME FIELDS IN THE STATUS WORD ARE:
61680000    % [23:1] = 1 IF ABNORMAL CONDITION SENSED BY ADAPTER.
61690000    % [24:1] = 1 IF READ READY.
61700000    % [28:1] = 1 IF BUFFER OVERFLOW (ON INPUT).
61710000    % [30:1] = 1 IF NOT READY.
61720000    S ← STATUS(STAT[USER],0);
61730000    IF BOOLEAN(S.[30:1]) THEN GO TO BADNEWS;
61740000    IF TIME(1) - TIMEX > WAITTIME THEN GO TO BADNEWS;
61750000    IF S.[23:2]≠0 OR BOOLEAN(S.[28:1]) THEN WHEN(5);
61760000 %  WAIT 5 SECS IF WRITE DID NOT TIME OUT
61770000    GO TO LOOP;
61780000 BADNEWS:
61790000    OUTPUT ← FALSE;
61800000    FINDUSERS;
61810000 EXIT:   END OUTPUT;
61820000 %========== END DATA COM OUTPUT PROCEDURE ==============================
61830000 %********POP*************************************************************
61840000 BOOLEAN PROCEDURE POP(STLOC); VALUE STLOC; ALPHA STLOC;
61850000 BEGIN ALPHA T, ST;
61860000 INTEGER I;
61870000 % IF INFORM THEN INFORMA(2,STLOC);
61880000 ST ← VALU[STLOC].LINK;
61890000 IF ST = 0 THEN POP ← FALSE ELSE
61900000    BEGIN VALU[STLOC] ← T ← VALU[ST];
61910000    IF T.CH ≥ 3 THEN    % RESET BACK-POINTER TO SYMB TABLE.
61920000       BEGIN I ← T.CH - 2;
61930000       MV(2,STLOC,6,DATA[T.R,I.W],I.C);
61940000       END;
61950000    VALU[STLOC].LINK ← VALU[ST].LINK;  % LINK TO NEXT CELL ON STACK.
61960000    RETURNCELL(ST);     % PUT BACK ON AVSL.
61970000    POP ← TRUE;
61980000    END;
61990000 END POP;
62000000 %********POP*************************************************************
62010000 %********PROCESSCONTROLCARD**********************************************
62020000 % PROCESSCONTROLCARD USES CHAR, MOVECHARS, CONTROLPARAMETER.
62030000 PROCEDURE PROCESSCONTROLCARD(A);
62040000    ALPHA ARRAY A[0];
62050000 BEGIN INTEGER I, P;
62060000 LABEL
62070000    E,              % END OF PROCEDURE
```

```
62080000     ILLEGAL,          % ILLEGAL CONTROL CARD
62090000     NOFILE,           % FILE REFERENCED WHICH CANT BE READ
62100000     PARAMERR,         % ERROR IN PARAMETER TO CONTROL CARD
62110000     PARAMMISSING};    % CANT FIND A NEEDED PARAMETER TO CONTROL CARD
62120000 %
62130000 P ← I ← 1 + SKIPCHAR(" ",A[*],1);
62140000 DO P ← P + 1 UNTIL P - I > 7 OR CHARTYPE[CHAR(A[*],P)] ≠ 3;
62150000 IF P - I > 7 THEN GO TO ILLEGAL;
62160000 AA ← 0;
62170000 MV(P-I,A[*],I,AA,8-P+I);
62180000 FOR I ← NUMCONTROLCARDS STEP -1 UNTIL 0 DO
62190000     IF AA = CONTROLCARD[I] THEN
62200000     BEGIN CASE I OF
62210000 %********CONTROL CARDS*****************
62220000     BEGIN GO TO ILLEGAL;
62230000 % 1: -DEBUG      TURNS ON DEBUG MODE
62240000     BEGIN DEBUGGING ← TRUE;
62250000     IF NEXTSOURCERECORD = 0 THEN NEXTSOURCERECORD ← 1;
62260000     DEBUGRULELIMITEXISTS ← ALIMITEXISTS ← TRUE;
62270000     % THIS WILL CAUSE SUSPENSION AT THE FIRST INSTRUCTION.
62280000     END;
62290000 % 2: -PCC      CAUSES ALL CONTROL CARDS TO BE PRINTED
62300000     PCC ← TRUE;
62310000 % 3: -LIST     CAUSES THE PROGRAM TO BE LISTED.
62320000     BEGIN IF LISTSPACES←CONTROLPARAMETER(A,P) < 0 THEN LISTSPACES ← 0;
62330000     LSTF ← TRUE;
62340000     END;
62350000 % 4: -UNLIST     CAUSES THE LISTING TO STOP
62360000     LSTF ← FALSE;
62370000 % 5: -PUNCH      CAUSES THE PROGRAM DECK TO BE PUNCHED
62380000     PUNCHF ← TRUE;
62390000 % 6: -EJECT      EJECTS THE LINE PRINTER TO THE NEXT PAGE
62400000     BEGIN I ← CONTROLPARAMETER(A,P);
62410000     IF I < -1 OR I > 11 THEN GO TO PARAMERR;
62420000     IF I < 1 THEN I ← 1;
62430000     IF LSTF THEN WRITE(PRINT[I]);
62440000     END;
62450000 % 7: -SPACE <INTEGER>      PRODUCES <INTEGER> BLANK LINES IN THE LISTING
62460000     BEGIN
62470000     I ← CONTROLPARAMETER(A,P);
62480000     FOR I ← I-1 WHILE I ≥ 0 DO WRITE(PRINT,FBL);
62490000     END;
62500000 % 8: -WIDTH <INTEGER>      SETS THE NUMBER OF CHARS IN EACH CARD IMAGE
62510000     % TO BE CONSIDERED PART OF AN INSTRUCTION; ALL CHARS AFTER THE NTH
62520000     % ARE TO BE IGNORED BY THE COMPILER.
62530000     % IF THIS CARD ISNT USED, THEN FIELDSIZE IS ASSUMED TO BE 72.
62540000     BEGIN
62550000     IF I ← CONTROLPARAMETER(A,P) > 80 OR I < 0
62560000        THEN GO TO PARAMERR;
62570000     FIELDSIZE ← I;
62580000     END;
62590000 % 9: -26       TURNS ON CHARACTER SET CONVERSION'FROM THE 026 KEYPUNCH
62600000     BEGIN CONVERTF ← TRUE;
62610000     CONVERTSTRINGS ← FALSE;
62620000     END;
62630000 % 10: -3600      CAUSES CHAR SET CONVERSION AND TRANSLATION OF I/O
62640000     % STRINGS FROM CDC 3600 SNOBOL.
```

```
62650000    CONVERTF ← CONVERTSTRINGS ← TRUE;
62660000 % 11: -B5500    CANCELS THE EFFECT OF PREVIOUS -26 AND -3600 CARDS.
62670000    CONVERTF ← CONVERTSTRINGS ← FALSE;
62680000 % 12: -DEFINE A,B,C     ACTS LIKE THE INTRINSIC FUNCTION DEFINE,
62690000    % BUT SETS UP THE FUNCTION AT COMPILE TIME.  THIS SAVES THE SPACE
62700000    % THAT A RUN-TIME DEFINITION WOULD TAKE FOR LITERALS AND INSTRUC-
62710000    % TION CODE, PLUS THE TIME NEEDED FOR COMPILATION.
62720000    % A, B, AND C MUST BE LITERALS.
62730000    BEGIN INTEGER ARRAY ST[1:3];
62740000    INTEGER J;
62750000    LABEL DEF1, DEFERR;
62760000    %
62770000    IF NOT SYMBTABLSETUP THEN INITIALIZESYMBTABL;
62780000    FOR J ← 1,2,3 DO ST[J] ← 0;
62790000    FOR J ← 1,2,3 DO
62800000       BEGIN P ← P + SCANCHAR("""",",",A[P,W],P,C);
62810000       IF P ≥ INSTSIZE THEN GO TO DEF1;
62820000       IF CHAR(A[P,W],P,C) = "," THEN
62830000          BEGIN ST[J] ← 0;
62840000          P ← P + 1;
62850000          END ELSE
62860000          BEGIN P ← P + 1;
62870000          I ← P + SCANCHAR("""",QMARK,A[P,W],P,C);
62880000          IF I ≥ INSTSIZE THEN GO TO DEFERR;
62890000          ST[J] ← NEWCELL(0);
62900000          AA ← STRING(I-P,ST[J]);
62910000          MOVE(I-P,A[P,W],P,C,DATA[AA,R,AA,W],AA,C);
62920000          VALU[ST[J]] ← AA;
62930000          P ← I + SCANCHAR(",",""",A[I,W],I,C+1) + 1;
62940000          IF CHAR(A[P,W],P,C) = "," THEN P ← P + 1;
62950000          IF P ≥ INSTSIZE THEN GO TO DEF1;
62960000          END;
62970000       END;
62980000 DEF1: IF NOT SNBLDEFINE(ST[1],ST[2],ST[3]) THEN GO TO PARAMERR;
62990000    FOR J ← 1,2,3 DO IF ST[J] ≠ 0 THEN RETURNCELL(ST[J]);
63000000    GO TO E;
63010000 DEFERR: MESSAGE0(19); % MISSING QUOTE
63020000    MESSAGETTO(19);
63030000    END DEFINE;
63040000 % 13: -WAIT <N>     SETS THE WAITING TIME FOR REMOTE I/O TO N SECONDS.
63050000    % IF N IS NOT > 0, THEN STANDARDWAITTIME IS USED.
63060000    BEGIN AA ← CONTROLPARAMETER(A,P);
63070000    IF AA < 0 THEN GO TO PARAMERR;
63080000    WAITTIME ← AA × 60;
63090000    END;
63100000 % 14 -LIMIT <I> LSINTEGER>     PUTS A LIMIT OF <INTEGER> TO THE
63110000    % QUANTITY <I>; THIS LIMIT IS CHECKED AT THE START OF THE EXECUTION
63120000    % OF EACH INSTRUCTION.  THE CURRENT VALUES OF <I> ARE:
63130000    %     RULES PUTS A LIMIT TO THE NUMBER OF STATEMENTS EXECUTED.
63140000    %     PROCESS PUTS A LIMIT (IN SECONDS) TO CPU TIME
63150000    %     IO PUTS A LIMIT (IN SECONDS) TO I/O TIME
63160000    BEGIN LABEL LOOP;
63170000 LOOP: P ← P + 1;
63180000    P ← P + SKIPCHAR(" ",A[*],P);
63190000    IF P > INSTSIZE THEN GO TO PARAMERR;
63200000    I ← SCANCHAR(" ","=",A[*],P);
63210000    IF I > INSTSIZE THEN GO TO PARAMERR;
```

