```
? COMPILE DUMP/ANALYZE ALGOL LIBRARY

          PACKET   956
          INPUT    597 CARDS FROM ZIP
          TIME    1041
          DATE   77311 MONDAY, 11/07/77



 *** BURROUGHS B5700 DCMCP MARK XVI.0.178  AND INTRINSICS MARK XVI.0.132 ***

     #NO MESSAGES TODAY

10:41:18  ? COMPILE DUMP/ANALYZE ALGOL LIBRARY
10:41:18       ? ALGOL FILE TAPE= SYMBOL/DUMPANL DISK SERIAL
10:41:19       ? FILE LINE= LINE PRINT OR BACK UP
10:41:19  ? DATA CARD
10:41:25            5:ALGOL/DUMP= 2 BOJ 1041 11/04/77
10:41:27            CDB IN CARD DB:ALGOL/DUMP= 2
10:41:31            DKA IN SER SYMBOL DUMPANL:ALGOL/DUMP= 2
10:41:32            PBD0957 OUT 011 LINE:ALGOL/DUMP= 2
10:41:45            DKA OUT RDM DUMP ANALYZE:ALGOL/DUMP= 2
10:41:46            DKA OUT SER DSK2 SITE:ALGOL/DUMP= 2
10:41:46            DKA OUT SER DSK1 SITE:ALGOL/DUMP= 2
10:44:49            CDB REL CARD DB:ALGOL/DUMP= 2
10:44:49  ? END,
10:44:49            DKA REL SYMBOL DUMPANL:ALGOL/DUMP= 2
10:44:51            DKA OUT SER DSRT1 SITE:ALGOL/DUMP= 2
10:45:10            DKA OUT SER DSRT2 SITE:ALGOL/DUMP= 2
10:45:25            DKA REL DSRT1 SITE:ALGOL/DUMP= 2
10:45:25            DKA REL DSRT2 SITE:ALGOL/DUMP= 2
10:45:26            DKA OUT SER DSRT1 SITE:ALGOL/DUMP= 2
10:46:10            DKA OUT SER DSRT2 SITE:ALGOL/DUMP= 2
10:46:55            DKA REL DSRT1 SITE:ALGOL/DUMP= 2
10:46:55            DKA REL DSRT2 SITE:ALGOL/DUMP= 2
10:46:56            DKA REL DSK2 SITE:ALGOL/DUMP= 2
10:46:56            DKA REL DSK1 SITE:ALGOL/DUMP= 2
10:46:57            PBD0957 REL 011 LINE 5931:ALGOL/DUMP= 2
10:46:57            DKA REL DUMP ANALYZE:ALGOL/DUMP= 2
10:46:58            ALGOL/DUMP= 2 SYNTX ERR 1046
10:47:01            FOR ALGOL/DUMP= 2: PROCESS= 257 SECS, IO= 143 SECS, OLAY= 5
10:47:02            PKT#0956 REMOVED
```

DUMP /ANALYZE

SOURCE FILE: SYMBOL /DUMPANL

```
COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE                    *
         * FILE ID: SYMBOL/DUMPANL  TAPE ID: SYMBOL2/FILE000             *
         * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION         *
         * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED               *
         * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON             *
         * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF               *
         * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232                *
         *                                                               *
         * COPYRIGHT (C) 1971, 1972, 1974                                *
         *       BURROUGHS CORPORATION                                   *
         *  AA320206   AA332366   AA386657                               *;
                    DUMP/ANALYZE
                  PATCHED 6/4/74 11/21/73
    SEVERAL OPTIONS ARE AVAILABLE FOR CONTROLLING THE ANALYSIS
OF A MEMORY DUMP.  THEY MAY BE CLASSIFIED INTO  FOUR CATEGORIES:

I.  THOSE WHICH SPECIFY THE FORMAT OF THE PRINTOUT OF MEMORY:
SPLASH DUMP,OCTAL DUMP,ALPHA/OCTAL DUMP,OR ALPHA DUMP;
EITHER SINGLY OR DOUBLY SPACED.

II.  THOSE WHICH INCLUDE OR EXCLUDE AREAS OF MEMORY NOT OTHERWISE
INCLUDED OR EXCLUDED:
1.  INCLUSION OF AVAILABLE AREAS.
2.  EXCLUSION OF:
A.  MCP CODE AREAS.
B.  NORMAL STATE CODE AREAS.
C.  NORMAL STATE AREAS OTHER THAN THOSE RELATING TO ONE
SPECIFIC MIX INDEX WHERE THE MIX INDEX IS SPECIFIED BY THE USER.
D.  ENTIRE CONTENTS OF MEMORY
III.  THOSE WHICH CAUSE OMISSION OF CERTAIN SECTIONS OF ANALYSIS:
1.  OMISSION OF THE PRINTOUT OF MCP PRT IDENTIFIERS, SORTED
ALPHABETICALLY AND BY PRT LOCATION.

IV.  THOSE WHICH CONTROL THE DUMPING OF STACKS:
1.  EXCLUSION OF ALL NORMAL STATE STACKS EXCEPTING THE ONE
BELONGING TO THE SAME MIX INDEX AS THE ONE SPECIFIED BY THE USER.
2.  INCLUSION OF A USER SELECTED "STACK" AREA WHICH THE
ANALYZER WOULD BE UNABLE TO LOCATE WITHOUT THE USER"S ASSISTANCE.

    THE MAJORITY OF OPTIONS MAY BE SET USING A COMMON CONTROL
CARD; A FEW MAY BE SET ONLY VIA THE "IN" MESSAGE TO ALTER THE
CONTENTS OF CERTAIN PRT CELLS WHICH HAVE BEEN RESERVED  FOR
CONTROL FUNCTIONS.
```

```
00000001 T  0001:0000:0
00000002 T  0001:0000:0
00000003 T  0001:0000:0
00000004 T  0001:0000:0
00000005 T  0001:0000:0
00000006 T  0001:0000:0
00000007 T  0001:0000:0
00000008 T  0001:0000:0
00000009 T  0001:0000:0
00000010 T  0001:0000:0
00000011 T  0001:0000:0
00000012 T  0001:0000:0
00000013 T  0001:0000:0
00000015 T  0001:0000:0
00000020 T  0001:0000:0
00000025 T  0001:0000:0
00000030 T  0001:0000:0
00000040 T  0001:0000:0
00000050 T  0001:0000:0
00000055 T  0001:0000:0
00000060 T  0001:0000:0
00000070 T  0001:0000:0
00000080 T  0001:0000:0
00000110 T  0001:0000:0
00000120 T  0001:0000:0
00000130 T  0001:0000:0
00000140 T  0001:0000:0
00000150 T  0001:0000:0
00000155 T  0001:0000:0
00000160 T  0001:0000:0
00000170 T  0001:0000:0
00000180 T  0001:0000:0
00000205 T  0001:0000:0
00000210 T  0001:0000:0
00000220 T  0001:0000:0
00000230 T  0001:0000:0
00000240 T  0001:0000:0
00000250 T  0001:0000:0
00000255 T  0001:0000:0
00000260 T  0001:0000:0
00000270 T  0001:0000:0
00000280 T  0001:0000:0
00000290 T  0001:0000:0
00000295 T  0001:0000:0
```

THE COMMON CONTROL CARD IS ASSUMED TO CONTAIN AN EIGHT DIGIT NUMBER--IF LESS THAN EIGHT APPEAR, LEADING ZEROES ARE AUTOMATICALLY SUPPLIED. THE ANALYZER SEPARATES THE 8 DIGITS INTO TWO GROUPS:

A. THE 5 LEFT-MOST DIGITS ARE EXPECTED TO BE *OCTAL* DIGITS WHICH REPRESENT THE TOP-OF-STACK ADDRESS OF THE USER SPECIFIED "STACK" AREA. IF ANY DIGIT EXCEEDS "7" OR IF ALL FIVE ARE "0", ONLY STACKS LOCATED BY THE ANALYZER WILL BE DUMPED. OTHERWISE, ONE ADDITIONAL "STACK", PROVIDING IT ALREADY HAS NOT BEEN DUMPED, WILL BE DUMPED STARTING AT THIS ADDRESS AND CONTINUING UNTIL 200(DECIMAL) WORDS BELOW IT HAVE BEEN PRINTED(THE STACK IS ASSUMED TO BE A CONTROL STATE STACK FOR PURPOSES OF STACK ANALYSIS). SHOULD MORE OR LESS STACK BE DESIRED, THE AMOUNT (IN DECIMAL) OF STACK TO ANALYZE CAN BE ENTERED INTO PRT CELL 27(OCTAL) USING THE "IN" MESSAGE.
IN MOST CASES, THE TOP-OF-STACK VALUE USED MAY BE THE SETTING OF THE "S" REGISTER TAKEN FROM THE DISPLAY PANEL AT THE TIME OF THE HANG. IF NO TOP-OF-STACK IS SPECIFIED THE ANALYZER WILL OBTAIN ONE FROM THE CONTROL STATE MSCW LOCATED IN CELL 7. THE STACK OBTAINED IN THIS MANNER IS GENERALLY OF INTEREST, ESPECIALLY WHEN LINKS HAVE BEEN CLOBBERED.

B. THE RIGHT-MOST 3 DIGITS ARE INTERPRETED AS DECIMAL DIGITS AND SERVE TO SET THE MAJORITY OF ANALYZER OPTIONS. GENERALLY SPEAKING, THESE OPTIONS ARE INTEGRAL POWERS OF 2, ARE INDEPENDENT OF ONE ANOTHER AND MAY THEREFORE BE USED ADDITIVELY. THAT IS, THE COMBINED FUNCTIONS OF COMMON VALUES 2,16,32, AND 64 COULD BE INVOKED BY "COMMON=114"(2+16+32+64). THE EXCEPTIONS TO THIS RULE WILL BE ELABORATED AFTER EACH COMMON OPTION IS DESCRIBED IN DETAIL.

COMMON CARD OPTIONS

NOTE: IN THE DESCRIPTION OF EACH OPTION, IT IS ASSUMED THAT A STANDARD MEMORY DUMP ANALYSIS IS OBTAINED *EXCEPT* FOR THE DIFFERENCES GENERATED BY THAT PARTICULAR OPTION. A "STANDARD" ANALYSIS IS ONE OBTAINED WITH NO COMMON VALUE(I. E. COMMON=0).

COMMON=0.
YIELDS A STANDARD ANALYSIS. AN MCP/PRT FILE IS REQUIRED.

COMMON=1.
YIELDS A "SPLASH",UNANALYZED DUMP OF MEMORY. NO MCP/PRT IS REQUIRED. A "COMMENT" ENTERED WHEN THE DUMP TAPE WAS CREATED IS ALSO PRINTED.

COMMON=2.
DUMPS ALL AVAILABLE AREAS.

COMMON=4.
EXCLUDES ALL NORMAL STATE OBJECT CODE.
THIS INCLUDES ALL PROGRAM SEGMENTS ASSOCIATED WITH A MIX INDEX AND ALL INTRINSIC SEGMENTS.

COMMON=8.
EXCLUDES ALL MCP OBJECT CODE.

00000300 T 0001:0000:0
00000310 T 0001:0000:0
00000320 T 0001:0000:0
00000330 T 0001:0000:0
00000335 T 0001:0000:0
00000340 T 0001:0000:0
00000350 T 0001:0000:0
00000360 T 0001:0000:0
00000370 T 0001:0000:0
00000380 T 0001:0000:0
00000390 T 0001:0000:0
00000400 T 0001:0000:0
00000410 T 0001:0000:0
00000420 T 0001:0000:0
00000430 T 0001:0000:0
00000435 T 0001:0000:0
00000440 T 0001:0000:0
00000445 T 0001:0000:0
00000450 T 0001:0000:0
00000451 T 0001:0000:0
00000452 T 0001:0000:0
00000453 T 0001:0000:0
00000455 T 0001:0000:0
00000460 T 0001:0000:0
00000470 T 0001:0000:0
00000480 T 0001:0000:0
00000490 T 0001:0000:0
00000500 T 0001:0000:0
00000510 T 0001:0000:0
00000520 T 0001:0000:0
00000530 T 0001:0000:0
00000540 T 0001:0000:0
00000550 T 0001:0000:0
00000560 T 0001:0000:0
00000570 T 0001:0000:0
00000580 T 0001:0000:0
00000590 T 0001:0000:0
00000600 T 0001:0000:0
00000610 T 0001:0000:0
00000620 T 0001:0000:0
00000625 T 0001:0000:0
00000630 T 0001:0000:0
00000640 T 0001:0000:0
00000650 T 0001:0000:0
00000660 T 0001:0000:0
00000665 T 0001:0000:0
00000670 T 0001:0000:0
00000680 T 0001:0000:0
00000765 T 0001:0000:0
00000770 T 0001:0000:0
00000780 T 0001:0000:0
00000790 T 0001:0000:0
00000800 T 0001:0000:0
00000815 T 0001:0000:0
00000820 T 0001:0000:0
00000830 T 0001:0000:0
00000845 T 0001:0000:0

COMMON=16.
    CAUSES CERTAIN ARRAYS SELECTED BY THE USER TO BE DUMPED.
SEE THE COMMENT AT SEQUENCE #00807480 FOR DETAILS.

COMMON=32.
    INHIBITS THE DUMP OF THE MCP PRT IDENTIFIERS SORTED
ALPHABETICALLY AND BY PRT LOCATION.

COMMON=64.
    YIELDS AN OCTAL-ONLY DUMP OF MEMORY FORMATTED SIX WORDS/LINE.

COMMON=128.
    YIELDS AN ALPHA/OCTAL DUMP OF MEMORY. EACH PRINTED LINE
CONTAINS FOUR WORDS OF OCTAL FOLLOWED, ON THE SAME LINE, BY
THEIR ALPHA EQUIVALENT.

COMMON=256.
                        LAST PATCHED 12/11/73
    YIELDS AN ALPHA DUMP OF MEMORY FORMATTED TWELVE WORDS
OF ALPHA PER PRINTED LINE.

COMMON=384.
    COMPLETELY INHIBITS THE PRINTOUT OF THE CONTENTS OF MEMORY.
THIS DOES NOT IMPLY THAT MEMORY IS NOT ANALYZED: VERIFICATION
OF MEMORY LINKS IS MADE, THE BED AND SLATE ARE CHECKED,
STACKS ARE LOCATED,ETC.

COMMON=512
    CAUSES THE CONTENTS OF THE "ARGH" ARRAY TO BE PRINTED UP TO THE
LAST VALID WORD POINTED TO BY "YECH". THIS INFORMATION IS PRESENT
ONLY WHEN AN MCP PATCH,THE "ROTO" PATCH, HAS BEEN COMPILED INTO THE
MCP.

PRT CELLS WHICH EXERCISE CONTROL FUNCTIONS

NOTE: PRT LOCATIONS SPECIFIED REFER TO OCTAL PRT LOCATIONS FOR
USE IN AN "IN" MESSAGE.

PRT 25.
    THIS CELL IS NORMALLY SET FROM THE RIGHT-MOST 3 DIGITS WHICH
APPEAR ON A COMMON CARD; HOWEVER, THEY MAY BE EXPLICITLY SET
AND/OR MODIFIED DURING EXECUTION OF THE ANALYZER VIA THE "IN"
MESSAGE.

PRT 26.
    THIS CELL CAN BE USED TO ENABLE A DUMP-BY-MIX-INDEX. IF SET
TO A NON-ZERO DECIMAL NUMBER EQUAL TO THE MIX INDEX OF A JOB IN
THE MIX AT THE TIME THE DUMP IS TAKEN, IT WILL CAUSE MEMORY AREAS
BELONGING TO OTHER MIX INDEXES TO BE EXCLUDED FROM THE ANALYSIS.
NOTE THAT IF THE CONTENTS OF THIS CELL DOES NOT CORRESPOND TO A
VALID MIX INDEX, *NO* NORMAL STATE STACKS WILL BE DUMPED NOR
WILL ANY NORMAL STATE AREAS ASSOCIATED WITH ANY MIX INDEX BE
PRINTED.

PRT 27.
    THIS CELL MAY BE USED ONLY IN CONJUNCTION WITH THE

00000850 T 0001:0000:0
00000860 T 0001:0000:0
00000870 T 0001:0000:0
00000875 T 0001:0000:0
00000880 T 0001:0000:0
00000890 T 0001:0000:0
00000900 T 0001:0000:0
00000905 T 0001:0000:0
00000910 T 0001:0000:0
00000920 T 0001:0000:0
00000945 T 0001:0000:0
00000950 T 0001:0000:0
00000960 T 0001:0000:0
00000970 T 0001:0000:0
00000980 T 0001:0000:0
00000985 T 0001:0000:0
00000990 T 0001:0000:0
00000999 T 0001:0000:0
00001000 T 0001:0000:0
00001010 T 0001:0000:0
00001015 T 0001:0000:0
00001020 T 0001:0000:0
00001030 T 0001:0000:0
00001040 T 0001:0000:0
00001050 T 0001:0000:0
00001060 T 0001:0000:0
00001070 T 0001:0000:0
00001072 T 0001:0000:0
00001073 T 0001:0000:0
00001074 T 0001:0000:0
00001075 T 0001:0000:0
00001076 T 0001:0000:0
00001077 T 0001:0000:0
00001078 T 0001:0000:0
00001080 T 0001:0000:0
00001085 T 0001:0000:0
00001090 T 0001:0000:0
00001100 T 0001:0000:0
00001110 T 0001:0000:0
00001120 T 0001:0000:0
00001130 T 0001:0000:0
00001140 T 0001:0000:0
00001150 T 0001:0000:0
00001160 T 0001:0000:0
00001165 T 0001:0000:0
00001170 T 0001:0000:0
00001180 T 0001:0000:0
00001190 T 0001:0000:0
00001200 T 0001:0000:0
00001210 T 0001:0000:0
00001220 T 0001:0000:0
00001230 T 0001:0000:0
00001240 T 0001:0000:0
00001250 T 0001:0000:0
00001255 T 0001:0000:0
00001260 T 0001:0000:0
00001270 T 0001:0000:0

PREVIOUSLY DESCRIBED COMMON CARD CONVENTION OF SPECIFYING AN OCTAL STACK ADDRESS AS THE FIRST 5 DIGITS OF THE COMMON VALUE. IF MADE NON-ZERO, THEN THE DECIMAL VALUE ENTERED IN CELL 27 WILL DETERMINE THE AMOUNT OF STACK DUMPED BELOW THE TOP-OF-STACK VALUE. IF LEFT ZERO, THE DEFAULT AMOUNT OF STACK DUMPED(AND ANALYZED) WILL BE 200(DECIMAL) WORDS.

PRT 30.
THIS CELL, IF SET TO 1, WILL CAUSE THE PRINTOUT OF MEMORY TO BE DOUBLE-SPACED. OTHERWISE, THE PRINTOUT IS SINGLE-SPACED.

PRECEDENCE OF OPTIONS AND SUGGESTIONS FOR COMBINING THEM

AS STATED EARLIER, COMMON OPTIONS MAY BE USED SEPARATELY OR IN COMBINATION. ALTHOUGH MOST COMBINATIONS ARE ALLOWED, THERE ARE IMPORTANT EXCEPTIONS. THE RULES ARE:

1. A COMMON VALUE OF 1 PRECLUDES THE USE OF ALL OTHERS.
2. A COMMON VALUE OF 512 PRECLUDES THE USE OF ALL OTHERS BUT 1.
3. A COMMON VALUE OF 384 PRECLUDES THE USE OF ALL REMAINING COMMON VALUES EXCEPT FOR 16 AND 32. NOTE THAT 384 IS THE SUM OF 128+256.
4. COMMON VALUES 2,4,8,16, AND 32 ARE TOTALLY INDEPENDENT AND ADDITIVE IN ALL POSSIBLE COMBINATIONS.

SOME SPECIFIC EXAMPLES OF COMMON OPTIONS

A. "COMMON=35721432".
THIS CAUSES A "STACK" AT 35721 TO BE DUMPED ALONG WITH OTHER STACKS; NONE OF MEMORY IS PRINTED(384), THE MCP PRT IDENTIFIERS ARE NOT PRINTED(32), AND SELECTED ARRAYS ARE DUMPED(16). THIS COMBINATION MIGHT BE USED TO PROVIDE A MINIMUM OF OUTPUT AFTER A MEMORY DUMP HAS BEEN ONCE ANALYZED BUT AFTERWARDS FOUND TO HAVE MISSED A STACK(IN THIS CASE, LOCATED NEAR ADDRESS 35721).

B. "COMMON=12".
THIS ELIMINATES ALL OBJECT CODE FROM THE PRINTOUT OF MEMORY.

C. "COMMON=66".
THIS ENSURES THAT AVAILABLE AREAS ARE ALWAYS PRINTED(2) AND PROVIDES THAT MEMORY IS PRINTED IN OCTAL FORMAT ONLY(64).

D. "COMMON=8".
THIS TOGETHER WITH PRT 26 SET TO A VALID MIX INDEX REDUCES THE AMOUNT OF OUTPUT OBTAINED DURING A MIX-ONLY DUMP. NO MCP CODE IS PRINTED LEAVING ONLY NON-MCP CODE AREAS BESIDES THOSE RELATING TO THE MIX INDEX BEING DUMPED. FINALLY, THE ONLY NORMAL STATE STACK DUMPED IS THE ONE BELONGING TO THE PARTICULAR MIX INDEX.