```
IF I = 2 THEN
   BEGIN IF EQ(2,WORDS[8],3,A[*],P)
      THEN I ← 0 ELSE GO TO LOOP;
   END ELSE
IF I = 5 THEN
   BEGIN IF EQ(5,WORDS[8],5,A[*],P)
      THEN I ← 1 ELSE GO TO LOOP;
   END ELSE
IF I = 7 THEN
   BEGIN IF EQ(7,WORDS[9],2,A[*],P)
      THEN I ← 2 ELSE GO TO LOOP;
   END ELSE
   GO TO LOOP;
   IF (P←CONTROLPARAMETER(A,P)) < 0 THEN GO TO PARAMERR;
   CASE I OF
      BEGIN
      BEGIN IOLIMIT ← P × 60;
      IOLIMITEXISTS ← TRUE;
      END;
      BEGIN RULELIMIT ← P;
      RULELIMITEXISTS ← TRUE;
      END;
      BEGIN CPULIMIT ← P × 60;
      CPULIMITEXISTS ← TRUE;
      END
      END CASES;
ALIMITEXISTS ← TRUE;
END LMT;
% 15: -SIZE <N>      TELLS THE COMPILER HOW MANY SCATTER AREAS TO SET
% ASIDE IN SYMBTABL.  THIS CONTROL CARD MUST APPEAR BEFORE ANY SNOBOL
% INSTRUCTIONS; THE VALUE OF <N> SHOULD BE APPROXIMATELY THE NUMBER
% OF INSTRUCTIONS IN THE PROGRAM
BEGIN DEFINE DUMMY=#;
IF SYMBTABLSETUP THEN GO TO ILLEGAL;
I ← CONTROLPARAMETER(A,P);
IF I < 0 THEN GO TO PARAMERR;
IF I < 150 THEN BEGIN SCATTERNO ← 0; GO TO E; END;
IF I < 300 THEN BEGIN SCATTERNO ← 1; GO TO E; END;
IF I < 475 THEN BEGIN SCATTERNO ← 2; GO TO E; END;
IF I < 600 THEN BEGIN SCATTERNO ← 3; GO TO E; END;
IF I < 800 THEN BEGIN SCATTERNO ← 4; GO TO E; END;
SCATTERNO ← 5;
END;
% 16: -SET <NAME> <LITERAL>      SETS THE VARIABLE <NAME> TO
% HAVE THE INITIAL VALUE OF <LITERAL>.
BEGIN INTEGER P1, P2;
IF NOT SYMBTABLSETUP THEN INITIALIZESYMBTABL;
% FIND NAME:
I ← P ← P + SKIPCHAR(" ",A[*],P);
WHILE CHARTYPE[CHAR(A[P.W],P.C)] = 3 DO P ← P + 1;
IF P ≥ INSTSIZE OR P = I THEN GO TO PARAMMISSING;
AA ← ENTERST(P-I,A[*],I,"SYMB");
% FIND VALUE:
P1 ← P + SCANCHAR(""","""",A[P.W],P.C);
IF P1 ≥ INSTSIZE THEN GO TO PARAMMISSING;
P ← P1 ← P1 + 1;
WHILE P < INSTSIZE DO
```

```
63790000        BEGIN P2 ← P + SCANCHAR("'","'",A[P,W],P,C);
63800000        IF P2 ≥ INSTSIZE THEN  % MISSING QUOTE:
63810000           BEGIN MESSAGEO(19);
63820000           MESSAGETTO(19);
63830000           GO TO E;
63840000           END;
63850000        P ← P2 + SCANCHAR("'","'",A[P2,W],P2,C+1) + 1;
63860000        IF P < INSTSIZE THEN   % SQUEEZE INSTRUCTION:
63870000           BEGIN MOVE(INSTSIZE-P-1,A[P,W],P,C+1,A[P2,W],P2,C);
63880000           INSTSIZE ← INSTSIZE - (P - P2 + 1);
63890000           P ← P2 + 1;
63900000           END;
63910000        END;
63920000     % P1, P2 NOW DELIMIT THE VALUE.
63930000     AB ← STRING(P2-P1,AA);
63940000     MOVE(P2-P1,A[P1,W],P1,C,FIRSTCHAR(AB));
63950000     VALTABL[AA,STR,AA,STW] ← AB;
63960000     END SET;
63970000  % 17: -LIBRARY <FID>/<MFID>     CREATES A LIBRARY COPY OF THE
63980000     % SNOBOL PROGRAM BEING COMPILED, WITH THE NAME <MFID>/<FID>.
63990000     % THIS LIBRARY FILE CAN BE LOADED FOR A LATER RUN WITH THE CONTROL
64000000     % CARD -LOAD <MFID>/<FID>.
64010000     BEGIN DEFINE DUMMY =#;
64020000     IF ENTRY = 0 THEN GO TO ILLEGAL;
64030000     IF LOADF THEN IF SLASTLABEL ≠ 0 THEN
64040000        BEGIN I ← PTR;
64050000        IF NOT (GTS AND GTF) THEN
64060000           BEGIN
64070000           STORECHARS(IF GT THEN 2 ELSE 3,"/-E",IF GT THEN 6 ELSE 5);
64080000           IF NOT GT THEN I ← I + 1;
64090000           IF NOT GTS THEN
64100000              BEGIN SLOC ← I;
64110000              MV(2,SLOC,6,CODE[MARKER,W],MARKER,C+3);
64120000              END;
64130000           IF NOT GTF THEN
64140000              BEGIN FLOC ← I;
64150000              MV(2,FLOC,6,CODE[MARKER,W],MARKER,C+5);
64160000              END;
64170000           END;
64180000        AA ← STRING(PTR,SLASTLABEL);
64190000        MOVE(PTR,CODE[*],0,FIRSTCHAR(AA));
64200000        VALU[SLASTLABEL],LOC ← AA;
64210000        IF PTR > MAXINSTSIZE THEN MAXINSTSIZE ← PTR;
64220000        PTR ← I;
64230000        END;
64240000     P ← P + SKIPCHAR(" ",A[*],P);
64250000     I ← P + SCANCHAR("/"," ",A[*],P);
64260000     IF I ≥ FIELDSIZE OR CHAR(A[I,W],I,C) ≠ "/" THEN GO TO PARAMERR;
64270000     AA ← 0;
64280000     MV(7,BLANKS,1,AA,1);
64290000     AB ← AA;
64300000     MV(MIN(7,I-P),A[P,W],P,C,AA,1);
64310000     P ← I + 1;
64320000     I ← P + SCANCHAR(" ",QMARK,A[P,W],P,C);
64330000     IF I ≤ P OR I > INSTSIZE THEN GO TO PARAMERR;
64340000     MV(MIN(7,I-P),A[P,W],P,C,AB,1);
64350000     CREATELIBRARY(AA,AB);
```

```
64360000     END LIBRARY;
64370000 % 18: -LOAD <A>/<B>      CAUSES THE DISK FILE <A>/<B> TO BE
64380000     % CONSIDERED A COMPILED SNOBOL PROGRAM, AND LOADED.  THIS CARD
64390000     % IS ILLEGAL IF INSTRUCTIONS, -DEFINE OR -SET CARDS HAVE BEEN
64400000     % ALREADY ENCOUNTERED.  IN ADDITION, ONLY ONE -LOAD CARD IS
64410000     % ACCEPTED PER JOB.
64420000     BEGIN DEFINE DUMMY=#;
64430000     IF SYMBTABLSETUP THEN GO TO ILLEGAL;
64440000     P ← P + SKIPCHAR(" ",A[*],P);
64450000     I ← P + SCANCHAR("/"," ",A[*],P);
64460000     IF I ≥ INSTSIZE  OR CHAR(A[I,W],I,C) ≠ "/" THEN GO TO PARAMERR;
64470000     AA ← 0;
64480000     MV(7,BLANKS,1,AA,1);
64490000     AB ← AA;
64500000     MV(MIN(7,I-P),A[P,W],P,C,AA,1);
64510000     P ← I + 1;
64520000     I ← P + SCANCHAR(" ",QMARK,A[P,W],P,C);
64530000     IF I ≤ P OR I > INSTSIZE THEN GO TO PARAMERR;
64540000     MV(MIN(7,I-P),A[P,W],P,C,AB,1);
64550000     LOADLIBRARY(AA,AB);
64560000     IF DEATH THEN GO TO ENDTERPRET;
64570000     END LOAD;
64580000 % 19: -COMPILE <A>/<B>      TELLS THE LOADER TO COMPILE THE DISK FILE
64590000     % <A>/<B> INTO THE PROGRAM AT THE POINT WHERE THIS CARD APPEARS.
64600000     BEGIN FILE IN PROG DISK SERIAL (1,10,30);
64610000     BOOLEAN RMT, BUFFULL;
64620000     ALPHA ARRAY BUF[0:10], X[0:6];
64630000     INTEGER S1, S2;
64640000     ALPHA AA, AB;
64650000     %
64660000     P ← P + SKIPCHAR(" ",A[*],P);
64670000     I ← P + SCANCHAR("/"," ",A[*],P);
64680000     IF I ≥ INSTSIZE OR CHAR(A[I,W],I,C) ≠ "/" THEN GO TO PARAMERR;
64690000     AA ← 0;
64700000     MV(7,BLANKS,1,AA,1);
64710000     AB ← AA;
64720000     MV(S1←MIN(7,I-P),A[P,W],P,C,AA,1);
64730000     P ← I + 1;
64740000     I ← P + SCANCHAR(" ",QMARK,A[P,W],P,C);
64750000     IF I ≤ P OR I > INSTSIZE THEN GO TO PARAMERR;
64760000     MV(S2←MIN(7,I-P),A[P,W],P,C,AB,1);
64770000     FILL PROG WITH AA,AB;
64780000     SEARCH(PROG,X[*]);
64790000     IF X[0] ≤ 0 THEN GO TO NOFILE;
64800000     IF BUFFULL←BUFFERFULL THEN MOVEWDS(10,BUFFER[0],BUF[0]);
64810000     READ(PROG,10,BUFFER[*]) [NOFILE:NOFILE];
64820000     BUFFERFULL ← TRUE;
64830000     LOADERLEVEL ← LOADERLEVEL + 1;
64840000     RMT ← PROGRAMFROMREMOTE; PROGRAMFROMREMOTE ← FALSE;
64850000     LOADER(PROG);
64860000     WRITE(BUFOUT[*],ENDPROG);
64870000     MV(S1,AA,1,BUFOUT[*],22);
64880000     MV(1,SLASH,7,BUFOUT[*],22+S1);
64890000     MV(S2,AB,1,BUFOUT[*],23+S1);
64900000     WRITE(PRINT,5,BUFOUT[*]);
64910000     IF DATACOMF THEN
64920000        BEGIN MV(2,CRLF,5,DCWRITE[*],0);
```

```
64930000         MV(21+S1+S2,BUFOUT[*],2,DCWRITE[*],2);
64940000         MV(3,CRLF,5,DCWRITE[*],23+S1+S2);
64950000         IF OUTPUT THEN;
64960000         END;
64970000     IF BUFFERFULL+BUFFULL THEN MOVEWDS(10,BUF[0],BUFFER[0]);
64980000     LOADERLEVEL ← LOADERLEVEL - 1;
64990000     PROGRAMFROMREMOTE ← RMT;
65000000     END COMPILE CARD;
65010000 % 20: -INFORM     TURNS ON INFORMATIVE DIAGNOSTICS.  IF THE SYSTEM-
65020000     % DEBUGGING FLAG IS TRUE, THEN THE SYSTEM DEBUGGING AIDS
65030000     % ARE ENABLED.
65040000     BEGIN INFORM ← SYSTEMDEBUGGING;
65050000     PRINTMESSAGES ← TRUE;
65060000     END;
65070000 % 21: -SILENCE    TURNS OFF DIAGNOSTIC MESSAGES TURNED ON BY -INFORM
65080000     INFORM ← PRINTMESSAGES ← FALSE;
65090000     END CONTROL CARDS CASE STATEMENT;
65100000 %********CONTROL CARDS******************
65110000     GO TO E;
65120000     END;
65130000 ILLEGAL:
65140000     MESSAGE0(20);
65150000     MESSAGETTO(20);
65160000     MESSAGES ← MESSAGES + 1;
65170000     GO TO E;
65180000 PARAMMISSING:
65190000     MESSAGE0(21);
65200000     MESSAGETTO(21);
65210000     MESSAGES ← MESSAGES + 1;
65220000     GO TO E;
65230000 NOFILE:
65240000     MESSAGE0(11);
65250000     MESSAGETTO(11);
65260000     GO TO E;
65270000 PARAMERR:
65280000     MESSAGE0(10);
65290000     MESSAGETTO(10);
65300000     MESSAGES ← MESSAGES + 1;
65310000 E: END PROCESSCONTROLCARD;
65320000 %********PROCESSCONTROLCARD**************************************************
65330000 %********PUSH****************************************************************
65340000 BOOLEAN PROCEDURE PUSH(STLOC,NEWVAL);
65350000     VALUE STLOC, NEWVAL; ALPHA STLOC, NEWVAL;
65360000 BEGIN ALPHA ST, T, T1;
65370000 INTEGER I;
65380000 LABEL RETURN;
65390000 % IF INFORM THEN INFORMA(1,STLOC);
65400000 ST ← NEWCELL(STLOC.STR);
65410000 IF DEATH THEN
65420000     BEGIN PUSH ← FALSE; GO TO RETURN; END;
65430000 VALU[ST] ← T1 ← VALU[STLOC];
65440000 IF T1.CH ≥ 3 THEN  % IF < 3 THERE IS NO STRING BEING POINTED TO.
65450000     BEGIN I ← T1.CH - 2;
65460000     MV(2,ST,6,DATA[T1.R,I,W],I,C);
65470000     END;
65480000 VALU[ST].LINK ← VALU[STLOC].LINK;
65490000 VALU[STLOC] ← 0 & T1 [1:1:4] & ST CLINK;
```