PROCESSING MULTIFILE DUMP TAPES

THE ANALYZER WILL AUTOMATICALLY PROCESS, IN A GIVEN RUN, ALL FILES ON A "DPMT" TAPE IF LABEL EQUATION IS MADE TO ANY FILE BEYOND THE FIRST ONE. THE ORDER OF ANALYSIS IS OPPOSITE THAT OF CREATION. FOR EXAMPLE, IF LABEL EQUATION IS MADE TO THE FOURTH FILE, "FILE MDUMP=MEMORY/DUMP004", THE FOURTH FILE IS FIRST ANALYZED, THE THIRD NEXT, THE SECOND AFTER THAT ONE AND THE FIRST FILE LAST. AS PROCESSING OF A NEW FILE IS BEGUN, A MESSAGE IS WRITTEN TO THE CONSOLE SO THAT THE RUN

```
00001280 T  0001:0000:0
00001290 T  0001:0000:0
00001300 T  0001:0000:0
00001310 T  0001:0000:0
00001320 T  0001:0000:0
00001330 T  0001:0000:0
00001335 T  0001:0000:0
00001340 T  0001:0000:0
00001350 T  0001:0000:0
00001360 T  0001:0000:0
00001370 T  0001:0000:0
00001380 T  0001:0000:0
00001390 T  0001:0000:0
00001400 T  0001:0000:0
00001410 T  0001:0000:0
00001420 T  0001:0000:0
00001430 T  0001:0000:0
00001435 T  0001:0000:0
00001440 T  0001:0000:0
00001450 T  0001:0000:0
00001460 T  0001:0000:0
00001530 T  0001:0000:0
00001540 T  0001:0000:0
00001550 T  0001:0000:0
00001560 T  0001:0000:0
00001565 T  0001:0000:0
00001560 T  0001:0000:0
00001565 T  0001:0000:0
00001570 T  0001:0000:0
00001580 T  0001:0000:0
00001590 T  0001:0000:0
00001600 T  0001:0000:0
00001610 T  0001:0000:0
00001620 T  0001:0000:0
00001630 T  0001:0000:0
00001640 T  0001:0000:0
00001650 T  0001:0000:0
00001660 T  0001:0000:0
00001670 T  0001:0000:0
00001690 T  0001:0000:0
00001700 T  0001:0000:0
00001710 T  0001:0000:0
00001730 T  0001:0000:0
00001740 T  0001:0000:0
00001750 T  0001:0000:0
00001760 T  0001:0000:0
00001770 T  0001:0000:0
00001790 T  0001:0000:0
00001800 T  0001:0000:0
00001900 T  0001:0000:0
00003000 T  0001:0000:0
00003005 T  0001:0000:0
00003010 T  0001:0000:0
00003020 T  0001:0000:0
00003030 T  0001:0000:0
00003040 T  0001:0000:0
00003050 T  0001:0000:0
00003060 T  0001:0000:0
00003070 T  0001:0000:0
```

```
MAY BE DISCONTINUED IF NO FURTHER FILES ARE TO BE ANALYZED.                    00003080 T  0001:0000:0
                                                                               00003090 T  0001:0000:0
    A COMMON VALUE MAY BE INCLUDED IN THE COMMENT PORTION OF THE               00003100 T  0001:0000:0
REMARKS ENTERED AFTER A "DPMT" COMMAND OR AFTER KEYING IN THE UNIT             00003110 T  0001:0000:0
WHEN USING THE MEMORY DUMP DECK.  THE SYNTAX IS: "COMMON="<COMMON VALUE>       00003120 T  0001:0000:0
, WHERE "COMMON" MAY BE ABBREVIATED AS "COM" AND MUST BEGIN NO LATER           00003130 T  0001:0000:0
THAN THE 140"TH CHARACTER OF THE MESSAGE.  A COMMON VALUE ENTERED              00003140 T  0001:0000:0
IN THIS MANNER OVERRIDES ANY WHICH MAY BE SUPPLIED BY LABEL EQUATION.          00003150 T  0001:0000:0
ONE EXAMPLE IS: "DPMT COMMON=36; ALL JOBS APPEAR ASLEEP."                      00003160 T  0001:0000:0
                                                                               00003170 T  0001:0000:0
BEGIN                                                                          00003800 T  0001:0000:0
                                                                    START OF SEGMENT **********    2
BOOLEAN COMMON;                                                                00003900 T  0002:0000:0
INTEGER PRT26; DEFINE ONEMIX=PRT26#; % FOR DUMPING ONLY ONE JOB                00004000 T  0002:0000:0
INTEGER PRT27; DEFINE MYSTACKSIZE=PRT27#; % IF ≠0, AMT BELOW MYSTACKADR        00004100 T  0002:0000:0
BOOLEAN PRT30; DEFINE DOUBLESPACE=PRT30#;% DBL-SPACE DUMP OF MEMORY            00004200 T  0002:0000:0
BOOLEAN PRT31;% RFE                                                            00004300 T  0002:0000:0
BOOLEAN PRT32;% USED FOR DEBUGGING THE ANALYZER                                00004400 T  0002:0000:0
INTEGER MYSTACKADR;% IF COMMON GTR MAXCOMMONVALUE, IT PNTS TO A STACK          00004500 T  0002:0000:0
DEFINE DUMPAVAIL=COMMON.[46:1]#; % COMMON=2 DUMPS AVAILABLE AREAS              00005000 T  0002:0000:0
DEFINE NONORMALCODE=COMMON.[45:1]#;% COMMON=4 DONT DUMP TYPE 1,7,13 CODE       00006000 T  0002:0000:0
DEFINE NOMCPCODE=COMMON.[44:1]#;% 8=DONT DUMP MCP CODE(TYPE=1,MIX=0)           00007000 T  0002:0000:0
DEFINE DUMPTABLES=COMMON.[43:1]#;% 16=DUMP SELECTED ARRAYS                     00008000 T  0002:0000:0
DEFINE DONTDUMPRT=COMMON.[42:1]#;%COMMON=32 STOPS DP OF PRT                    00009000 T  0002:0000:0
DEFINE DUMPOCTALONLY=COMMON.[41:1]#;% 64=UNCONDITIONAL OCTAL ONLY DUMP         00010000 T  0002:0000:0
DEFINE DUMPALPHAOCTAL=COMMON.[40:1]#;% 128=DUMP MEMORY IN ALPHA/OCTAL          00011000 T  0002:0000:0
DEFINE DUMPALPHAONLY=COMMON.[39:1]#;% 256=DUMP MEMORY IN ALPHA ONLY            00012000 T  0002:0000:0
DEFINE NODUMP=REAL(COMMON).[39:2]=3#;% 384=OMIT DUMP OF MEMORY                 00012500 T  0002:0000:0
DEFINE DUMPCESSPOOLONLY=COMMON.[38:1]#;% 512=DUMP ONLY THE "ARGH" ARRAY        00013000 T  0002:0000:0
DEFINE MAXCOMMONVALUE=1023#;%                                                  00016000 T  0002:0000:0
FILE IN DISK DISK SERIAL "MCP" "PRT"(2,30);                                    00016500 T  0002:0000:0
FILE IN MDUMP 2(2,533); % DISK=533, TAPE=513                                   00016510 T  0002:0003:3
FILE SPO 11(1,10);                                                             00016520 T  0002:0007:2
FILE P 4(3,17);                                                      %103-     00016530 P  0002:0010:1
PROCEDURE PRINTCOMMONVALUES;                                                   00017000 T  0002:0014:0
BEGIN                                                                          00017010 T  0002:0014:0
    INTEGER I;                                                                 00017020 T  0002:0014:0
                                                                    START OF SEGMENT **********    3
    SWITCH FORMAT COMINFO:=                                                    00017030 T  0003:0000:0
                                                                    START OF SEGMENT **********    4
    (///"THE COMMON VALUES AVAILABLE ARE:"/),                                  00017040 T  0004:0000:0
    ("COMMON=0 YIELDS A STANDARD MEMORY DUMP AND ANALYSIS,"),                   00017050 T  0004:0000:0
    ("COMMON=1 YIELDS A SPLASH DUMP WITH NO ANALYSIS,"),                       00017060 T  0004:0000:0
    ("COMMON=2 PRINTS AVAILABLE AREAS,"),                                      00017070 T  0004:0000:0
    ("COMMON=4 OMITS NORMAL STATE CODE SEGMENTS,"),                            00017080 T  0004:0000:0
    ("COMMON=8 OMITS MCP CODE SEGMENTS,"),                                     00017100 T  0004:0000:0
    ("COMMON=16 CAUSES CERTAIN USER DEFINED ARRAYS TO BE PRINTED,"),           00017110 T  0004:0000:0
    ("COMMON=32 OMITS THE PRINTING OF THE SORTED MCP PRT IDENTIFIERS."),       00017120 T  0004:0000:0
    ("COMMON=64 CAUSES MEMORY TO BE PRINTED ENTIRELY IN OCTAL."),              00017130 T  0004:0000:0
    ("COMMON=128 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA/OCTAL."),       00017140 T  0004:0000:0
    ("COMMON=256 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA."),             00017150 T  0004:0000:0
    ("COMMON=384 CAUSES NONE OF THE CONTENTS OF MEMORY TO BE PRINTED."),       00017160 T  0004:0000:0
    ("COMMON=512 DISPLAYS THE CONTENTS OF THE ARGH ARRAY."/),                  00017170 T  0004:0000:0
    ("FOR ADDITIONAL INFORMATION, CONSULT THE FIRST SEVERAL ",                 00017180 T  0004:0000:0
    "PAGES OF THE FILE:SYMBOL/DUMPANL,");                                      00017190 T  0004:0000:0
                                                                      4 IS  190 LONG, NEXT SEG    3
    FOR I:=0 STEP 1 UNTIL 13 DO WRITE(P,COMINFO[I]);                           00017200 T  0003:0000:0
```

```
END PRINTCOMMONVALUES;

BOOLEAN DONTPRINTLINKS,MYSTACKDUMPED;
FORMAT FINI(49("*")," END OF DUMP ANALYZE ",49("*"));


INTEGER PRTMAX,INTMAX,INFOMAX;
DEFINE PRTBASE=129#;%FIRST PRT CELL ALLOCATED BY ESPOL
DEFINE ACTUALPRTBASE=112#;% FIRST PRT CELL AS PER MCP DEFINE
DEFINE PRTSMAX=100#; % UPPER LIMIT OF PRTS ARRAY
REAL LEVEL,SUBLEVEL,VERSION;
INTEGER KLASS;% IDENTIFIER CLASS,AS DETERMINED BY ESPOL
INTEGER NAMESIZE,NAMSSIZE;
DEFINE NSNAME[NSNAME1]=NAME[NSNAME1],[8:10]#;
INTEGER INAMESIZE,INAMSSIZE;
DEFINE ISNAME[ISNAME1]=INAME[ISNAME1],[8:10]#;
ARRAY SEGZERO[0:29];
BOOLEAN  MCP;% FILL/PRT DETERMINED WHAT MODULES WERE INCLUDED
STREAM PROCEDURE MOVE(S,D,W); VALUE W;
BEGIN SI:=S; DI:=D; DS:= W WDS ; END MOVE;
PROCEDURE BUSTCOMMON;
BEGIN
    REAL MY,D;

    ARRAY T[0:0];
    FORMAT F("%",I5,",",X1);

    REAL STREAM PROCEDURE OCTDEC(C); VALUE C;
        BEGIN SI←LOC C; DI←LOC OCTDEC; DS←8DEC END;
    BOOLEAN STREAM PROCEDURE NOT5OCTADES(C); VALUE C;
        BEGIN SI←LOC C; 5(IF SC GTR "7" THEN BEGIN TALLY←1; JUMP OUT
        END ELSE SI←SI+1); NOT5OCTADES←TALLY END;
    STREAM PROCEDURE DECOCT(D,O1,O2); VALUE D;
        BEGIN SI←LOC D; DI←O1;DS←5OCT;DI←O2;DS←3OCT END;
%
    IF NOT5OCTADES(D←OCTDEC(COMMON))THEN D,[1:29]←0;
    DECOCT(D,MY,COMMON);
    IF MY=0 THEN MYSTACKADR←-1  ELSE
    BEGIN % CONVERT FIRST 5 DIGITS OF COM AS IF THEY WERE OCTAL DIGITS
        WRITE(T[*],F,MY);
        READ(T[*],/,MYSTACKADR);
    END;
END BUSTCOMMON;

PROCEDURE READARRAY(WORDSIZE,ANAME,BASE);
VALUE WORDSIZE,BASE;
INTEGER WORDSIZE,BASE;
ARRAY ANAME[*];
BEGIN
    INTEGER I,SEGS,N;

    ARRAY BUF[0:29];
    N:=WORDSIZE MOD 30;
    SEGS:=(WORDSIZE+29) DIV 30;
    SEGS:=SEGS-1;
    FOR I:=0 STEP 1 UNTIL SEGS DO
```

```
                00017210 T  0003:0007:2
          3 IS    10 LONG, NEXT SEG    2
                00018000 T  0002:0014:0
                00023000 T  0002:0014:0
        START OF SEGMENT **********    5
          5 IS    13 LONG, NEXT SEG    2
                00024000 T  0002:0014:0
                00025000 T  0002:0014:0
                00026000 T  0002:0014:0
                00027000 T  0002:0014:0
                00028000 T  0002:0014:0
                00029000 T  0002:0014:0
                00030000 T  0002:0014:0
                00031000 T  0002:0014:0
                00032000 T  0002:0014:0
                00033000 T  0002:0014:0
                00034000 T  0002:0014:0
                00035000 T  0002:0015:3
                00036000 T  0002:0015:3
                00037000 T  0002:0015:3
                00040100 T  0002:0018:0
                00040110 T  0002:0018:0
                00040120 T  0002:0018:0
        START OF SEGMENT **********    6
                00040130 T  0006:0000:0
                00040140 T  0006:0001:3
        START OF SEGMENT **********    7
          7 IS     7 LONG, NEXT SEG    6
                00040150 T  0006:0001:3
                00040160 T  0006:0001:3
                00040170 T  0006:0004:1
                00040180 T  0006:0004:1
                00040190 T  0006:0006:1
                00040200 T  0006:0009:2
                00040210 T  0006:0009:2
                00040220 T  0006:0011:3
                00040230 T  0006:0011:3
                00040240 T  0006:0017:2
                00040250 T  0006:0018:0
                00040260 T  0006:0020:1
                00040270 T  0006:0021:2
                00040280 T  0006:0029:2
                00040290 T  0006:0037:3
                00040300 T  0006:0037:3
          6 IS    42 LONG, NEXT SEG    2
                00041000 T  0002:0018:0
                00042000 T  0002:0018:0
                00043000 T  0002:0018:0
                00044000 T  0002:0018:0
                00045000 T  0002:0018:0
                00046000 T  0002:0018:0
        START OF SEGMENT **********    8
                00047000 T  0008:0000:0
                00048000 T  0008:0001:3
                00049000 T  0008:0003:2
                00050000 T  0008:0004:1
                00051000 T  0008:0006:0
```

```
    BEGIN
         READ(DISK,30,BUF[*]);
         MOVE(BUF,ANAME[BASE+30*I],IF I ≠SEGS OR N=0 THEN 30 ELSE N);
    END;
    IF PRT32 THEN BEGIN WRITE(P[PAGE]); WRITE(P[DBL],<"BASE=",
    I5," SIZE=",I5>,BASE,WORDSIZE); FOR I:=BASE STEP 1 UNTIL

    BASE+WORDSIZE -1 DO BEGIN MOVE(ANAME[ I],BUF[0],1);
    WRITE(P,1,BUF[*]) END END;
END OF READARRAY;

BUSTCOMMON;
COMMENT:TO GET OPTIONS,USE"(MIX)IN25" AND/OR COMMON CONTROL CARD;
IF COMMON THEN NAMESIZE:=NAMSSIZE:=INAMESIZE:=INAMSSIZE:=1 ELSE
BEGIN
READARRAY(30,SEGZERO[*],0);
CLOSE(DISK);
LEVEL ← SEGZERO[0],[42:6];                                          %101-
SUBLEVEL:="00"; % 2 DIGITS
VERSION:="D"; % 1 LETTER                                            %110-
COMMENT EACH TIME ANALYZER IS CHANGED INCREMENT ALPHABETICALLY;
MCP:=BOOLEAN(SEGZERO[1]);% COMPILE-TIME OPTIONS
NAMESIZE:=SEGZERO[2]+1;% # OF MCP PRT CELLS(ABOVE @200)
NAMSSIZE:=SEGZERO[3];% # OF WORDS THE MCP IDENTIFIERS OCCUPY
INAMESIZE:=SEGZERO[4]+1;% # OF INTRINSICS
INAMSSIZE:=SEGZERO[5];% # OF WORDS THE INTRINSIC IDENTIFIERS OCCUPY
END SEGZEROSTUFF;
INFOMAX:=4*PRTSMAX+3;
% % % % % % % % % % % % % INNER BLOCK BEGINS % % % % % % % % % % % % %
BEGIN
    LABEL EOPROG;

    LABEL RESTART,XX;
    INTEGER I;
    REAL R;
    REAL AUXTYPE;
    BOOLEAN SGLTOG;
    BOOLEAN NODUMPTOG;
    BOOLEAN NOMO,MULTI,REPEATING,RELOAD;
    ARRAY PRTNAME[0:4];%
    ARRAY TPNAME[0:2];
    DEFINE TDMFID=TPNAME[0]#,
           TDFID=TPNAME[1]#,
           FILETYPE=TPNAME[2]#;
    ARRAY PRTS[0:PRTSMAX];
    ARRAY INFO[0:INFOMAX+1];
    ARRAY NAME[PRTBASE:PRTBASE+NAMESIZE -1];
    ARRAY NAMS[0:NAMSSIZE-1];
    ARRAY INAME[0:INAMESIZE-1];
    ARRAY INAMS[0:INAMSSIZE-1];
    ARRAY XNAME[ACTUALPRTBASE:201];
    ARRAY XNAMS[0:18];
    DEFINE XSNAME[XSNAME1]=XNAME[XSNAME1],[8:10]#;
    INTEGER MIXMAX;%MIXMAX OBTAINED IN "GETPRTENTRIES"
    REAL PRTO; BOOLEAN PRTOK; % PRTO CONTAINS PRT[*,*] ENTRY,       %103-
    REAL VJOBNUM,VBED;
    DEFINE STAXMAX=80#;%
```

```
                                                                    00052000 T  0008:0007:2
                                                                    00053000 T  0008:0007:2
                                                                    00054000 T  0008:0011:2
                                                                    00055000 T  0008:0017:3
                                                                    00056000 T  0008:0019:3
                                                                    00057000 T  0009:0026:0
                                                                9 IS    7 LONG, NEXT SEG    8
                                                                    00058000 T  0008:0034:0
                                                                    00059000 T  0008:0041:2
                                                                    00060000 T  0008:0046:0
                                                                8 IS   50 LONG, NEXT SEG    2
                                                                    00061000 T  0002:0018:0
                                                                    00063000 T  0002:0019:3
                                                                    00064000 T  0002:0019:3
                                                                    00065000 T  0002:0022:1
                                                                    00066000 T  0002:0023:2
                                                                    00067000 T  0002:0024:1
                                                                    00068000 P  0002:0026:1
                                                                    00068100 T  0002:0028:0
                                                                    00068200 P  0002:0028:1
                                                                    00068300 T  0002:0029:3
                                                                    00069000 T  0002:0029:3
                                                                    00070000 T  0002:0030:1
                                                                    00071000 T  0002:0032:0
                                                                    00072000 T  0002:0033:2
                                                                    00073000 T  0002:0034:1
                                                                    00074000 T  0002:0035:3
                                                                    00078000 T  0002:0035:3
                                                                    00079000 T  0002:0037:2
                                                                    00080000 T  0002:0037:2
                                                                    00080400 T  0002:0037:2
                                                          START OF SEGMENT *********   10
                                                                    00080500 T  0010:0000:0
                                                                    00081000 T  0010:0000:0
                                                                    00082000 T  0010:0000:0
                                                                    00082050 T  0010:0000:0
                                                                    00082100 T  0010:0000:0
                                                                    00082110 T  0010:0000:0
                                                                    00082150 T  0010:0000:0
                                                                    00082200 T  0010:0000:0
                                                                    00082250 T  0010:0002:1
                                                                    00082300 T  0010:0005:2
                                                                    00082310 T  0010:0005:2
                                                                    00082330 T  0010:0005:2
                                                                    00083000 T  0010:0005:2
                                                                    00084000 T  0010:0007:3
                                                                    00085000 T  0010:0011:3
                                                                    00086000 T  0010:0016:1
                                                                    00087000 T  0010:0020:1
                                                                    00088000 T  0010:0024:1
                                                                    00089000 T  0010:0028:1
                                                                    00090000 T  0010:0031:2
                                                                    00091000 T  0010:0033:3
                                                                    00092000 T  0010:0033:3
                                                                    00092100 C. 0010:0033:3
                                                                    00093000 T  0010:0033:3
                                                                    00094000 T  0010:0033:3
```

```
    ARRAY STAX[0:STAXMAX-1]; INTEGER MAXSTK,BEDSTK;
COMMENT:THE DEFINES BELOW MUST CORRESPOND TO THOSE IN FILL/PRT;
    DEFINE BREAKOUT      = MCP.[47:1]#,
           CHECKLINK     = MCP.[46:1]#,
           DATACOM       = MCP.[45:1]#,
           DCLOG         = MCP.[44:1]#,
           DCSPO         = MCP.[43:1]#,
           DEBUGGING     = MCP.[42:1]#,
           DFX           = MCP.[41:1]#,
           DISKLOG       = MCP.[40:1]#,
           DUMPP         = MCP.[39:1]#,
           INQUIRY       = MCP.[38:1]#,
           SAVERESULTS   = MCP.[37:1]#,
           SHAREDISK     = MCP.[36:1]#,
           STATISTICS    = MCP.[35:1]#,
           AUXMEMM       = MCP.[34:1]#,
           RJE           = MCP.[33:1]#,
           B6500LOAD     = MCP.[32:1]#,
           SEPTICTANK    = MCP.[31:1]#,
           PACKETS       = MCP.[30:1]#,
           MONITORR      = MCP.[29:1]#,
           MAXOPT        = 19#; % UPDATE AS OPTIONS ARE ADDED
    ARRAY MEMORY[0:63,0:511];
     DEFINE OCTADE = (DS:=3 RESET; 3(IF SB THEN DS:=SET ELSE
                     DS:=RESET;SKIP SB))#;
    DEFINE TYPMAX=49#; %
   DEFINE ARROW = "←"#;
    DEFINE B = BOOLEAN#;
    DEFINE M=MEMORY#, %
       FF=[18:15]#,%
       CF=[33:15]#,%
      CTF=[18:33:15]#,%
      CTC=[33:33:15]#,%
          ROW=[33:6]#, %
          COL=[39:9]#; %
    DEFINE DEFINEDMIXMAX=9#;%MIXMAX NOW OBTAINED FM PRT MOTHER
    DEFINE STATE              = PRTS[00]#,
           NSTATE             = PRTS[01]#,
           LSTATE             = PRTS[02]#,
           ESPBIT             = PRTS[03]#,
           AVAIL              = PRTS[04]#,
           MSTART             = PRTS[05]#,
           MEND               = PRTS[06]#,
           TOGLF              = PRTS[07]#,
           BED                = PRTS[08]#,
           PRT                = PRTS[12]#,
           JAR                = PRTS[13]#,
           INTRNSC            = PRTS[14]#,
           SHEET              = PRTS[15]#,
           JOBNUM             = PRTS[16]#,
           PRYOR              = PRTS[17]#,
           NFO                = PRTS[18]#,
           ISTACK             = PRTS[19]#,
           PROCTIME           = PRTS[20]#,
           IOTIME             = PRTS[21]#,
           CHANNEL            = PRTS[22]#,
           FINALQUE           = PRTS[23]#,
```

```
                                                                                00095000 T  0010:0033:3
                                                                                00096000 T  0010:0037:3
                                                                                00097000 T  0010:0037:3
                                                                                00098000 T  0010:0037:3
                                                                                00099000 T  0010:0037:3
                                                                                00100000 T  0010:0037:3
                                                                                00101000 T  0010:0037:3
                                                                                00102000 T  0010:0037:3
                                                                                00103000 T  0010:0037:3
                                                                                00104000 T  0010:0037:3
                                                                                00105000 T  0010:0037:3
                                                                                00106000 T  0010:0037:3
                                                                                00107000 T  0010:0037:3
                                                                                00108000 T  0010:0037:3
                                                                                00109000 T  0010:0037:3
                                                                                00110000 T  0010:0037:3
                                                                                00110100 T  0010:0037:3
                                                                                00110200 T  0010:0037:3
                                                                                00110300 T  0010:0037:3
                                                                                00110400 T  0010:0037:3
                                                                                00110500 T  0010:0037:3
                                                                                00111000 T  0010:0037:3
                                                                                00112000 T  0010:0037:3
                                                                    %103-       00112500 C  0010:0040:1
                                                                    %103-       00112600 C  0010:0040:1
                                                                    %109-       00113000 P  0010:0040:1
                                                                    %103-       00113150 C  0010:0040:1
                                                                    %103-       00113160 C  0010:0040:1
                                                                                00114000 T  0010:0040:1
                                                                                00115000 T  0010:0040:1
                                                                                00116000 T  0010:0040:1
                                                                                00117000 T  0010:0040:1
                                                                                00117100 T  0010:0040:1
                                                                                00118000 T  0010:0040:1
                                                                                00119000 T  0010:0040:1
                                                                                00120000 T  0010:0040:1
                                                                                00121000 T  0010:0040:1
                                                                                00122000 T  0010:0040:1
                                                                                00123000 T  0010:0040:1
                                                                                00124000 T  0010:0040:1
                                                                                00125000 T  0010:0040:1
                                                                                00126000 T  0010:0040:1
                                                                                00127000 T  0010:0040:1
                                                                                00128000 T  0010:0040:1
                                                                                00129000 T  0010:0040:1
                                                                                00130000 T  0010:0040:1
                                                                                00131000 T  0010:0040:1
                                                                                00132000 T  0010:0040:1
                                                                                00133000 T  0010:0040:1
                                                                                00134000 T  0010:0040:1
                                                                                00135000 T  0010:0040:1
                                                                                00136000 T  0010:0040:1
                                                                                00137000 T  0010:0040:1
                                                                                00138000 T  0010:0040:1
                                                                                00139000 T  0010:0040:1
                                                                                00140000 T  0010:0040:1
                                                                                00141000 T  0010:0040:1
```

```
LOCATQUE          = PRTS[24]#,
IOQUEAVAIL        = PRTS[25]#,
IOQUE             = PRTS[26]#,
UNIT              = PRTS[27]#,
TINU              = PRTS[28]#,
WAITQUE           = PRTS[29]#,
NEXTWAIT          = PRTS[30]#,
FIRSTWAIT         = PRTS[31]#,
LABELTABLE        = PRTS[32]#,
MULTITABLE        = PRTS[33]#,
RDCTABLE          = PRTS[34]#,
OPTION            = PRTS[35]#,
MESSAGEHOLDER     = PRTS[36]#,
PRNTABLE          = PRTS[37]#,
INITIALIZE        = PRTS[38]#,
P1MIX             = PRTS[39]#,
P2MIX             = PRTS[40]#,
NOTHINGTODO       = PRTS[41]#,
STACKOVERFLOW     = PRTS[42]#,
RETURN            = PRTS[43]#,
DIRECTORYBUILDER  = PRTS[44]#,
DCQPTSTACK        = PRTS[45]#,
SAVERESULT         = PRTS[46]#,
AUXDATA            = PRTS[47]#,
AUXCODE            = PRTS[48]#,
EUIO               = PRTS[49]#,
PEOIO              = PRTS[50]#,
AVTABLE            = PRTS[51]#,
REPLY              = PRTS[52]#,
TRANSACTION        = PRTS[53]#,
DALOC              = PRTS[54]#,
MEMASK             = PRTS[55]#,
CTABLE             = PRTS[56]#,
FS                 = PRTS[57]#,
DBARRAY            = PRTS[58]#,
ATTACHED           = PRTS[59]#,
STATION            = PRTS[60]#,
DCQARA             = PRTS[61]#,
USERSTA            = PRTS[62]#,
TUSTABYMIX         = PRTS[63]#,
QTIMES             = PRTS[64]#,
EUQ                = PRTS[65]#,
CIDROW             = PRTS[66]#,
ARGH               = PRTS[67]#,
YECH               = PRTS[68]#,
DIRECTORYFREE      = PRTS[69]#,
SYSNQ              = PRTS[70]#,
STATIONMESSAGEHOLDER
                   = PRTS[71]#,
LOOKQ              = PRTS[72]#,
PINGQ              = PRTS[73]#,
DC19Q              = PRTS[74]#,
ILL                = PRTS[75]#,
RJEWAITQ           = PRTS[76]#,
NOPROCESSTOG       = PRTS[77]#,
MIXMASK            = PRTS[78]#,
INFOMASK1          = PRTS[79]#,
INFOMASK2          = PRTS[80]#,
```

```
00142000 T  0010:0040:1
00143000 T  0010:0040:1
00144000 T  0010:0040:1
00145000 T  0010:0040:1
00146000 T  0010:0040:1
00147000 T  0010:0040:1
00148000 T  0010:0040:1
00149000 T  0010:0040:1
00150000 T  0010:0040:1
00151000 T  0010:0040:1
00152000 T  0010:0040:1
00153000 T  0010:0040:1
00154000 T  0010:0040:1
00155000 T  0010:0040:1
00156000 T  0010:0040:1
00157000 T  0010:0040:1
00158000 T  0010:0040:1
00159000 T  0010:0040:1
00160000 T  0010:0040:1
00161000 T  0010:0040:1
00162000 T  0010:0040:1
00163000 T  0010:0040:1
00164100 T  0010:0040:1
00164110 T  0010:0040:1
00164120 T  0010:0040:1
00164130 T  0010:0040:1
00164140 T  0010:0040:1
00164150 T  0010:0040:1
00164160 T  0010:0040:1
00164170 T  0010:0040:1
00164180 T  0010:0040:1
00164190 T  0010:0040:1
00164200 T  0010:0040:1
00164210 T  0010:0040:1
00164220 T  0010:0040:1
00164230 T  0010:0040:1
00164240 T  0010:0040:1
00164250 T  0010:0040:1
00164260 T  0010:0040:1
00164270 T  0010:0040:1
00164275 T  0010:0040:1
00164280 T  0010:0040:1
00164285 T  0010:0040:1
00164290 T  0010:0040:1
00164295 T  0010:0040:1
00164300 T  0010:0040:1
00164305 T  0010:0040:1
00164310 T  0010:0040:1
00164311 T  0010:0040:1
00164315 T  0010:0040:1
00164320 T  0010:0040:1
00164325 T  0010:0040:1
00164330 T  0010:0040:1
00164335 T  0010:0040:1
00164340 T  0010:0040:1
00164345 T  0010:0040:1
00164350 T  0010:0040:1
00164355 T  0010:0040:1
```

```
            CCMASK1          = PRTS[81]#,
            CCMASK2          = PRTS[82]#,
            CORE             = PRTS[83]#,
            DISKOUNT         = PRTS[84]#,
            EUW              = PRTS[85]#,
            PUNTER           = PRTS[86]#,
            LQUE             = PRTS[87]#,
            LQAVAIL          = PRTS[88]#,
            TAR              = PRTS[89]#,
            IOQUESLOTS       = PRTS[90]#,
            DUMMY            = DUMMY#;%
PROCEDURE FILLINFO;
FILL INFO[*] WITH %
"SLATE   ",0,0,26,                          % 00

"NSLATE  ",0,0,22,                          % 01
"LSLATE  ",0,0,22,                          % 02
"ESPBIT  ",0,0,10,                          % 03
"AVAIL   ",0,0,23,                          % 04
"MSTART  ",0,0,23,                          % 05
"MEND    ",0,0,23,                          % 06
"TOGLE   ",0,0,22,                          % 07
"BED     ",0,0,26,                          % 08
0,0,0,0,                                    % 09
0,0,0,0,                                    % 10
0,0,0,0,                                    % 11
"PRT     ",0,0,26,                          % 12
"JAR     ",0,0,26,                          % 13
"INTRNSC ",0,0,26,                          % 14
"SHEET   ",0,0,26,                          % 15
"JOBNUM  ",0,0,22,                          % 16
"PRYOR   ",0,0,26,                          % 17
"NFO     ",0,0,26,                          % 18
"ISTACK  ",0,0,26,                          % 19
"PROCTIME",0,0,26,                          % 20
"IOTIME  ",0,0,26,                          % 21
"CHANNEL ",0,0,26,                          % 22
"FINALQUE",0,0,26,                          % 23
"LOCATQUE",0,0,26,                          % 24
"IOQUEAVA","IL      ",0,22,                 % 25
"IOQUE   ",0,0,26,                          % 26
"UNIT    ",0,0,26,                          % 27
"TINU    ",0,0,26,                          % 28
"WAITQUE ",0,0,26,                          % 29
"NEXTWAIT",0,0,22,                          % 30
"FIRSTWAI","T       ",0,22,                 % 31
"LABELTAB","LE      ",0,26,                 % 32
"MULTITAB","LE      ",0,26,                 % 33
"RDCTABLE",0,0,26,                          % 34
"OPTION  ",0,0,22,                          % 35
"MESSAGEH","OLDER   ",0,22,                 % 36
"PRNTABLE",0,0,26,                          % 37
"INITIALI","ZE      ",0,10,                 % 38
"P1MIX   ",0,0,22,                          % 39
"P2MIX   ",0,0,22,                          % 40
"NOTHINGT","ODO     ",0,32,                 % 41
"STACKOVE","RFLOW   ",0,32,                 % 42
```

```
                00164360 T  0010:0040:1
                00164365 T  0010:0040:1
                00164370 T  0010:0040:1
                00164380 T  0010:0040:1
                00164390 T  0010:0040:1
                00164400 T  0010:0040:1
                00164410 T  0010:0040:1
                00164420 T  0010:0040:1
                00164430 T  0010:0040:1
                00164440 T  0010:0040:1
                00164999 T  0010:0040:1
                00165000 T  0010:0040:1
                00166000 T  0010:0040:1
                00167000 T  0010:0041:3
START OF SEGMENT **********   11
                00168000 T  0010:0042:1
                00169000 T  0010:0042:1
                00170000 T  0010:0042:1
                00171000 T  0010:0042:1
                00172000 T  0010:0042:1
                00173000 T  0010:0042:1
                00174000 T  0010:0042:1
                00175000 T  0010:0042:1
                00176000 T  0010:0042:1
                00177000 T  0010:0042:1
                00178000 T  0010:0042:1
                00179000 T  0010:0042:1
                00180000 T  0010:0042:1
                00181000 T  0010:0042:1
                00182000 T  0010:0042:1
                00183000 T  0010:0042:1
                00184000 T  0010:0042:1
                00185000 T  0010:0042:1
                00186000 T  0010:0042:1
                00187000 T  0010:0042:1
                00188000 T  0010:0042:1
                00189000 T  0010:0042:1
                00190000 T  0010:0042:1
                00191000 T  0010:0042:1
                00192000 T  0010:0042:1
                00193000 T  0010:0042:1
                00194000 T  0010:0042:1
                00195000 T  0010:0042:1
                00196000 T  0010:0042:1
                00197000 T  0010:0042:1
                00198000 T  0010:0042:1
                00199000 T  0010:0042:1
                00200000 T  0010:0042:1
                00201000 T  0010:0042:1
                00202000 T  0010:0042:1
                00203000 T  0010:0042:1
                00204000 T  0010:0042:1
                00205000 T  0010:0042:1
                00206000 T  0010:0042:1
                00207000 T  0010:0042:1
                00208000 T  0010:0042:1
                00209000 T  0010:0042:1
```

```
"RETURN  ",0,0,32,                      % 43
"DIRECTOR","YBUILDER",0,10,             % 44
"DCQPTSTA","CK      ",0,22,             % 45
"SAVERESU","LT      ",0,26,               % 46
"AUXDATA ",0,0,26,                        % 47
"AUXCODE ",0,0,26,                        % 48
"EUIO    ",0,0,26,                        % 49
"PEUIO   ",0,0,26,                        % 50
"AVTABLE ",0,0,26,                        % 51
"REPLY   ",0,0,26,                        % 52
"TRANSACT","ION     ",0,26,               % 53
"DALOC   ",0,0,26,                        % 54
"MEMASK  ",0,0,26,                        % 55
"CTABLE  ",0,0,26,                        % 56
"FS      ",0,0,26,                        % 57
"DBARRAY ",0,0,26,                        % 58
"ATTACHED",0,0,26,                        % 59
"STATION ",0,0,26,                        % 60
"DCQARA  ",0,0,26,                        % 61
"USERSTA ",0,0,26,                        % 62
"TUSTABYM","IX      ",0,26,               % 63
"QTIMES  ",0,0,26,                        % 64
"EUQ     ",0,0,26,                        % 65
"CIDROW  ",0,0,26,                        % 66
"ARGH    ",0,0,26,                        % 67
"YECH    ",0,0,22,                        % 68
"DIRECTOR","YFREE   ",0,22,               % 69
"SYSNO   ",0,0,22,                        % 70
"STATIONM","ESSAGEHO","LDER    ",22,      % 71
"LOOKQ   ",0,0,22,                        % 72
"PINGO   ",0,0,22,                        % 73
"DC19Q   ",0,0,22,                        % 74
"ILL     ",0,0,22,                        % 74
"RJEWAITQ",0,0,22,                        % 76
"NOPROCES","STOG    ",0,22,               % 77
"MIXMASK ",0,0,22,                        % 78
"INFOMASK","1       ",0,22,               % 79
"INFOMASK","2       ",0,22,               % 80
"CCMASK1 ",0,0,22,                        % 81
"CCMASK2 ",0,0,22,                        % 82
"CORE    ",0,0,22,                        % 83
"DISKOUNT",0,0,23,                        % 84
"EUW     ",0,0,22,                        % 85
"PUNTER  ",0,0,26,                        % 86
"LQUE    ",0,0,26,                        % 87
"LQAVAIL ",0,0,22,                        % 88
"TAR     ",0,0,26,                        % 89
"IOQUESLO","TS      ",0,22,               % 90
0;%

COMMENT*****************************************************************
    FOR EACH IDENTIFIER DEFINED IN THE ARRAY "PRTS", THERE MUST BE
    A CORRESPONDING 4 WORD ENTRY IN "INFO".  THE FIRST THREE WORDS
    ARE RESERVED FOR THE IDENTIFIER TEXT.  THE FOURTH MUST CONTAIN
    THE CLASS OF THE IDENTIFIER AS DETERMINED BY THE ESPOL COMPILER.
    THIS CLASS APPEARS IN THE FIRST FOUR COLUMNS OF THE STUFF CARD
    FOR THIS IDENTIFIER.  IF THE CLASS IS NOT KNOWN, A ZERO MAY BE
```

```
00210000 T  0010:0042:1
00211000 T  0010:0042:1
00212000 T  0010:0042:1
00212100 T  0010:0042:1
00212200 T  0010:0042:1
00212300 T  0010:0042:1
00212400 T  0010:0042:1
00212450 T  0010:0042:1
00212500 T  0010:0042:1
00212550 T  0010:0042:1
00212600 T  0010:0042:1
00212625 T  0010:0042:1
00212650 T  0010:0042:1
00212675 T  0010:0042:1
00212700 T  0010:0042:1
00212725 T  0010:0042:1
00212750 T  0010:0042:1
00212775 T  0010:0042:1
00212800 T  0010:0042:1
00212825 T  0010:0042:1
00212850 T  0010:0042:1
00212875 T  0010:0042:1
00212880 T  0010:0042:1
00212885 T  0010:0042:1
00212890 T  0010:0042:1
00212895 T  0010:0042:1
00212900 T  0010:0042:1
00212905 T  0010:0042:1
00212910 T  0010:0042:1
00212915 T  0010:0042:1
00212920 T  0010:0042:1
00212925 T  0010:0042:1
00212930 T  0010:0042:1
00212935 T  0010:0042:1
00212940 T  0010:0042:1
00212945 T  0010:0042:1
00212950 T  0010:0042:1
00212955 T  0010:0042:1
00212960 T  0010:0042:1
00212965 T  0010:0042:1
00212970 T  0010:0042:1
00212980 T  0010:0042:1
00212990 T  0010:0042:1
00212992 T  0010:0042:1
00212994 T  0010:0042:1
00212996 T  0010:0042:1
00212998 T  0010:0042:1
00212999 T  0010:0042:1
00213000 T  0010:0042:1
11 IS  365 LONG, NEXT SEG   10
00214000 T  0010:0043:2
00215000 T  0010:0043:2
00216000 T  0010:0043:2
00217000 T  0010:0043:2
00218000 T  0010:0043:2
00219000 T  0010:0043:2
00220000 T  0010:0043:2
```

```
USED IN THE "INFO" ENTRY FOR IT. HOWEVER USING A CLASS OF ZERO WILL
INCREASE THE TIME NEEDED TO LOCATE THE PRT ADDRESS OF AN
IDENTIFIER.  NOTE THAT ALTHOUGH ARRAY "PRTS" MAY CONTAIN GAPS
"INFO" MUST CONTAIN A DUMMY ENTRY COMPRISED OF ALL ZEROES(SEE,FOR
INSTANCE, PRTS[9], PRTS[10], AND PRTS[11]).
FOR CONVENIENCE, THE CLASS NUMBERS THAT THE ESPOL COMPILER MAY
ASSIGN AND WHICH MAY APPEAR ON A STUFF CARD ARE LISTED BELOW:
     PROCID              =10#,
     STRPROCID           =12#,
     BOOSTRPROCID        =13#,
     REALSTRPROCID       =14#,
     INTSTRPROCID        =15#,
     BOOPROCID           =17#,
     REALPROCID          =18#,
     INTPROCID           =19#,
     BOOID               =21#,
     REALID              =22#,
     INTID               =23#,
     BOOARRAYID          =25#,
     REALARRAYID         =26#,
     INTARRAYID          =27#,
     NAMEID              =30#,
     INTNAMEID           =31#,
     LABELID             =32#.
*********************************************************************;
PROCEDURE SETUPXNAMEANDXNAMS;
BEGIN
     STREAM PROCEDURE FILLXNAMS(XNAMS);

     BEGIN
          DI:=XNAMS;
          DS:= 8LIT"NT1     ";
          DS:= 8LIT"NT2     ";
          DS:= 8LIT"NT3     ";
          DS:= 8LIT"NT4     ";
          DS:= 8LIT"NT5     ";
          DS:= 8LIT"NT6     ";
          DS:= 8LIT"NT7     ";
          DS:= 8LIT"DATE    ";
          DS:= 8LIT"CLOCK   ";
          DS:= 8LIT"XCLOCK  ";
          DS:= 8LIT"READY   ";
          DS:= 8LIT"--------";
          DS:= 8LIT"KLUMP   ";
          DS:=16LIT"FIRSTDECK       ";
          DS:= 8LIT"LASTDECK";
          DS:= 8LIT"DIRDSK  ";
          DS:= 8LIT"MEMORY  ";
          DS:= 8LIT"RRRMECH ";
     END OF FILLXNAMS;
     ARRAY T[ACTUALPRTBASE:213];
     INTEGER I,J;
     FILL T[*] WITH % XNAMES
          123,1,00,

          120,1,01,
          119,1,02,
```

```
00221000 T  0010:0043:2
00222000 T  0010:0043:2
00223000 T  0010:0043:2
00224000 T  0010:0043:2
00225000 T  0010:0043:2
00226000 T  0010:0043:2
00227000 T  0010:0043:2
00228000 T  0010:0043:2
00229000 T  0010:0043:2
00230000 T  0010:0043:2
00231000 T  0010:0043:2
00232000 T  0010:0043:2
00233000 T  0010:0043:2
00234000 T  0010:0043:2
00235000 T  0010:0043:2
00236000 T  0010:0043:2
00237000 T  0010:0043:2
00238000 T  0010:0043:2
00239000 T  0010:0043:2
00240000 T  0010:0043:2
00241000 T  0010:0043:2
00242000 T  0010:0043:2
00243000 T  0010:0043:2
00244000 T  0010:0043:2
00245000 T  0010:0043:2
00246000 T  0010:0043:2
00247000 T  0010:0043:2
00248000 T  0010:0043:2
START OF SEGMENT **********   12
00249000 T  0012:0000:0
00250000 T  0012:0000:0
00251000 T  0012:0000:0
00252000 T  0012:0001:3
00253000 T  0012:0002:1
00254000 T  0012:0004:0
00255000 T  0012:0005:2
00256000 T  0012:0006:1
00257000 T  0012:0007:3
00258000 T  0012:0009:2
00259000 T  0012:0010:0
00260000 T  0012:0011:3
00261000 T  0012:0012:1
00262000 T  0012:0014:0
00263000 T  0012:0015:2
00264000 T  0012:0016:1
00265000 T  0012:0018:1
00266000 T  0012:0020:0
00267000 T  0012:0021:2
00268000 T  0012:0022:1
00269000 T  0012:0023:3
00270000 T  0012:0024:0
00271000 T  0012:0025:3
00272000 T  0012:0025:3
00273000 T  0012:0026:0
START OF SEGMENT **********   13
00274000 T  0012:0027:3
00275000 T  0012:0027:3
```

```
                127,1,03,
                125,1,04,
                124,1,05,
                126,1,06,
                128,1,07,
                112,1,08,
                113,1,09,
                114,1,10,
                115,1,11,
                116,1,12,
                117,2,13,
                118,1,15,
                122,1,16,
                129,1,17,
                121,1,18;%

    FILLXNAMS(XNAMS);
    J:=ACTUALPRTBASE-1;
    FOR I:=ACTUALPRTBASE STEP 3 UNTIL 211 DO
    XNAME[J:=J+1]:=0&T[I][8:38:10]&T[I+1][18:33:15]&T[I+2][33:33:15];
END SETTING UP XNAMEANDXNAMS;

REAL LINKTYPE;
ARRAY AREATYPE[0:TYPMAX+1], ATP[0:(TYPMAX+1)*2];                    %103-
BOOLEAN ARRAYSFILLED;                                               %103-
PROCEDURE FILLAREATYPE;
COMMENT***AREATYPE[LINKTYPE], WHERE LINKTYPE IS THE TYPE OF MEMORY
LINK TO BE PRINTED, CONTAINS A CODE WHICH DETERMINES THE FORMAT FOR
PRINTING THE AREA : 1=OCTAL, 2=ALPHA/OCTAL, 3=ALPHA ONLY.
DUMPMEMORYANDNOTESTACKS DETERMINES LINKTYPE FROM [3:6] OF A PRIMARY
LINK.  PRINTCORE THEN OBTAINS AREATYPE[LINKTYPE] AND PRINTS THE AREA
IN THE APPROPRIATE FORMAT.  ELEMENTS 0 THRU TYPMAX CORRESPOND TO
POSSIBLE LINK TYPES.  ELEMENT TYPMAX+1 IS USED TO CONTAIN  THE DEFAULT
FORMAT CODE USED IN PRINTING AN AREA WHICH DOES NOT BEGIN WITH A LINK
OR CONTAINS A CLOBBERED LINK.  THE CONTENTS OF AREATYPE ARE NOT
USED IF DUMPOCTALONLY, DUMPALPHOCTAL OR DUMPALPHAONLY ARE SET OR
IF AN AVAILABLE AREA IS BEING PRINTED WHOSE CORRESPONDING FORMAT
CODE IS NOT NEGATIVE.  FOR EXAMPLE, A NEGATIVE CODE SUCH AS -2 CAUSES
BOTH IN-USE AND AVAILABLE AREAS TO BE FORMATTED AS ALPHA/OCTAL.  A
CODE OF +2 WOULD CAUSE ONLY IN-USE AREAS TO BE PRINTED IN ALPHA/OCTAL-
AVAILABLE AREAS, IF PRINTED, WOULD BE PRINTED IN OCTAL ONLY.****;
IF NOT ARRAYSFILLED THEN BEGIN                                      %103-
FILL  AREATYPE[*] WITH % 1=OCTAL, 2=ALPHA/OCTAL,3=ALPHA
2,% UNKNOWN(0)

1,% CODE(1)
2,% DATA(2)
2,% IOBUF(3)
1,% ALGFIB(4)
2,% INQBUF(5)
2,% COBFIB(6)
1,% INTSEG(7)
2,% HEADER(8)                                                       %109-
1,% (9)
2,% LBL EQN(10)                                                     %109-
1,% (11)
2,% STACK(12)
```

```
                                                    00276000 T  0012:0027:3
                                                    00277000 T  0012:0027:3
                                                    00278000 T  0012:0027:3
                                                    00279000 T  0012:0027:3
                                                    00280000 T  0012:0027:3
                                                    00281000 T  0012:0027:3
                                                    00282000 T  0012:0027:3
                                                    00283000 T  0012:0027:3
                                                    00284000 T  0012:0027:3
                                                    00285000 T  0012:0027:3
                                                    00286000 T  0012:0027:3
                                                    00287000 T  0012:0027:3
                                                    00288000 T  0012:0027:3
                                                    00289000 T  0012:0027:3
                                                    00290000 T  0012:0027:3
                                          13 IS   54 LONG, NEXT SEG   12
                                                    00291000 T  0012:0027:3
                                                    00292000 T  0012:0028:1
                                                    00293000 T  0012:0029:3
                                                    00294000 T  0012:0031:2
                                                    00295000 T  0012:0041:2
                                          12 IS   45 LONG, NEXT SEG   10
                                                    00295100 T  0010:0043:2
                                                    00295110 P  0010:0043:2
                                                    00295115 C  0010:0051:3
                                                    00295120 T  0010:0051:3
                                                    00295130 T  0010:0051:3
                                                    00295140 T  0010:0051:3
                                                    00295150 T  0010:0051:3
                                                    00295160 T  0010:0051:3
                                                    00295170 T  0010:0051:3
                                                    00295180 T  0010:0051:3
                                                    00295190 T  0010:0051:3
                                                    00295200 T  0010:0051:3
                                                    00295210 T  0010:0051:3
                                                    00295220 T  0010:0051:3
                                                    00295230 T  0010:0051:3
                                                    00295240 T  0010:0051:3
                                                    00295250 T  0010:0051:3
                                                    00295260 T  0010:0051:3
                                                    00295270 T  0010:0051:3
                                                    00295275 C  0010:0051:3
                                                    00295280 T  0010:0053:2
                                                    00295290 T  0010:0053:3
                                        START OF SEGMENT **********   14
                                                    00295300 T  0010:0054:1
                                                    00295310 T  0010:0054:1
                                                    00295320 T  0010:0054:1
                                                    00295330 T  0010:0054:1
                                                    00295340 T  0010:0054:1
                                                    00295350 T  0010:0054:1
                                                    00295360 T  0010:0054:1
                                                    00295370 P  0010:0054:1
                                                    00295380 T  0010:0054:1
                                                    00295390 P  0010:0054:1
                                                    00295400 T  0010:0054:1
                                                    00295410 T  0010:0054:1
```

```
1,% (13)
1,% (14)
1,% (15)
1,% (16)
2,% SPOUT MSG(17)
2,% UVROW(18)
2,% JAROW(19)
2,% CIDROW(20)
1,% (21)
1,% (22)
2,% (23)
2,% (24)
1,% DALOCROW(25)
2,% SHEFT(26)
1,% (27)
2,% KEY;NBUFFER(28)
1,% FSROW(29)
2,% (30)
1,% AVTABLE(31)
2,% TRACE TABLE(32)
1,% SEGDICT(33)
1,% STACK-PRT(34)
2,% MCP-TABLES(35)
1,% IRSTACK(36)
2,% FPB(37)
2,% CC(38)
2,% LABEL(39)
2,% MDUMP(40)
2,% ESPnISK(41)
2,% LOG(42)
2,% (43)
2,% (44)
2,% (45)
2,% (46)
2,% (47)
2,% (48)
2;%

FILL ATP[*] WITH % NAMES OF MEMORY AREA TYPES.
    "UNKNOWN←","          ", % TYPE 0

    "CODE←   ","          ", % TYPE 1
    "DATA←   ","          ", % TYPE 2
    "I/O BUFF","ER←       ", % TYPE 3
    "ALGOL FI","B←        ", % TYPE 4
    "INQUIRY ","BUFFER←   ", % TYPE 5
    "COBOL FI","B←        ", % TYPE 6
    "REENTRAN","T INTRNS",   % TYPE 7
    "DISK HEA","DER←      ", % TYPE 8
    "MAINTENA","NCE←      ", % TYPE 9
    "LBL.EQN.","  ENTRY←   ", % TYPE 10
    "SEGMENT ","ZERO←     ", % TYPE 11
    "STACK←  ","          ", % TYPE 12
    "INTRINSI","C←        ", % TYPE 13
    "SCRATCH ","DIRECTRY",   % TYPE 14
    "OPSET←  ","          ", % TYPE 15
    "DIRECTOR","Y TOP←    ", % TYPE 16
```

```
                00295420 T  0010:0054:1
                00295430 T  0010:0054:1
                00295440 T  0010:0054:1
                00295450 T  0010:0054:1
%109-           00295460 P  0010:0054:1
%109-           00295470 P  0010:0054:1
%109-           00295480 P  0010:0054:1
                00295490 T  0010:0054:1
                00295500 T  0010:0054:1
                00295510 T  0010:0054:1
%109-           00295511 C  0010:0054:1
%109-           00295512 C  0010:0054:1
%109-           00295513 C  0010:0054:1
%109-           00295514 C  0010:0054:1
%109-           00295515 C  0010:0054:1
%109-           00295516 C  0010:0054:1
%109-           00295517 C  0010:0054:1
%109-           00295518 C  0010:0054:1
%109-           00295519 C  0010:0054:1
%109-           00295520 P  0010:0054:1
%109-           00295521 C  0010:0054:1
%109-           00295522 C  0010:0054:1
%109-           00295523 C  0010:0054:1
%109-           00295524 C  0010:0054:1
%109-           00295525 C  0010:0054:1
%109-           00295530 C  0010:0054:1
%109-           00295535 C  0010:0054:1
%109-           00295540 C  0010:0054:1
%109-           00295545 C  0010:0054:1
%109-           00295550 C  0010:0054:1
%109-           00295555 C  0010:0054:1
%109-           00295560 C  0010:0054:1
%109-           00295565 C  0010:0054:1
%109-           00295570 C  0010:0054:1
%109-           00295575 C  0010:0054:1
%109-           00295580 C  0010:0054:1
%109-           00295585 C  0010:0054:1
      14 IS   50 LONG, NEXT SEG   10
%103-           00295700 C  0010:0054:1
%103-           00295705 C  0010:0055:2
    START OF SEGMENT **********   15
%103-           00295710 C  0010:0056:1
%103-           00295715 C  0010:0056:1
%103-           00295720 C  0010:0056:1
%103-           00295725 C  0010:0056:1
%103-           00295730 C  0010:0056:1
%103-           00295735 C  0010:0056:1
%103-           00295740 C  0010:0056:1
%103-           00295745 C  0010:0056:1
%103-           00295750 C  0010:0056:1
%103-           00295755 C  0010:0056:1
%103-           00295760 C  0010:0056:1
%103-           00295765 C  0010:0056:1
%103-           00295770 C  0010:0056:1
%103-           00295775 C  0010:0056:1
%103-           00295780 C  0010:0056:1
%103-           00295785 C  0010:0056:1
```

```
"SPOUT ME","SSAGE←  ", % TYPE 17
"LABELS← ","        ", % TYPE 18
"JAR ROW←","        ", % TYPE 19
"CID ROW←","        ", % TYPE 20
"INQUIRY ","INPUT←  ", % TYPE 21
"INTRINSI","C ARRAY←", % TYPE 22
"RJF INPU","T←      ", % TYPE 23
"DC QUFUE","←       ", % TYPE 24
"DALOC RO","W←      ", % TYPE 25
"SHEET EN","TRY←    ", % TYPE 26
"STATION ","ROW←    ", % TYPE 27
"KEYIN BU","FFER←   ", % TYPE 28
"FS ROW← ","        ", % TYPE 29
"DC 19 QU","EUE←    ", % TYPE 30
"AVAIL DI","SK TBL← ", % TYPE 31
"TRACE TA","BLF←    ", % TYPE 32
"SEGMENT ","DICT,←  ", % TYPE 33
"STACK-PR","T←      ", % TYPE 34
"MCP TABL","ES←     ", % TYPE 35
"I R STAC","K←      ", % TYPE 36
"FILE PAR","AM BLOCK", % TYPE 37
"CONTROL ","CARD←   ", % TYPE 38
"LABEL RE","CORD←   ", % TYPE 39
"MDUMP BU","FFER←   ", % TYPE 40
"ESP DISK","←       ", % TYPE 41
"LOG INFO","←       ", % TYPE 42
"TYPE 43←","        ", % TYPE 43
"TYPE 44←","        ", % TYPE 44
"TYPE 45←","        ", % TYPE 45
"TYPE 46←","        ", % TYPE 46
"TYPE 47←","        ", % TYPE 47
"TYPE 48←","        ", % TYPE 48
"@@@@@@@@","@@@@@@@@"; % END OF TABLE.

ARRAYSFILLED := TRUE;
END;
    DEFINE NEXTPAGE=WRITE(PIPAGE])#;
    FORMAT STARS(20("******")),

           STARZ(*("*"));

FORMAT MCPHDR("DCMCP,XVI.",A1,".",A2," COMPILE TIME OPTIONS:");


SWITCH FORMAT MCPOPT:=(        "BREAKOUT")
                      ,(    "CHECKLINK")
                      ,(    "DATACOM")
                      ,(    "DCLOG")
                      ,(    "DCSPO")
                      ,(    "DEBUGGING")
                      ,(    "DFX")
                      ,(    "DISKLOG")
                      ,(    "DUMP")
                      ,(    "INQUIRY")
                      ,(    "SAVERESULTS")
                      ,(    "SHAREDISK")
                      ,(    "STATISTICS")
```

```
%103-       00295790 C  0010:0056:1
%103-       00295795 C  0010:0056:1
%103-       00295800 C  0010:0056:1
%103-       00295805 C  0010:0056:1
%103-       00295810 C  0010:0056:1
%103-       00295815 C  0010:0056:1
%103-       00295820 C  0010:0056:1
%103-       00295825 C  0010:0056:1
%103-       00295830 C  0010:0056:1
%103-       00295835 C  0010:0056:1
%103-       00295840 C  0010:0056:1
%103-       00295845 C  0010:0056:1
%103-       00295850 C  0010:0056:1
%103-       00295855 C  0010:0056:1
%103-       00295860 C  0010:0056:1
%103-       00295865 C  0010:0056:1
%103-       00295870 C  0010:0056:1
%103-       00295875 C  0010:0056:1
%103-       00295880 C  0010:0056:1
%103-       00295885 C  0010:0056:1
%103-       00295890 C  0010:0056:1
%103-       00295895 C  0010:0056:1
%103-       00295900 C  0010:0056:1
%103-       00295905 C  0010:0056:1
%103-       00295910 C  0010:0056:1
%103-       00295915 C  0010:0056:1
%103-       00295920 C  0010:0056:1
%103-       00295925 C  0010:0056:1
%103-       00295930 C  0010:0056:1
%103-       00295935 C  0010:0056:1
%103-       00295940 C  0010:0056:1
%103-       00295945 C  0010:0056:1
%103-       00295985 C  0010:0056:1
      15 IS  100 LONG, NEXT SEG   10
%103-       00295990 C  0010:0056:1
%103-       00295995 C  0010:0057:2
            00296000 T  0010:0057:3
            00297000 T  0010:0057:3
    START OF SEGMENT **********   16
            00298000 T  0016:0057:3
      16 IS   12 LONG, NEXT SEG   10
%101-       00299000 P  0010:0057:3
    START OF SEGMENT **********   17
      17 IS   12 LONG, NEXT SEG   10
            00300000 T  0010:0057:3
    START OF SEGMENT **********   18
            00301000 T  0018:0057:3
            00302000 T  0018:0057:3
            00303000 T  0018:0057:3
            00304000 T  0018:0057:3
            00305000 T  0018:0057:3
            00306000 T  0018:0057:3
            00307000 T  0018:0057:3
            00308000 T  0018:0057:3
            00309000 T  0018:0057:3
            00310000 T  0018:0057:3
            00311000 T  0018:0057:3
            00312000 T  0018:0057:3
```

```
                                ,(    "AUXMEM")
                                ,(    "RJE")
                                ,(    "B6500LOAD")
                                ,(    "SEPTICTANK")
                                ,(    "PACKETS")
                                ,(    "MONITOR")
                                                  ;

ALPHA ARRAY ID[0:2];%USED TO CONTAIN AN IDENT WHOSE PRT LOC IS SOUGHT
PROCEDURE TABLELOOKUP(ID,LOC);ARRAY ID[0];INTEGER LOC;
BEGIN
INTEGER I,N;

BOOLEAN STREAM PROCEDURE GOTIDENT(A,N,B);VALUE N;
BEGIN S1:=A;DI:=B;TALLY:=0;
    IF N SC = DC THEN TALLY:=1;
    GOTIDENT:=TALLY;
END GOTIDENT;
KLASS:=KLASS-1;
FOR I:= 129 STEP 1 UNTIL PRTMAX DO
IF N:=NAME[I].[18:30] NEQ 0 THEN
IF N.FF=LOC OR LOC=0 THEN
IF KLASS LEQ 0 OR KLASS=NAME[I].[3:5] THEN
IF GOTINENT(NAMS[N.CF],N:=8*N.FF,ID[0]) THEN
BEGIN
        LOC:=I;
        I:=PRTMAX+2;%FORCE FALL THRU
END;
 IF I GEQ PRTMAX +2 THEN ELSE LOC:=-1;
END OF TABLELOOKUP;

DEFINE GETIDLOC(GETIDLOC1,GETIDLOC2,GETIDLOC3,GETIDLOC4)=
        BEGIN
            FILL ID[*] WITH GETIDLOC1,GETIDLOC2,GETIDLOC3;
            TABLELOOKUP(ID,GETIDLOC4);
        END#;%
PROCEDURE FIXDEFINES;
BEGIN COMMENT**************************************************************
    THIS PROCEDURE IS RESPONSIBLE FOR FINDING THE PRT LOCATION OF AN
    IDENTIFIER DEFINED IN "PRTS" FOR WHICH THERE IS A CORRESPONDING
    ENTRY IN "INFO".  IT DOES THIS BY CALLING "TABLELOOKUP" TO
    PERFORM A LINEAR SEARCH OF "NAMES" TO LOCATE AN IDENTIFIER IN
    "NAMS" AND COMPARE IT WITH THE ONE GIVEN IN "INFO".  IF AN
    IDENTIFIER CAN NOT BE LOCATED IN "NAMS", THE "PRTS" ELEMENT IS
    LEFT UNCHANGED--OTHERWISE, THE ADDRESS OBTAINED IS STORED
    IN "PRTS",*****************************************************;
    INTEGER I,LOC;

    INTEGER STREAM PROCEDURE IDLENGTH(A);
        BEGIN LOCAL L; SI:=LOC L; DI:=A;
          3(IF SC=DC THEN JUMP OUT ELSE BEGIN TALLY:=TALLY+1;
            SI:=LOC L END);   IDLENGTH:=TALLY;
        END IDLENGHTH;
    FILLINFO;
    FOR I:=0 STEP 4 UNTIL INFOMAX DO
    IF (LOC:=IDLENGTH(INFO[I]))=0 THEN ELSE BEGIN
        MOVE(INFO[I],ID[0],3); KLASS:=INFO[I+3];
```

```
 00313000 T  0018:0057:3
 00313100 T  0018:0057:3
 00313200 T  0018:0057:3
 00313300 T  0018:0057:3
 00313400 T  0018:0057:3
 00313500 T  0018:0057:3
 00314000 T  0018:0057:3
  18 IS  108 LONG, NEXT SEG   10
 00315000 T  0010:0057:3
 00316000 T  0010:0060:1
 00317000 T  0010:0060:1
 00318000 T  0010:0060:1
START OF SEGMENT *********   19
 00319000 T  0019:0000:0
 00320000 T  0019:0000:0
 00321000 T  0019:0000:1
 00322000 T  0019:0001:3
 00323000 T  0019:0002:0
 00324000 T  0019:0003:2
 00325000 T  0019:0004:0
 00326000 T  0019:0005:2
 00327000 T  0019:0007:3
 00328000 T  0019:0010:0
 00329000 T  0019:0013:3
 00330000 T  0019:0018:1
 00331000 T  0019:0019:2
 00332000 T  0019:0020:0
 00333000 T  0019:0021:2
 00334000 T  0019:0023:3
 00335000 T  0019:0027:2
  19 IS   30 LONG, NEXT SEG   10
 00336000 T  0010:0060:1
 00337000 T  0010:0060:1
 00338000 T  0010:0060:1
 00339000 T  0010:0060:1
 00340000 T  0010:0060:1
 00341000 T  0010:0060:1
 00342000 T  0010:0060:1
 00343000 T  0010:0060:1
 00344000 T  0010:0060:1
 00345000 T  0010:0060:1
 00346000 T  0010:0060:1
 00347000 T  0010:0060:1
 00348000 T  0010:0060:1
 00349000 T  0010:0060:1
 00350000 T  0010:0060:1
 00351000 T  0010:0060:1
START OF SEGMENT *********   20
 00352000 T  0020:0000:0
 00353000 T  0020:0000:0
 00354000 T  0020:0000:1
 00355000 T  0020:0002:0
 00356000 T  0020:0003:2
 00357000 T  0020:0004:0
 00358000 T  0020:0005:3
 00359000 T  0020:0007:2
 00360000 T  0020:0010:1
```

```
            TABLELOOKUP(ID,LOC);
            PRTS[I/4]:=IF LOC GTR 0 THEN LOC ELSE PRTS[I/4] END;
        KLASS:=0;
        IF PRT32,[46:1] THEN FOR I:=0 STEP 1 UNTIL PRTSMAX DO
        WRITE(P,<"PRTS[",I3,"]= ",I4>,I,PRTS[I]);

END FIXING DEFINES;

BOOLEAN STREAM PROCEDURE BITON(W,B);
    VALUE B;
BEGIN
    SI:=W;SKIP B SB;
    IF SB THEN TALLY:=1;
    BITON:=TALLY;
END OF BITON;
PROCEDURE NEXTITEM;
    BEGIN
        WRITE(P);
        WRITE(P[DBL],STARS);
     END;
PROCEDURE PRINTMCPOPTIONS;
BEGIN
     WRITE(P[DBL],MCPHDR,LEVEL,SUBLEVEL);
    FOR I:=13,0,15,1,2,3,4,5,6,7,8,9,18,17,14,10,16,11,12 DO
    IF BITON(MCP,47-I) THEN WRITE(P,MCPOPT[I]);
    WRITE(P[DBL]);
END OF PRINTMCPOPTIONS;

DEFINE    DATE      =(119)#,
          CLOCK     =(120)#,
          XCLOCK    =(121)#;
    BOOLEAN ARRAY MODON[0:7];
    INTEGER ARRAY ITD[0:9];
    ARRAY RTD[0:5];
    DEFINE TIMEANALYZED           = ITD[0]#,
           DATEANALYZED           = ITD[1]#,
           TIMETAKEN              = ITD[2]#,
           DATETAKEN              = ITD[3]#,
           TIMELASTHL             = ITD[4]#,
           SINCSLASTHL            = ITD[5]#,
           MINUTES                = ITD[6]#,
           SECONDS                = ITD[7]#,
           DAYS                   = ITD[8]#,
           YEARS                  = ITD[9]#;
    DEFINE DATX                   = RTD[0]#,