```
65500000 IF NEWVAL ≠ 0 THEN  % ASSIGN NEW VALUE
65510000    BEGIN IF STLOC = NEWVAL     % PUSH(N,N) CALLED.
65520000       THEN NEWVAL ← ST;        % RE-ASSIGN OLD VALUE.
65530000    ST ← VALU[NEWVAL].S;
65540000    T ← STRING(ST,STLOC);
65550000    IF DEATH THEN
65560000       BEGIN PUSH ← FALSE; GO TO RETURN; END;
65570000    NEWVAL ← VALU[NEWVAL];
65580000    MOVE(ST,FIRSTCHAR(NEWVAL),FIRSTCHAR(T));
65590000    VALU[STLOC].LOC ← T;
65600000    END;
65610000 PUSH ← TRUE;
65620000 RETURN:
65630000 END PUSH;
65640000 %********PUSH***********************************************************
65650000 %********RETURNCELL*****************************************************
65660000 PROCEDURE RETURNCELL(N); VALUE N; ALPHA N;
65670000 BEGIN DEFINE DUMMY=#;
65680000 % IF INFORM THEN INFORMA(3,N);
65690000 NAME[N] ← 0 & NEXTCELL[N,STR] CLINK;
65700000 VALU[N] ← 0;
65710000 NEXTCELL[N,STR] ← N;
65720000 END RETURNCELL;
65730000 %********RETURNCELL*****************************************************
65740000 %********RETURNTEMP*****************************************************
65750000 PROCEDURE RETURNTEMPS;
65760000 BEGIN INTEGER I; ALPHA ST,P;
65770000 % IF INFORM THEN INFORMO(7);
65780000 FOR I ← 0 STEP 1 UNTIL 15 DO IF USEDST[I] THEN
65790000    IF TEMPLIST[I] ≠ 0 THEN
65800000       BEGIN ST ← TEMPLIST[I];
65810000       P ← NEXTCELL[I];
65820000       TEMPLIST[I] ← 0;
65830000       NEXTCELL[I] ← NAME[ST].LINK;
65840000       NAME[ST] ← 0 & P CLINK;
65850000       END;
65860000 END RETURNTEMPS;
65870000 %********RETURNTEMP*****************************************************
65880000 %********SCATTER********************************************************
65890000 ALPHA PROCEDURE SCATTER(SIZE,LOC,P,X);
65900000 VALUE SIZE, P, X;
65910000 INTEGER SIZE, P;
65920000 ALPHA ARRAY LOC[0];
65930000 ALPHA X;
65940000 BEGIN ALPHA I, J, K;
65950000 J ← IF SIZE = 0 THEN 1/X ELSE SIZE/X;
65960000 FOR I ← 1 STEP 1 UNTIL SIZE DO
65970000    J ← J + X / (I + CHAR(LOC[(K←P+I-1).W],K.C));
65980000 K ← J.[8:30];
65990000 I.STR ← ENTIER(K MOD (SCATTERNO+1));
66000000 J.[1:9] ← 0;
66010000 %  LITERALS ARE SCATTERED INTO A DIFFERENT AREA THAN OTHER QUANTITIES:
66020000 I.STW ← IF X = "LIT"
66030000    THEN SCATTERSIZE + ENTIER(J MOD 16)
66040000    ELSE ENTIER(J MOD SCATTERSIZE);
66050000 IF I = 0 THEN I ← 1;
66060000 SCATTER ← I;
```

```
66070000 END SCATTER;
66080000 %********SCATTER*************************************************************
66090000 %********SEARCHST************************************************************
66100000 ALPHA PROCEDURE SEARCHST(N,L,P,X);
66110000 VALUE N, P, X;
66120000 INTEGER N, P;
66130000 ALPHA ARRAY L[0];
66140000 ALPHA X;
66150000 BEGIN ALPHA J, K;
66160000 INTEGER I, XTEST;
66170000 LABEL FOUND, FAIL;
66180000 %
66190000 XTEST ← IF X = "SYMB" THEN 0 ELSE
66200000         IF X = "FCT"  THEN 2 ELSE
66210000         IF X = "INST" THEN 3 ELSE 0;
66220000 I ← SCATTER(N,L,P,X);
66230000 WHILE I ≠ 0 DO
66240000    BEGIN IF (IF N ≠ (K←NAME[I]).S
66250000                THEN FALSE
66260000                ELSE EQUAL(N,L,P,FIRSTCHAR(K)))
66270000          THEN IF (IF N = 0 THEN K.LOC ≠ 0 ELSE TRUE)
66280000               THEN IF K.TYPE = XTEST
66290000                    THEN GO TO FOUND;
66300000    I ← K.LINK;
66310000    END;
66320000 FAIL:
66330000    I ← -1;
66340000 FOUND:
66350000    SEARCHST ← I;
66360000 END SEARCHST;
66370000 %********SEARCHST************************************************************
66380000 %********SNBLDEFINE*********************************************************
66390000 % SNBLDEFINE SETS UP PROGRAM-DEFINED FUNCTIONS.  THE THREE PARAMETERS
66400000 % SHOULD BE THE SYMBOL TABLE LOCATIONS OF THE THREE PARAMETERS FOR
66410000 % THE SNOBOL DEFINE.
66420000 % SNBLDEFINE USES CHAR, SCANCHAR, ENTERST, MIN, STRING, MOVECHARS.
66430000 BOOLEAN PROCEDURE SNBLDEFINE(ST1,ST2,ST3);
66440000 VALUE ST1, ST2, ST3;
66450000 ALPHA ST1, ST2, ST3;
66460000 %
66470000 BEGIN
66480000 DEFINE PARAMMAX = 30 #;  % MAX NUMBER OF PARAMS & LOC VARS ALLOWED.
66490000 INTEGER I, P1, P2, LVNO;
66500000 ALPHA PROT, ENT, LVARS, STLOC;
66510000 INTEGER ARRAY LV[0:PARAMMAX];
66520000 LABEL PROTERR, LVERR, RENVOI, TOOMANY;
66530000 IF ST1 = 0 THEN
66540000    BEGIN % NO FIRST PARAM (PROTOTYPE).
66550000    MESSAGE0(4);
66560000    MESSAGETTO(4);
66570000    GO TO RENVOI;
66580000    END;
66590000 PROT ← VALU[ST1];
66600000 ENT  ← IF ST2.STW = 0 THEN 0 ELSE VALU[ST2];
66610000 LVARS← IF ST3.STW = 0 THEN 0 ELSE VALU[ST3];
66620000 %
66630000 % FIND NAME:
```

```
66640000 P1 ← PROT.CH + SKIPCHAR(" ",DATA[PROT.R,PROT.W],PROT.C);
66650000 P2 ← P1 + SCANCHAR("(",QMARK,DATA[PROT.R,P1.W],P1.C);
66660000 IF P2 - P1 > PROT.S THEN
66670000    BEGIN MESSAGEO(3);
66680000    MESSAGETT@(3);
66690000    GO TO RENVOI;
66700000    END;
66710000 STLOC ← ENTERST(P2-P1,DATA[PROT.R,*],P1,"FCT");
66720000 LV[LVNO←0] ← ENTERST(P2-P1,DATA[PROT.R,*],P1,"SYMB");
66730000 IF PRINTMESSAGES THEN % CHECK WHETHER ALREADY DEFINED:
66740000    IF VALU[STLOC].LINK ≠ 0 THEN
66750000       IF LOADF THEN
66760000          BEGIN MESSAGEO(13); MESSAGETTO(13); END ELSE
66770000             BEGIN MESSAGEI(21,INSTNO); MESSAGETTI(21,INSTNO); END;
66780000 %
66790000 % FIND ENTRY POINT:
66800000 ENT ← IF ENT.S = 0
66810000    THEN ENTERST(P2-P1,DATA[PROT.R,*],P1,"INST")
66820000    ELSE ENTERST(ENT.S,DATA[ENT.R,*],ENT.CH,"INST");
66830000 %
66840000 % PROCESS FORMAL PARAMETERS:
66850000 I ← PROT.CH + SCANCHAR(")",QMARK,DATA[PROT.R,PROT.W],PROT.C);
66860000 IF I < P2 - PROT.CH THEN GO TO PROTERR;
66870000 P1 ← P2 + 1;
66880000 WHILE P1 < I DO
66890000    BEGIN P1 ← P1 + SKIPCHAR(" ",DATA[PROT.R,P1.W],P1.C);
66900000    P2 ← P1 + MIN(
66910000       SCANCHAR(",",")",DATA[PROT.R,P1.W],P1.C),
66920000       SCANCHAR(" ",QMARK,DATA[PROT.R,P1.W],P1.C));
66930000    IF P2 > I THEN P2 ← I;
66940000    IF P1 = P2 THEN GO TO PROTERR;
66950000    IF LVNO ← LVNO+1 > PARAMMAX THEN GO TO TOOMANY;
66960000    LV[LVNO] ← ENTERST(P2-P1,DATA[PROT.R,*],P1,"SYMB");
66970000    P2 ← P2 + SKIPCHAR(" ",DATA[PROT.R,P2.W],P2.C);
66980000    IF P2 < I AND CHAR(DATA[PROT.R,P2.W],P2.C) ≠ "," THEN GO TO PROTERR;
66990000    P1 ← P2 + 1;
67000000    END;
67010000 %
67020000 % PROCESS LOCAL VARIABLES:
67030000 I ← LVARS.CH + LVARS.S;
67040000 P1 ← LVARS.CH;
67050000 I ← P1 + LVARS.S;
67060000 WHILE P1 < I DO
67070000    BEGIN P1 ← P1 + SKIPCHAR(" ",DATA[LVARS.R,P1.W],P1.C);
67080000    P2 ← P1 + MIN(
67090000       SCANCHAR(",",")",DATA[LVARS.R,P1.W],P1.C),
67100000       SCANCHAR(" ",QMARK,DATA[LVARS.R,P1.W],P1.C));
67110000    IF P2 > I THEN P2 ← I;
67120000    IF LVNO←LVNO+1 > PARAMMAX THEN GO TO TOOMANY;
67130000    LV[LVNO] ← ENTERST(P2-P1,DATA[LVARS.R,*],P1,"SYMB");
67140000    P2 ← P2 + SKIPCHAR(" ",DATA[LVARS.R,P2.W],P2.C);
67150000    IF P2 < I AND CHAR(DATA[LVARS.R,P2.W],P2.C) ≠ "," THEN GO TO LVERR;
67160000    P1 ← P2 + 1;
67170000    END;
67180000 % FINISH SYMBOL TABLE ENTRY:
67190000 I ← STRING((LVNO+1)×2,STLOC);
67200000 VALU[STLOC] ← I & 1[1:47:1] & ENT CLINK;
```