           XCLOCX                 = RTD[1]#,
           CLOCX                  = RTD[2]#,
           TEMP                   = RTD[3]#,
           KINDS                  = RTD[4]#,
           MCPVERSION             = RTD[5]#;
    FORMAT OUT FMXX("DATE ANALYZED ",I2,"/",I2,"/",I2/

    "TIME ANALYZED ",I2,":",I2,":",I2),
    X1("MCP VERSION NUMBER ",I2/
    "DATE TAKEN    ",I2,"/",I2,"/",I2/
    "TIME TAKEN    ",I2,":",I2,":",I2),
    X1MARKXI("MCP VERSION NUMBER ",A1,A*/
                "DATE TAKEN    ",I2,"/",I2,"/",I2 /
                "TIME TAKEN    ",I2,":",I2,":",I2),
```

```
00361000 T  0020:0014:0
00362000 T  0020:0015:3
00363000 T  0020:0022:0
00364000 T  0020:0023:2
00365000 T  0020:0025:2
 21 IS    7 LONG, NEXT SEG   20
00366000 T  0020:0036:1
 20 IS   39 LONG, NEXT SEG   10
00395000 T  0010:0060:1
00396000 T  0010:0060:1
00397000 T  0010:0060:1
00398000 T  0010:0061:2
00399000 T  0010:0061:3
00400000 T  0010:0062:1
00401000 T  0010:0062:1
00402000 T  0010:0063:3
00403000 T  0010:0063:3
00404000 T  0010:0063:3
00405000 T  0010:0068:0
00406000 T  0010:0071:2
00407000 T  0010:0071:2
00408000 T  0010:0071:2
00409000 T  0010:0071:2
00410000 T  0010:0081:2
00411000 T  0010:0119:3
00412000 T  0010:0126:1
00413000 T  0010:0130:1
00414000 T  0010:0132:0
00415000 T  0010:0132:0
00416000 T  0010:0132:0
00417000 T  0010:0132:0
00418000 T  0010:0135:3
00419000 T  0010:0138:0
00420000 T  0010:0140:1
00421000 T  0010:0140:1
00422000 T  0010:0140:1
00423000 T  0010:0140:1
00424000 T  0010:0140:1
00425000 T  0010:0140:1
00426000 T  0010:0140:1
00427000 T  0010:0140:1
00428000 T  0010:0140:1
00429000 T  0010:0140:1
00430000 T  0010:0140:1
00431000 T  0010:0140:1
00432000 T  0010:0140:1
00433000 T  0010:0140:1
00434000 T  0010:0140:1
00435000 T  0010:0140:1
00436000 T  0010:0140:1
START OF SEGMENT **********   22
00437000 T  0022:0140:1
00438000 T  0022:0140:1
00439000 T  0022:0140:1
00440000 T  0022:0140:1
00441000 T  0022:0140:1
00442000 T  0022:0140:1
00443000 T  0022:0140:1
```

```
   FMX2("TIME OF THE LAST HALT-LOAD ",I2,":",I2,":",I2),
   SYSID("DUMP TAKEN ON SYSTEM ",A1),
   FTAPDSK(A4," FILE CONTAINING DUMP IS ",A1,A6,"/",A1,A6),
   FCOMVAL("COMMON VALUE USED IS ",I5),
    FANAL("ANALYZER VERSION=",A1),
FMX3("TIME SINCE LAST HALT-LOAD  ",I2,":",I2,":",I2);

     FORMAT BADBED("********* BAD BED ENTRY *********");


     FORMAT BADDATE ("BAD DATE TAKEN ....."),

            BADXCLOCK ("BAD TIME TAKEN ....."),
            BADCLOCK("BAD TIME OF H/L");

        FORMAT BADCELL3("WORD 3 HAS THE FLAG BIT ON.....");


     PROCEDURE TIMES (WHEN,HRS,MIN,SEC);
     REAL WHEN;  INTEGER SEC,MIN,HRS;
     BEGIN
     INTEGER T;

IF WHEN LSS 0 OR WHEN GTR 5184000 THEN HRS:=MIN:=SEC:=100 %OVERFLOW
            ELSE BEGIN T:=WHEN;
            HRS:=T DIV 216000;
            MIN:=T DIV 3600 MOD 60;
            SEC:=T DIV 60 MOD 60;        END;
END OF TIMES PROCEDURE;

PROCEDURE DATES(ADATE,MONTH,DAY,YEAR);
VALUE ADATE;
ALPHA ADATE;
INTEGER MONTH,DAY,YEAR ;
   BEGIN
     REAL Y,D,M;

     LABEL ON;
     ARRAY DAYTABLE [0:11];
     STREAM PROCEDURE CONV (YEAR,DAY,DAT );
      VALUE DAT ;
      BEGIN
SI:= LOC DAT;
SI := SI +3;
DI := YEAR; DS := 2 OCT;
DI := DAY; DS := 3 OCT;
      END;
     FILL DAYTABLE [*] WITH
      0,31,59,90,120,151,181,212,243,273,304,334;


     CONV (Y,D,ADATE);
     IF ((Y MOD 4)=0) AND (Y≠0) THEN
       BEGIN
         IF D =60 THEN
           BEGIN
M := 1; GO TO ON;
```

```
        00444000 T  0022:0140:1
        00444100 T  0022:0140:1
        00444200 T  0022:0140:1
        00444300 T  0022:0140:1
        00444400 T  0022:0140:1
        00445000 T  0022:0140:1
  22 IS  136 LONG, NEXT SEG   10
        00446000 T  0010:0140:1
START OF SEGMENT **********   23
  23 IS    9 LONG, NEXT SEG   10
        00447000 T  0010:0140:1
START OF SEGMENT **********   24
        00448000 T  0024:0140:1
        00449000 T  0024:0140:1
  24 IS   20 LONG, NEXT SEG   10
        00450000 T  0010:0140:1
START OF SEGMENT **********   25
  25 IS    9 LONG, NEXT SEG   10
        00451000 T  0010:0140:1
        00452000 T  0010:0140:1
        00453000 T  0010:0140:1
        00454000 T  0010:0140:1
START OF SEGMENT **********   26
        00455000 T  0026:0000:0
        00456000 T  0026:0003:2
        00457000 T  0026:0007:3
        00458000 T  0026:0009:2
        00459000 T  0026:0011:2
        00460000 T  0026:0013:2
  26 IS   18 LONG, NEXT SEG   10
        00461000 T  0010:0140:1
        00462000 T  0010:0140:1
        00463000 T  0010:0140:1
        00464000 T  0010:0140:1
        00465000 T  0010:0140:1
        00466000 T  0010:0140:1
START OF SEGMENT **********   27
        00467000 T  0027:0000:0
        00468000 T  0027:0000:0
        00469000 T  0027:0001:3
        00470000 T  0027:0001:3
        00471000 T  0027:0001:3
        00472000 T  0027:0003:2
        00473000 T  0027:0003:2
        00474000 T  0027:0003:3
        00475000 T  0027:0004:0
        00476000 T  0027:0004:1
        00477000 T  0027:0004:1
        00478000 T  0027:0005:3
START OF SEGMENT **********   28
  28 IS   12 LONG, NEXT SEG   27
        00479000 T  0027:0006:1
        00480000 T  0027:0008:0
        00481000 T  0027:0010:0
        00482000 T  0027:0010:1
        00483000 T  0027:0011:3
        00484000 T  0027:0012:0
```

```
          END;
        IF D > 60 THEN D:=D-1;
      END;
 FOR M := 0 STEP 1 UNTIL 11 DO
      IF DAYTABLE [M] GEQ D THEN GO TO ON;
    ON:
 MONTH := M;
  IF M=0 THEN D:=0 ELSE %GO AHEAD
 DAY := D - DAYTABLE[M-1];
 YEAR :=Y;
   END OF PROCEDURE DATE;  %RCC

INTEGER MAXMOD,MAXCOR;
INTEGER TABLESLOC;
BOOLEAN LNKSOK,AVALNKOK,SOMOKF,SOMOKB;
% UTILITY PROCEDURES
REAL PROCEDURE OCTAL(N);%
    VALUE N; %
    INTEGER N;%
    % N.[1:23]=0 SO THAT IF N CONTAINS AT MOST A HALF-WORD THEN
    % OCTAL IF PRINTED USING O FORMAT, OR A FORMAT FOR FEWER OCTADES
    % WILL BE THE OCTAL REPRESENTATION OF N
OCTAL:=N.[45:3]&(IF N>7 THEN OCTAL(N.[24:21]) ELSE 0)[3:9:39];
REAL STREAM PROCEDURE CHRS(AT,SKIPPING,MANY);
    VALUE SKIPPING,MANY; %
    % RETURNING THE 7 OR LESS CHRS REQUIRED
    BEGIN
        SI:=AT; SI:=SI+SKIPPING;
        DI:=LOC CHRS; DS:=8 LIT"0"; DI:=DI-MANY;
        DS:=MANY CHR;
    END CHRS;
    INTEGER PROCEDURE HIHALF(LOC);
        VALUE LOC;
        INTEGER LOC; %
    HIHALF:=CHRS(M[LOC.ROW,LOC.COL],0,4); %
    INTEGER PROCEDURE LOHALF(LOC); %
        VALUE LOC;
        INTEGER LOC; %
    LOHALF:=CHRS(M[LOC.ROW,LOC.COL],4,4); %
    BOOLEAN STREAM PROCEDURE FLGBIT(WORD);
    BEGIN
        SI:=WORD; %
        IF SB THEN TALLY:=1; %
        FLGBIT:=TALLY; %
    END FLGBIT; %
ARRAY THISROW,LASTROW[0:11]; %
SAVE ARRAY ALINE[0:18];
ARRAY BLINE[0:18];
BOOLEAN NOTPRINTCALL,AVALNK;
BOOLEAN STREAM PROCEDURE COMPAREWORDS(S,D,W);
    VALUE W;
BEGIN
    LABEL XIT;
    SI:=S;DI:=D;
    W(IF 8 SC NEQ DC THEN JUMP OUT TO XIT);
    TALLY:=1;
    XIT:COMPAREWORDS:=TALLY;
```

```
00485000 T  0027:0013:2
00486000 T  0027:0013:2
00487000 T  0027:0015:3
00488000 T  0027:0015:3
00489000 T  0027:0017:2
00490000 T  0027:0020:1
00491000 T  0027:0021:2
00492000 T  0027:0022:0
00493000 T  0027:0024:0
00494000 T  0027:0026:1
00495000 T  0027:0027:3
27 IS   32 LONG, NEXT SEG   10
00496000 T  0010:0140:1
00497000 T  0010:0140:1
00498000 T  0010:0140:1
00528000 T  0010:0140:1
00529000 T  0010:0140:1
00530000 T  0010:0140:1
00531000 T  0010:0140:1
00532000 T  0010:0140:1
00533000 T  0010:0140:1
00534000 T  0010:0140:1
00535000 T  0010:0140:1
00536000 T  0010:0148:0
00537000 T  0010:0148:0
00538000 T  0010:0148:0
00539000 T  0010:0148:0
00540000 T  0010:0149:2
00541000 T  0010:0149:3
00542000 T  0010:0151:3
00543000 T  0010:0152:0
00544000 T  0010:0153:2
00545000 T  0010:0153:2
00546000 T  0010:0153:2
00547000 T  0010:0153:2
00548000 T  0010:0160:0
00549000 T  0010:0160:0
00550000 T  0010:0160:0
00551000 T  0010:0160:0
00552000 T  0010:0167:2
00553000 T  0010:0167:2
00554000 T  0010:0168:0
00555000 T  0010:0168:0
00556000 T  0010:0169:2
00557000 T  0010:0169:2
00558000 T  0010:0170:0
00559000 T  0010:0174:1
00559050 T  0010:0177:2
00559100 T  0010:0179:3
00560000 T  0010:0179:3
00561000 T  0010:0179:3
00562000 T  0010:0179:3
00563000 T  0010:0180:0
00564000 T  0010:0180:0
00565000 T  0010:0180:1
00566000 T  0010:0182:1
00567000 T  0010:0183:2
```

```
END COMPARING WORDS;
%
DEFINE PRINT(PRINT1,PRINT2)=IF NODUMP THEN ELSE
        PRINTCORE(PRINT1,PRINT2)#;
%
    PROCEDURE PRINTCORE(FROM,TOO);
        VALUE FROM,TOO; %
        INTEGER FROM,TOO; %
    BEGIN %
        DEFINE FORI = FOR I := 0 STEP 1 UNTIL Z1 DO#; %

STREAM PROCEDURE BUILD(FROM,LINE,XA,NA,XB,NB,NC,AL,AO); %
VALUE FROM,NA,NB,NC,AL,AO; %
        BEGIN  DI:=LINE;SI:=LOC FROM;SI:=SI+5;SKIP 3SB;
        5 OCTADE; DS := LIT " "; AO( SI := XA; %
                NA(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));SI:=XB;
                NB(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));
        NC(DS:=19LIT" "); DS:=LIT" "); %
     AL(SI:=XA;NA(DS:=LIT" ";DS:=8 CHR); %
        SI:=XB;NB(DS:=LIT" ";DS:=8 CHR); %
        NC(DS:=9LIT" "); DS:=6 LIT " "); END BUILD; %
STREAM PROCEDURE MV(A,B); BEGIN SI:=A;DI:=B;DS:= WDS END MV;
STREAM PROCEDURE EXPAND(ALINE,BLINE,N); VALUE N;
        BEGIN DI:=BLINE; 18(DS:=8LIT" "); DI:=BLINE; DI:=DI+7;
              SI:=ALINE; SI:=SI+7;
              N(2(8 OCTADE; DI:=DI+1); SI:=SI+1);
         END EXPAND;
        BOOLEAN MATCH,FIRST,LAST,STARD;
        INTEGER I,R,STARCOUNT;
         BOOLEAN AL,AO;      INTEGER Z,Z1; %
        LABEL EDITLINE;
         IF DUMPALPHAONLY THEN AL:=TRUE ELSE
         IF DUMPALPHAOCTAL THEN AO:=TRUE ELSE
         IF NOT DUMPOCTALONLY THEN
         IF(I:=AREATYPE[IF NOTPRINTCALL THEN LINKTYPE ELSE
              TYPMAX+1]) GTR 0 AND AVALNK THEN ELSE
         IF I:=ABS(I)=3 THEN AL:=TRUE ELSE
         IF I=2 THEN AO:=TRUE;
        Z1:=(Z:=IF AL THEN 12 ELSE IF AL:=AO THEN 4 ELSE 6)-1;
         IF NOT (AL OR AO) THEN AO:=TRUE;
        AO:=AO AND TRUE; AL:=AL AND TRUE; %
        STARCOUNT:=IF Z=12 THEN 114 ELSE IF Z=6 THEN 120 ELSE 119;
        FIRST:=TRUE;
        START := FROM;                                          %103=

ERROR NUMBER 100 -- START,
        TOO:=TOO,[32:16]; FROM:=FROM,[32:16];
        IF TOO>FROM THEN %
        DO %
        BEGIN %
         IF NOT LAST := (TOO - FROM LEQ Z) THEN % NOT LAST LINE
            BEGIN
                IF FIRST THEN
                BEGIN
IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],LASTROW,Z)
ELSE %
                FORI MV(M[(FROM+I).ROW,(FROM+I).COL],
```

```
00568000 T  0010:0183:2
00568100 T  0010:0184:0
00568200 T  0010:0184:0
00568300 T  0010:0184:0
00568400 T  0010:0184:0
00569000 T  0010:0184:0
00570000 T  0010:0184:0
00571000 T  0010:0184:0
00572000 T  0010:0184:0
00573000 T  0010:0184:0
START OF SEGMENT **********   29
00576000 T  0029:0000:0
00577000 T  0029:0000:0
00578000 T  0029:0000:0
00579000 T  0029:0001:2
00580000 T  0029:0005:3
00581000 T  0029:0011:2
00582000 T  0029:0016:1
00583000 T  0029:0022:0
00584000 T  0029:0025:2
00585000 T  0029:0027:3
00586000 T  0029:0031:3
00586100 T  0029:0033:2
00586110 T  0029:0033:2
00586120 T  0029:0035:3
00586130 T  0029:0036:0
00586140 T  0029:0041:2
00587000 T  0029:0041:2
00588000 T  0029:0041:2
00589000 T  0029:0041:2
00589100 T  0029:0041:2
00590000 T  0029:0041:2
00591000 T  0029:0044:0
00592000 T  0029:0046:1
00593000 T  0029:0048:0
00594000 T  0029:0050:0
00594100 T  0029:0053:2
00594200 T  0029:0056:0
00594300 T  0029:0058:1
00594400 T  0029:0064:0
00595000 T  0029:0066:0
00596000 T  0029:0068:1
00597000 T  0029:0073:3
00597100 C  0029:0074:0
            <<<<<<<<<<<<<<<<<<<<
00598000 T  0029:0074:0
00599000 T  0029:0076:1
00600000 T  0029:0077:3
00601000 T  0029:0078:0
00602000 T  0029:0078:0
00603000 T  0029:0080:0
00604000 T  0029:0080:1
00605000 T  0029:0080:1
00606000 T  0029:0081:2
00607000 T  0029:0087:2
00608000 T  0029:0087:3
```

```
                    LASTROW[I]);
                  STARD:=FALSE;
               IF Z=12 THEN GO EDITLINE;
                  FIRST:=FALSE;
                  END
                  ELSE
                  BEGIN
IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],THISROW,Z)
ELSE %
                  FORI MV(M[(FROM+I).ROW,(FROM+I).COL],
                        THISROW[I]);
         IF (MATCH:=COMPAREWORDS(THISROW[0],LASTROW[0],Z)) %
                        AND NOT STARD THEN
                  BEGIN
                       IF DOUBLESPACE THEN
                       WRITE(P[DBL],STARZ,STARCOUNT) ELSE
                       WRITE(P,STARZ,STARCOUNT);
                      STARD:=TRUE;
                  END;
                  IF NOT MATCH THEN
                  BEGIN
                       STARD:=FALSE;
                    MOVE(THISROW,LASTROW,Z); %
                  END;
                  END;
              END;% IF NOT LAST
              IF LAST    OR
                   NOT STARD OR
                   NOT MATCH THEN
              BEGIN
EDITLINE:
          R := MIN(Z,TOO-FROM); %
          IF(FROM+R).ROW NEQ FROM.ROW THEN % CROSS ROW ROUND
          BUILD(FROM,ALINE[*],M[FROM.ROW,FROM.COL],512-FROM.COL,
              M[(FROM+R).ROW,0],(FROM+R).COL,IF R LSS Z THEN %
         Z-R ELSE 0,AL,A0) ELSE %  STILL IN SAME ROW OF M ARRAY
          BUILD(FROM ,ALINE[*],M[FROM.ROW,FROM.COL],R,
              M[0,0],0,IF R LSS Z THEN Z-R ELSE 0,AL,A0); %
          IF Z=12 THEN IF FIRST THEN
          BEGIN
               FIRST:=FALSE;
               IF DONTPRINTLINKS THEN ELSE
               BEGIN
               EXPAND(ALINE,BLINE,2+REAL(AVALNK));
               WRITE(P[DBL],18,BLINE[*]);
               END;
          END;
          IF DOUBLESPACE THEN WRITE(P[DBL],18,ALINE[*]) ELSE
          WRITE(P,18,ALINE[*]);
             END; %
        END UNTIL (FROM := FROM + Z) GEQ TOO; %
        WRITE(P); %
    END PRINT; %

FORMAT ITEM(A5," = ",2(0,X1),A5,X89);
```

```
00609000 T  0029:0092:1
00610000 T  0029:0095:3
00610100 T  0029:0096:0
00611000 T  0029:0097:3
00612000 T  0029:0098:0
00613000 T  0029:0098:0
00614000 T  0029:0098:0
00615000 T  0029:0098:1
00616000 T  0029:0104:1
00617000 T  0029:0105:2
00618000 T  0029:0110:1
00619000 T  0029:0113:3
00620000 T  0029:0116:0
00621000 T  0029:0117:2
00622000 T  0029:0117:3
00623000 T  0029:0117:3
00624000 T  0029:0126:0
00625000 T  0029:0134:0
00626000 T  0029:0135:2
00627000 T  0029:0135:2
00628000 T  0029:0135:3
00629000 T  0029:0136:0
00630000 T  0029:0136:1
00631000 T  0029:0138:1
00632000 T  0029:0138:1
00633000 T  0029:0138:1
00634000 T  0029:0138:1
00635000 T  0029:0138:1
00636000 T  0029:0139:3
00637000 T  0029:0140:0
00637100 T  0029:0140:1
00638000 T  0029:0141:2
00639000 T  0029:0144:1
00640000 T  0029:0146:1
00641000 T  0029:0152:0
00642000 T  0029:0156:0
00643000 T  0029:0159:2
00644000 T  0029:0163:2
00644100 T  0029:0168:0
00644200 T  0029:0169:3
00644300 T  0029:0170:0
00644310 T  0029:0171:2
00644320 T  0029:0171:3
00644400 T  0029:0172:0
00644500 T  0029:0174:1
00644510 T  0029:0178:1
00644600 T  0029:0178:1
00645000 T  0029:0178:1
00646000 T  0029:0183:3
00647000 T  0029:0188:1
00648000 T  0029:0188:1
00649000 T  0029:0190:1
00650000 T  0029:0194:1
  29 IS  199 LONG, NEXT SEG   10
00651000 T  0010:0184:0
START OF SEGMENT **********   30
  30 IS   11 LONG, NEXT SEG   10
```

```
BOOLEAN PROCEDURE PDATADESC(AT,WHAT); VALUE AT;                                  00651100 T  0010:0184:0
                                      INTEGER AT; REAL WHAT;FORWARD;             00651200 T  0010:0184:0
ARRAY LINE[0:16];                                                    %103-       00652000 P  0010:0184:0
PROCEDURE DISPLAYIT(WHAT,RANGE,ADR);                                             00653000 T  0010:0187:3
    VALUE WHAT,RANGE,ADR;                                                        00654000 T  0010:0187:3
    INTEGER WHAT,ADR;                                                            00655000 T  0010:0187:3
    BOOLEAN RANGE;                                                               00656000 T  0010:0187:3
BEGIN                                                                            00657000 T  0010:0187:3
    INTEGER H,L,T;                                                               00658000 T  0010:0187:3
                                                                  START OF SEGMENT ********** 31
    BOOLEAN PP;                                                                  00658100 T  0031:0000:0
    T:=IF PP:=(ADR=0) THEN WHAT ELSE ADR;                                        00658500 T  0031:0000:0
    IF NOT PP THEN IF PDATADESC(T,H) AND H.[8:10] NEQ 0 THEN ELSE                00658600 T  0031:0003:2
    RANGE:=FALSE;                                                                00658700 T  0031:0007:2
    WRITE(LINE[*],ITEM,OCTAL(T),                                                 00659000 T  0031:0008:1
            OCTAL(H:=HIHALF(T)),                                                 00660000 T  0031:0016:0
            OCTAL(L:=LOHALF(T)),                                                 00661000 T  0031:0019:3
            IF RANGE THEN OCTAL(H,[32:10]+L.CF-1)                                00662000 T  0031:0023:2
            ELSE "     ");                                                       00663000 T  0031:0026:1
    IF 129≤WHAT AND WHAT≤PRTMAX THEN                                             00664000 T  0031:0033:2
    MOVE(NAMS[NAME[WHAT].CF],LINE[4],                                            00665000 T  0031:0035:2
             NAME[WHAT].FF);                                                     00666000 T  0031:0038:0
IF SGLTOG THEN WRITE(P,17,LINE[*]) ELSE                                          00666500 T  0031:0040:0
    IF PP THEN                                                                   00666900 T  0031:0045:2
WRITE(P[DBL],17,LINE[*]);                                                        00667000 T  0031:0045:3
END DISPLAYIT;                                                                   00668000 T  0031:0050:1
                                                                  31 IS   53 LONG, NEXT SEG   10
DEFINE DISPLAY(DISPLAY1,DISPLAY2)=                                               00668100 T  0010:0187:3
       DISPLAYIT(DISPLAY1,DISPLAY2,0)#;                                          00668200 T  0010:0187:3
BOOLEAN PROCEDURE OPERAND(AT,WHAT); %                                            00669000 T  0010:0187:3
    VALUE AT; %                                                                  00670000 T  0010:0187:3
    INTEGER AT; %                                                                00671000 T  0010:0187:3
    REAL WHAT; %                                                                 00672000 T  0010:0187:3
BEGIN %                                                                          00673000 T  0010:0187:3
    INTEGER R,C; %                                                               00674000 T  0010:0187:3
                                                                  START OF SEGMENT ********** 32
    IF FLGBIT(M[R:=AT.ROW,C:=AT.COL]) THEN %                                     00675000 T  0032:0000:0
    OPERAND:=FALSE %                                                             00676000 T  0032:0004:0
    ELSE %                                                                       00677000 T  0032:0005:2
    BEGIN %                                                                      00678000 T  0032:0005:3
        WHAT:=M[R,C]; %                                                          00679000 T  0032:0006:0
        OPERAND:=TRUE; %                                                         00680000 T  0032:0008:0
    END; %                                                                       00681000 T  0032:0008:1
END OPERAND; %                                                                   00682000 T  0032:0008:1
                                                                  32 IS   12 LONG, NEXT SEG   10
BOOLEAN PROCEDURE DESCRIPTOR(AT,WHAT,KIND);                                      00683000 T  0010:0187:3
    VALUE AT;                                                                    00684000 T  0010:0187:3
    INTEGER AT;                                                                  00685000 T  0010:0187:3
    REAL WHAT,KIND;                                                              00686000 T  0010:0187:3
BEGIN                                                                            00687000 T  0010:0187:3
    INTEGER R,C;                                                                 00688000 T  0010:0187:3
                                                                  START OF SEGMENT ********** 33
    IF (KIND:=OCTAL(CHRS(M[R:=AT.ROW,C:=AT.COL],0,1)))≥"40" THEN                 00689000 T  0033:0000:0
    BEGIN                                                                        00690000 T  0033:0006:1
        DESCRIPTOR:=TRUE;                                                        00691000 T  0033:0007:2
        WHAT:=CHRS(M[R,C],1,7);                                                  00692000 T  0033:0007:3
    END;                                                                         00693000 T  0033:0011:2
```

```
    END DESCRIPTOR;

    BOOLEAN PROCEDURE PDATADESC(AT,WHAT);
        VALUE AT;
        INTEGER AT;
        REAL WHAT;
    BEGIN
        INTEGER KIND;

        IF DESCRIPTOR(AT,WHAT,KIND) AND
        KIND LSS "60" AND KIND GEQ "50" THEN
        PDATADESC:=TRUE;
    END;

BOOLEAN PROCEDURE CONTROLDESC(AT,WHAT);
    VALUE AT;
    INTEGER AT;
    REAL WHAT;
BEGIN
    INTEGER TYP;

    CONTROLDESC:=DESCRIPTOR(AT,WHAT,TYP) AND
                 TYP.[40:1]=1  AND
                 TYP.[45:1]=0;
END CONTROLDESC;

    INTEGER MINLNK,MINBAD,MAXBAD,MAXLNK; %
BOOLEAN NEEDCHECKAVAILNKS;%FALSE IF DPMT DUMP
 ARRAY COMMT[0:19]; BOOLEAN COMNT;
      PROCEDURE MCPENTRIES;
        BEGIN
        SGLTOG:=TRUE;
        DISPLAY(BED,FALSE);
        DISPLAY(PRT,FALSE);
        DISPLAY(JAR,FALSE);
        DISPLAY(INTRNSC,FALSE);
        DISPLAY(SHEET,FALSE);
        DISPLAY(JOBNUM,FALSE);
        DISPLAY(NFO,FALSE);
        DISPLAY(ISTACK,FALSE);
        DISPLAY(WAITQUE,FALSE);
        DISPLAY(P1MIX,FALSE);
        DISPLAY(P2MIX,FALSE);
        SGLTOG:=FALSE;
        END;
  PROCEDURE GETPRTENTRIES;
   BEGIN
      REAL ADR,WC,PRTROW,L;

      REAL ADRR;
      INTEGER I,K;
      ARRAY MIX[0:40];
      FORMAT PRTOUT(X10,"PRT LOCATIONS:"/

            X20,"MIX",X20,"PRT"/),
         F1( X20, A2,X20, A5);

      FORMAT BADPRT("********BAD PRT DESCRIPTOR*******");
```

```
        00694000 T  0033:0011:2
  33 IS   14 LONG, NEXT SEG   10
        00695000 T  0010:0187:3
        00696000 T  0010:0187:3
        00697000 T  0010:0187:3
        00698000 T  0010:0187:3
        00699000 T  0010:0187:3
        00700000 T  0010:0187:3
START OF SEGMENT **********   34
        00701000 T  0034:0000:0
        00702000 T  0034:0001:3
        00703000 T  0034:0003:3
        00704000 T  0034:0004:1
  34 IS    8 LONG, NEXT SEG   10
        00705000 T  0010:0187:3
        00706000 T  0010:0187:3
        00707000 T  0010:0187:3
        00708000 T  0010:0187:3
        00709000 T  0010:0187:3
        00710000 T  0010:0187:3
START OF SEGMENT **********   35
        00711000 T  0035:0000:0
        00712000 T  0035:0001:3
        00713000 T  0035:0003:2
        00714000 T  0035:0005:2
  35 IS    8 LONG, NEXT SEG   10
        00715000 T  0010:0187:3
        00716000 T  0010:0187:3
        00717000 T  0010:0187:3
        00718000 T  0010:0190:0
        00719000 T  0010:0190:0
        00719100 T  0010:0190:0
        00720000 T  0010:0191:3
        00721000 T  0010:0193:2
        00722000 T  0010:0194:1
        00723000 T  0010:0196:0
        00724000 T  0010:0197:3
        00725000 T  0010:0199:2
        00726000 T  0010:0200:1
        00727000 T  0010:0202:0
        00728000 T  0010:0203:3
        00729000 T  0010:0205:2
        00730000 T  0010:0206:1
        00730100 T  0010:0208:0
        00731000 T  0010:0209:2
        00732000 T  0010:0209:2
        00733000 T  0010:0209:2
        00734000 T  0010:0209:2
START OF SEGMENT **********   36
        00735000 T  0036:0000:0
        00736000 T  0036:0000:0
        00737000 T  0036:0000:0
        00738000 T  0036:0001:3
START OF SEGMENT **********   37
        00739000 T  0037:0001:3
        00740000 T  0037:0001:3
  37 IS   20 LONG, NEXT SEG   36
        00741000 T  0036:0001:3
```

```
            LABEL XIT;
            LABEL NEXT;
      IF NOT(PDATADESC(PRT,L) AND L.CF NEQ 0 AND L.[8:10] GTR 0
            AND L.[8:10] LSS 41) THEN BEGIN
            MIXMAX:=DEFINEDMIXMAX;%                                   FOR LINK CK
              WRITE (P,BADPRT); GO XIT END;
           ADR:= L.[33:15];
           WC:= L.[8:10];
           K:=-1;
           MIXMAX:=WC-1;
          FOR I:= 1 STEP 1 UNTIL (WC-1) DO BEGIN
           IF PDATADESC(ADR+I,PRTROW) THEN ELSE PRTROW:=0;
              IF PRTROW =0 THEN
                 GO TO NEXT;
               K:=K+1;
               MIX[K]:= I; % SAVE MIX NUMBER
               MIX[K:=K+1]:= PRTROW;
               NEXT:
              END; % OF FINDING PRTS & VALID MIXES;
            WRITE(P,PRTOUT);
            WRITE(P,F1,FOR I:=0 STEP 1 UNTIL K DO OCTAL(MIX[I],[33:15]));
             XIT:
           END OF GETPRTENTRIES;

PROCEDURE DUMPARRAY(PRTLOC); VALUE PRTLOC; INTEGER PRTLOC;
BEGIN
    INTEGER LOC,SIZE,I;

    FORMAT F(X2,I4,X2,2(O,X1));


    DISPLAY(PRTLOC,TRUE);
    IF PRTLOC GEQ 129 AND PRTLOC LEQ PRTMAX THEN
    IF PDATADESC(PRTLOC,LOC) THEN
    IF SIZE:=LOC.[8:10] NEQ 0 THEN
    IF LOC:=LOC.CF GTR 0 THEN
    BEGIN
        FOR I:=0 STEP 1 UNTIL SIZE-1 DO
        WRITE(P,F,I,OCTAL(HIHALF(LOC+I)),OCTAL(LOHALF(LOC+I)));
        WRITE(P[DBL]);
    END;
END DUMPARRAY;

PROCEDURE DUMPCESSPOOL;
BEGIN
    FORMAT HDR("D25-ABN=ABNORMAL"/"D24-RR =READ READY"/

                    "D23-BFL=BUFFER FINAL LOCATION"/
                    "D21-WR =WRITE READY"/"D20-BSY=BUSY"/
                    "D18-NTR=DTCU NOT READY"/
                    "NOTE:BSY AND NTR=ADAPTER/DTTU NOT READY"///);

    FORMAT WHATFILE("CORE PORTION OF SEPTIC /",A1,A6,//);
```

```
START OF SEGMENT **********   38
  38 IS    9 LONG, NEXT SEG   36
        00742000 T  0036:0001:3
        00743000 T  0036:0001:3
        00744000 T  0036:0001:3
        00745000 T  0036:0005:2
        00746000 T  0036:0008:0
        00747000 T  0036:0009:2
        00748000 T  0036:0012:1
        00749000 T  0036:0013:3
        00750000 T  0036:0015:2
        00751000 T  0036:0016:0
        00752000 T  0036:0017:2
        00753000 T  0036:0021:3
        00754000 T  0036:0024:1
        00755000 T  0036:0025:3
        00756000 T  0036:0026:0
        00757000 T  0036:0027:2
        00758000 T  0036:0028:1
        00759000 T  0036:0030:1
        00760000 T  0036:0031:2
        00761000 T  0036:0031:3
        00762000 T  0036:0034:1
        00763000 T  0036:0048:0
        00763010 T  0036:0049:2
  36 IS   54 LONG, NEXT SEG   10
        00763020 T  0010:0209:2
        00763022 T  0010:0209:2
        00763024 T  0010:0209:2
START OF SEGMENT **********   39
        00763026 T  0039:0000:0
START OF SEGMENT **********   40
  40 IS   10 LONG, NEXT SEG   39
        00763028 T  0039:0000:0
        00763030 T  0039:0001:2
        00763032 T  0039:0003:2
        00763034 T  0039:0004:1
        00763036 T  0039:0006:1
        00763038 T  0039:0009:2
        00763040 T  0039:0009:3
        00763042 T  0039:0013:3
        00763044 T  0039:0028:1
        00763046 T  0039:0032:1
        00763048 T  0039:0032:1
  39 IS   35 LONG, NEXT SEG   10
        00763100 T  0010:0209:2
        00763105 T  0010:0209:2
        00763110 T  0010:0209:2
START OF SEGMENT **********   41
START OF SEGMENT **********   42
        00763115 T  0042:0000:0
        00763120 T  0042:0000:0
        00763125 T  0042:0000:0
        00763130 T  0042:0000:0
  42 IS   40 LONG, NEXT SEG   41
        00763135 T  0041:0000:0
START OF SEGMENT **********   43
```

```
REAL I,B,FROM; BOOLEAN LO;
FILE SEPTIC DISK SERIAL (2,60,MYUSE=INPUT);
DEFINE BUMPI = I:=I+1#;
DEFINE INTSP=INFO#;
ARRAY CC,NAM[0:3];
LABEL LOOP,EOJ;
STREAM PROCEDURE INIT(A);
BEGIN DI:=A; DS:=32 LIT
     "PASS INTACT, INTWRITE   READ    ";
END INIT;
     STREAM PROCEDURE FILL(A,B,C,D,E); VALUE B,C,D;
     BEGIN DI:=E; 15(DS:=8LIT" "); DI:=E; SI:=A;
          DS:=WDS; SI:=LOC B; DI:=DI+1;
          2(DS:=2DEC; A:=DI;DI:=DI-2;DS:=FILL;DI:=A;DS:=LIT"/");
          DI:=DI-1; DS:=4LIT" ";
          SI:=SI+6; SKIP 2 SB; % START AT D25
          IF SB THEN DS:=3LIT"ABN" ELSE DI:=DI+3; SKIP SB; %D25
          DI:=DI+3;
          IF SB THEN DS:=3LIT"RR " ELSE DI:=DI+3; SKIP SB; %D24
          DI:=DI+3;
          IF SB THEN DS:=3LIT"BFL" ELSE DI:=DI+3; SKIP SB; %D23
          DI:=DI+3;
          SKIP SB;%IGNORE D22
          IF SB THEN DS:=3LIT"WR " ELSE DI:=DI+3; SKIP SB; %D21
          DI:=DI+3;
          IF SB THEN DS:=3LIT"BSY" ELSE DI:=DI+3; SKIP SB; %D20
          DI:=DI+3;
          SKIP SB; %IGNORE D19
          IF SB THEN DS:=3LIT"NTR" ELSE DI:=DI+3; %D18
     END FLL;
PROCEDURE TIM(A,CC); VALUE A; REAL A; ARRAY CC[0];
BEGIN INTEGER I,J;

     STREAM PROCEDURE FF(A,B);
     BEGIN SI:=A; DI:=B;
          4(DS:=2 DEC; DS:=LIT ":");
          DI:=DI-1; DS:=5 LIT " ";
     END FF;
     FOR I:=3 STEP -1 UNTIL 0 DO
     BEGIN CC[I]:=J:=A MOD 60;
          A:=A DIV 60;
     END;
     FF(CC,LINE[7]);
END TIM;

BOOLEAN STREAM PROCEDURE GLUNK(A);
BEGIN SI:=A; IF SC=" " THEN TALLY:=1;
     GLUNK:=TALLY;
END GLUNK;
PROCEDURE MMM(B,I); VALUE B; REAL I,B;
BEGIN REAL C;

     STREAM PROCEDURE MOVE(A,B);
     BEGIN SI:=A; DI:=B;
          8(IF TOGGLE THEN DS:=LIT " " ELSE
          IF SC="+" THEN DS:=CHR ELSE DS:=CHR);
```

```
43 IS    11 LONG, NEXT SEG    41
          00763140 T  0041:0000:0
          00763145 T  0041:0000:0
          00763150 T  0041:0006:0
          00763155 T  0041:0006:0
          00763160 T  0041:0006:0
          00763165 T  0041:0008:0
          00763170 T  0041:0008:0
          00763175 T  0041:0010:0
          00763180 T  0041:0010:1
          00763185 T  0041:0014:1
          00763190 T  0041:0014:1
          00763195 T  0041:0014:1
          00763200 T  0041:0017:3
          00763205 T  0041:0018:0
          00763210 T  0041:0020:1
          00763215 T  0041:0021:3
          00763220 T  0041:0022:0
          00763225 T  0041:0024:0
          00763230 T  0041:0024:0
          00763235 T  0041:0026:0
          00763240 T  0041:0026:1
          00763245 T  0041:0028:1
          00763250 T  0041:0028:1
          00763255 T  0041:0029:2
          00763260 T  0041:0031:2
          00763265 T  0041:0031:2
          00763270 T  0041:0033:2
          00763275 T  0041:0033:3
          00763280 T  0041:0033:3
          00763285 T  0041:0035:3
          00763290 T  0041:0035:3
          00763295 T  0041:0035:3