```
67210000 FOR P1 ← 0 STEP 1 UNTIL LVNO DO
67220000    MOVE(2,LV[P1],6,DATA[I,R,(I+2×P1),W],(I+2×P1),C);
67230000 IF TRACEALL THEN
67240000    BEGIN VALU[STLOC].IOUSE ← 3;
67250000    IO[STLOC] ← 0 & 3 CFILNO & 3 CIOUSE & 2 CIOTYPE;
67260000    END;
67270000 SNBLDEFINE ← TRUE;
67280000 GO TO RENVOI;
67290000 TOOMANY:
67300000    SNBLDEFINE ← FALSE;
67310000    MESSAGEI(22,PARAMMAX);
67320000    MESSAGETTI(24,PARAMMAX);
67330000    GO TO RENVOI;
67340000 PROTERR:
67350000    SNBLDEFINE ← FALSE;
67360000    MESSAGETTO(2);
67370000    MESSAGEO(2); GO TO RENVOI;
67380000 LVERR:
67390000    SNBLDEFINE ← FALSE;
67400000    MESSAGEO(1);
67410000    MESSAGETTO(1);
67420000 RENVOI:
67430000 END SNBLDEFINE;
67440000 %********SNBLDEFINE*****************************************************
67450000 %********SNBLIN*********************************************************
67460000 BOOLEAN PROCEDURE SNBLIN(ST);
67470000 VALUE ST; ALPHA ST;
67480000 BEGIN LABEL RETURN, FAIL, EOF, PAR, NOFILE,
67490000    DATACOMM, INREAD, LOOK, ILLEGAL;
67500000 ALPHA STN, STV, SIO;
67510000 INTEGER SIZE, TYPE, T, UNIT;
67520000 SWITCH READSW ←
67530000    INREAD,
67540000    LOOK,
67550000    DATACOMM;
67560000 %
67570000 STV ← VALU[ST];
67580000 SIO ← IO[ST];
67590000 IF NOT BOOLEAN(STV.INUSE) THEN GO TO ILLEGAL;
67600000 TYPE ← SIO.IOTYPE;
67610000 UNIT ← SIO.FILNO;
67620000 % IF INFORM THEN IF UNIT > 0 THEN INFORMI(6,UNIT);
67630000 GO TO READSW[TYPE];
67640000 GO TO ILLEGAL;
67650000 %
67660000 DATACOMM:
67670000    BEGIN % INPUT FROM TELETYPE:
67680000    IF NOT DATACOMF THEN
67690000       BEGIN FINDUSERS;
67700000       IF NOT DATACOMF THEN GO TO NOFILE;
67710000       END;
67720000    IF NOT INPUT THEN GO TO FAIL;
67730000    VALU[ST].LOC ← T ← IF STV.S ≥ DCSIZE
67740000       THEN STV.LOC & DCSIZE CONCS % USE OLD STRING
67750000       ELSE STRING(DCSIZE,ST);     % NEED BIGGER STRING
67760000    MOVE(DCSIZE,DCREAD[*],0,FIRSTCHAR(T));
67770000    SNBLIN ← TRUE;
```

```
67780000     BREAK ← FALSE;
67790000     GO TO RETURN;
67800000     END;
67810000 INREAD:
67820000     BEGIN COMMENT INPUT FROM IO[UNIT];
67830000     IOEOF[UNIT] ← FALSE;
67840000     SIZE ← IOSIZE[UNIT];
67850000     IF LOOKF[UNIT] THEN
67860000        BEGIN LOOKF[UNIT] ← FALSE;
67870000        MOVE(SIZE,NEXTRECORD[UNIT,*],0,BUFFER[*],0);
67880000        END ELSE
67890000        % READ NEXT RECORD:
67900000        IF RECORD[UNIT] ≥ 0 THEN
67910000           BEGIN
67920000           READ(IOFILE[UNIT][RECORD[UNIT]],1023,BUFFER[*])[EOF:PAR];
67930000           RECORD[UNIT] ← -1;
67940000           END ELSE
67950000           BEGIN SPACE(IOFILE[UNIT],IOSPACE[UNIT])[EOF:PAR];
67960000           READ(IOFILE[UNIT],1023,BUFFER[*])[EOF:PAR];
67970000           END;
67980000     VALU[ST].LOC ← T ←
67990000        IF STV.S < SIZE
68000000           THEN STRING(SIZE,ST)
68010000           ELSE STV.LOC & SIZE CONCS;
68020000     MOVE(SIZE,BUFFER[*],0,FIRSTCHAR(T));
68030000     SNBLIN ← TRUE;
68040000     GO TO RETURN;
68050000     END;
68060000 LOOK:
68070000     BEGIN % NON-READING INPUT (LOOK) FROM IOFILE[UNIT];
68080000     SIZE ← IOSIZE[UNIT];
68090000     IF NOT LOOKF[UNIT] THEN
68100000        BEGIN IF RECORD[UNIT] ≥ 0 THEN
68110000           READ(IOFILE[UNIT][RECORD[UNIT]],1023,NEXTRECORD[UNIT,*])
68120000              [EOF:PAR] ELSE
68130000           BEGIN SPACE(IOFILE[UNIT],IOSPACE[UNIT])[EOF:PAR];
68140000           READ(IOFILE[UNIT],1023,NEXTRECORD[UNIT,*])[EOF:PAR];
68150000           END;
68160000        IOEOF[UNIT] ← FALSE;
68170000        LOOKF[UNIT] ← TRUE;
68180000        END;
68190000     VALU[ST].LOC ← T ←
68200000        IF STV.S < SIZE
68210000           THEN STRING(SIZE,ST)
68220000           ELSE STV.LOC & SIZE CONCS;
68230000     MOVE(SIZE,NEXTRECORD[UNIT,*],0,FIRSTCHAR(T));
68240000     SNBLIN ← TRUE;
68250000     GO TO RETURN;
68260000     END;
68270000 PAR:
68280000     BEGIN DEFINE DUMMY=#;
68290000     WRITE(BUFOUT[*],FPARITY);
68300000     STN ← NAME[ST];
68310000     MOVE(STRINGLOC(STN),BUFOUT[*],32);
68320000     WRITE(PRINT,FBL);
68330000     WRITE(PRINT,17,BUFOUT[*]);
68340000     GO TO FAIL;
```

```
68350000     END;
68360000 ILLEGAL:
68370000     INFORMA(5,ST);
68380000     DEATH ← SYSTEMERROR ← INFORM ← TRUE;
68390000     GO TO ENDTERPRET;
68400000 NOFILE:
68410000     IF PRINTMESSAGES THEN
68420000     BEGIN DEFINE DUMMY=#;
68430000     STN ← NAME[ST];
68440000     WRITE(BUFOUT[*],FCLOSEDR);
68450000     MOVE(STN.S,FIRSTCHAR(STN),BUFOUT[*],36);
68460000     WRITE(PRINT,FBL);
68470000     WRITE(PRINT,17,BUFOUT[*]);
68480000     END;
68490000     GO TO FAIL;
68500000 EOF:
68510000     IOEOF[UNIT] ← TRUE;
68520000 FAIL:
68530000     SNBLIN ← FALSE;
68540000 RETURN:
68550000 END SNBLIN;
68560000 %********SNBLIN*************************************************************
68570000 %********SNBLOUT************************************************************
68580000 BOOLEAN PROCEDURE SNBLOUT(ST); VALUE ST; ALPHA ST;
68590000 BEGIN
68600000 LABEL RETURN, FAIL, EOF, PAR, NOFILE, SYSPOT,
68610000     DATACOMM, IOWRITE, TRACE, ILLEGAL;
68620000 SWITCH WRITESW ←
68630000         IOWRITE,
68640000         TRACE,
68650000         DATACOMM,
68660000         ILLEGAL,
68670000         SYSPOT;
68680000 ALPHA STN, STV, SIO, AA, AB;
68690000 INTEGER SIZE, TYPE, UNIT, IS, ICH, IR, T, I1, I2;
68700000 %
68710000 STV ← VALU[ST];
68720000 SIO ← IO[ST];
68730000 IF NOT BOOLEAN(STV.OUTUSE) THEN GO TO ILLEGAL;
68740000 TYPE ← SIO.IOTYPE;
68750000 UNIT ← SIO.FILNO;
68760000 IS ← STV.S;
68770000 ICH ← STV.CH;
68780000 IR ← STV.R;
68790000 % IF INFORM THEN IF UNIT>0 THEN INFORMI(13,UNIT);
68800000 %
68810000 GO TO WRITESW[TYPE];
68820000 GO TO ILLEGAL;
68830000 %
68840000 DATACOMM:
68850000     BEGIN COMMENT OUTPUT TO DATACOMM UNIT;
68860000     IF NOT DATACOMF THEN
68870000         BEGIN FINDUSERS;
68880000         IF NOT DATACOMF THEN GO TO NOFILE;
68890000         END;
68900000     WHILE IS ≥ 100 DO
68910000         BEGIN MOVE(100,DATA[IR,ICH,W],ICH,C,DCWRITE[*],0);
```

```
68920000        MV(1,ARROW,7,DCWRITE[12],5);
68930000        IF NOT OUTPUT THEN GO TO FAIL;
68940000        IS ← IS - 100;
68950000        ICH ← ICH + 100;
68960000        END;
68970000     IF IS > 0 THEN
68980000        BEGIN MOVE(IS,DATA[IR,ICH,W],ICH,C,DCWRITE[*],0);
68990000        MV(1,ARROW,7,DCWRITE[IS,W],IS,C);
69000000        IF NOT OUTPUT THEN GO TO FAIL;
69010000        END;
69020000     SNBLOUT ← TRUE;
69030000     GO TO RETURN;
69040000     END;
69050000  IOWRITE:
69060000     BEGIN % OUTPUT TO FILE # UNIT.
69070000     IOEOF[UNIT] ← FALSE;
69080000     T ← (IOSIZE[UNIT]-1),W + 1;
69090000     SIZE ← IOSIZE[UNIT];
69100000     FOR I1←IOSPACE[UNIT] STEP -1 UNTIL 1 DO WRITE(IOFILE[UNIT]);
69110000     IF IS > SIZE THEN IF BOOLEAN(SIO,OVFL) THEN
69120000        BEGIN % WRITE AS MANY FULL RECORDS AS POSSIBLE;
69130000        MOVEWDS(1,BLANKS,BUFOUT[*]);
69140000        DO BEGIN
69150000           MOVE(SIZE,DATA[IR,ICH,W],ICH,C,BUFOUT[*],0);
69160000           IF RECORD[UNIT] ≥ 0
69170000              THEN WRITE(IOFILE[UNIT][RECORD[UNIT]],T,BUFOUT[*])[EOF:PAR]
69180000              ELSE WRITE(IOFILE[UNIT],T,BUFOUT[*])[EOF:PAR];
69190000           RECORD[UNIT] ← -1;
69200000           ICH ← ICH + SIZE;
69210000           IS ← IS - SIZE;
69220000           END UNTIL IS ≤ SIZE;
69230000        END ELSE IS ← SIZE;
69240000     CLEAR(BUFOUT,T);
69250000     MOVE(IS,DATA[IR,ICH,W],ICH,C,BUFOUT[*],0);
69260000     IF RECORD[UNIT] ≥ 0
69270000        THEN WRITE(IOFILE[UNIT][RECORD[UNIT]],T,BUFOUT[*])[EOF:PAR]
69280000        ELSE WRITE(IOFILE[UNIT][NO],T,BUFOUT[*])[EOF:PAR];
69290000     RECORD[UNIT] ← -1;
69300000     SNBLOUT ← TRUE;
69310000     GO TO RETURN;
69320000     END;
69330000  TRACE:
69340000     BEGIN DEFINE DUMMY=#;
69350000     T ← (IOSIZE[UNIT]-1),W + 1;
69360000     SIZE ← IOSIZE[UNIT];
69370000     STN ← NAME[ST];
69380000     WRITE(IOFILE[UNIT],FBL);
69390000     I1 ← MIN(STN.S,63);        % SIZE OF NAME
69400000     I2 ← I1 + 3 - I1.[46:2];  % SIZE OF FIELD FOR NAME.
69410000     WRITE(BUFOUT[*],FTRACE,INSTNO,I2);
69420000     MOVE(I1,FIRSTCHAR(STN),BUFOUT[*],8);
69430000     IF SIZE - I2 - 12 > STV.S THEN
69440000        BEGIN MOVE(STRINGLOC(STV),BUFOUT[*],I2+12);
69450000        MV(1,QUOTE,7,BUFOUT[(I1+STV.S+I2+12),W],I1,C);
69460000        WRITE(IOFILE[UNIT][NO],T,BUFOUT[*]) [EOF:PAR];
69470000        END ELSE
69480000        BEGIN
```