START OF SEGMENT **********   44
          00763300 T  0044:0000:0
          00763305 T  0044:0000:0
          00763310 T  0044:0000:1
          00763315 T  0044:0001:3
          00763320 T  0044:0003:2
          00763325 T  0044:0003:2
          00763330 T  0044:0005:2
          00763335 T  0044:0007:2
          00763340 T  0044:0008:1
          00763345 T  0044:0010:1
          00763350 T  0044:0012:0
 44 IS    15 LONG, NEXT SEG    41
          00763355 T  0041:0035:3
          00763360 T  0041:0035:3
          00763365 T  0041:0037:2
          00763370 T  0041:0037:2
          00763375 T  0041:0038:0
          00763380 T  0041:0038:0
START OF SEGMENT **********   45
          00763385 T  0045:0000:0
          00763390 T  0045:0000:0
          00763395 T  0045:0000:1
          00763400 T  0045:0001:3
```

```
            END MOVE;
            FOR C:=0 STEP 1 UNTIL R-1 DO
            MOVE(INTSP[BUMPI],LINE[C]);
            WRITE(P,B,LINE[*]);
        END;

        %%%%%%%%%%%%%%%%%%%START
        INIT(NAM);
        IF OPERAND(TOGLE,I) AND BOOLEAN(I,[13:1]) THEN
        IF PDATADESC(ARGH,FROM) AND FROM,[8:10]=128 THEN
        IF OPERAND((FROM:=FROM.CF)+127,I) THEN LO:=TRUE
        ELSE GO TO EOJ ELSE GO TO EOJ ELSE GO TO EOJ;
        WRITE(P[PAGE]);
        WRITE(P,WHATFILE,I,[6:6],I);
        DISPLAY(ARGH,TRUE);
        WRITE(P,HDR);
        GO EOJ; % VOID THIS CARD TO DUMP SEPTIC DISK FILE
        FILL SEPTIC WITH "SEPTIC ",1;
        WHILE LO DO
        BEGIN READ(SEPTIC,60,INTSP[*])[EOJ]; I:=0;
LOOP:
            IF (B:=INTSP[I]),[9:9]=0 THEN GO TO EOJ;
            FILL(NAM[B,[7:2]],B,[9:4],B,[14:4],B,[21:12],LINE);
            TIM(INTSP[BUMPI],[18:30],CC);
            WRITE(P,10,LINE[*]);
            IF (B:=B,[33:15])≠0 THEN
            BEGIN WHILE B GTR 14 DO
                BEGIN MMM(14,I); B:=B-14; END;
                MMM(B,I);
            END;
            WRITE(P);
            IF I≠59 THEN
            IF GLUNK(INTSP[BUMPI]) THEN ELSE GO TO LOOP;
        END;
EOJ:  IF LO THEN
        BEGIN CLOSE(SEPTIC); LO:=FALSE;
            IF OPERAND(FROM+125,I) AND I NEQ 0 THEN
            IF OPERAND(FROM+62,I) AND (I=0 OR I=64) THEN
            BEGIN FROM:=FROM+I;
                IF (B:=512-(R:=FROM,COL)) LSS (I:=60) THEN
                BEGIN MOVE(M[FROM,ROW,R],INTSP,B);
                    R:=0; I:=60-B; FROM:=FROM+60;
                END ELSE B:=0;
                MOVE(M[FROM,ROW,R],INTSP[B],I);
                I:=0; GO TO LOOP;
END; END; END;

INTEGER STREAM PROCEDURE MCPCNT(A); VALUE A;
BEGIN S1:=LOC A;  SI:=SI+1;
    7(IF TOGGLE THEN TALLY:=TALLY+1 ELSE IF SC NEQ "0" THEN
      TALLY:=TALLY+1;SI:=SI+1);    MCPCNT:=TALLY;
END OF MCPCNT;%
INTEGER PRTCODE; % SET TO 0,1 OR 2 BY CHECKPRTFILE
PROCEDURE DATIME;
BEGIN INTEGER I; BOOLEAN B;

    STREAM PROCEDURE PRTFILEMESS(P,L,C);    VALUE C;
```

```
                00763405 T  0045:0003:2
                00763410 T  0045:0003:3
                00763415 T  0045:0008:0
                00763420 T  0045:0011:3
                00763425 T  0045:0015:3
          45 IS   18 LONG, NEXT SEG   41
                00763430 T  0041:0038:0
                00763435 T  0041:0038:0
                00763440 T  0041:0040:0
                00763445 T  0041:0042:0
                00763450 T  0041:0045:3
                00763455 T  0041:0049:2
                00763460 T  0041:0049:3
                00763465 T  0041:0053:3
                00763470 T  0041:0064:0
                00763475 T  0041:0065:3
                00763480 T  0041:0068:1
                00763485 T  0041:0069:2
                00763490 T  0041:0073:2
                00763495 T  0041:0073:3
                00763500 T  0041:0079:3
                00763505 T  0041:0080:0
                00763510 T  0041:0082:1
                00763515 T  0041:0086:1
                00763520 T  0041:0090:0
                00763525 T  0041:0094:0
                00763530 T  0041:0096:0
                00763535 T  0041:0098:0
                00763540 T  0041:0102:0
                00763545 T  0041:0103:2
                00763550 T  0041:0103:2
                00763555 T  0041:0107:2
                00763560 T  0041:0107:3
                00763565 T  0041:0111:2
                00763570 T  0041:0111:3
                00763575 T  0041:0112:0
                00763580 T  0041:0115:2
                00763585 T  0041:0117:3
                00763590 T  0041:0121:3
                00763595 T  0041:0123:3
                00763600 T  0041:0126:1
                00763605 T  0041:0130:0
                00763610 T  0041:0133:3
                00763615 T  0041:0134:1
                00763620 T  0041:0137:3
                00763625 T  0041:0139:2
          41 IS  144 LONG, NEXT SEG   10
                00765000 T  0010:0209:2
                00766000 T  0010:0209:2
                00767000 T  0010:0210:1
                00768000 T  0010:0212:0
                00769000 T  0010:0213:2
                00769900 T  0010:0214:0
                00770000 T  0010:0214:0
                00771000 T  0010:0214:0
      START OF SEGMENT **********   46
                00771100 T  0046:0000:0
```

```
BEGIN LABEL MAYBE,BAD,FIX,XIT;                                                  00771105 T  0046:0000:0
      SI:=P; DI:=L; DS:=4 WDS;                                                  00771110 T  0046:0000:0
      CI:=CI+C;                                                                 00771115 T  0046:0000:1
      GO XIT; % 0                                                               00771120 T  0046:0001:2
      GO MAYBE; % 1                                                             00771125 T  0046:0001:3
      GO BAD; % 2                                                               00771130 T  0046:0001:3
      MAYBE:DS:=9LIT"(APPEARS "; GO FIX;                                        00771135 T  0046:0002:0
      BAD:  DS:=12LIT"(DEFINITELY "; GO FIX;                                    00771140 T  0046:0003:3
      FIX  :DS:=10LIT"INCORRECT)";                                              00771145 T  0046:0006:0
      XIT  :DS:=32LIT" ";                                                       00771150 T  0046:0007:3
END PRTFILEMESS;                                                                00771155 T  0046:0012:0
  IF NOT COMMON THEN BEGIN                                                      00772000 T  0046:0012:1
    IF NOT OPERAND(3,MCPVERSION) THEN WRITE(P,BADCELL3);                        00773000 T  0046:0014:0
     IF FLGBIT(M[DATE,ROW,DATE,COL]) THEN                                       00774000 T  0046:0019:2
       WRITE(P,BADDATE) ELSE                                                    00775000 T  0046:0022:0
         BEGIN                                                                  00776000 T  0046:0025:3
            DATX := M[DATE,ROW,DATE,COL];                                       00777000 T  0046:0026:0
            DATES(DATX,DATETAKEN,DAYS,YEARS);                                   00778000 T  0046:0029:3
         END;  %  OF DATE                                                      00779000 T  0046:0032:0
    IF FLGBIT(M[XCLOCK,ROW,XCLOCK,COL]) THEN                                    00780000 T  0046:0032:0
 WRITE (P,BADXCLOCK) ELSE                                                       00781000 T  0046:0035:2
         BEGIN                                                                  00782000 T  0046:0038:1
            XCLOCX := M[XCLOCK,ROW,XCLOCK,COL];                                 00783000 T  0046:0039:2
            TIMES(XCLOCX,TIMETAKEN,MINUTES,SECONDS);                            00784000 T  0046:0042:1
         END;    %     OF XCLOCK                                              00785000 T  0046:0045:2
    PRTFILEMESS(PRTNAME,LINE,PRTCODE);                                          00785400 T  0046:0045:2
        WRITE(P,8,LINE[*]);                                                     00785500 T  0046:0046:1
         WRITE(P,FANAL,VERSION);                                                00785550 T  0046:0051:2
        WRITE(P,FTAPDSK,FILETYPE,TDMFID,[6:6],TDMFID,                           00785600 T  0046:0058:0
                                    TDFID,[6:6],TDFID);                         00785610 T  0046:0068:0
        WRITE(P,FCOMVAL,REAL(COMMON));                                          00785700 T  0046:0075:2
    WRITE(P,X1MARKXI,IF B:=(I:=MCPCNT(MCPVERSION) GTR 6)                        00786000 T  0046:0084:0
            THEN MCPVERSION,[6:6] ELSE " ", IF B THEN 6 ELSE IF I=0             00787000 T  0046:0091:2
            THEN 1 ELSE I,MCPVERSION,DATETAKEN,DAYS,YEARS,                      00788000 T  0046:0098:0
            TIMETAKEN,MINUTES,SECONDS);                                         00789000 T  0046:0107:2
      END;                                                                      00790000 T  0046:0114:0
      IF SHAREDISK THEN                                                         00790100 T  0046:0114:0
      IF OPERAND(SYSNO,I) AND I GEQ 0 AND I LEQ 3 THEN                          00790110 T  0046:0115:2
      WRITE(P,SYSID,I+17);                                                      00790120 T  0046:0118:1
       DATES(TIME(0),DATEANALYZED,DAYS,YEARS);                                  00791000 T  0046:0129:2
        TIMES(TIME(1),TIMEANALYZED,MINUTES,SECONDS);                            00792000 T  0046:0132:0
 WRITE(P,FMXX,DATEANALYZED,DAYS,YEARS,TIMEANALYZED,MINUTES,SECONDS);            00793000 T  0046:0137:3
      IF COMMON THEN WRITE(P);                                                  00794000 T  0046:0153:2
      IF NOT COMMON THEN BEGIN                                                  00795000 T  0046:0158:0
      IF FLGBIT(M[CLOCK,ROW,CLOCK,COL]) THEN                                    00796000 T  0046:0159:2
        WRITE(P,BADCLOCK) ELSE                                                  00797000 T  0046:0162:0
            BEGIN                                                               00798000 T  0046:0165:3
               CLOCX := M[CLOCK,ROW,CLOCK,COL];                                 00799000 T  0046:0166:0
            IF XCLOCX LSS CLOCX THEN XCLOCX:=XCLOCX+5184000;                    00800000 T  0046:0169:3
            TIMES(XCLOCX-CLOCX,TIMELASTHL,MINUTES,SECONDS);                     00801000 T  0046:0173:2
               WRITE(P,FMX2,TIMELASTHL,MINUTES,SECONDS);                        00802000 T  0046:0180:0
               TIMES(CLOCX,SINCSLASTHL,MINUTES,SECONDS);                        00803000 T  0046:0191:2
               WRITE(P[DBL],FMX3,SINCSLASTHL,MINUTES,SECONDS);                  00804000 T  0046:0193:3
          END ; %OF CLOCK                                                       00805000 T  0046:0205:2
      END;                                                                      00806000 T  0046:0205:2
END;% OF DATIME                                                                 00807000 T  0046:0205:2
                                                                    46 IS  208 LONG, NEXT SEG   10
```

```
PROCEDURE PRINTARRAYROW(PRTLOC,INX); VALUE PRTLOC,INX;                          00807100 T  0010:0214:0
                                     INTEGER PRTLOC,INX;                        00807110 T  0010:0214:0
BEGIN                                                                           00807120 T  0010:0214:0
    INTEGER LOC,SIZE;                                                           00807130 T  0010:0214:0
                                                                        START OF SEGMENT **********   47
    BOOLEAN TOG,COMSAVE,DBLSAVE;                                                00807140 T  0047:0000:0
    LABEL AGAIN;                                                                00807150 T  0047:0000:0
    STREAM PROCEDURE FIXLINE(LINE,INX); VALUE INX;                              00807160 T  0047:0000:0
    BEGIN LOCAL A,B;                                                            00807170 T  0047:0000:0
         SI:=LINE;4(SI:=SI+8);                                                  00807180 T  0047:0000:0
         32(IF SC=" " THEN JUMP OUT ELSE SI:=SI+1);                             00807190 T  0047:0001:2
         A:=SI; DI:=A; DS:=LIT"["; A:=DI;                                       00807200 T  0047:0003:2
         SI:=LOC INX; DS:=4 DEC; B:=DI;                                         00807210 T  0047:0004:0
         DI:=A; DS:=3 FILL; DI:=B;                                              00807230 T  0047:0005:2
         DS:=3LIT",*]";                                                         00807240 T  0047:0005:3
    END FIXLINE;                                                                00807250 T  0047:0006:1
    IF TOG:=BOOLEAN(INX,[1:1]) THEN                                             00807260 T  0047:0006:1
    DISPLAY(PRTLOC,TRUE);                                                       00807270 T  0047:0008:0
    INX:=ABS(INX);                                                              00807275 T  0047:0010:0
    COMSAVE:=COMMON;                                                            00807280 T  0047:0011:2
    COMMON:=FALSE;                                                              00807290 T  0047:0011:3
    DBLSAVE:=DOUBLESPACE;                                                       00807295 T  0047:0012:1
    DOUBLESPACE:=TRUE;                                                          00807300 T  0047:0013:2
    IF PRTLOC GEQ 129 AND PRTLOC LEQ PRTMAX THEN                                00807310 T  0047:0014:0
AGAIN;                                                                          00807320 T  0047:0015:3
    IF PDATADESC(PRTLOC,LOC) THEN                                               00807330 T  0047:0017:2
    IF (SIZE:=LOC,[8:10]) NEQ 0 THEN                                            00807340 T  0047:0018:0
    IF TOG THEN PRINT(LOC:=LOC,CF,MIN(MAXCOR,LOC+SIZE)) ELSE                    00807350 T  0047:0020:0
    BEGIN                                                                       00807360 T  0047:0028:1
         DISPLAYIT(PRTLOC,TRUE,PRTLOC:=LOC,CF+INX);                             00807370 T  0047:0029:2
         FIXLINE(LINE,INX);                                                     00807380 T  0047:0031:3
         WRITE(P[DBL],15,LINE[*]);                                              00807390 T  0047:0033:3
         TOG:=TRUE;                                                             00807400 T  0047:0037:3
         GO AGAIN;                                                              00807410 T  0047:0038:1
    END;                                                                        00807420 T  0047:0039:2
    COMMON:=COMSAVE;                                                            00807430 T  0047:0039:2
    DOUBLESPACE:=DBLSAVE;                                                       00807435 T  0047:0039:3
END PRINTARRAYROW;                                                              00807440 T  0047:0040:1
                                                                      47 IS   44 LONG, NEXT SEG   10
DEFINE PRINTARRAY(PRINTARRAY1)=                                                 00807450 T  0010:0214:0
       PRINTARRAYROW(PRINTARRAY1,*1)#;                                          00807460 T  0010:0214:0
DEFINE PA=PRINTARRAY#,PAROW=PRINTARRAYROW#;                                     00807470 T  0010:0214:0
COMMENT; THE FOLLOWING PROCEDURE IS INTENDED TO BE MODIFIED TO SUIT             00807480 T  0010:0214:0
INDIVIDUAL USER DEBUGGING REQUIREMENTS.  IT ALLOWS AN ARRAY,                    00807482 T  0010:0214:0
EITHER 1 OR 2 DIMENSIONAL, TO BE DISPLAYED WITH A MINIMUM OF EFFORT.            00807484 T  0010:0214:0
IN GENERAL, A 3 CARD PATCH IS NECESSARY TO THE ANALYZER:  ONE CARD TO           00807486 T  0010:0214:0
DEFINE AN ELEMENT IN "PRTS", SEQUENCE #00121000,ANOTHER TO MAKE AN ENTRY        00807488 T  0010:0214:0
IN "FILINFO",SEQUENCE #00165000, AND A THIRD TO CALL BELOW ON                   00807490 T  0010:0214:0
"PRINTARRAY"(OR "PA") FOR 1 DIMENSIONAL ARRAYS OR ON                            00807492 T  0010:0214:0
"PRINTARRAYROW"(OR "PAROW") FOR 2 DIMENSIONAL ARRAYS.                           00807494 T  0010:0214:0
NOTE THAT THE PROCEDURE "PRINTSELECTEDARRAYS" IS CALLED ONLY WHEN               00807496 T  0010:0214:0
COMMON=16;                                                                      00807498 T  0010:0214:0
PROCEDURE PRINTSELECTEDARRAYS;                                                  00807500 T  0010:0214:0
BEGIN                                                                           00807510 T  0010:0214:0
    INTEGER I;                                                                  00807515 T  0010:0214:0
                                                                        START OF SEGMENT **********   48
    IF DUMPTABLES THEN                                                          00807520 T  0048:0000:0
```

```
        BEGIN
             WRITE(PRPAGE]);
        IF DFX THEN PA(EUQ);
        PA(MEMASK);
        PA(QTIMES);
        PA(STATION);
        FOR I:=0 STEP 1 UNTIL 15 DO PAROW(STATION,I);
        END;
        DUMPCESSPOOL;
    END PRINTSELECTEDARRAYS;

    BOOLEAN BADCOMMENT;
    FORMAT COMNTPAR("---PARITY ERROR OCCURRED IN COMMENTS BLOCK...");


        PROCEDURE LOAD;
        BEGIN
            LABEL EOT;

            LABEL EF,PR,IGPAR;
            LABEL FLAG,PAR,BADTM,IGNOREPAR;%
            FORMAT BADEOF("---INVALID EOF AFTER BLOCK # ",I2),

                   PRTFILE("PRT FILE USED IS ",A1,A6,"/",A1,A6,X8),
                   PARERR ("---IRRECOVERABLE PARITY IN BLOCK # ",I2),
                   FLAGERR("---FLAG BIT IN WORD ZERO OF BLOCK # ",I2);

        ARRAY A[0:532];
        STREAM PROCEDURE MOVER(S,D);
        BEGIN
            SI:=S; DI:=D;
            16(DS:=32 WDS);
        END MOVER;
        STREAM PROCEDURE GETCOMMON(COMMT,N);
        BEGIN LOCAL X,Y; LABEL XIT,HIT,GETNUM;
        DI:=N; DS:=8LIT"+0000001"; SI:=COMMT;
        7(20(IF SC="C" THEN BEGIN SI:=SI+1;
             IF SC="O" THEN BEGIN SI:=SI+1;
             IF SC="M" THEN JUMP OUT 2 TO HIT ELSE SI:=SI+1
             END ELSE SI:=SI+1 END ELSE SI:=SI+1));
        GO XIT;
        HIT:15(IF SC="=" THEN JUMP OUT TO GETNUM ELSE SI:=SI+1); GO XIT;
        GETNUM:SI:=SI+1;
        DI:=LOC X;
        10(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);
        5(IF SC GEQ "0" THEN IF SC LEQ "9" THEN
        BEGIN TALLY:=TALLY+1; DS:=CHR END ELSE JUMP OUT ELSE JUMP OUT);
        Y:=TALLY;
        SI:=LOC X;
        DI:=N;
        DS:=Y OCT;
        XIT: END GETCOMMON;
        INTEGER I;
        ALPHA N1,N2;
        COMNT:=BADCOMMENT:=FALSE;
        IF RELOAD THEN
        FOR I:=0 STEP 1 UNTIL 63 DO UNLOCK(M[A[0],ROW,*]);
```

```
        00807530 T  0048:0000:1
        00807540 T  0048:0001:2
        00807550 T  0048:0005:2
        00807560 T  0048:0008:0
        00807570 T  0048:0009:3
        00807575 T  0048:0011:2
        00807580 T  0048:0012:1
        00807890 T  0048:0017:3
        00807899 T  0048:0017:3
        00807900 T  0048:0018:0
  48 IS   21 LONG, NEXT SEG   10
        00808000 T  0010:0214:0
        00809000 T  0010:0214:0
START OF SEGMENT **********   49
  49 IS   11 LONG, NEXT SEG   10
        00810000 T  0010:0214:0
        00811000 T  0010:0214:0
        00812005 T  0010:0214:0
START OF SEGMENT **********   50
        00812010 T  0050:0000:0
        00813000 T  0050:0000:0
        00814000 T  0050:0000:0
START OF SEGMENT **********   51
        00814100 T  0051:0000:0
        00815000 T  0051:0000:0
        00816000 T  0051:0000:0
  51 IS   41 LONG, NEXT SEG   50
        00817000 T  0050:0000:0
        00818000 T  0050:0001:3
        00819000 T  0050:0001:3
        00820000 T  0050:0003:2
        00821000 T  0050:0003:3
        00822000 T  0050:0004:0
        00822100 T  0050:0004:1
        00822110 T  0050:0004:1
        00822120 T  0050:0005:2
        00822130 T  0050:0006:1
        00822140 T  0050:0008:1
        00822150 T  0050:0009:3
        00822155 T  0050:0011:2
        00822160 T  0050:0013:2
        00822170 T  0050:0013:2
        00822175 T  0050:0016:0
        00822180 T  0050:0017:2
        00822190 T  0050:0017:3
        00822200 T  0050:0020:0
        00822210 T  0050:0021:3
        00822220 T  0050:0025:2
        00822230 T  0050:0025:2
        00822240 T  0050:0025:3
        00822250 T  0050:0025:3
        00822260 T  0050:0026:0
        00823000 T  0050:0027:2
        00823100 T  0050:0027:2
        00824000 T  0050:0027:2
        00824100 T  0050:0029:2
        00824110 T  0050:0029:3
```

```
    IF MULTI THEN                                                                   00824200 T  0050:0036:1
    READ REVERSE(MDUMP,20,COMMT[*])[BADTM:IGPAR];                                   00824210 T  0050:0036:1
    IF FALSE THEN IGPAR:BADCOMMENT:=TRUE;                                           00824220 T  0050:0043:3
FOR I:=0 STEP 1 UNTIL 63 DO                                                         00825000 T  0050:0045:3
    BEGIN                                                                           00826000 T  0050:0047:2
        IF MULTI THEN READ REVERSE(MDUMP,533,A[*])[BADTM:PAR] ELSE                  00826990 T  0050:0047:2
        READ(MDUMP,533,A[*])[BADTM:PAR];%                                           00827000 T  0050:0054:0
        IF FLGBIT(A) THEN GO FLAG ELSE                                              00828000 T  0050:0061:2
        IF MODON[A[0],[33:3]]:=NOT BOOLEAN(A[0],[1:1]) THEN                         00829000 T  0050:0062:1
        BEGIN                                                                       00830000 T  0050:0066:0
            MOVER(A[1],M[A[0],ROW,0]);                                              00832000 T  0050:0066:1
            LOCK(M[A[0],ROW,*]);                                                    00832100 T  0050:0069:3
        END ELSE IF A[0],[3:5]=0 THEN I:=I+7; %                                     00833000 T  0050:0073:2
        COMMENT ONE BLOCK IS WRITTEN PER ABSENT MOD;                                00833010 T  0050:0076:1
    END;                                                                            00834000 T  0050:0076:1
    TDMFID:=MDUMP.MFID;                                                             00834050 T  0050:0079:2
    TDFID:=MDUMP.FID;                                                               00834060 T  0050:0082:1
    FILETYPE:=IF MDUMP.TYPE=2 THEN "TAPE" ELSE "DISK";                              00834070 T  0050:0086:1
    IF NOT MULTI THEN                                                               00834075 T  0050:0092:0
    IF FILETYPE="TAPE" AND TDMFID NEQ 0 THEN                                        00834080 T  0050:0092:1
    IF TDFID.[30:18]=1 THEN ELSE MULTI:=TRUE;                                       00834085 T  0050:0095:3
    IF REPEATING THEN IF TDFID.[30:18]=1 THEN CLOSE(MDUMP)                          00834095 T  0050:0104:1
    ELSE CLOSE(MDUMP,*) ELSE                                                        00834097 T  0050:0108:1
    IF (AUXTYPE:=A[0],[3:5])=0 THEN                                                 00834100 T  0050:0111:3
    IF FILETYPE="TAPE" THEN                                                         00834200 T  0050:0114:0
    READ(MDUMP,20,COMMT[*])[EOT:IGNOREPAR] ELSE % DISK                              00835000 T  0050:0115:3
    MOVE(A[513],COMMT,20);                                                          00835100 T  0050:0122:0
     IF FALSE THEN IGNOREPAR:BADCOMMENT+TRUE;                                       00836000 T  0050:0125:3
    COMNT := TRUE;                                                                  00837000 T  0050:0127:3
     GETCOMMON(COMMT,N1);                                                           00837050 T  0050:0128:1
     IF N1 GEQ 0 THEN COMMON:=BOOLEAN(N1); N1:=0;                                   00837055 T  0050:0130:0
      IF MULTI THEN NOMO:=TRUE ELSE                                                 00837095 T  0050:0133:2
      READ(MDUMP[NO])[EOT:PAR];                                                     00837100 T  0050:0134:1
      IF FALSE THEN                                                                 00837200 T  0050:0141:2
      BEGIN                                                                         00837300 T  0050:0141:2
      EOT: CLOSE(MDUMP);                                                            00838000 T  0050:0141:3
      NOMO:=TRUE;                                                                   00838010 T  0050:0143:3
      END;                                                                          00838100 T  0050:0144:1
      IF COMMON OR RELOAD THEN                                                      00839000 T  0050:0144:1
      ELSE                                                                          00840000 T  0050:0145:2
      BEGIN                                                                         00841000 T  0050:0145:3
      IF PRT32 THEN READARRAY(30,SEGZERO[*],0) ELSE                                 00843000 T  0050:0146:0
       SPACE(DISK,1);                                                               00844000 T  0050:0148:1
       READARRAY(NAMESIZE,NAME[*],PRTBASE);                                         00845000 T  0050:0152:1
       READARRAY(NAMSSIZE,NAMS[*],0);                                               00846000 T  0050:0154:0
      READARRAY(INAMESIZE,INAME[*],0);                                              00847000 T  0050:0156:0
       READARRAY(INAMSSIZE,INAMS[*],0);                                             00848000 T  0050:0157:3
      CLOSE(DISK);                                                                  00849000 T  0050:0159:3
      N1:=DISK.MFID; N2:=DISK.FID;                                                  00849100 T  0050:0161:2
      WRITE(PRTNAME[*],PRTFILE,N1.[6:6],N1,N2,[6:6],N2);                            00849200 T  0050:0167:3
      END;                                                                          00850000 T  0050:0185:2
          IF RELOAD THEN ELSE                                                       00850100 T  0050:0185:2
          BEGIN                                                                     00850200 T  0050:0186:0
      MAXMOD:=7; %                                                                  00851000 T  0050:0186:1
      WHILE NOT MODON[MAXMOD] DO MAXMOD:=MAXMOD-1;                                  00852000 T  0050:0187:2
      MAXCOR:=4096*(MAXMOD+1)-1; %                                                  00853000 T  0050:0191:2
IF COMMON THEN ELSE                                                                 00854000 T  0050:0193:2
```

```
        BEGIN
            PRTMAX:=PRTBASE+NAMESIZE-1;
            INTMAX:=INAMESIZE-1;
            FOR I:=PRTBASE STEP 1 UNTIL PRTMAX DO NSNAME[I]:=
            NSNAME[I]+PRTBASE;
            FOR I:=0 STEP 1 UNTIL INTMAX DO ISNAME[I]:=
            ISNAME[I]+PRTBASE;
            FIXDEFINES;
            SETUPXNAMEANDXNAMS;
            IF MYSTACKADR LSS 0 THEN % LETS USE IT
            IF CONTROLDESC(7,I) THEN % CHECK FOR MSCW IN CELL 7
            IF I.CF=0 AND I:=I.FF GEQ 0 THEN % ASSUME VALID
            MYSTACKADR:=I+30; % MAY NEED MORE THAN 30
        END;
                END;
         IF FALSE THEN BEGIN
         BADTM:WRITE(SPO,BADEOF,I); GO EOPROG;
         PAR:   WRITE(SPO,PARERR,I+1); GO EOPROG;
         FLAG:  WRITE(SPO,FLAGERR,I+1); GO EOPROG;  END;
        END LOAD;

    PROCEDURE CHECKPRTFILE;%
    %SIGNALS USER IF MCP/PRT FILE *MIGHT* BE THE WRONG ONE
    BEGIN
    REAL LOC,TYP,I;

    REAL MYTYPE;
    LABEL BAD,MAYBE;
    DEFINE CHECKFORLABELDESCRIPTOR=IF LOC LSS 0 THEN GO BAD ELSE
            IF DESCRIPTOR(LOC,I,TYP) AND TYP=MYTYPE THEN ELSE GO MAYBE#;
    DEFINE CHECKFORPROGRAMDESCRIPTOR=CHECKFORLABELDESCRIPTOR#;
    FORMAT BADPRT("---YOUR MCP/PRT FILE IS WRONG..."),

    INCOMPAT("---MCP/PRT FILE NOT COMPATIBLE WITH MCPS PRT IN MEMORY...");

    DEFINE ERRMESS(ERRMESS1)=BEGIN WRITE(SPO,ERRMESS1);
                  WRITE(P[PAGE],ERRMESS1) END#;%
    MYTYPE:="76";
    IF (LOC:=NOTHINGTODO) NEQ 0 THEN
    CHECKFORLABELDESCRIPTOR;
    IF (LOC:=STACKOVERFLOW) NEQ 0 THEN
    CHECKFORLABELDESCRIPTOR;
    IF (LOC:=RETURN) NEQ 0 THEN
    CHECKFORLABELDESCRIPTOR;
    MYTYPE:="75";
    IF (LOC:=DIRECTORYBUILDER)=0 THEN LOC:="1;
    CHECKFORPROGRAMDESCRIPTOR;
    IF FALSE THEN MAYBE:BEGIN ERRMESS(INCOMPAT); PRTCODE:=1 END;
    IF FALSE THEN BAD:BEGIN ERRMESS(BADPRT); PRTCODE:=2 END;
    END OF CHECKPRTFILE;

    PROCEDURE DUMPMCPSPRT;
    BEGIN
    FORMAT HDR(X42,"D C M C P S  P R T"/

                X42,"= - - - - - - - - -"///,
                2("PRT",X5,"CONTENTS",X16,"NAME",X28)/),
```

```
00855000 T  0050:0194:0
00856000 T  0050:0196:0
00857000 T  0050:0197:3
00858000 T  0050:0199:2
00859000 T  0050:0201:3
00860000 T  0050:0207:2
00861000 T  0050:0209:2
00862000 T  0050:0214:0
00863000 T  0050:0214:1
00863100 T  0050:0215:2
00863110 T  0050:0215:3
00863120 T  0050:0217:2
00863130 T  0050:0221:2
00864000 T  0050:0222:1
00864100 T  0050:0222:1
00865000 T  0050:0222:1
00866000 T  0050:0223:3
00867000 T  0050:0233:3
00868000 T  0050:0245:3
00869000 T  0050:0257:3
  50 IS  262 LONG, NEXT SEG   10
00870000 T  0010:0214:0
00871000 T  0010:0214:0
00872000 T  0010:0214:0
00873000 T  0010:0214:0
START OF SEGMENT **********   52
00874000 T  0052:0000:0
00875000 T  0052:0000:0
00876000 T  0052:0000:0
00877000 T  0052:0000:0
00878000 T  0052:0000:0
00879000 T  0052:0000:0
START OF SEGMENT **********   53
00880000 T  0053:0000:0
  53 IS   22 LONG, NEXT SEG   52
00881000 T  0052:0000:0
00882000 T  0052:0000:0
00883000 T  0052:0000:0
00884000 T  0052:0000:1
00885000 T  0052:0002:0
00886000 T  0052:0006:1
00887000 T  0052:0008:0
00888000 T  0052:0012:1
00889000 T  0052:0014:0
00890000 T  0052:0018:1
00891000 T  0052:0019:3
00892000 T  0052:0022:1
00893000 T  0052:0026:1
00894000 T  0052:0034:1
00895000 T  0052:0042:1
  52 IS   45 LONG, NEXT SEG   10
00896000 T  0010:0214:0
00897000 T  0010:0214:0
00898000 T  0010:0214:0
START OF SEGMENT **********   54
START OF SEGMENT **********   55
00899000 T  0055:0000:0
00900000 T  0055:0000:0
```

```
        DASHES(120("-")),
        F(X8,"MEMORY MODS: ",8I1),
        PRTITEM(2(A5," = ",0,X1,0,X39));

INTEGER LOC,N;
LABEL EXIT;
IF DUMPPROCESSPOOLONLY THEN DONTDUMPRT:=TRUE;
IF NOT COMMON THEN
BEGIN
CHECKPRTFILE;
IF NOT DONTDUMPRT THEN BEGIN DATIME; WRITE(P[DBL]) END;
PRINTMCPOPTIONS;
IF DONTDUMPRT THEN ELSE
BEGIN
   WRITE(P[DBL],HDR);
   FOR LOC:=ACTUALPRTBASE STEP 1 UNTIL PRTBASE DO
   BEGIN
       WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),
          OCTAL(LOHALF(LOC)),OCTAL(N:=XSNAME[LOC]),
          OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));
       MOVE(XNAMS[XNAME[LOC],CF],LINE[4],XNAME[LOC],FF);
       MOVE(XNAMS[XNAME[N],CF],LINE[12],XNAME[N],FF);
       WRITE(P,15,LINE[*]);
       IF DONTDUMPRT THEN GO EXIT;
   END;
   WRITE(P,DASHES);
   FOR LOC:=PRTBASE+1 STEP 1 UNTIL PRTMAX DO
   BEGIN
       WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),
          OCTAL(LOHALF(LOC)),OCTAL(N:=NSNAME[LOC]),
          OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));
       MOVE(NAMS[NAME[LOC],CF],LINE[4],NAME[LOC],FF);
       MOVE(NAMS[NAME[N],CF],LINE[12],NAME[N],FF);
       WRITE(P,15,LINE[*]);
       IF DONTDUMPRT THEN GO EXIT;
   END;
EXIT:
   NEXTPAGE;
END;
END;
WRITE(P[DBL],F,FOR N:=0 STEP 1 UNTIL 7 DO
      [IF MODON[N] THEN N ELSE 10]);
DATIME;
IF DUMPPROCESSPOOLONLY THEN ELSE
IF NOT COMMON THEN
BEGIN
    MCPENTRIES;
    GETPRTENTRIES; NEXTPAGE;
END;
PRTOK := PDATADESC(PRT,PRTO);
END OF DUMP OF MCPS PRT;

    PROCEDURE CHECKMEMORYLINKS;
    BEGIN %
        BOOLEAN ZEROK; REAL V0; %

        BOOLEAN MSTARTOK; REAL VMSTART; %
```

```
                                00901000 T  0055:0000:0
                                00902000 T  0055:0000:0
                                00903000 T  0055:0000:0
                          55 IS   50 LONG, NEXT SEG   54
                                00904000 T  0054:0000:0
                                00904100 T  0054:0000:0
                                00904500 T  0054:0000:0
                                00905000 T  0054:0003:2
                                00906000 T  0054:0003:3
                                00907000 T  0054:0004:0
                                00908000 T  0054:0004:1
                                00909000 T  0054:0010:1
                                00910000 T  0054:0011:2
                                00911000 T  0054:0012:0
                                00912000 T  0054:0012:1
                                00913000 T  0054:0015:3
                                00914000 T  0054:0017:2
                                00915000 T  0054:0017:2
                                00916000 T  0054:0027:2
                                00917000 T  0054:0034:0
                                00918000 T  0054:0043:2
                                00919000 T  0054:0047:3
                                00920000 T  0054:0052:0
                                00920100 T  0054:0056:1
                                00921000 T  0054:0058:0
                                00922000 T  0054:0060:0
                                00923000 T  0054:0063:2
                                00924000 T  0054:0067:3
                                00925000 T  0054:0067:3
                                00926000 T  0054:0078:0
                                00927000 T  0054:0085:2
                                00928000 T  0054:0094:0
                                00929000 T  0054:0098:1
                                00930000 T  0054:0103:2
                                00930100 T  0054:0107:3
                                00931000 T  0054:0109:2
                                00931100 T  0054:0109:3
                                00932000 T  0054:0110:0
                                00933000 T  0054:0114:0
                                00934000 T  0054:0114:0
                                00935000 T  0054:0114:0
                                00936000 T  0054:0119:2
                                00937000 T  0054:0128:0
                                00937500 T  0054:0128:1
                                00938000 T  0054:0130:0
                                00939000 T  0054:0131:2
                                00940000 T  0054:0131:3
                                00941000 T  0054:0132:0
                                00942000 T  0054:0136:1
                          %103- 00942100 C  0054:0136:1
                                00943000 T  0054:0138:0
                        54 IS  141 LONG, NEXT SEG   10
                                00944000 T  0010:0214:0
                                00945000 T  0010:0214:0
                                00946000 T  0010:0214:0
                      START OF SEGMENT **********   56
                                00947000 T  0056:0000:0
```

```
 BOOLEAN NOWRAPAROUND;                                                                                                00947100 T  0056:0000:0
 REAL LINK,VLINK,PREVLINK;                                                                                            00948000 T  0056:0000:0
 INTEGER AVAILN,AVAILT,AVAILM;                                                                                        00949000 T  0056:0000:0
 BOOLEAN PROCEDURE LINKOK(WORD); %                                                                                    00950000 T  0056:0000:0
     VALUE WORD; %                                                                                                    00951000 T  0056:0000:0
     REAL WORD; %                                                                                                     00952000 T  0056:0000:0
 LINKOK:=WORD.[3:6]≤TYPMAX AND %                                                                                      00953000 T  0056:0000:0
     WORD.[9:6]≤MIXMAX      AND                                                                                       00954000 T  0056:0001:2
     WORD.[15:3]=0; %                                                                                                 00955000 T  0056:0002:1
FORMAT RANGE(X8,"PRIMARY LINKS FROM ",A5," TO ",A5," ARE OK"),                                                        00956000 T  0056:0007:2
                                                                                                  START OF SEGMENT **********   57
       FCORE(A5," = ",2(O,X1),X6,"CORE",X8,"FACTOR = ",F4.2,                                                          00956100 T  0057:0007:2
       ", MAX CORE = ",A5,"(",I5,")","USING ",A5,"(",I5,")"),                                                         00956110 T  0057:0007:2
        BAD(X8,A6," LINK AT ",A5," IS NOT OK"),                                                                       00957000 T  0057:0007:2
 BADPRIMARY(X8,"PRIMARY LINK AT ",A5," IS NOT OK"),                                                                   00958000 T  0057:0007:2
        AVLBAD(X8,"AVAILABLE STORAGE TOTALS DO NOT CROSS CHECK"),                                                     00959000 T  0057:0007:2
        AVL(X8,"AVAILABLE LINKS ARE OK, TOTALING",                                                                    00960000 T  0057:0007:2
            I4,"(DECIMAL) AREAS OF ",A5,"(",I5,                                                                       00961000 T  0057:0007:2
            ") WORDS UP TO ",A5,"(",I5,") WORDS EACH");                                                               00962000 T  0057:0007:2
                                                                                                  57 IS   99 LONG, NEXT SEG   56
 INTEGER N,T,M; %                                                                                                     00963000 T  0056:0007:2
 DISPLAY(0,FALSE);                                                                                                    00964000 T  0056:0007:2
 IF MSTART>0 THEN DISPLAY(MSTART,FALSE);                                                                              00965000 T  0056:0009:2
 IF MEND>0 THEN DISPLAY(MEND,FALSE);                                                                                  00966000 T  0056:0012:0
 DISPLAY(AVAIL,FALSE);                                                                                                00967000 T  0056:0015:2
 IF OPERAND(CORE,T) AND T.[1:3]=0 THEN                                                                                00967100 T  0056:0016:1
 BEGIN                                                                                                                00967110 T  0056:0019:3
     WRITE(LINE[*],FCORE,OCTAL(CORE),OCTAL(HIHALF(CORE)),                                                             00967120 T  0056:0020:0
          OCTAL(LOHALF(CORE)),T.[4:14]/100,                                                                           00967130 T  0056:0030:1
          OCTAL((N:=T.CF×64).CF),N,                                                                                   00967140 T  0056:0036:1
          OCTAL((N:=T.FF×64).CF),N);                                                                                  00967150 T  0056:0042:1
     WRITE(P[DBL],15,LINE[*]);                                                                                        00967160 T  0056:0051:2
 END ELSE                                                                                                             00967170 T  0056:0055:3
 DISPLAY(CORE,FALSE);                                                                                                 00967180 T  0056:0055:3
 ZEROK:=OPERAND(0,V0)          AND                                                                                    00968000 T  0056:0057:3
     V0.[1:2]=1                AND                                                                                    00969000 T  0056:0058:1
V0.[03:15]=0             AND                                                                                         00970000 T  0056:0060:0
     V0.FF=(MAXLNK:=MAXCOR-2) AND                                                                                     00971000 T  0056:0061:3
     V0.CF GTR 1023 AND V0.CF LSS 4096 AND                                                                            00972000 T  0056:0064:0
     OPERAND(V0.CF,T) AND LINKOK(T) AND T.FF=0;                                                                       00972400 T  0056:0067:2
 IF MSTART=0 THEN MSTARTOK:=FALSE ELSE                                                                                00972500 T  0056:0071:3
 MSTARTOK:=OPERAND(MSTART,VMSTART) AND                                                                                00973000 T  0056:0074:0
     VMSTART.[1:35]=0          AND %                                                                                  00974000 T  0056:0077:2
     VMSTART GTR 1023 AND VMSTART LSS 4096;                                                                           00975000 T  0056:0078:1
 IF LNKSOK:=MSTARTOK OR ZEROK THEN                                                                                    00976000 T  0056:0081:2
 MINLNK:=LINK:=IF MSTARTOK AND ZEROK THEN MAX(V0.CF,VMSTART)                                                          00976100 T  0056:0082:1
 ELSE IF MSTARTOK THEN VMSTART ELSE                                                                                   00976200 T  0056:0085:2
 IF ZEROK THEN V0.CF ELSE 0;                                                                                          00976300 T  0056:0089:2
 AVALNKOK:=TRUE;                                                                                                      00978000 T  0056:0092:1
  N:=T:=0;                                                                                                            00978100 T  0056:0093:2
 WHILE LNKSOK AND MAXLNK>PREVLINK DO %                                                                                00979000 T  0056:0094:1
 IF LNKSOK:=(OPERAND(LINK,VLINK) AND                                                                                  00980000 T  0056:0096:1
              LINKOK(VLINK)       AND %                                                                               00981000 T  0056:0097:3
              VLINK.FF=PREVLINK   AND %                                                                               00982000 T  0056:0098:1
             (IF LINK=MAXLNK                                                                                          00983000 T  0056:0100:0
              THEN VLINK.CF=0 %                                                                                       00984000 T  0056:0100:1
              ELSE VLINK.CF>LINK))THEN                                                                                00985000 T  0056:0102:0
```

```
        BEGIN %                                                                 00986000 T  0056:0105:2
            PREVLINK:=LINK; %                                                   00987000 T  0056:0105:3
            LINK:=VLINK.CF; %                                                   00988000 T  0056:0106:1
            IF AVALNKOK THEN %                                                  00989000 T  0056:0107:3
            IF BOOLEAN(VLINK,[1:1]) THEN %                                      00990000 T  0056:0108:0
            IF AVALNKOK:=MAXLNK>PREVLINK THEN %                                 00991000 T  0056:0109:2
            IF AVALNKOK:=(OPERAND(PREVLINK+1,VLINK) AND %                       00992000 T  0056:0111:2
                          VLINK,[1:17]=0) THEN %                                00993000 T  0056:0113:2
            BEGIN %                                                             00994000 T  0056:0115:2
                AVAILN:=AVAILN+1; %                                             00995000 T  0056:0115:3
                AVAILT:=AVAILT+VLINK.FF; %                                      00996000 T  0056:0116:1
                AVAILM:=MAX(AVAILM,VLINK,FF); %                                 00997000 T  0056:0118:1
            END; %                                                              00998000 T  0056:0122:0
        END; %                                                                  00999000 T  0056:0122:0
    IF NOT ZEROK THEN WRITE(P,BADPRIMARY,0);                                    00999100 T  0056:0124:0
    IF LNKSOK THEN WRITE(P,RANGE,IF ZEROK THEN 0 ELSE OCTAL(MINLNK),            01000000 T  0056:0133:2
    OCTAL(MAXLNK))                ELSE                                          01001000 T  0056:0141:3
        BEGIN %                                                                 01002000 T  0056:0147:2
            IF SOMOKF:=PREVLINK>MINLNK THEN                                     01003000 T  0056:0147:3
            BEGIN %                                                             01004000 T  0056:0149:2
            WRITE(P,RANGE,IF ZEROK THEN 0 ELSE OCTAL(MINLNK),                   01005000 T  0056:0149:3
                  OCTAL(PREVLINK));                                             01006000 T  0056:0157:3
            WRITE(P,BADPRIMARY,OCTAL(LINK));%                                   01007000 T  0056:0163:2
            END; %                                                              01008000 T  0056:0173:2
            MINBAD:=LINK;                                                       01009000 T  0056:0173:2
            PREVLINK:=0; %                                                      01010000 T  0056:0174:0
            LINK:=MAXLNK; %                                                     01011000 T  0056:0174:1
            WHILE OPERAND(LINK,VLINK) AND %                                     01012000 T  0056:0175:3
                LINKOK(VLINK)           AND                                     01013000 T  0056:0177:2
                VLINK.CF=PREVLINK       AND %                                   01014000 T  0056:0178:0
                LINK>VLINK.FF           DO %                                    01015000 T  0056:0179:3
            BEGIN %                                                             01016000 T  0056:0181:3
                PREVLINK:=LINK; %                                               01017000 T  0056:0181:3
                LINK:=VLINK.FF; %                                               01018000 T  0056:0182:0
            END; %                                                              01019000 T  0056:0183:3
            IF PREVLINK=0 THEN MAXBAD:=MAXCOR+1 ELSE %                          01020000 T  0056:0184:0
            WRITE(P,RANGE,OCTAL(MAXLNK),OCTAL(MAXBAD:=PREVLINK)); %             01021000 T  0056:0186:1
            WRITE(P,BADPRIMARY,OCTAL(LINK));%                                   01022000 T  0056:0199:2
            SOMOKB:=PREVLINK>0;                                                 01023000 T  0056:0209:2
    END; %                                                                      01024000 T  0056:0210:1
    NOWRAPAROUND:=TRUE;                                                         01025000 T  0056:0210:1
    IF AVALNKOK:=(OPERAND(AVAIL,PREVLINK) AND                                   01026000 T  0056:0211:2
        PREVLINK=MAXLNK+1         AND %                                         01027000 T  0056:0212:1
        OPERAND(PREVLINK,VLINK) AND %                                           01028000 T  0056:0214:0
        VLINK,[1:17]=0            AND %                                         01029000 T  0056:0215:2
        VLINK.FF=32767)           THEN %                                        01030000 T  0056:0216:1
    BEGIN %                                                                     01031000 T  0056:0218:1
        WHILE AVALNKOK AND %                                                    01032000 T  0056:0219:2
              NOWRAPAROUND AND                                                  01032100 T  0056:0220:0
              (IF LNKSOK THEN AVAILN GTR N ELSE TRUE) AND                       01033000 T  0056:0220:1
              (IF LNKSOK THEN AVAILT GTR T ELSE TRUE) AND                       01034000 T  0056:0223:2
        (IF LNKSOK THEN AVAILM GEQ M  ELSE TRUE) DO                             01035000 T  0056:0225:3
        IF NOWRAPAROUND:=(MAXCOR GTR LINK:=VLINK.CF                             01036000 T  0056:0228:1
                          AND LINK GTR 0) THEN                                  01036100 T  0056:0229:2
        IF AVALNKOK:=(OPERAND(LINK-1,VLINK) AND %                               01037000 T  0056:0232:0
            LINKOK(VLINK)           AND %                                       01038000 T  0056:0234:0
            BOOLEAN(VLINK,[1:1])  AND %                                         01039000 T  0056:0235:2
```

```
                OPERAND(LINK+1,VLINK) AND %                                    01040000 T  0056:0236:0
                VLINK=PREVLINK        AND %                                    01041000 T  0056:0237:3
                OPERAND(LINK,VLINK)   AND %                                    01042000 T  0056:0238:1
                VLINK.[1:17]=0        AND %                                    01043000 T  0056:0240:0
                VLINK.CF#LINK)        THEN %                                   01044000 T  0056:0241:3
            BEGIN %                                                            01045000 T  0056:0243:3
                N:=N+1;                                                        01046000 T  0056:0244:0
                T:=T+VLINK.FF; %                                               01047000 T  0056:0245:2
                M:=MAX(M,VLINK.FF); %                                          01048000 T  0056:0247:2
                PREVLINK:=LINK; %                                              01049000 T  0056:0250:1
            END; %                                                             01050000 T  0056:0251:2
            IF NOT AVALNKOK THEN WRITE(P,BAD,"AVALBL",OCTAL(LINK)) ELSE        01051000 T  0056:0253:2
            IF LNKSOK AND                                                      01051100 T  0056:0266:0
              NOT(AVALNKOK:=                                                   01052000 T  0056:0267:2
                LNKSOK AND                                                     01052100 T  0056:0267:2
                AVAILN=N AND %                                                 01053000 T  0056:0267:2
                AVAILT=T AND %                                                 01054000 T  0056:0268:0
                AVAILM=M AND %                                                 01055000 T  0056:0269:2
                VLINK.CF=MAXLNK+1) THEN WRITE(P,AVLBAD) ELSE %                 01056000 T  0056:0270:0
            WRITE(P,AVL,N,OCTAL(T),T,OCTAL(M),M);                              01057000 T  0056:0276:1
          TABLESLOC:=IF LNKSOK AND OPERAND(MAXLNK,T)                           01058000 T  0056:0293:2
                       THEN T.FF ELSE -1;                                      01059000 T  0056:0294:0
        END ELSE WRITE(P,BAD,"AVALBL",OCTAL(MAXLNK+1));                        01060000 T  0056:0297:3
        WRITE(P[DBL])); %                                                      01061000 T  0056:0311:2
END; %                                                                         01062000 T  0056:0315:2
                                                                    56 IS  321 LONG, NEXT SEG   10

ARRAY MIXSTK[0:43];                                                            01063000 T  0010:0214:0
PROCEDURE GETSTACKSFROMTHEBED;                                                 01064000 T  0010:0216:1
BEGIN                                                                          01065000 T  0010:0216:1
    INTEGER I;                                                                 01066000 T  0010:0216:1
                                                                    START OF SEGMENT *********   58
    INTEGER MIX;                                                               01067000 T  0058:0000:0
INTEGER M; BOOLEAN B;                                                          01068000 T  0058:0000:0
FORMAT DUPBED(X8,"DUPLICATE BED ENTRY FOR MIX=",I2,                            01069000 T  0058:0000:0
                                                                    START OF SEGMENT *********   59
            " ,CHECK BED AT ADDRESS = ",A5,                                    01070000 T  0059:0000:0
            " (THIS ENTRY IGNORED)");                                          01071000 T  0059:0000:0
                                                                      59 IS   20 LONG, NEXT SEG   58
    REAL V,TYP;                                                                01072000 T  0058:0000:0
    IF OPERAND(JOBNUM,VJOBNUM) THEN                                            01073000 T  0058:0000:0
    IF VJOBNUM.[1:8]=0 AND                                                     01074000 T  0058:0001:2
       VJOBNUM.[47:1]=0 AND                                                    01075000 T  0058:0003:2
       PDATADESC(BED,VBED) AND                                                 01076000 T  0058:0004:1
       VJOBNUM≤VBED.[8:10] THEN                                                01077000 T  0058:0006:0
    BEGIN                                                                      01078000 T  0058:0007:3
        VBED:=VBED.CF;                                                         01079000 T  0058:0008:0
        FOR I:=0 STEP 2 UNTIL VJOBNUM DO                                       01080000 T  0058:0009:2
        IF DESCRIPTOR(VBED+I,V,TYP) THEN                                       01081000 T  0058:0010:0
    IF B:=((MIX:=V.[6:2]&TYP[43:45:3]) GTR 0                                   01082000 T  0058:0011:3
        AND (M:=MIXSTK[MIX]) NEQ 0)    THEN                                    01083000 T  0058:0014:1
WRITE(P,DUPBED,MIX,OCTAL(VBED+I))                                              01084000 T  0058:0017:2
    ELSE                                                                       01085000 T  0058:0024:1
    BEGIN                                                                      01086000 T  0058:0028:0
        BEDSTK:=MAXSTK:=MAXSTK+1;                                              01087000 T  0058:0028:1
            MIXSTK[MIX]:=BEDSTK;                                               01088000 T  0058:0030:1
            STAX[BEDSTK]:=                                                     01089000 T  0058:0031:3
                V,FF&                                                          01090000 T  0058:0032:0
```

```
                    I[8:38:10]&
                    (REAL(MIX>0))[7:47:1];
                END;
            END;
        IF B THEN WRITE(P[DBL]);
        END GETSTACKSFROMTHEBED;

    PROCEDURE DUMPMCPSAVECODE(FROM,TOO); VALUE FROM, TOO; REAL FROM,TOO;
      FORWARD;
        PROCEDURE DUMPMEMORYANDNOTESTACKS(FROM,TOO);
            VALUE FROM,TOO;
            INTEGER FROM,TOO;
        BEGIN
            BOOLEAN BEDDED;

            DEFINE T=LINKTYPE#;
            REAL L;
              FORMAT ITEM(X2,A3," =",2(X4,A5));


              REAL R,ADR,Q,LL; BOOLEAN INTR,QT;
            REAL X;
            REAL ISTACKV;BOOLEAN ISTACKOK,IRSTACK;
            BOOLEAN MCPSAVE,MXNOTO;
            DEFINE MXO=NOT MXNOTO#;
            REAL DCQPTSTACK,DCQPTSTACKV;
            BOOLEAN DCSTACK,DCHIT;
            DEFINE DC=DCQPTSTACK#,DCV=DCQPTSTACKV#;
            INTEGER N,STK;
            REAL MX;
            DEFINE OCTALWORD = 8 OCTADE; DI := DI + 1; 8 OCTADE;
                               DI := DI + 2#;
            STREAM PROCEDURE BUILDHEADER(LINE,FROM,WHICH,TYPE,NAME,MIX,
                                         SIZE,MOM,LINK1,LINK2,LINK3);
              VALUE FROM,WHICH,TYPE,MIX,SIZE,MOM,LINK1,LINK2,LINK3;
            BEGIN
              LOCAL SW; % = 1 IF BUILDING AVAIL AREA HEADER,
              LABEL SAVEAREA, OLAY, AVAIL, MCPCODE, INTCODE;
              LABEL MOVNAM, MOVTYP, MOVSIZ;
              DI := LINE;
              17 (DS := 8 LIT " ");
              DI := LINE;
              SI := LOC FROM; SI := SI + 5; SKIP 3 SB; 5 OCTADE;
              DS := LIT " ";
              CI := CI + WHICH;
                GO TO OLAY; % CASE 0
                GO TO SAVEAREA; % CASE 1
                GO TO AVAIL; % CASE 2
                GO TO MCPCODE; % CASE 3
                GO TO INTCODE; % CASE 4
            SAVEAREA:
              DS := 4 LIT "SAVE";
              GO TO MOVTYP;
            OLAY:
              DS := 4 LIT "OLAY";
            MOVTYP:
              DS := 12 LIT " AREA, TYPE=";
```

```
                01091000 T  0058:0033:2
                01092000 T  0058:0034:0
                01093000 T  0058:0036:0
                01094000 T  0058:0038:0
                01095000 T  0058:0038:0
                01096000 T  0058:0043:2
          58 IS   47 LONG, NEXT SEG   10
                01096500 C  0010:0216:1
%106-           01096600 C  0010:0216:1
                01097000 T  0010:0216:1
                01098000 T  0010:0216:1
                01099000 T  0010:0216:1
                01100000 T  0010:0216:1
                01101000 T  0010:0216:1
       START OF SEGMENT *********   60
                01101100 T  0060:0000:0
                01102000 T  0060:0000:0
                01103000 T  0060:0000:0
       START OF SEGMENT *********   61
         61 IS   10 LONG, NEXT SEG   60
                01104000 T  0060:0000:0
%103-           01105000 P  0060:0000:0
                01106000 T  0060:0000:0
                01107100 T  0060:0000:0
                01107200 T  0060:0000:0
                01108000 T  0060:0000:0
                01109000 T  0060:0000:0
                01110000 T  0060:0000:0
                01112000 T  0060:0000:0
                01113000 T  0060:0000:0
%103-           01115100 C  0060:0000:0
%103-           01115200 C  0060:0000:0
%103-           01115300 C  0060:0000:0
%103-           01115400 C  0060:0000:0
%103-           01115500 C  0060:0000:0
%103-           01115600 C  0060:0000:0
%103-           01115700 C  0060:0000:0
%103-           01115800 C  0060:0000:0
%103-           01115850 C  0060:0000:0
%103-           01115900 C  0060:0000:0
%103-           01116000 P  0060:0000:0
%103-           01116100 C  0060:0002:0
%103-           01116200 C  0060:0002:0
%103-           01116300 C  0060:0005:3
%103-           01116400 C  0060:0006:0
%103-           01116500 C  0060:0006:1
%103-           01116600 C  0060:0007:2
%103-           01116700 C  0060:0007:2
%103-           01116800 C  0060:0007:3
%103-           01116900 C  0060:0007:3
%103-           01117000 P  0060:0008:0
%103-           01117100 C  0060:0008:0
%103-           01117200 C  0060:0008:1
%103-           01117300 C  0060:0009:2
%103-           01117400 C  0060:0009:2
%103-           01117500 C  0060:0009:3
%103-           01117600 C  0060:0010:0
```

```
  SI := LOC TYPE;                                                    %103-    01117700 C  0060:0011:3
  DS := 2 DEC;                                                       %103-    01117800 C  0060:0012:0
  DS := LIT "(";                                                     %103-    01117900 C  0060:0012:0
  SI := NAME;                                                        %103-    01118000 P  0060:0012:1
  16 (IF SC = ARROW THEN JUMP OUT ELSE DS := CHR);                   %103-    01118100 C  0060:0013:2
  DS := 7 LIT "), MIX=";                                             %103-    01118200 C  0060:0015:2
  SI := LOC MIX;                                                     %103-    01118300 C  0060:0016:0
  DS := 2 DEC;                                                       %103-    01118400 C  0060:0016:1
  GO TO MOVSIZ;                                                      %103-    01118500 C  0060:0016:1
AVAIL:                                                               %103-    01118600 C  0060:0017:2
  DS := 14 LIT "AVAILABLE AREA";                                     %103-    01118700 C  0060:0017:2
  TALLY := 1;                                                        %103-    01118800 C  0060:0019:2
  SW := TALLY;                                                       %103-    01118900 C  0060:0019:2
  GO TO MOVSIZ;                                                      %103-    01119000 P  0060:0019:3
MCPCODE:                                                             %103-    01119100 C  0060:0019:3
  DS := 8 LIT "MCP CODE";                                            %103-    01119200 C  0060:0020:0
  GO TO MOVNAM;                                                      %103-    01119300 C  0060:0021:2
INTCODE:                                                             %103-    01119400 C  0060:0021:3
  DS := 9 LIT "INTRINSIC";                                           %103-    01119500 C  0060:0022:0
MOVNAM:                                                              %103-    01119600 C  0060:0023:3
  DS := 3 LIT " = ";                                                 %103-    01119700 C  0060:0024:0
  SI := NAME;                                                        %103-    01119800 C  0060:0024:1
  TYPE ( 8 ( IF SC = " " THEN JUMP OUT 2 TO MOVSIZ                   %103-    01119900 C  0060:0025:2
                                        ELSE DS := CHR ) );          %103-    01119950 C  0060:0027:3