```
69490000         MOVE(SIZE-I2-12,FIRSTCHAR(STV),BUFOUT[*],12+I2);
69500000         WRITE(IOFILE[UNIT][NO],T,BUFOUT[*]) [EOF:PAR];
69510000         STV.S ← STV.S - SIZE + I2 + 12;
69520000         STV.CH ← STV.CH + SIZE - I2 - 12;
69530000         WHILE STV.S ≥ SIZE DO
69540000            BEGIN MOVE(SIZE,FIRSTCHAR(STV),BUFOUT[*],0);
69550000            WRITE(IOFILE[UNIT],FBL);
69560000            WRITE(IOFILE[UNIT][NO],T,BUFOUT[*])[EOF:PAR];
69570000            STV.S ← STV.S - SIZE;
69580000            STV.CH ← STV.CH + SIZE;
69590000            END;
69600000         CLEAR(BUFOUT,T);
69610000         MOVE(STV.S,FIRSTCHAR(STV),BUFOUT[*],0);
69620000         MV(1,QUOTE,7,BUFOUT[(STV.S),W],(STV.S),C);
69630000         WRITE(IOFILE[UNIT],FBL);
69640000         WRITE(IOFILE[UNIT][NO],T,BUFOUT[*]) [EOF:PAR];
69650000         END;
69660000      SNBLOUT ← TRUE;
69670000      GO TO RETURN;
69680000      END;
69690000 SYSPOT:
69700000      BEGIN DEFINE DUMMY =#;
69710000      IF STV.S = 0 THEN GO TO RETURN;
69720000      AA ← CHAR(DATA[STV.R,STV.W],STV.C);
69730000      IF AA = "+" THEN IOSPACE[3] ← 0 ELSE
69740000      IF AA = " " THEN IOSPACE[3] ← 1 ELSE
69750000      IF AA = "0" THEN IOSPACE[3] ← 2 ELSE
69760000      IF AA = "1" THEN
69770000         BEGIN
69780000         WRITE(PRINT[PAGE]);
69790000         IOSPACE[3] ← 1;
69800000         END ELSE
69810000      IF AA = "*" THEN   % NO SKIP AT END OF PAGE
69820000         BEGIN RECORD[3] ← 6;
69830000         IOSPACE[3] ← 1;
69840000         END;
69850000      STN ← TEMPCELL;
69860000      AB ← STRING(STV,S-1,STN);
69870000      MOVE(AB.S,FIRSTCHAR(STV)+1,FIRSTCHAR(AB));
69880000      AA ← VALU[PRINTLOC];
69890000      AA.LOC ← AB;
69900000      VALU[STN] ← AA;
69910000      IO[STN] ← IO[PRINTLOC];
69920000      IF SNBLOUT(STN) THEN
69930000         BEGIN
69940000         IOSPACE[3] ← 1;
69950000         SNBLOUT ← TRUE;
69960000         GO TO RETURN;
69970000         END ELSE
69980000         BEGIN
69990000         IOSPACE[3] ← 1;
70000000         GO TO FAIL;
70010000         END;
70020000      END;
70030000 ILLEGAL:
70040000      INFORMA(5,ST);
70050000      DEATH ← SYSTEMERROR ← INFORM ← TRUE;
```

```
70060000     GO TO ENDTERPRET;
70070000 NOFILE:
70080000     IF PRINTMESSAGES THEN
70090000     BEGIN DEFINE DUMMY=#;
70100000     STN ← NAME[ST];
70110000     WRITE(BUFOUT[*],FCLOSEDW);
70120000     MOVE(STN.S,FIRSTCHAR(STN),BUFOUT[*],35);
70130000     WRITE(PRINT,FBL);
70140000     WRITE(PRINT,17,BUFOUT[*]);
70150000     END;
70160000     GO TO FAIL;
70170000 PAR:
70180000     IF PRINTMESSAGES THEN
70190000     BEGIN DEFINE DUMMY=#;
70200000     STN ← NAME[ST];
70210000     WRITE(BUFOUT[*],FPARITY);
70220000     MOVE(STN.S,FIRSTCHAR(STN),BUFOUT[*],32);
70230000     WRITE(PRINT,FBL);
70240000     WRITE(PRINT,17,BUFOUT[*]);
70250000     END;
70260000     GO TO FAIL;
70270000 EOF:
70280000     IOEOF[UNIT] ← TRUE;
70290000 FAIL:
70300000     SNBLOUT ← FALSE;
70310000 RETURN:
70320000 END SNBLOUT;
70330000 %********SNBLOUT*************************************************
70340000 %********STORECHARS*********************************************
70350000 PROCEDURE STORECHARS(N,LOC,INC);
70360000 VALUE N, INC; INTEGER N, INC; ALPHA LOC;
70370000 BEGIN DEFINE DUMMY =#;
70380000 IF PTR + N > 8180 THEN
70390000     BEGIN
70400000     SYNTAXERR(42,0);
70410000     END ELSE
70420000     BEGIN
70430000     IF N ≤ 63
70440000        THEN MV(N,LOC,INC,CODE[PTR.W],PTR.C)
70450000        ELSE MOVE(N,LOC,INC,CODE[PTR.W],PTR.C);
70460000     PTR ← PTR + N;
70470000     END;
70480000 END STORECHARS;
70490000 %********STORECHARS*********************************************
70500000 %********STRING*************************************************
70510000 % STRING RETURNS THE LOCATION OF AN UNUSED STRING OF SIZE N.
70520000 % IF NECESSARY, THE GARBAGE COLLECTOR IS CALLED, OR NEW ROWS ARE
70530000 % MADE AVAILABLE.
70540000 ALPHA PROCEDURE STRING(N,STLOC);
70550000 VALUE N, STLOC;
70560000 INTEGER N; ALPHA STLOC;
70570000 %
70580000 BEGIN LABEL L, E, FAIL;
70590000 INTEGER I;
70600000 IF DPNTR[RW] + N ≥ 8181 THEN
70610000     BEGIN INTEGER J; J ← 8181 - N;
70620000     FOR I ← 0 STEP 1 UNTIL DATASIZE DO
```

```
70630000        IF DPNTR[I] < J AND USEDROW[I] THEN
70640000           BEGIN RW ← I; J ← DPNTR[I] END;
70650000     IF DPNTR[RW] + N ≥ 8181 THEN
70660000        BEGIN GARBAGECOLLECTOR;
70670000        FOR I ← 0 STEP 1 UNTIL DATASIZE DO
70680000           IF DPNTR[I] < J AND USEDROW[I] THEN
70690000              BEGIN RW ← I; J ← DPNTR[I] END;
70700000        IF DPNTR[RW] + N ≥ 8181 THEN
70710000           BEGIN
70720000           IF N ≥ 8181 THEN GO TO FAIL ELSE
70730000           FOR I ← 0 STEP 1 UNTIL DATASIZE DO
70740000              IF NOT USEDROW[I] THEN
70750000                 BEGIN USEDROW[I] ← TRUE;
70760000                 RW ← I; GO TO L;
70770000                 END;
70780000           END ELSE GO TO L;
70790000        GO TO FAIL;
70800000        END;
70810000     END;
70820000 L: STLOC.[30:6] ← QMARK;
70830000 MV(3,STLOC,5,DATA[RW,(I←DPNTR[RW]),W],I,C);
70840000 STRING ← (I+3) & RW CONCR & N CONCS;
70850000 DPNTR[RW] ← I + N + 3;
70860000 GO TO E;
70870000 FAIL:
70880000     MESSAGEI(12,N);
70890000     MESSAGETTI(12,N);
70900000     COMMENT OUT OF SPACE IN DATA[*,*];
70910000     WRITE(PRINT,MESSAGE[1]);
70920000     DEATH ← TRUE;
70930000 E:
70940000 END STRING;
70950000 %********STRING*************************************************************
70960000 %********STRINGDUMP*********************************************************
70970000 PROCEDURE STRINGDUMP(INSTNO); VALUE INSTNO; INTEGER INSTNO;
70980000 BEGIN
70990000 INTEGER I, J, T, SN;
71000000 ALPHA STN, STV;
71010000 WRITETIME(0,0);
71020000 MESSAGEO(6);
71030000 IF PRINTMESSAGES THEN MESSAGETTO(6);
71040000 FOR I ← 0 STEP 1 WHILE I ≤ STRMAX AND USEDST[I] DO
71050000    FOR J ← 0 STEP 1 UNTIL STWMAX DO
71060000       IF (STN←NAMTABL[I,J]).[1:2] = 0 THEN IF STN.LOC > 1 THEN
71070000             BEGIN
71080000             SN ← MIN(STN,S,63);
71090000             T ← MAX(SN+3-SN.[46:2],8);
71100000             STV ← VALTABL[I,J];
71110000             WRITE(BUFOUT[*],FTRACE,INSTNO,T);
71120000             MOVE(SN,FIRSTCHAR(STN),BUFOUT[*],8);
71130000             IF 120 - T > STV.S THEN
71140000                BEGIN MOVE(STV.S,FIRSTCHAR(STV),
71150000                   BUFOUT[T.W],12+T.C);
71160000                MV(1,QUOTE,7,BUFOUT[(I1←STV.S+T+12).W],I1.C);
71170000                WRITE(PRINT,17,BUFOUT[*]);
71180000                END ELSE
71190000                BEGIN
```

```
71200000                  MOVE(120-T,FIRSTCHAR(STV),
71210000                     BUFOUT[*],12+T);
71220000                  WRITE(PRINT,17,BUFOUT[*]);
71230000                  STV.S ← STV.S - 120 + T;
71240000                  STV.CH ← STV.CH + 120 - T;
71250000                  WHILE STV.S ≥ 132 DO
71260000                     BEGIN MOVE(132,FIRSTCHAR(STV),
71270000                        BUFOUT[*],0);
71280000                     WRITE(PRINT,17,BUFOUT[*]);
71290000                     STV.S ← STV.S-132;
71300000                     STV.CH ← STV.CH + 132;
71310000                     END;
71320000                  CLEAR(BUFOUT,17);
71330000                  MOVE(STV.S,FIRSTCHAR(STV),BUFOUT[*],0);
71340000                  MV(1,QUOTE,7,BUFOUT[(STV.S),W],(STV.S),C);
71350000                  WRITE(PRINT,17,BUFOUT[*]);
71360000                  END;
71370000               END;
71380000 MESSAGE0(7);
71390000 FOR I ← 0 STEP 1 WHILE I ≤ STRMAX AND USEDST[I] DO
71400000    FOR J ← 0 STEP 1 UNTIL STWMAX DO
71410000       IF (STN ← NAMTABL[I,J]).[1:2] = 3 THEN
71420000          BEGIN SN ← MIN(STN.S,63);
71430000          T ← MAX(SN+3-SN.[46:2],8);
71440000          I1 ← VALTABL[I,J].LINK;
71450000          WRITE(BUFOUT[*],FTRACEL,T,DIGITS(I1),I1);
71460000          MV(SN,FIRSTCHAR(STN),BUFOUT[*],3);
71470000          WRITE(PRINT,17,BUFOUT[*]);
71480000          END;
71490000 WRITETIME(0,0);
71500000 END STRINGDUMP;
71510000 %********STRINGDUMP*************************************************
71520000 %********SYNTAXERR*************************************************
71530000 PROCEDURE SYNTAXERR(N,P); VALUE N,P; INTEGER N,P;
71540000 BEGIN
71550000 DEFINE DUMMY = #;
71560000 P ← P MOD FIELDSIZE;
71570000 IF P > 0 THEN
71580000    BEGIN
71590000    CLEAR(BUFOUT,17);
71600000    MOVE(1,QMARK,7,BUFOUT[P,W],P,C+8);
71610000    WRITE(PRINT,17,BUFOUT[*]);
71620000    END;
71630000 WRITE(PRINT,MESSAGE[N]);
71640000 PTR ← MARKER + 8;
71650000 STORECHARS(1,"X",7);
71660000 MESSAGES ← MESSAGES + 1;
71670000 ERRORS ← ERRORS + 1;
71680000 END SYNTAXERR;
71690000 %********SYNTAXERR*************************************************
71700000 %********TEMPCEL***************************************************
71710000 % TEMPCELL PRODUCES A TEMPORARY CELL IN THE SYMBOL TABLE.
71720000 % THIS CELL CAN GENERALLY BE ASSUMED TO BE GONE AFTER THE END OF AN
71730000 % INSTRUCTION. (THIS IS ACTUALLY NOT ALWAYS TRUE, BUT IT IS RATHER
71740000 % DIFFICULT TO PREDICT WHEN A CELL WILL STAY AROUND LONGER.)  SEE ALSO
71750000 % THE PROCEDURE NEWCELL FOR GETTING PERMANENT CELLS.
71760000 ALPHA PROCEDURE TEMPCELL;
```

```
71770000 BEGIN INTEGER ST, I, J;
71780000 LABEL FAIL, RETURN;
71790000 DEFINE ROW=TEMPROW#;
71800000 %
71810000 % IF INFORM THEN INFORMI(7,ROW);
71820000 ST ← NEXTCELL[ROW];
71830000 IF ST = 0 THEN
71840000    BEGIN
71850000    FOR ROW ← 0 STEP 1 WHILE ST = 0 DO
71860000       IF ROW > 15 THEN GO TO FAIL ELSE
71870000       IF USEDST[ROW]
71880000          THEN ST ← NEXTCELL[ROW]
71890000          ELSE BEGIN NEWSTROW(ROW);
71900000             ST ← 0 & ROW CONCSTR;
71910000             END;
71920000    ROW ← ROW - 1;  % THE "STEP" WAS EXECUTED ONE TIME TOO MANY.
71930000    END;
71940000 NEXTCELL[ROW] ← NAME[ST].LINK;
71950000 IF (I←TEMPLIST[ROW]) = 0 THEN
71960000    BEGIN TEMPLIST[ROW] ← ST;
71970000    NAME[ST] ← 0 & ST CLINK;
71980000    END ELSE
71990000    BEGIN J ← NAME[I].LINK;
72000000    NAME[I].LINK ← ST;
72010000    NAME[ST] ← 0 & J CLINK;
72020000    END;
72030000 VALU[ST] ← 0;
72040000 TEMPCELL ← ST;
72050000 % IF USEDCELL[ROW] < ST THEN USEDCELL[ROW] ← ST;
72060000 GO TO RETURN;
72070000 FAIL:
72080000    MESSAGEO(14);
72090000    MESSAGETTO(14);
72100000    WRITE(PRINT,MESSAGE[1]);
72110000    DEATH ← TRUE;
72120000 RETURN:
72130000 % IF INFORM THEN INFORMIA(0,ROW,ST);
72140000 END TEMPCEL;
72150000 %********TEMPCEL*************************************************
72160000 %********TEMPVAL*************************************************
72170000 % TEMPVAL PRODUCES A TEMPORARY SYMB TABLE CELL WHOSE VALUE IS THE
72180000 % STRING REPRESENTATION OF THE INTEGER I.
72190000 ALPHA PROCEDURE TEMPVAL(I);
72200000 VALUE I; INTEGER I;
72210000 BEGIN ALPHA ARRAY NUM[0:1];
72220000 ALPHA AA, AB; INTEGER J;
72230000 %
72240000 WRITE(NUM[*],FI16,I);
72250000 J ← 16 - SKIPCHAR(" ",NUM[0],0);
72260000 TEMPVAL ← AA ← TEMPCELL;
72270000 AB ← STRING(J,AA);
72280000 MV(J,NUM[0],16-J,DATA[AB.R,AB.W],AB.C);
72290000 VALTABL[AA.STR,AA.STW] ← AB;
72300000 END TEMPVAL;
72310000 %********TEMPVAL*************************************************
72320000 %********TRACEFCTCALL********************************************
72330000 BOOLEAN PROCEDURE TRACEFCTCALL(F);
```

```
72340000     VALUE F;
72350000     INTEGER F;  % SYMBOL TABLE LOC OF FUNCTION
72360000 BEGIN
72370000 LABEL FAIL, FIN;
72380000 ALPHA FCTN, FIO, PV;
72390000 INTEGER I, J, SIZE, P, PS, PC, PR;
72400000 DEFINE WRITERECORD = IF WRITEBUFF(BUFOUT,P,SIZE) THEN GO TO FAIL #;
72410000 %
72420000 FCTN ← NAME[F];
72430000 FIO ← IO[F];
72440000 IF FIO.IOTYPE = 2 THEN  % OUTPUT TO FILE.
72450000     BEGIN
72460000     UNIT ← FIO.FILNO;
72470000     SIZE ← IOSIZE[UNIT];
72480000     I ← MIN(63,FCTN,S);
72490000     WRITE(BUFOUT[*],FTRACEFCTCALL,INSTNO,I);
72500000     MV(I,FIRSTCHAR(FCTN),BUFOUT[*],8);
72510000     P ← I + 8;
72520000     IF SP ≤ MKS THEN P ← P + 1 ELSE
72530000     FOR I ← MKS STEP 1 WHILE I < SP DO
72540000        BEGIN  % MOVE PARAM INTO BUFOUT[*];
72550000        IF P←P+1 ≥ SIZE THEN WRITERECORD;
72560000        MV(1,QUOTE,7,BUFOUT[P,W],P,C);
72570000        P ← P + 1;
72580000        PV ← VALU[PST[I]];
72590000        PR ← PV.R;
72600000        PS ← PV.S;
72610000        PC ← PV.CH;
72620000        WHILE PS←P ≥ SIZE DO
72630000           BEGIN  % MOVE IN PART OF PARAM, AND WRITE:
72640000           IF P < SIZE THEN
72650000              BEGIN
72660000              MOVE(SIZE-P,DATA[PR,PC,W],PC,C,BUFOUT[P,W],P,C);
72670000              PS ← PS - SIZE + P;
72680000              PC ← PC + SIZE - P;
72690000              END;
72700000           WRITERECORD;
72710000           END;
72720000        MOVE(PS,DATA[PR,PC,W],PC,C,BUFOUT[P,W],P,C);
72730000        P ← P + PS;
72740000        IF P+1 ≥ SIZE THEN WRITERECORD;
72750000        MV(1,QUOTE,7,BUFOUT[P,W],P,C);
72760000        IF (P←P+1)+1 ≥ SIZE THEN WRITERECORD;
72770000        MV(1,COMMA,7,BUFOUT[P,W],P,C);
72780000        END;
72790000     % CHANGE LAST "," TO ")";
72800000     AA ← ")";
72810000     MV(1,AA,7,BUFOUT[P,W],P,C);
72820000     WRITERECORD;
72830000     TRACEFCTCALL ← TRUE;
72840000     GO TO FIN;
72850000     END FILE OUTPUT ELSE
72860000     % CALL TRACING FUNCTION;
72870000     ;  % NOT YET IMPLEMENTED.
72880000 FAIL;
72890000     TRACEFCTCALL ← FALSE;
72900000 FIN;
```

```
72910000 END TRACEFCTCALL;
72920000 %********TRACEFCTCALL************************************************
72930000 %********TRACEFCTRETURN*********************************************
72940000 BOOLEAN PROCEDURE TRACEFCTRETURN(F,SUC);
72950000    VALUE F, SUC;
72960000    INTEGER F;  % SYMBOL TABLE LOC OF FUNCTION
72970000    BOOLEAN SUC;    % TRUE IF SUCCEEDED.
72980000 BEGIN
72990000 LABEL FAIL, FIN, FAUT;
73000000 INTEGER I, P, SV, CV, RV, SIZE;
73010000 ALPHA FCTN, FIO;
73020000 DEFINE WRITERECORD = IF WRITEBUFF(BUFOUT,P,SIZE) THEN GO TO FAIL #;
73030000 %
73040000 FCTN ← NAME[F];
73050000 FIO ← IO[F];
73060000 IF FIO.IOTYPE = 2 THEN  % OUTPUT TO FILE
73070000    BEGIN
73080000    UNIT ← FIO.FILNO;
73090000    SIZE ← IOSIZE[UNIT];
73100000    I ← MIN(63,FCTN,S);
73110000    WRITE(BUFOUT[*],FTRACEFCTRET,INSTNO,I);
73120000    MV(I,FIRSTCHAR(FCTN),BUFOUT[*],8);
73130000    P ← I + 8;
73140000    WRITE(IOFILE[UNIT],FBL) [FAIL:FAIL];
73150000    IF SUC THEN
73160000       BEGIN  % WRITE OUT VALUE:
73170000       P ← P + 6;
73180000       AA ← VALU[PST[MKS]];  % = VALUE.
73190000       SV ← AA.S;
73200000       CV ← AA.CH;
73210000       RV ← AA.R;
73220000       WHILE SV+P ≥ SIZE DO
73230000          BEGIN
73240000          MOVE(SIZE-P,DATA[RV,CV,W],CV,C,BUFOUT[P,W],P,C);
73250000          CV ← CV + SIZE - P;
73260000          SV ← SV - SIZE + P;
73270000          WRITERECORD;
73280000          END;
73290000       IF SV > 0 THEN
73300000          MOVE(SV,DATA[RV,CV,W],CV,C,BUFOUT[P,W],P,C);
73310000       IF P+P+SV < SIZE THEN  % ROOM FOR QUOTE
73320000          BEGIN MV(1,QUOTE,7,BUFOUT[P,W],P,C);
73330000          WRITE(IOFILE[UNIT][NO],1023,BUFOUT[*]) [FAIL:FAIL];
73340000          END ELSE
73350000          % PUT QUOTE ON NEXT LINE:
73360000          BEGIN WRITE(IOFILE[UNIT],1023,BUFOUT[*])[FAIL:FAIL];
73370000          MV(1,QUOTE,7,BUFOUT[*],0);
73380000          MV(7,BLANKS,0,BUFOUT[*],1);
73390000          WRITE(IOFILE[UNIT][NO],1,BUFOUT[*])[FAIL:FAIL];
73400000          END;
73410000       END SUCCESS CASE ELSE
73420000       % WRITE "FAILED.":
73430000       BEGIN AA ← "FAILED.";
73440000       MV(7,AA,1,BUFOUT[P,W],P,C+3);
73450000       WRITE(IOFILE[UNIT][NO],1023,BUFOUT[*])[FAIL:FAIL];
73460000       END;
73470000    TRACEFCTRETURN ← TRUE;
```