MOVSIZ:                                                              %103-    01120000 P  0060:0028:1
  DS := 7 LIT ", SIZE=";                                             %103-    01120100 C  0060:0029:2
  SI := LOC SIZE;                                                    %103-    01120200 C  0060:0030:0
  SI := SI + 5;                                                      %103-    01120300 C  0060:0030:1
  SKIP 3 SB;                                                         %103-    01120400 C  0060:0030:1
  5 OCTADE;                                                          %103-    01120500 C  0060:0031:2
  DS := LIT "(";                                                     %103-    01120600 C  0060:0033:3
  SI := LOC SIZE;                                                    %103-    01120700 C  0060:0034:0
  DS := 5 DEC;                                                       %103-    01120800 C  0060:0034:1
  DS := LIT ")";                                                     %103-    01120850 C  0060:0034:1
  MOM (DS := 9 LIT ", MOM AT ";                                      %103-    01120900 C  0060:0035:2
       SI := LOC LINK2;                                              %103-    01120950 C  0060:0037:3
       SI := SI + 5;                                                 %103-    01121000 P  0060:0038:0
       SKIP 3 SB;                                                    %103-    01121050 C  0060:0038:0
       5 OCTADE);                                                    %103-    01121100 C  0060:0038:1
  DS := 9 LIT "  LINKS: ";                                          %103-    01121150 C  0060:0041:3
  SI := LOC LINK1;                                                   %103-    01121300 C  0060:0043:2
  OCTALWORD;                                                         %103-    01121400 C  0060:0043:2
  SI := LOC LINK2;                                                   %103-    01121500 C  0060:0049:2
  OCTALWORD;                                                         %103-    01121600 C  0060:0049:3
  SW (SI := LOC LINK3;                                               %103-    01121700 C  0060:0055:3
      OCTALWORD);                                                    %103-    01121800 C  0060:0056:1
 END OF BUILDHEADER;                                                 %103-    01121900 C  0060:0063:2
NOTPRINTCALL:=TRUE;                                                           01122100 T  0060:0063:2
IF DATACOM THEN                                                               01123000 T  0060:0064:1
BEGIN                                                                         01124000 T  0060:0065:3
     DCSTACK:=DC GTR 129 AND OPERAND(DC,DCV)                                  01125000 T  0060:0066:0
                 AND DCV NEQ 0 AND DCV.[1:32]=0                               01126000 T  0060:0067:2
                 AND (DCV:=DCV.CF) GTR 1023 AND                               01127000 T  0060:0069:3
                 OPERAND(DCV,R) AND R=0 AND                                   01128000 T  0060:0072:1
                 OPERAND(DCV-1,Q) AND Q.[9:6]=0 AND                           01129000 T  0060:0074:1
           (Q.[3:6]=0 OR Q.[3:6]=12) AND Q.[2:1]=1;                           01130000 T  0060:0078:0
END OF LOCATING DCQPTSTACK AFTER MANY CHECKS;                                 01131000 T  0060:0083:2
```

```
 ISTACKOK:=PDATADESC(ISTACK,ISTACKV) AND                                  01132000 T  0060:0083:2
           ISTACKV,[8:10] GTR 0    AND                                    01133000 T  0060:0084:0
           (ISTACKV:=ISTACKV,CF) GTR 0;                                   01134000 T  0060:0085:3
 INTR:=PDATADESC(INTRNSC,ADR) AND ADR,FF=0;                               01135000 T  0060:0088:0
   ADR := ADR,CF;                                                %103-    01136000 P  0060:0091:3
 WHILE FROM<TOO DO                                                        01143000 T  0060:0092:1
 BEGIN                                                                    01144000 T  0060:0094:0
 IRSTACK:=FALSE;                                                          01145000 T  0060:0094:0
DCHIT:=FALSE;                                                             01146000 T  0060:0095:2
     N:=(L:=M[FROM,ROW,FROM,COL]),CF;                                     01149000 T  0060:0095:3
     IF T:=L,[3:6] GTR TYPMAX THEN L,[3:6]:=T:=TYPMAX+1;% BAD TYP         01149100 T  0060:0099:3
     MXNOTO:=(MX:=L,[9:6]) NEQ 0;                                         01149200 T  0060:0104:1
     IF B(L,[1:1]) THEN % AVAILABLE AREA                           %103-  01151000 P  0060:0106:1
       IF NOT NODUMP THEN                                          %103-  01151100 C  0060:0107:3
         BEGIN                                                     %103-  01151200 C  0060:0109:2
           QT := OPERAND(FROM+2,Q); % GET LINK 3,                  %103-  01151300 C  0060:0109:3
           BUILDHEADER(LINE[*],FROM,2,0,X,0,IF OPERAND(FROM+1,X)           01151400 C  0060:0111:3
                     AND NOT B(X,[1:17]) THEN X,FF ELSE 0,%103-            01151500 C  0060:0115:2
                     0, L, X, Q);                                %103-  01151600 C  0060:0118:1
           WRITE(P[DBL],17,LINE[*]);                             %103-  01151700 C  0060:0120:0
           AVALNK := TRUE;                                       %103-  01151800 C  0060:0124:0
           IF DUMPAVAIL OR NEEDCHECKAVAILNKS THEN                %103-  01151900 C  0060:0125:2
             PRINT(FROM+3,N);                                    %103-  01152000 P  0060:0126:0
           AVALNK := FALSE;                                      %103-  01152100 C  0060:0130:1
         END                                                     %103-  01152200 C  0060:0131:2
       ELSE % NO DUMP                                            %103-  01152300 C  0060:0131:2
         ELSE % IN USE AREA,                                     %103-  01152400 C  0060:0131:2
         BEGIN                                                           01162000 T  0060:0131:3
QT:=OPERAND(FROM+1,Q);                                                   01163000 T  0060:0132:0
 MCPSAVE:=QT AND Q NEQ 0 AND L,[2:13]=4096 AND Q,FF NEQ 0                 01164000 T  0060:0134:0
      AND Q,CF GEQ 129 AND Q,CF LEQ PRTMAX;                              01164100 T  0060:0137:3
 DCHIT:=(DCV-1)=FROM;                                                    01164200 T  0060:0142:0
 IRSTACK:= (CONTROLDESC(FROM+2,R) AND                                    01164300 T  0060:0143:3
      R,FF=ISTACKV+1) OR                                                 01164400 T  0060:0145:2
      (CONTROLDESC(FROM+3,R) AND                                         01164500 T  0060:0147:2
      R,FF=ISTACKV+1);                                                   01164600 T  0060:0148:1
 IF NOT NODUMP THEN BEGIN                                      %103-  01164900 P  0060:0151:3
   IF MXNOTO THEN % CHECK FOR STACK OR SEGMENT DICT,            %103-  01166000 P  0060:0153:2
     IF T = 2 THEN % MIGHT BE STACK-PRT,                        %103-  01166100 C  0060:0153:3
       IF OPERAND(FROM+2,X) AND X="BBBBBBBB" AND                %103-  01166200 C  0060:0154:1
          ((CONTROLDESC(FROM+3,X) AND X,FF = 0) OR              %103-  01166300 C  0060:0157:3
           (OPERAND(FROM+3,X) AND X=0 AND CONTROLDESC(FROM+4,X) %103-  01166400 C  0060:0160:1
            AND X,FF = 0)) THEN % IT IS STACK-PRT,              %103-  01166500 C  0060:0164:1
         T := 34                                                %103-  01166600 C  0060:0167:2
       ELSE                                                     %103-  01166700 C  0060:0167:3
     ELSE                                                       %103-  01166800 C  0060:0168:0
       IF T = 1 THEN % MIGHT BE SEGMENT DICT,                   %103-  01166900 C  0060:0171:2
         IF B(L,[2:1]) AND OPERAND(FROM+2,X) AND X,[1:37] = 0   %103-  01167000 P  0060:0172:0
            AND OPERAND(FROM+3,X) AND X,[1:2] LEQ 2 AND         %103-  01167100 C  0060:0176:1
            X,[3:5] = 0 AND X,CF ≠ 0 THEN % IT IS SEGMENT DICT, %103-  01167200 C  0060:0180:1
           T := 33                                              %103-  01167300 C  0060:0183:3
         ELSE                                                   %103-  01167400 C  0060:0184:0
       ELSE                                                     %103-  01167500 C  0060:0184:1
   ELSE % MIX = 0                                               %103-  01167600 C  0060:0185:2
     IF(FROM=TABLESLOC AND T=0)OR(PRTO,CF>FROM AND PRTO,CF<N)THEN          01167700 C  0060:0185:3
       T := 35                                                  %103-  01167800 C  0060:0191:2
     ELSE                                                       %103-  01167900 C  0060:0191:3
```

```
            IF FROM = ADR-2 THEN % INTRINSIC ARRAY,                          %103-
              T := 22                                                        %103-
            ELSE                                                             %103-
              IF (DCSTACK AND DCHIT) OR L.[2:1] = 1 AND (T=0 OR T=12)        %103-
                 AND MX0 AND ISTACKOK AND IRSTACK THEN                       %103-
                T := 36;                                                     %103-
          IF MX0 AND ((T=1 AND QT) OR MCPSAVE) THEN                          %103-
            BUILDHEADER(LINE[*], FROM, 3, NAME[Q.CF],FF,                     %103-
                        NAMS[NAME[Q.CF],CF], 0, N-FROM-2, 0, L, Q, 0)        %103-
           ELSE                                                              %103-
            IF QT AND T = 7 AND INTR AND OPERAND(ADR+(LL:=Q.[8:10]),R)
                AND R := R.CF > 1023 AND LL LEQ INTMAX THEN                  %103-
              BUILDHEADER(LINE[*], FROM, 4, INAME[LL],FF,                    %103-
                          INAMS[INAME[LL],CF], 0, N-FROM-2, 0, L, Q, 0)
            ELSE                                                             %103-
              BUILDHEADER(LINE[*],FROM,L,[2:1],T,ATP[T+T],MX,                %103-
                          N-FROM-2, IF PDATADESC(Q,CF,X) AND                 %103-
                          X.CF = FROM+2 THEN 1 ELSE 0, L, Q, 0);             %103-
           WRITE(P[DBL],17,LINE[*]);                                         %103-
        END;
          STK:=-1;    BEDDED:=FALSE;
            WHILE BEDSTK GEQ STK:=STK+1 AND NOT BEDDED DO
             IF BEDDED:=
        (LL:=STAX[STK].CF GTR FROM AND N GTR LL) THEN
        IF STAX[STK],FF=0 THEN STAX[STK],FF:=FROM+2;
    IF NOT BEDDED AND  (IRSTACK  OR
            T=12 OR
            (DCSTACK AND DCHIT))  THEN
    STAX[MAXSTK:=MAXSTK+1]:=(N-1)&(FROM+2)[18:33:15]
                           &1[6:47:1];
        NOTPRINTCALL:=NOTPRINTCALL AND NOT MCPSAVE;
            IF (ONEMIX NEQ 0 AND ONEMIX ≠ MX AND MX ≠ 0) OR                  %103-
               (NONORMALCODE AND ((T=7 OR T=13) OR (T=1 AND MX ≠ 0)))        %103-
               OR (NOMCPCODE AND (L.[3:12]=64 OR MCPSAVE))  THEN             %103-
            ELSE                                                             %103-
              IF T = 35 THEN                                                 %106-
                DUMPMCPSAVECODE(FROM+2,N)                                    %106-
              ELSE                                                           %106-
                PRINT(FROM+2,N);                                             %106-
        MCPSAVE:=FALSE;
                END;
                FROM:=N;
            END;
            NOTPRINTCALL:=FALSE;
        END DUMP MEMORY AND NOTE STACKS;

        ARRAY MCPROG[0:255];
        INTEGER MAXMCPROG,ESP;
        ARRAY OUTERBLOCK[0:0];%
        PROCEDURE SEQUENCE(ARAY,L,M);
            VALUE L,M;
            ARRAY ARAY[0];
            INTEGER LIM;
        BEGIN
            INTEGER T,L;

            REAL V;
            STREAM PROCEDURE MOVE(S,D,D32,M32);
```

```
01168000 P  0060:0192:0
01168100 C  0060:0194:0
01168200 C  0060:0194:1
01168300 C  0060:0195:2
01168400 C  0060:0199:3
01168500 C  0060:0201:3
01168550 C  0060:0203:2
01168600 C  0060:0205:3
01168700 C  0060:0209:3
01168800 C  0060:0214:1
01168900 C  0060:0214:1
01169000 P  0060:0219:2
01169100 C  0060:0222:1
01169200 C  0060:0226:0
01169300 C  0060:0229:3
01169400 C  0060:0230:0
01169500 C  0060:0234:0
01169550 C  0060:0236:1
01169600 C  0060:0241:2
01169650 C  0060:0245:3
01194000 T  0060:0245:3
01195000 T  0060:0247:2
01196000 T  0060:0251:2
01197000 T  0060:0251:2
01198000 T  0060:0254:1
01199000 T  0060:0260:0
01200000 T  0060:0261:2
01202000 T  0060:0262:0
01203000 T  0060:0263:2
01204000 T  0060:0267:2
01208100 T  0060:0268:1
01209000 P  0060:0270:0
01210000 P  0060:0272:1
01211000 P  0060:0277:3
01212000 P  0060:0280:1
01213000 C  0060:0281:2
01213100 C  0060:0282:1
01213200 C  0060:0284:0
01213300 C  0060:0284:1
01215100 T  0060:0288:1
01216000 T  0060:0289:3
01218000 T  0060:0289:3
01219000 T  0060:0290:0
01219100 T  0060:0290:1
01220000 T  0060:0291:3
          60 IS  299 LONG, NEXT SEG   10
01221000 T  0010:0216:1
01222000 T  0010:0219:2
01223000 T  0010:0219:2
01224000 T  0010:0221:3
01225000 T  0010:0221:3
01226000 T  0010:0221:3
01227000 T  0010:0221:3
01228000 T  0010:0221:3
01229000 T  0010:0221:3
START OF SEGMENT **********   62
01230000 T  0062:0000:0
01231000 T  0062:0000:0
```

```
            VALUE D32,M32;
        BEGIN
            SI:=S; DI:=D;
            D32(DS:=32 WDS);
            DS:=M32 WDS;
        END MOVE;
        T:=LIM;
        WHILE (T:=T-1)≥0 DO
        BEGIN
            I:=LIM;
            L:=(V:=ARAY[T]).CF;
            WHILE ARAY[I].CF>L DO
                I:=I-1;
            IF (L:=I-T)>0 THEN
            BEGIN
                MOVE(ARAY[T+1],ARAY[T],L,[37:6],L,[43:5]);
                ARAY[I]:=V;
            END;
        END;
    END SEQUENCE;

    INTEGER BADSAVEPRTLOC;
    PROCEDURE GETSORTANDLISTMCPROG(B); VALUE B; BOOLEAN B;
    BEGIN
        REAL R;

        INTEGER TYP;
    FORMAT TOTALPROCS(///"### A TOTAL OF ",I3," MCP PROCEDURES ",
           "WERE PRESENT IN MEMORY"),
           BADLOC(//"*** AT LEAST ",I4," MCP PRT LOCATIONS DO NOT ",
                  "CONTAIN PROGRAM OR DATA DESCRIPTORS FOR SAVE CODE ",
                  "AS THEY SHOULD..."),
           F(X8,"PRESENT MCP PROCEDURES"//
                "PRT",X5,"DESCRIPTOR",X8,"THRU"//);

        IF NOT B THEN % SORT/*DONT* LIST
        BEGIN
        IF DESCRIPTOR(ESPBIT,R,TYP) AND TYP="75" THEN
        MCPROG[MAXMCPROG:=0]:=
            (ESP:=R.CF)&
            (ESP+R.[8:10]-1)[18:33:15]&
            ESPBIT[8:38:10]
        ELSE MAXMCPROG:="1;
        IF MAXMCPROG NEQ -1 THEN
       OUTERBLOCK[0]:=(PRTMAX+1)&(ESP-1) CTF;
        FOR I:=129 STEP 1 UNTIL PRTMAX DO
        IF DESCRIPTOR(I,R,TYP) AND
         (TYP="75" OR TYP="77" OR TYP="74") AND
           R.CF≠ESP           THEN
        MCPROG[MAXMCPROG:=MAXMCPROG+1]:=
            R.CF&
            (R.CF+R.[8:10]-1)[18:33:15]&
            I[8:38:10];
        SEQUENCE(MCPROG,MAXMCPROG);
        END ELSE % LIST ONLY
        BEGIN
```

```
                01232000 T  0062:0000:0
                01233000 T  0062:0000:0
                01234000 T  0062:0000:0
                01235000 T  0062:0000:1
                01236000 T  0062:0002:0
                01237000 T  0062:0002:1
                01238000 T  0062:0002:1
                01239000 T  0062:0003:3
                01240000 T  0062:0006:0
                01241000 T  0062:0006:0
                01242000 T  0062:0007:2
                01243000 T  0062:0009:2
                01244000 T  0062:0011:2
                01245000 T  0062:0012:1
                01246000 T  0062:0014:1
                01247000 T  0062:0015:2
                01248000 T  0062:0018:1
                01249000 T  0062:0019:3
                01250000 T  0062:0019:3
                01251000 T  0062:0020:0
      62 IS   23 LONG, NEXT SEG   10
                01251500 T  0010:0221:3
                01252000 T  0010:0221:3
                01253000 T  0010:0221:3
                01254000 T  0010:0221:3
    START OF SEGMENT *********   63
                01255000 T  0063:0000:0
                01256000 T  0063:0000:0
    START OF SEGMENT *********   64
                01257000 T  0064:0000:0
                01257100 T  0064:0000:0
                01257110 T  0064:0000:0
                01257120 T  0064:0000:0
                01258000 T  0064:0000:0
                01259000 T  0064:0000:0
      64 IS   61 LONG, NEXT SEG   63
                01259500 T  0063:0000:0
                01259600 T  0063:0000:1
                01260000 T  0063:0001:2
                01261000 T  0063:0003:3
                01262000 T  0063:0005:2
                01263000 T  0063:0006:0
                01264000 T  0063:0008:0
                01265000 T  0063:0009:3
                01265900 T  0063:0011:3
                01266000 T  0063:0012:1
                01267000 T  0063:0015:3
                01268000 T  0063:0017:2
                01269000 T  0063:0018:0
                01270000 T  0063:0021:2
                01271000 T  0063:0022:1
                01272000 T  0063:0024:1
                01273000 T  0063:0025:3
                01274000 T  0063:0028:0
                01275000 T  0063:0031:3
                01275500 T  0063:0033:2
                01275600 T  0063:0033:2
```

```
            NEXTPAGE;                                                                    01276000 T  0063:0033:3
            WRITE(P,F);                                                                  01277000 T  0063:0037:3
            SGLTOG:=TRUE;                                                                01277100 T  0063:0040:1
            FOR I:=0 STEP 1 UNTIL MAXMCPROG DO                                           01278000 T  0063:0041:3
            DISPLAY(MCPROG[I],[8:10],TRUE);                                              01279000 T  0063:0043:2
            SGLTOG:=FALSE;                                                               01279100 T  0063:0047:2
            WRITE(P,TOTALPROCS,MAXMCPROG+1);                                             01280000 T  0063:0048:0
             IF BADSAVEPRTLOC NEQ 0 THEN WRITE(P,BADLOC,BADSAVEPRTLOC);                  01280300 T  0063:0057:2
            END;                                                                         01280500 T  0063:0066:0
        END GET SORT AND LIST PRESENT MCP PROGRAM SEGMENTS;                              01281000 T  0063:0066:0
                                                                                  63 IS   69 LONG, NEXT SEG   10
"DEFINE GETANDSORTMCPROG=                                                                01281400 T  0010:0221:3
        GETSORTANDLISTMCPROG(FALSE)#;                                                    01281410 T  0010:0221:3
 DEFINE LISTMCPROG=                                                                      01281420 T  0010:0221:3
        GETSORTANDLISTMCPROG(TRUE)#;                                                     01281430 T  0010:0221:3
 PROCEDURE DUMPMCPSAVECODE(FROM,TOO);                                          %106-     01281500 P  0010:0221:3
   VALUE FROM, TOO;                                                            %106-     01281510 P  0010:0221:3
   REAL FROM, TOO;                                                             %106-     01281520 P  0010:0221:3
 BEGIN                                                                         %106-     01281530 P  0010:0221:3
   ARRAY MCPDATA [0:127]; % HOLDS DESCRIPTORS POINTING INTO                    %106-     01281540 P  0010:0221:3
                                                                                  START OF SEGMENT **********   65
                          % AREA BETWEEN FROM AND TOO.                         %106-     01281550 P  0065:0001:3
   INTEGER I,J,K,MAXMCPDATA;                                                   %106-     01281560 P  0065:0001:3
   DEFINE SIZEF = [8:10]#;                                                     %106-     01281570 P  0065:0001:3
   SGLTOG := TRUE;                                                             %106-     01281580 P  0065:0001:3
   MAXMCPDATA := -1;                                                           %106-     01281590 P  0065:0002:1
 FOR I := PRTBASE STEP 1 UNTIL PRTMAX DO                                       %106-     01281600 P  0065:0003:3
   IF DESCRIPTOR(I,J,K) THEN                                                   %106-     01281610 P  0065:0005:2
     IF ((K="75" OR K="77" OR K="74") AND (J.CF≠ESP OR I=ESPBIT)) OR           %106-     01281620 P  0065:0006:0
        (K="50" AND J.SIZEF≠0) THEN                                            %106-     01281630 P  0065:0012:0
       IF J.CF GEQ FROM AND J.CF+J.SIZEF LEQ TOO THEN                          %106-     01281640 P  0065:0014:1
         MCPDATA[MAXMCPDATA:=MAXMCPDATA+1] := J.CF &                           %106-     01281650 P  0065:0019:2
           (J.CF+J.SIZEF-1) FF & I SIZEF& REAL(K="50") [1:1];                  %106-     01281660 P  0065:0021:3
 SEQUENCE(MCPDATA,MAXMCPDATA); % SORT DESCRIPTORS IN ORDER BY ADDRESS.          %106-     01281670 P  0065:0029:3
 I := 0;                                                                       %106-     01281680 P  0065:0031:2
 NOTPRINTCALL := TRUE & NOTPRINTCALL [1:46]; % STACK PREVIOUS VAL.             %106-     01281685 C  0065:0031:3
 WHILE FROM < TOO DO                                                           %106-     01281690 P  0065:0033:3
   IF I > MAXMCPDATA THEN                                                      %106-     01281700 P  0065:0035:2
     BEGIN                                                                     %106-     01281710 P  0065:0036:0
       PRINT(FROM,TOO);                                                        %106-     01281720 P  0065:0036:1
       FROM := TOO;                                                            %106-     01281730 P  0065:0039:3
     END                                                                       %106-     01281740 P  0065:0040:1
   ELSE                                                                        %106-     01281750 P  0065:0040:1
     IF FROM < (J:=MCPDATA[I].CF) THEN                                         %106-     01281760 C  0065:0040:1
       BEGIN                                                                   %106-     01281770 C  0065:0043:2
         PRINT(FROM,J);                                                        %106-     01281780 C  0065:0043:3
         FROM := J;                                                            %106-     01281790 C  0065:0046:1
       END                                                                     %106-     01281800 C  0065:0047:3
     ELSE                                                                      %106-     01281810 C  0065:0047:3
       IF FROM = J THEN                                                        %106-     01281820 C  0065:0047:3
         BEGIN                                                                 %106-     01281830 C  0065:0048:1
           IF BOOLEAN(LINKTYPE:=1+MCPDATA[I],[1:1]) AND NOMCPCODE THEN         %106-     01281835 C  0065:0049:2
           ELSE                                                                %106-     01281840 C  0065:0052:0
             BEGIN                                                             %106-     01281845 C  0065:0052:1
               DISPLAY(MCPDATA[I],SIZEF,TRUE);                                 %106-     01281850 C  0065:0053:2
               PRINT(J,MIN(TOO,MCPDATA[I],FF+1));                              %106-     01281855 C  0065:0055:2
             END;                                                              %106-     01281857 C  0065:0062:0
```

```
            FROM := MCPDATA[I].FF+1;
            I := I + 1;
          END
        ELSE % FROM > MCPDATA[I].CF
          I := I + 1;
  SGLTOG := FALSE;
  NOTPRINTCALL := NOTPRINTCALL.[1:46]; % POP VALUE
END OF DUMPMCPSAVECODE;

    ARRAY INTSP[0:INAMESIZE-1];
    INTEGER INTSPMAX;
    PROCEDURE GETSORTANDLISTINTRINSICS;
    BEGIN
        INTEGER IMAX,TYP;

        REAL ADR,R,L;
        REAL MIX;
        FORMAT ITEM(X2,A3," =",2(X4,A5)),

          BADINTO("*** BAD INTRNSC[0]"),
            F(X8,"PRESENT INTRINSICS"//
                 "INDEX",X7,"FROM",X5,"THRU"/));

        INTSPMAX:=-1;
        IF DESCRIPTOR(INTRNSC,ADR,TYP) AND TYP="50" THEN
        BEGIN
        IMAX:=ADR.[8:10]-1;
        ADR:=ADR.CF;
        IF OPERAND(ADR,R) AND R.[1:38]=0
        AND R NEQ 0 AND R LEQ IMAX THEN IMAX:=R ELSE MIX:=-1;
        IF IMAX>0 THEN
        FOR I:=1 STEP 1 UNTIL IMAX DO
        IF OPERAND(ADR+I,R) AND R.CF>1023 THEN
        INTSP[INTSPMAX:=INTSPMAX+1]:=
          IF I=17 THEN R.CF&(R.CF+3) CTF & I[8:38:10] ELSE
            R.CF&
            (IF OPERAND(R.CF-1,L) AND
                0<(L:=L.FF) AND
                L≤1023 THEN
            R.CF+L-1 ELSE R.CF)[18:33:15]&
            I[8:38:10];
    END;
    IF INTSPMAX≥0 THEN
    BEGIN
        NEXTPAGE;
        DISPLAY(INTRNSC,TRUE);
        IF MIX LSS 0 THEN WRITE(P[DBL],BADINTO);
        WRITE(P[DBL],F);
        SEQUENCE(INTSP,INTSPMAX);
        FOR I:=0 STEP 1 UNTIL INTSPMAX DO
        BEGIN
            WRITE(LINE[*],ITEM,OCTAL(L:=INTSP[I],[8:10]),
                  OCTAL(INTSP[I],CF),
                  OCTAL(INTSP[I],FF));
            IF L≤INTMAX THEN
            MOVE(INAMS[INAME[L],CF],LINE[4],
                       INAME[L],FF);
```

```
%106=          01281860 C  0065:0062:0
%106=          01281865 C  0065:0064:1
%106=          01281870 C  0065:0065:3
%106=          01281880 C  0065:0065:3
%106=          01281890 C  0065:0065:3
%106=          01281900 C  0065:0068:0
%106=          01281905 C  0065:0068:1
%106=          01281910 C  0065:0070:1
        65 IS   76 LONG, NEXT SEG   10
               01282000 T  0010:0221:3
               01283000 T  0010:0225:3
               01284000 T  0010:0225:3
               01285000 T  0010:0225:3
               01286000 T  0010:0225:3
     START OF SEGMENT **********   66
               01287000 T  0066:0000:0
               01287100 T  0066:0000:0
               01288000 T  0066:0000:0
     START OF SEGMENT **********   67
               01288100 T  0067:0000:0
               01289000 T  0067:0000:0
               01290000 T  0067:0000:0
      67 IS    31 LONG, NEXT SEG   66
               01291000 T  0066:0000:0
               01292000 T  0066:0001:2
               01293000 T  0066:0003:3
               01294000 T  0066:0004:0
               01295000 T  0066:0005:3
               01295100 T  0066:0007:2
               01295200 T  0066:0009:2
               01296000 T  0066:0014:1
               01297000 T  0066:0015:3
               01298000 T  0066:0017:2
               01299000 T  0066:0020:0
               01299100 T  0066:0022:0
               01300000 T  0066:0027:2
               01301000 T  0066:0027:3
               01302000 T  0066:0029:3
               01303000 T  0066:0031:3
               01304000 T  0066:0032:1
               01305000 T  0066:0036:1
               01306000 T  0066:0040:0
               01307000 T  0066:0040:0
               01308000 T  0066:0041:2
               01309000 T  0066:0041:3
               01310000 T  0066:0045:3
               01310100 T  0066:0047:2
               01311000 T  0066:0051:2
               01312000 T  0066:0054:0
               01313000 T  0066:0055:3
               01314000 T  0066:0057:2
               01315000 T  0066:0057:2
               01316000 T  0066:0065:2
               01317000 T  0066:0068:1
               01318000 T  0066:0075:2
               01319000 T  0066:0076:0
               01320000 T  0066:0078:1
```

```
            WRITE(P[DBL],7,LINE[*]);
        END;
    END;
END GET SORT AND LIST PRESENT INTRINSICS;

STREAM PROCEDURE MOVC(S,SP,D,DP,W,C);
    VALUE SP,DP,W,C;
BEGIN
    SI:=S; SI:=SI+SP;
    DI:=D; DI:=DI+DP;
    W(DS:=8 CHR); DS:=C CHR;
END MOVC;
BOOLEAN PROCEDURE WITHIN(ARAY,AMAX,ITEM,RESULT,PLUS);
    VALUE AMAX,ITEM;
    ARRAY ARAY[0];
    INTEGER AMAX,ITEM,RESULT,PLUS;
BEGIN
    BOOLEAN FOUND;

    LABEL EXIT;
    IF AMAX≥0 THEN
    FOR I:=0 STEP 1 UNTIL AMAX DO
    IF FOUND:=
        (ARAY[I].CF≤ITEM AND ITEM ≤ARAY[I].FF) THEN
    BEGIN
        RESULT:=ARAY[I].[8:10];
        PLUS:=ITEM-ARAY[I].CF;
        GO TO EXIT;
    END;
    EXIT: WITHIN:=FOUND;
  END WITHIN;

ARRAY PROGS[0:3,0:255]; INTEGER PROWS;                                  %104
  DEFINE CONVERTIT = MSG := DI;                                         %104
                     DI := LOC T;                                       %104
                     DI := DI + 1;                                      %104
                     DS := 7 DEC;                                       %104
                     DI := DI - 7;                                      %104
                     DS := 6 FILL;                                      %104
                     DI := MSG;                                         %104
                     SI := LOC T;                                       %104
                     SI := SI + 1;                                      %104
                     7 (IF SC = " " THEN SI := SI + 1 ELSE DS := CHR)   %104
                     #;                                                 %104
  STREAM PROCEDURE FORMATCODENAME(MSG,WHICH,SEG,ADR,SYL,NAME,N);        %104
    VALUE WHICH,SEG,ADR,SYL,N;                                          %104
    BEGIN                                                               %104
      LOCAL T;                                                          %104
      LABEL MOVADR, MCPCODE, INTRINSIC, MOVNAM, EXIT, MCPOUTERBLOCK;    %104
      DI := MSG;                                                        %104
      CI := CI + WHICH;                                                 %104
        GO TO MOVADR;                                                   %104
        GO TO MOVADR;                                                   %104
        GO TO MOVADR;                                                   %104
      SI := LOC SEG;                                                    %104
      CONVERTIT;                                                        %104
      DS := LIT ":";                                                    %104
```

```
                01321000 T  0066:0080:0
                01322000 T  0066:0084:0
                01323000 T  0066:0086:1
                01324000 T  0066:0086:1
          66 IS   90 LONG, NEXT SEG   10
                01325000 T  0010:0225:3
                01326000 T  0010:0225:3
                01327000 T  0010:0225:3