```
73480000     GO TO FIN;
73490000     END FILE OUTPUT ELSE
73500000     % CALL TRACING FUNCTION:
73510000     ;  % NOT YET IMPLEMENTED.
73520000 FAUT:
73530000 FAIL:
73540000     TRACEFCTRETURN ← FALSE;
73550000 FIN:
73560000 END TRACEFCTRETURN;
73570000 %********TRACEFCTRETURN*************************************************
73580000 %********TTINDEX*******************************************************
73590000 % ST SHOULD BE A SYMBOL TABLE CELL POINTING TO A TELETYPE ADDRESS IN
73600000 % THE STANDARD FORM <T.U.>/<B.A.>.  THE VALUE RETURNED IS THE SUBSCRIPT
73610000 % TO USE WITH ID[*] AND STAT[*] FOR THIS TELETYPE.
73620000 % IF THE TELETYPE ISNT ATTACHED, -1 IS RETURNED.
73630000 INTEGER PROCEDURE TTINDEX(ST);
73640000 VALUE ST;
73650000 ALPHA ST;
73660000 BEGIN LABEL FAIL, FIN;
73670000 INTEGER BA, TU, I, J, K, N, IR, IC;
73680000 FINDUSERS;
73690000 ST ← VALU[ST];
73700000 IR ← ST.R; IC ← ST.CH;
73710000 TU ← BA ← 0;
73720000 N ← ST.S;
73730000 I ← SCANCHAR("/",QMARK,DATA[IR,IC,W],IC.C);
73740000 IF I ≥ N OR CHAR(DATA[IR,IC.W],IC.C+I) ≠ "/" THEN GO TO FAIL;
73750000 FOR J ← 0 STEP 1 WHILE J < I DO
73760000     IF K ← CHAR(DATA[IR,IC.W],IC.C+J) > 9
73770000         THEN GO TO FAIL
73780000         ELSE TU ← TU×10 + K;
73790000 IF TU = 0 OR TU > 15 THEN GO TO FAIL;
73800000 FOR J ← I+1 STEP 1 WHILE J < N DO
73810000     IF K ← CHAR(DATA[IR,IC.W],IC.C+J) > 9
73820000         THEN GO TO FAIL
73830000         ELSE BA ← BA×10 + K;
73840000 IF BA > 15 THEN GO TO FAIL;
73850000 BA ← BA & TU[39:44:4];
73860000 I ← -1;
73870000 FOR J ← 0 STEP 1 WHILE J < NUMUSERS AND I < 0 DO
73880000     IF BA = STAT[J].[9:9] THEN I ← J;
73890000 IF I ≥ 0 THEN
73900000     BEGIN TTINDEX ← I;
73910000     GO TO FIN;
73920000     END;
73930000 FAIL:
73940000     TTINDEX ← -1;
73950000 FIN:
73960000 END TTINDEX;
73970000 %********TTINDEX*******************************************************
73980000 %********WRITEBUFF*****************************************************
73990000 BOOLEAN PROCEDURE WRITEBUFF(BUFOUT,P,SIZE);
74000000     VALUE SIZE;
74010000     INTEGER P, SIZE;
74020000     ARRAY BUFOUT[0];
74030000 BEGIN LABEL FAIL, FIN;
74040000 WRITE(IOFILE[UNIT],FBL);
```

```
74050000 WRITE(IOFILE[UNIT][NO],1023,BUFOUT[*]) [FAIL:FAIL];
74060000 P ← 0;
74070000 CLEAR(BUFOUT,SIZE.W);
74080000 WRITEBUFF ← FALSE;
74090000 GO TO FIN;
74100000 FAIL:
74110000    WRITEBUFF ← TRUE;     % TRUE IF OUTPUT FAILS
74120000 FIN:
74130000 END WRITEBUFF;
74140000 %********WRITEBUFF*************************************************************
74150000 %********WRITEDATA*************************************************************
74160000 % DUMP OF PROGRAM/DATA AREAS:
74170000 PROCEDURE WRITEDATA;
74180000 BEGIN INTEGER I;
74190000 PROCEDURE WRITEDATAROW(I); VALUE I; INTEGER I;
74200000 BEGIN INTEGER J; INTEGER ARRAY BUFFER[0:14];
74210000 J ← 0;
74220000 WRITE(BUFFER[*],MESSAGE[37],I);
74230000 MOVE(100,DATA[I,*],0,BUFFER[*],16);
74240000 MOVE(4,BLANKS,0,BUFFER[14],4);
74250000 WRITE(PRINT,MESSAGE[36]);
74260000 WRITE(PRINT,15,BUFFER[*]);
74270000 FILL BUFFER[*] WITH "        ","        ";
74280000 FOR J ← J + 100 WHILE J < 8084 DO
74290000    BEGIN MOVE(100,DATA[I,J,W],J,C,BUFFER[*],16);
74300000    WRITE(PRINT,15,BUFFER[*]);
74310000    END;
74320000 MOVE(84,DATA[I,1012],4,BUFFER[*],16);
74330000 FOR J ← 100,108 DO MOVE(8,BLANKS,0,BUFFER[J,W],J,C);
74340000 WRITE(PRINT,15,BUFFER[*]);
74350000 WRITE(PRINT,MESSAGE[36]);
74360000 END WRITEDATAROW;
74370000 WRITETIME(0,0);
74380000 WRITE(PRINT [PAGE]);
74390000 FOR I ← 0 STEP 1 UNTIL 31 DO
74400000    BEGIN WRITE(PRINT,MESSAGE[35],I,USEDROW[I],I,DPNTR[I]);
74410000    IF USEDROW[I] THEN WRITEDATAROW(I);
74420000    END;
74430000 WRITETIME(0,0);
74440000 END WRITEDATA;
74450000 %********WRITEDATA*************************************************************
74460000 %********WRITEINST*************************************************************
74470000 % WRITEINST PRINTS OUT THE CODE THAT HAS BEEN PRODUCED SO FAR
74480000 % BY THE COMPILER.
74490000 PROCEDURE WRITEINST;
74500000 BEGIN DEFINE DUMMY=#;
74510000 CLEAR(BUFOUT,17);
74520000 MOVE(MIN(PTR-MARKER,132),CODE[MARKER,W],MARKER,C,BUFOUT[*],0);
74530000 WRITE(PRINT,17,BUFOUT[*]);
74540000 END WRITEINST;
74550000 %********WRITEINST*************************************************************
74560000 %********WRITEST***************************************************************
74570000 % SYMBOL TABLE DUMP ROUTINE:
74580000 PROCEDURE WRITEST;
74590000 BEGIN
74600000 FORMAT F(X5,A2,X1,I4," * ",2(4I1,X1,I2,X1,I3,X1,I4,X1,I2,X1,
74610000       I4,"    "),A2,X12,I1,X4,I3,X3,I3,X4,L1,X4,I2),
```

```
74620000    F1(7("*"),X10,"NAMTABL[",I2,",*]",X16,"VALTABL[",I2,",*]",X27,
74630000       "IOUSE FILNO CYCLE OVFL IOTYPE"),
74640000    F2("NEXTCELL[",I2,"] = ",A2,"; TEMPLIST[",I2,"] = ",A2);
74650000 INTEGER I, J, K;
74660000 ALPHA L;
74670000 WRITE(PRINT[1]);
74680000 WRITETIME(0,0);
74690000 FOR I ← 0 STEP 1 UNTIL STRMAX DO IF USEDST[I] THEN
74700000    BEGIN
74710000    WRITE(PRINT,F2,I,NEXTCELL[I],I,TEMPLIST[I]);
74720000    WRITE(PRINT,F1,I,I);
74730000    FOR K ← 0 STEP 1 UNTIL STWMAX DO
74740000       WRITE(PRINT,F,(K&I CONCSTR),K,FOR L ←
74750000          NAMTABL[I,K],VALTABL[I,K] DO
74760000             [L.[1:1],
74770000              L.[2:1],
74780000              L.[3:1],
74790000              L.[4:1],
74800000              L.[5:4],
74810000              L.[9:8],
74820000              L.S,
74830000              L.R,
74840000              L.CH],
74850000          (K&I CONCSTR),
74860000          FOR L ← IOTABL[I,K] DO
74870000             [L.IOUSE,
74880000              L.FILNO,
74890000              L.CYCLE,
74900000              L.OVFL,
74910000              L.IOTYPE]);
74920000    END;
74930000 WRITETIME(0,0);
74940000 END WRITEST;
74950000 %********WRITEST*************************************************************
74960000 %********WRITESTACK**********************************************************
74970000 % WRITESTACK(N) PRINTS OUT THE CONTENTS OF THE STACK, FROM WORD 0
74980000 % TO WORD N.  N SHOULD BE ≤ STACKSIZE.
74990000 PROCEDURE WRITESTACK(N);
75000000 VALUE N;
75010000 INTEGER N;
75020000 BEGIN INTEGER I;
75030000 WRITE(PRINT,FSTACKHEAD);
75040000 FOR I ← 0 STEP 1 UNTIL N DO
75050000    WRITE(PRINT,FSTACKENTRY,
75060000       I,
75070000       PTYPE[I],
75080000       PST[I],
75090000       PNAME[I],
75100000       PLOC[I].S, PLOC[I].R, PLOC[I].CH,
75110000       PSIZE[I],
75120000       PPOINT[I],
75130000       PMINLEFT[I],
75140000       PBACK[I]);
75150000 END WRITESTACK;
75160000 %********WRITESTACK**********************************************************
75170000 %********WRITETIME***********************************************************
75180000 % F IS NO LONGER IN USE HERE.
```

```
75190000 PROCEDURE WRITETIME(F,N); VALUE F,N; INTEGER F,N;
75200000 BEGIN REAL T, TS;
75210000 INTEGER TH, TM;
75220000 T ← TIME(1) / 60;
75230000 TH ← ENTIER(T / 3600);
75240000 TM ← ENTIER((T - TH × 3600) / 60);
75250000 TS ← T - TH × 3600 - TM × 60;
75260000 IF N ≤ 0 THEN
75270000    WRITE(PRINT,FTIME[0],TIME(2)/60,TIME(3)/60) ELSE
75280000 IF N ≤ 1 THEN
75290000    BEGIN ALPHA   D, M, Y, X; ALPHA ARRAY MM[1:12];
75300000    X ← DATE;
75310000    M ← X.[1:5] × 10 + X.[6:6];
75320000    D ← X.[18:6] × 10 + X.[24:6];
75330000    Y ← X.[36:12];
75340000    FILL MM[*] WITH
75350000       "JAN","FEB","MAR","APR","MAY","JUN",
75360000       "JUL","AUG","SEP","OCT","NOV","DEC";
75370000    WRITE(PRINT,FTIME[1],VERSION,DIGITS(D),D,MM[M],Y,TH,TM,TS);
75380000    END ELSE
75390000 IF N ≤ 3 THEN
75400000    WRITE(PRINT,FTIME[N],TH,TM,TS) ELSE
75410000 IF N ≤ 5 THEN
75420000    WRITE(PRINT,FTIME[N],TH,TM,TS,DIGITS(LEVEL),LEVEL,
75430000       DIGITS(INSTNO),INSTNO);
75440000 END WRITETIME;
75450000 %********WRITETIME*************************************************
75460000 %*************************************************************************
75470000 % END OF PROCEDURE DECLARATIONS
75480000 BEGIN COMMENT INITIALIZATION;
75490000 ALPHA ARRAY USERS[0:31];
75500000 %
75510000 % B5500 CHARACTER SET:
75520000 %    0    1    2    3    4    5    6    7    8    9
75530000 %    #    @  QMARK   :    >    ≥    +    A    B    C
75540000 %    D    E    F    G    H    I    .    [    &    (
75550000 %    <    ←    ×    J    K    L    M    N    O    P
75560000 %    Q    R    $    *    -    )    ;    ≤  BLANK   /
75570000 %    S    T    U    V    W    X    Y    Z    ,    %
75580000 %    ≠    =    ]    "
75590000 %
75600000 FILL CONVERTVAL[*] WITH
75610000    "0","1","2","3","4","5","6","7","8","9",
75620000    "=",""","!","≤","%","[","<","A","B","C",
75630000    "D","E","F","G","H","I",".",")","+","≥",
75640000    12, "]",12, "J","K","L","M","N","O","P",
75650000    "Q","R","$","*","-",12, 12, ">"," ","/",
75660000    "S","T","U","V","W","X","Y","Z",",","(",
75670000    "]",12, 12, "&";
75680000 %
75690000 %
75700000 FILL CHARTYPE[*] WITH
75710000    3,  3,  3,  3,  3,  3,  3,  3,  3,  3,
75720000    0,  0,  13, 9,  0,  0,  10, 3,  3,  3,
75730000    3,  3,  3,  3,  3,  3,  3,  0,  0,  5,
75740000    0,  2,  10, 3,  3,  3,  3,  3,  3,  3,
75750000    3,  3,  8,  7,  10, 6,  14, 0,  1,  12,
```

```
75760000     3,  3,  3,  3,  3,  3,  3,  3,  11, 0,
75770000     0,  2,  0,  4;
75780000 %
75790000 FILL DOTTYPE[*] WITH
75800000     0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
75810000     0,  0,  0,  0,  0,  0,  2,  0,  0,  0,
75820000     0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
75830000     0,  0,  2,  0,  0,  0,  0,  1,  0,  0,
75840000     0,  0,  0,  2,  2,  0,  0,  0,  0,  2,
75850000     3,  0,  0,  0,  0,  0,  0,  0,  0,  0,
75860000     0,  0,  0,  0;
75870000 % VALUES FOR DOTTYPE:
75880000 %  0 ERROR
75890000 %  1  .N
75900000 %  2  .+, .-, .x, ./, .*
75910000 %  3  .S
75920000 %
75930000 FILL OPLEVEL[*] WITH
75940000     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
75950000     0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
75960000     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
75970000     0, 0, 3, 0, 0, 0, 0, 0, 0, 0,
75980000     0, 0, 0, 4, 2, 0, 0, 0, 0, 3,
75990000     0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
76000000     0, 0, 0, 0;
76010000 FILL WORDS[*] WITH
76020000     "UNANCHOR","OUNDUMP ","INTEGERT","RUNCATIO","NDISK IN",
76030000     "FORMSILE","NCENDFRE","TURNDATA","COMIORUL","ESPROCES",
76040000     "SAVERELE","ASEPURGE","TELETYPE","DEBUGRUN","ERRDUMPT",
76050000     "RUNCATED","VERFLOWQ","UOTEQMAR","KARROWRE","ADPRINTP",
76060000     "UNCHLOOK","SYSPOTCA","RDNEWDIS","K       ";
76070000 FILL MNEMONIC[0,*] WITH
76080000     6,
76090000     "ON","TO","BY","AT","IN","OFF";
76100000 FILL MNEMONIC[1,*] WITH
76110000     2,
76120000     "FROM","TO";
76130000 FILL CONTROLCARD[*] WITH 0,
76140000     "DEBUG",
76150000     "PCC",
76160000     "LIST",
76170000     "UNLIST",
76180000     "PUNCH",
76190000     "EJECT",
76200000     "SPACE",
76210000     "WIDTH",
76220000     "26",
76230000     "3600",
76240000     "B5500",
76250000     "DEFINE",
76260000     "WAIT",
76270000     "LIMIT",
76280000     "SIZE",
76290000     "SET",
76300000     "LIBRARY",
76310000     "LOAD",
76320000     "COMPILE",
```

```
76330000     "INFORM",
76340000     "SILENCE";
76350000 FILL INTRINSFCT[3,*] WITH
76360000     8,".EQ",".NE",".LT",".LE",".GT",".GE","POP","EOF";
76370000 FILL INTRINSNDX[3,*] WITH
76380000     9, 21, 19, 16, 15, 14, 25, 8;
76390000 FILL INTRINSFCT[4,*] WITH
76400000     17,
76410000     ".NUM","PUSH","PAGE","SIZE","MODE",
76420000     "DUMP","DATE","TRIM","TIME",
76430000     "FILE","LOOK","WAIT","FILL","USER",
76440000     "SEEK","C@PY";
76450000 FILL INTRINSNDX[4,*] WITH
76460000     22, 26, 24, 30, 20, 7, 4, 36, 32,
76470000     12, 18, 39, 13, 42, 43, 48;
76480000 FILL INTRINSFCT[5,*] WITH
76490000     10,
76500000     "SPACE","UNEQL","TRACE","CLOSE","RULES",
76510000     "LEVEL","OPSYN","CALLF",".RANF","CLEAR";
76520000 FILL INTRINSNDX[5,*] WITH
76530000     31, 38, 46, 3, 29, 17, 23, 2, 50, 51;
76540000 FILL INTRINSFCT[6,*] WITH
76550000     14,
76560000     ".REMDR","ANCHOR","UNANCH","EQUALS","DEFINE",
76570000     "DETACH","TRACEF","REWIND","TRACEL","STATUS",
76580000     "TRACES","RECORD","ASSIGN","SEARCH";
76590000 FILL INTRINSNDX[6,*] WITH
76600000     27, 1, 37, 10, 5, 6, 34, 28, 35, 41,
76610000     33, 45, 47, 49;
76620000 FILL INTRINSFCT[7,*] WITH
76630000     4,"EXECUTE","COMPILE","SUSPEND","RELEASE";
76640000 FILL INTRINSNDX[7,*] WITH
76650000     11, 0, 40, 44;
76660000 FILL MONTHS[*] WITH 31,28,31,30,31,30,31,31,30,31,30,31;
76670000 IF ((TEMP←TIME(0)).[24:6] + TEMP.[18:6] × 10) MOD 4 = 0 THEN
76680000     MONTHS[2] ← 29;
76690000 %
76700000 %
76710000 %
76720000 RANDNO ← 8388607;  % INITIALIZE RANDOM NUMBER GENERATOR
76730000 CRLF ← "≤≠←";
76740000 ARROW ← "←"; SLASH ← "/";
76750000 EQSIGN ← "=";
76760000 QMARK ← 12; QUOTE ← """; BLANK ← " "; COLON ← ":";
76770000 STAR ← "*"; COMMA ← ",";
76780000 STOPPER ← QUOTE & QMARK[36:42:6] & BLANK[30:42:6];
76790000 BLANKS ← "    "; MOVE(4,BLANKS,4,BLANKS,0);
76800000 TEENYNEG ← REAL(NOT FALSE);
76810000 ANCHORMODE ← 2;
76820000 TRACEALL ← BOOLEAN(COMMON.[47:1]);
76830000 DUMPALL ← BOOLEAN(COMMON.[46:1]);
76840000 SYSTEMDEBUGGING ← BOOLEAN(COMMON.[45:1]);
76850000 USEDROW[0] ← TRUE;
76860000 EXECUTE ← TRUE;
76870000 FIELDSIZE ← 72;
76880000 WAITTIME ← 18000;  % 5 MINUTES
76890000 FINDUSERS;  % LOCATE ATTACHED TELETYPES
```

```
76900000 IF DATACOMF THEN
76910000    BEGIN
76920000    IF NUMUSERS = 1 THEN PROGRAMFROMREMOTE ← TRUE;
76930000    FOR USER ← 0 STEP 1 UNTIL NUMUSERS-1 DO
76940000       BEGIN
76950000       WRITE(DCWRITE[*],FTTHELLO,VERSION);
76960000       IF OUTPUT THEN;
76970000       IF NOT PROGRAMFROMREMOTE THEN
76980000          BEGIN WRITE(DCWRITE[*],FFROMREMOTE,12);
76990000          IF OUTPUT THEN IF INPUT THEN
77000000             BEGIN
77010000             WHILE J←CHAR(DCREAD[*],SKIPCHAR(" ",DCREAD[*],0))≠"Y"
77020000                   AND J ≠ "N" DO
77030000                BEGIN WRITE(DCWRITE[*],FYESORNO);
77040000                IF OUTPUT THEN;
77050000                IF INPUT THEN;
77060000                END;
77070000             IF J = "Y" THEN
77080000                BEGIN PROGRAMFROMREMOTE ← TRUE;
77090000                MAINUSER ← USER;
77100000             END ELSE
77110000             BEGIN MV(3,CRLF,5,DCWRITE[*],0);
77120000             IF OUTPUT THEN;
77130000             END;
77140000             END;
77150000          END;
77160000       END;
77170000    USER ← MAINUSER;
77180000    END;
77190000 IF NOT PROGRAMFROMREMOTE THEN
77200000    BEGIN READ(PROGRAM,10,BUFFER[*]);
77210000    BUFFERFULL ← TRUE;
77220000    END;
77230000 WRITETIME(0,1);
77240000 LOADF ← TRUE;
77250000 LOADER(PROGRAM);
77260000 LOADF ← FALSE;
77270000 IF PUNCHF THEN CLOSE(PUNCH);
77280000 %
77290000 WRITE(PRINT,FERRS,DIGITS(ERRORS),ERRORS);
77300000 WRITETIME(0,2);
77310000 WRITETIME(0,0);
77320000 IF INFORM THEN
77330000    BEGIN WRITEST;
77340000    WRITEDATA;
77350000    END;
77360000 %
77370000 %
77380000 IF EXECUTE THEN
77390000    BEGIN WRITETIME(0,3);
77400000    WRITE(PRINT,FASTERISKS);
77410000    %
77420000    INTERPRETER;
77430000    %
77440000    MESSAGEO(22);
77450000    MESSAGETTI(14,INSTNO);
77460000    RESULT ← FALSE;
```

```
77470000     WRITE(PRINT,FASTERISKS);
77480000     WRITETIME(0,5);
77490000     END;
77500000 GO TO ENDOFRUN;
77510000 %
77520000 %
77530000 END INITIALIZATION;
77540000 %
77550000 ABORTION:     % FATAL SYSTEM ERROR.
77560000     MESSAGETTO(7);
77570000     RESULT ← FALSE;
77580000     SYSTEMERROR ← TRUE;
77590000 %
77600000 ENDTERPRET:
77610000     WRITE(PRINT,FASTERISKS);
77620000     WRITETIME(0,IF RESULT THEN 4 ELSE 5);
77630000     IF RESULT THEN MESSAGETTI(23,INSTNO);
77640000 ENDOFRUN:
77650000     BEGIN DEFINE DUMMY=#;
77660000     IF NOT RESULT THEN MESSAGETTI(10,INSTNO);
77670000     IF TRACEALL THEN INFORM ← TRUE;
77680000     WRITETIME(0,0);
77690000     WRITE(PRINT,FRULES,RULES[0]);
77700000     IF DUMPALL THEN DMPST ← DMPDATA ← DMPSTR ← TRUE;
77710000     IF SYSTEMERROR THEN
77720000        BEGIN WRITE(PRINT[DBL],FBL);
77730000        FOR I ← 0,1,2,3 DO WRITE(PRINT[NO],FSENDCOPY);
77740000        INFORM ← TRUE;
77750000        WRITE(PRINT[DBL],FBL);
77760000        END;
77770000     WRITE(PRINT,FGCS,DIGITS(GCS),GCS,DIGITS(GCTIMECP)+2,GCTIMECP,
77780000        DIGITS(GCTIMEIO)+2,GCTIMEIO);
77790000     IF INFORM OR DMPST THEN WRITESTACK(STACKSIZE);
77800000     IF INFORM OR DMPST THEN WRITEST;
77810000     IF INFORM OR DMPDATA THEN WRITEDATA;
77820000     IF INFORM OR DMPSTR OR (ERRDUMP AND NOT RESULT)
77830000        THEN STRINGDUMP(INSTNO);
77840000     END ENDOFRUN SEGMENT;
77850000 END ALGOL PROCEDURES;
77860000 END END END GLOBAL STREAM PROCEDURES;
77870000 %*********************************************************************
77880000 END PROGRAM.
99999999 END;END.        LAST CARD ON OCRDING TAPE

      LABEL  000000000PRNT   00177105?USER=SITE   ; EXECUTE COPY   /SITE                                    COPY   /SITE
```