                01328000 T  0010:0226:0
                01329000 T  0010:0226:1
                01330000 T  0010:0227:3
                01331000 T  0010:0229:3
                01332000 T  0010:0229:3
                01333000 T  0010:0229:3
                01334000 T  0010:0229:3
                01335000 T  0010:0229:3
                01336000 T  0010:0229:3
                01337000 T  0010:0229:3
       START OF SEGMENT **********   68
                01338000 T  0068:0000:0
                01339000 T  0068:0000:0
                01340000 T  0068:0000:1
                01341000 T  0068:0002:0
                01342000 T  0068:0002:0
                01343000 T  0068:0006:0
                01344000 T  0068:0006:1
                01345000 T  0068:0008:0
                01346000 T  0068:0010:1
                01347000 T  0068:0011:2
                01348000 T  0068:0013:2
                01349000 T  0068:0014:1
          68 IS   18 LONG, NEXT SEG   10
                01349005 C  0010:0229:3
                01349010 C  0010:0232:1
                01349020 C  0010:0232:1
                01349030 C  0010:0232:1
                01349040 C  0010:0232:1
                01349050 C  0010:0232:1
                01349060 C  0010:0232:1
                01349070 C  0010:0232:1
                01349080 C  0010:0232:1
                01349090 C  0010:0232:1
                01349100 C  0010:0232:1
                01349110 C  0010:0232:1
                01349120 C  0010:0232:1
                01349130 C  0010:0232:1
                01349140 C  0010:0232:1
                01349150 C  0010:0234:0
                01349160 C  0010:0234:0
                01349170 C  0010:0234:0
                01349180 C  0010:0234:0
                01349190 C  0010:0234:1
                01349200 C  0010:0235:2
                01349210 C  0010:0235:2
                01349220 C  0010:0235:3
                01349230 C  0010:0235:3
                01349240 C  0010:0239:3
```

```
MOVADR:                                                                 %104-       01349250 C  0010:0240:0
  SI := LOC ADR;                                                        %104-       01349260 C  0010:0241:2
  CONVERTIT;                                                            %104-       01349270 C  0010:0241:2
  DS := LIT ":";                                                        %104-       01349280 C  0010:0245:2
  SI := LOC SYL;                                                        %104-       01349290 C  0010:0245:3
  CONVERTIT;                                                            %104-       01349300 C  0010:0246:0
  DS := 4 LIT " OF ";                                                   %104-       01349310 C  0010:0250:0
  CI := CI + WHICH;                                                     %104-       01349320 C  0010:0250:1
    GO TO MCPCODE;                                                      %104-       01349330 C  0010:0251:2
    GO TO MCPOUTERBLOCK;                                                %104-       01349340 C  0010:0251:3
    GO TO INTRINSIC;                                                    %104-       01349350 C  0010:0251:3
  DS := 12 LIT "USER PROGRAM";                                          %104-       01349360 C  0010:0252:0
  GO TO EXIT;                                                           %104-       01349370 C  0010:0253:3
MCPOUTERBLOCK:                                                          %104-       01349371 C  0010:0254:0
  DS := 15 LIT "MCP OUTER BLOCK";                                       %104-       01349372 C  0010:0254:0
  GO TO EXIT;                                                           %104-       01349374 C  0010:0256:0
MCPCODE:                                                                %104-       01349380 C  0010:0256:1
  DS := 8 LIT "MCP CODE";                                               %104-       01349390 C  0010:0257:2
  GO TO MOVNAM;                                                         %104-       01349400 C  0010:0258:0
INTRINSIC:                                                              %104-       01349410 C  0010:0258:1
  DS := 9 LIT "INTRINSIC";                                              %104-       01349420 C  0010:0259:2
MOVNAM:                                                                 %104-       01349430 C  0010:0260:1
  DS := 3 LIT " - ";                                                    %104-       01349440 C  0010:0261:2
  SI := NAME;                                                           %104-       01349450 C  0010:0261:3
  N ( 8 (IF SC = " " THEN JUMP OUT 2 TO EXIT ELSE DS := CHR ) );        %104-       01349460 C  0010:0262:0
EXIT:                                                                   %104-       01349470 C  0010:0265:3
  DS := LIT ARROW;                                                      %104-       01349480 C  0010:0266:0
END OF FORMATCODENAME;                                                  %104-       01349490 C  0010:0266:1
BOOLEAN PROCEDURE WITHINCODESEGMENT(ADDRESS,SEG,ADR,SYL,MSG);           %104-       01349500 C  0010:0266:1
  VALUE ADDRESS, SYL;                                                   %104-       01349510 C  0010:0266:1
  INTEGER ADDRESS,SEG,ADR,SYL;                                          %104-       01349520 C  0010:0266:1
  ARRAY MSG[*];                                                         %104-       01349530 C  0010:0266:1
BEGIN                                                                   %104-       01349540 C  0010:0266:1
  INTEGER I;                                                            %104-       01349550 C  0010:0266:1
                                                                                START OF SEGMENT **********   69
  BOOLEAN FLAG;                                                         %104-       01349555 C  0069:0000:0
  I := -1;                                                              %104-       01349556 C  0069:0000:0
  IF WITHINCODESEGMENT := WITHIN(MCPROG,MAXMCPROG,ADDRESS,SEG,ADR) THEN             01349560 C  0069:0001:2
    FORMATCODENAME(MSG[*],0,SEG,ADR,SYL,NAMS[NAME[SEG],CF],             %104-       01349570 C  0069:0004:0
      NAME[SEG],FF)                                                     %104-       01349580 C  0069:0008:1
  ELSE                                                                  %104-       01349590 C  0069:0010:0
    IF WITHINCODESEGMENT := WITHIN(OUTERBLOCK,0,ADDRESS,SEG,ADR) THEN               01349600 C  0069:0010:1
      FORMATCODENAME(MSG[*],1,SEG,ADR:=ADDRESS,SYL,SYL,SYL)             %104-       01349610 C  0069:0014:0
    ELSE                                                                %104-       01349620 C  0069:0018:0
      IF WITHINCODESEGMENT := WITHIN(INTSP,INTSPMAX,ADDRESS,SEG,ADR)THEN            01349630 C  0069:0018:0
        FORMATCODENAME(MSG[*],2,SEG,ADR,SYL,INAMS[INAME[SEG],CF],       %104-       01349640 C  0069:0021:2
          INAME[SEG],FF)                                                %104-       01349650 C  0069:0025:3
      ELSE                                                              %104-       01349660 C  0069:0027:2
        WHILE (I:=I+1) LEQ PROWS AND NOT FLAG DO                        %104-       01349670 C  0069:0027:2
          IF WITHINCODESEGMENT:=FLAG:=WITHIN(PROGS[I,*],PROGS[I,255],               01349680 C  0069:0031:2
              ADDRESS,SEG,ADR) THEN                                     %104-       01349690 C  0069:0033:3
            FORMATCODENAME(MSG[*],3,SEG,ADR,SYL,SYL,SYL);               %104-       01349700 C  0069:0036:0
END OF WITHINCODESEGMENT;                                               %104-       01349710 C  0069:0039:3
                                                                              69 IS   43 LONG, NEXT SEG   10
%                                                                       %104-       01350000 P  0010:0266:1
    INTEGER PROCEDURE SPREAD(AT,F,C);VALUE AT;                                      01351000 T  0010:0266:1
        INTEGER AT,F,C;          FORWARD;                                           01352000 T  0010:0266:1
```

```
REAL LOCIRCW;
PROCEDURE DUMPSTACK(INX);
    VALUE INX;
    INTEGER INX;
BEGIN
    INTEGER TOS,BOS,SEG,ADR,H,L,PRO,I;

    INTEGER FUNNYMSCW, WHICH;
    REAL V,R;
    REAL TOSORIG,X,Y; BOOLEAN WOOPS;
    LABEL ERROWT;
    LABEL DUMPIT;
    LABEL SKIPEEE;
    LABEL SQUEEZE,CUTBACK,XIT;
    DEFINE CD=WOOPS #;
    DEFINE DELTA=50#;
    DEFINE MAXSTACKSIZE=512#;% MAX STACK BELOW TOS WE"LL DUMP
    DEFINE SPEC=BOOLEAN(STAX[INX],[4:1])#;
    BOOLEAN NORMAL, RCW;
    FORMAT HD(X8,"STACK FROM ",A5," DOWN TO ",A5),

          TRITEM(A5," = ",A6,2(X1,A5)),
          S(X8,"BED[",A3,"] =",A6,X1,A5,X1,A5,", MASK= ",
            A6,2(X1,A5)),
          C(X8,3A6,X1,A4,"(",I4,")","  + ",A4,"(",I4,":",I1,")");

STREAM PROCEDURE FORMATRCW(LINE, F, C, NAME);
  VALUE F, C;
BEGIN
  LOCAL T;
  DI := LINE;
  DI := DI + 32;
  13 (DS := 8 LIT " ");
  DI := LINE;
  DI := DI + 30;
  DS := 10 LIT "--RCW:  F=";
  SI := LOC F;
  SI := SI + 5;
  SKIP 3 SB;
  5 OCTADE;
  DS := 4 LIT ", C=";
  SI := LOC C;
  SI := SI + 5;
  SKIP 3 SB;
  5 OCTADE;
  DS := 2 LIT " (";
  SI := NAME;
  63 (IF SC ≠ ARROW THEN DS := CHR ELSE JUMP OUT);
  DS := LIT ")";
END OF FORMATRCW;
STREAM PROCEDURE FORMATMSCW(LINE, R, MSFF, SALF, F);
  VALUE R, MSFF, SALF, F;
BEGIN
  DI := LINE;
  DI := DI + 32;
  13 (DS := 8 LIT " ");
  DI := LINE;
```

```
                                    01352500 T  0010:0266:1
                                    01353000 T  0010:0266:1
                                    01354000 T  0010:0266:1
                                    01355000 T  0010:0266:1
                                    01356000 T  0010:0266:1
                                    01357000 T  0010:0266:1
                                START OF SEGMENT **********   70
%104-                               01357100 P  0070:0000:0
                                    01358000 T  0070:0000:0
                                    01359000 T  0070:0000:0
                                    01360000 T  0070:0000:0
                                    01360100 T  0070:0000:0
                                    01361000 T  0070:0000:0
                                    01362000 T  0070:0000:0
                                    01363000 T  0070:0000:0
                                    01364000 T  0070:0000:0
                                    01365000 T  0070:0000:0
                                    01365100 T  0070:0000:0
%104-                               01366000 P  0070:0000:0
                                    01367000 T  0070:0000:0
                                START OF SEGMENT **********   71
                                    01368000 T  0071:0000:0
                                    01369000 T  0071:0000:0
                                    01370000 T  0071:0000:0
                                    01370010 C  0071:0000:0
                                  71 IS   56 LONG, NEXT SEG   70
%104-                               01370020 C  0070:0000:0
%104-                               01370030 C  0070:0000:0
%104-                               01370040 C  0070:0000:0
%104-                               01370050 C  0070:0000:0
%104-                               01370060 C  0070:0000:0
%104-                               01370070 C  0070:0000:0
%104-                               01370080 C  0070:0000:1
%104-                               01370090 C  0070:0002:0
%104-                               01370100 C  0070:0002:1
%104-                               01370110 C  0070:0002:1
%104-                               01370120 C  0070:0004:0
%104-                               01370130 C  0070:0004:1
%104-                               01370140 C  0070:0004:1
%104-                               01370150 C  0070:0005:2
%104-                               01370160 C  0070:0007:3
%104-                               01370170 C  0070:0008:1
%104-                               01370180 C  0070:0008:1
%104-                               01370190 C  0070:0009:2
%104-                               01370200 C  0070:0009:2
%104-                               01370210 C  0070:0012:0
%104-                               01370220 C  0070:0012:1
%104-                               01370230 C  0070:0012:1
%104-                               01370240 C  0070:0015:2
%104-                               01370250 C  0070:0015:3
%104-                               01370260 C  0070:0015:3
%104-                               01370270 C  0070:0015:3
%104-                               01370280 C  0070:0015:3
%104-                               01370290 C  0070:0016:0
%104-                               01370300 C  0070:0016:0
%104-                               01370310 C  0070:0016:1
%104-                               01370320 C  0070:0018:0
```

```
    DI := DI + 32;                                                          %104-    01370330 C  0070:0018:1
    DS := 9 LIT "MSCW:  F=";                                                %104-    01370340 C  0070:0018:1
    SI := LOC F;                                                            %104-    01370350 C  0070:0020:0
    SI := SI + 5;                                                           %104-    01370360 C  0070:0020:1
    SKIP 3 SB;                                                              %104-    01370370 C  0070:0020:1
    5 OCTADE;                                                               %104-    01370380 C  0070:0021:2
    DS := 7 LIT ", MSFF=";                                                  %104-    01370390 C  0070:0023:3
    SI := LOC MSFF;                                                         %104-    01370400 C  0070:0025:2
    SI := SI + 7;                                                           %104-    01370410 C  0070:0025:2
    DS := CHR;                                                              %104-    01370420 C  0070:0025:3
    DS := 7 LIT ", SALF=";                                                  %104-    01370430 C  0070:0025:3
    SI := LOC SALF;                                                         %104-    01370440 C  0070:0027:2
    SI := SI + 7;                                                           %104-    01370450 C  0070:0027:2
    DS := CHR;                                                              %104-    01370460 C  0070:0027:3
    DS := 4 LIT ", R=";                                                     %104-    01370470 C  0070:0027:3
    SI := LOC R;                                                            %104-    01370480 C  0070:0028:1
    SI := SI + 6;                                                           %104-    01370490 C  0070:0028:1
    SKIP 3 SB;                                                              %104-    01370500 C  0070:0029:2
    3 OCTADE;                                                               %104-    01370510 C  0070:0029:2
    DS := 2 LIT "00";                                                       %104-    01370515 C  0070:0032:0
END;                                                                        %104-    01370520 C  0070:0032:1
STREAM PROCEDURE FORMATDESC(LINE, PRESENCEBIT, LENGTH, ADDRESS,             %104-    01370530 C  0070:0032:1
    MOMADDRESS, MOMFLG);                                                    %104-    01370540 C  0070:0032:1
  VALUE PRESENCEBIT, LENGTH, ADDRESS, MOMFLG, MOMADDRESS;                   %104-    01370550 C  0070:0032:1
BEGIN                                                                       %104-    01370560 C  0070:0032:1
  LABEL L1, THISISAMOM, NOMOM;                                              %104-    01370570 C  0070:0033:2
  DI := LINE;                                                               %104-    01370580 C  0070:0033:2
  DI := DI + 32;                                                            %104-    01370590 C  0070:0033:2
  13 (DS := 8 LIT " ");                                                     %104-    01370600 C  0070:0033:3
  DI := LINE;                                                               %104-    01370610 C  0070:0035:2
  DI := DI + 34;                                                            %104-    01370620 C  0070:0035:3
  PRESENCEBIT (DS := 7 LIT "PRESENT"; JUMP OUT TO L1);                      %104-    01370630 C  0070:0035:3
  DS := 11 LIT "NON-PRESENT";                                               %104-    01370640 C  0070:0038:1
L1:                                                                         %104-    01370650 C  0070:0040:1
  LENGTH (DS := 6 LIT " ARRAY"; JUMP OUT);                                  %104-    01370660 C  0070:0041:2
  DS := 21 LIT " DESCRIPTOR, ADDRESS=";                                     %104-    01370670 C  0070:0044:0
  SI := LOC ADDRESS;                                                        %104-    01370680 C  0070:0047:2
  SI := SI + 5;                                                             %104-    01370690 C  0070:0047:2
  SKIP 3 SB;                                                                %104-    01370700 C  0070:0047:3
  5 OCTADE;                                                                 %104-    01370710 C  0070:0047:3
  LENGTH (DS := 9 LIT ", LENGTH=";                                          %104-    01370720 C  0070:0050:1
          SI := LOC LENGTH;                                                 %104-    01370730 C  0070:0053:2
          SI := SI + 6;                                                     %104-    01370740 C  0070:0053:2
          4 OCTADE;                                                         %104-    01370750 C  0070:0053:3
          DS := LIT "(";                                                    %104-    01370760 C  0070:0056:0
          SI := LOC LENGTH;                                                 %104-    01370770 C  0070:0056:1
          DS := 4 DEC;                                                      %104-    01370780 C  0070:0057:2
          DS := LIT ")";                                                    %104-    01370790 C  0070:0057:2
          JUMP OUT);                                                        %104-    01370800 C  0070:0057:3
  CI := CI + MOMFLG;                                                        %104-    01370810 C  0070:0059:2
    GO TO NOMOM;                                                            %104-    01370820 C  0070:0059:3
    GO TO THISISAMOM;                                                       %104-    01370830 C  0070:0059:3
  DS := 9 LIT ", MOM AT ";                                                  %104-    01370840 C  0070:0060:0
  SI := LOC MOMADDRESS;                                                     %104-    01370850 C  0070:0061:3
  SI := SI + 5;                                                             %104-    01370860 C  0070:0061:3
  SKIP 3 SB;                                                                %104-    01370870 C  0070:0062:0
  5 OCTADE;                                                                 %104-    01370880 C  0070:0062:0
```

```
  GO TO NOMOM;                                                                %104-    01370890 C  0070:0065:2
THISISAMOM:                                                                   %104-    01370900 C  0070:0065:2
  DS := 16 LIT " = THIS IS A MOM";                                            %104-    01370910 C  0070:0066:0
NOMOM:                                                                        %104-    01370920 C  0070:0068:0
END OF FORMATDESC;                                                            %104-    01370930 C  0070:0069:2
STREAM PROCEDURE FORMATOP(LINE, VAL, LINE2);                                  %104-    01370940 C  0070:0069:2
  VALUE VAL;                                                                  %104-    01370950 C  0070:0069:2
BEGIN                                                                         %104-    01370960 C  0070:0069:2
  DI := LINE;                                                                 %104-    01370970 C  0070:0070:0
  DI := DI + 32;                                                              %104-    01370980 C  0070:0070:0
  13 ( DS := 8 LIT " ");                                                      %104-    01370990 C  0070:0070:1
  DI := LINE;                                                                 %104-    01371000 P  0070:0072:0
  DI := DI + 34;                                                              %104-    01371010 C  0070:0072:1
  DS := 18 LIT "OPERAND:  DECIMAL=";                                          %104-    01371020 C  0070:0072:1
  SI := LINE2;                                                                %104-    01371030 C  0070:0075:2
  32 (IF SC = " " THEN SI := SI + 1 ELSE DS := CHR);                          %104-    01371040 C  0070:0075:3
  DS := 6 LIT ", BCL=";                                                       %104-    01371050 C  0070:0077:2
  DS := LIT """;                                                              %104-    01371060 C  0070:0078:0
  SI := LOC VAL;                                                              %104-    01371070 C  0070:0078:1
  DS := 8 CHR;                                                                %104-    01371080 C  0070:0079:2
  DS := LIT """;                                                              %104-    01371090 C  0070:0079:2
END;                                                                          %104-    01371100 C  0070:0079:3
STREAM PROCEDURE FORMATPROGDESC(LINE,WHICH,PBIT,NAME);                        %104-    01371110 C  0070:0080:0
  VALUE WHICH,PBIT;                                                           %104-    01371120 C  0070:0080:0
BEGIN                                                                         %104-    01371130 C  0070:0080:0
  LABEL L1, ACCIDENTRY, WORDMODE, LABELDESC, MODE, REST;                      %104-    01371135 C  0070:0080:0
  DI := LINE;                                                                 %104-    01371140 C  0070:0080:0
  DI := DI + 32;                                                              %104-    01371150 C  0070:0080:0
  13 (DS := 8 LIT " ");                                                       %104-    01371160 C  0070:0080:1
  DI := LINE;                                                                 %104-    01371170 C  0070:0082:0
  DI := DI + 34;                                                              %104-    01371180 C  0070:0082:1
  PBIT (DS := 7 LIT "PRESENT"; JUMP OUT TO L1);                               %104-    01371190 C  0070:0082:1
  DS := 11 LIT "NON-PRESENT";                                                 %104-    01371200 C  0070:0085:3
L1:                                                                           %104-    01371210 C  0070:0087:3
  CI := CI + WHICH;                                                           %104-    01371220 C  0070:0088:0
    GO TO ACCIDENTRY;                                                         %104-    01371230 C  0070:0088:1
    GO TO WORDMODE;                                                           %104-    01371240 C  0070:0088:1
    GO TO LABELDESC;                                                          %104-    01371250 C  0070:0089:2
  DS := 10 LIT " CHARACTER";                                                  %104-    01371252 C  0070:0089:2
  GO TO MODE;                                                                 %104-    01371254 C  0070:0090:1
LABELDESC:                                                                    %104-    01371256 C  0070:0091:2
  DS := 6 LIT " LABEL";                                                       %104-    01371260 C  0070:0091:2
  GO TO REST;                                                                 %104-    01371270 C  0070:0092:0
ACCIDENTRY:                                                                   %104-    01371280 C  0070:0092:0
  DS := 17 LIT " ACCIDENTAL ENTRY";                                           %104-    01371290 C  0070:0093:2
  GO TO REST;                                                                 %104-    01371300 C  0070:0095:3
WORDMODE:                                                                     %104-    01371310 C  0070:0095:3
  DS := 5 LIT " WORD";                                                        %104-    01371320 C  0070:0096:0
MODE:                                                                         %104-    01371360 C  0070:0097:2
  DS := 13 LIT " MODE PROGRAM";                                               %104-    01371370 C  0070:0097:2
REST:                                                                         %104-    01371380 C  0070:0099:2
  DS := 11 LIT " DESCRIPTOR";                                                 %104-    01371390 C  0070:0099:2
  PBIT ( DS := 4 LIT " TO ";                                                  %104-    01371400 C  0070:0100:1
         SI := NAME;                                                          %104-    01371410 C  0070:0102:1
         63 (IF SC = ARROW THEN JUMP OUT ELSE DS := CHR) );                   %104-    01371420 C  0070:0102:1
END OF FORMATPROGDESC;                                                        %104-    01371430 C  0070:0105:2
%                                                                             %104-    01372000 P  0070:0105:3
```

```
            NEXTITEM;
            WRITE(P[DBL],HD,
                  IF (TOS:=STAX[INX],CF) LSS 127 THEN "*****"
                    ELSE OCTAL(TOS),
                IF (BOS:=STAX[INX],FF) LSS 64 THEN "*****"
                    ELSE OCTAL(BOS));
          IF BOS LSS 64 THEN BOS:=0;
          IF TOS LSS 127 THEN TOS:=0;
        IF TOS=0 THEN GO ERROWT;%F,G, IN EARLY SELECTRUN
        IF SPEC THEN GO DUMPIT;
            NORMAL:=BOOLEAN(STAX[INX],[7:1]);
            IF INX=0 OR INX>BEDSTK THEN
            BEGIN
         IF NORMAL THEN
         BEGIN
             WRITE(P[DBL],TRITEM,OCTAL(TOS),OCTAL(SPREAD(TOS,H,L)),
                   OCTAL(H),OCTAL(L));
             TOS:=TOS-1;
        END;
        IF NOT NORMAL AND BOS=64 THEN
FOR TOS:=127 STEP -1 UNTIL ACTUALPRTBASE DO
BEGIN
        WRITE(LINE[*],TRITEM,OCTAL(TOS),OCTAL(SPREAD(TOS,H,L)),
              OCTAL(H),OCTAL(L));
        MOVF(XNAMS[XNAME[TOS],CF],LINE[5],XNAME[TOS],FF);
        WRITE(P[DBL],8,LINE[*]);
END;
        IF BOOLEAN(STAX[INX],[6:1]) THEN BEGIN
            TOSORIG:=TOS;
        SQUEEZE:DO TOS:=TOS-1 UNTIL (WOOPS:=TOS LEQ BOS) OR
                                CONTROLDESC(TOS,X);
        IF WOOPS THEN BEGIN TOS:=TOS+10; GO XIT END;
        CUTBACK:IF (Y:=X,FF) GEQ BOS AND
                 Y LEQ BOS +1          AND
                CD:=CONTROLDESC(Y,X) THEN
                  BEGIN
                       TOS:=MIN(TOSORIG,TOS+DELTA);
                       GO XIT;
                  END
                  ELSE
                  IF Y GTR BOS +1 AND  Y LSS TOS
                      AND CD      THEN
                     GO CUTBACK;
            GO SQUEEZE;
        XIT;
        END;
                IF BOS=64 THEN
                IF OPERAND(107,V) AND V=(Y:="EEEE"&"EEEE"[1:25:23]) THEN
                BEGIN
                    FOR I:=108 STEP 1 UNTIL 111 DO
                    IF OPERAND(I,V) AND V NEQ Y THEN I:=112;
                    IF I=113 THEN FOR    TOS:=111 STEP -1 UNTIL 108 DO
                    WRITE(P[DBL],TRITEM,OCTAL(TOS),
                                        OCTAL(R:=SPREAD(TOS,H,L)),
                          OCTAL(H),OCTAL(L));
                END;
            SKIPEEE:
```

```
01373000 T  0070:0105:3
01374000 T  0070:0106:1
01375000 T  0070:0109:2
01376000 T  0070:0114:0
01377000 T  0070:0118:0
01378000 T  0070:0121:3
01379000 T  0070:0128:0
01380000 T  0070:0130:0
01381000 T  0070:0132:0
01381100 T  0070:0133:3
01382000 T  0070:0135:2
01383000 T  0070:0136:1
01384000 T  0070:0138:1
01385000 T  0070:0139:2
01386000 T  0070:0139:2
01387000 T  0070:0139:3
01388000 T  0070:0149:3
01389000 T  0070:0157:2
01390000 T  0070:0158:1
01391000 T  0070:0158:1
01392000 T  0070:0160:0
01393000 T  0070:0162:0
01394000 T  0070:0162:0
01395000 T  0070:0172:1
01396000 T  0070:0180:0
01397000 T  0070:0184:1
01398000 T  0070:0189:2
01399000 T  0070:0191:2
01400000 T  0070:0192:1
01401000 T  0070:0193:3
01402000 T  0070:0196:1
01403000 T  0070:0198:0
01404000 T  0070:0200:1
01405000 T  0070:0202:1
01406000 T  0070:0204:0
01407000 T  0070:0206:0
01408000 T  0070:0206:1
01409000 T  0070:0210:0
01410000 T  0070:0210:1
01411000 T  0070:0210:1
01412000 T  0070:0210:1
01413000 T  0070:0212:1
01414000 T  0070:0213:3
01415000 T  0070:0214:1
01416000 T  0070:0215:2
01417000 T  0070:0215:2
01418500 T  0070:0215:2
01418510 T  0070:0215:3
01418520 T  0070:0219:3
01418530 T  0070:0220:0
01418540 T  0070:0222:0
01418560 T  0070:0227:3
01418570 T  0070:0230:0
01418575 T  0070:0236:0
01418580 T  0070:0240:0
01418590 T  0070:0250:1
01418600 T  0070:0250:1
```

```
                IF OPERAND(TOS,V) THEN
IF ((V="EEEE"&"EEEE"[1:25:23]) AND NOT NORMAL) OR
                ((V="[[[["&"[[[["[1:25:23]) AND
                      NORMAL)              THEN
                WHILE OPERAND(TOS-1,R) AND R=V DO
                TOS:=TOS-1;
            END
            ELSE
            BEGIN
        WRITE(P[DBL],S,OCTAL(V←STAX[INX],[8:10]),
                OCTAL(SPREAD(V←V+VBED,CF,H,L)),
                OCTAL(H),OCTAL(L),
                OCTAL(R←SPREAD(V+1,H,L)),
                OCTAL(H),OCTAL(L));
                IF OCTAL(R,[30:6])="74" THEN
                IF WITHIN(MCPROG,MAXMCPROG,L,CF,SEG,ADR) THEN          %105-
                BEGIN
                    WRITE(LINE[*],C,"COMPLE","X SLEE","P AT  ",
                       OCTAL(SEG),SEG,OCTAL(ADR),ADR,R,[40:2]);
                    MOVE(NAMS[NAMF[SEG],CF],LINE[7],
                                NAMF[SEG],FF);
                    WRITE(P[DBL],15,LINE[*]);
                END
                ELSE WRITE(P);
            END;
            IF BOS=0 THEN BOS:=MAX(0,TOS-(IF NORMAL THEN MAXSTACKSIZE-1
                        ELSE 128)) ELSE
            IF BOS>TOS THEN BOS:=MAX(0,TOS-1);
            IF TOS-BOS GEQ MAXSTACKSIZE
                THEN BOS:=TOS-MAXSTACKSIZE+1;
      DUMPIT:%
            IF MYSTACKADR GTR 0 AND MYSTACKSIZE=0 THEN
            IF NOT MYSTACKDUMPED THEN
            IF MYSTACKADR LEQ TOS THEN
            IF MYSTACKADR GTR BOS THEN
            MYSTACKDUMPED:=TRUE;
            FOR I:=TOS STEP -1 UNTIL BOS DO
            BEGIN
                PRO:=-1;
RCW := FALSE;
WRITE(LINE[*],TRITEM,OCTAL(I),OCTAL(R:=SPREAD(I,H,L)),          %104-
  OCTAL(H),OCTAL(L));
IF CONTROLDESC(I,X) AND (X.CF ≠ 0) AND (X.FF GEQ BOS) AND       %104-
    (X.FF < TOS) AND I ≠ FUNNYMSCW THEN % IT LOOKS LIKE RCW.    %104-
   RCW := WITHINCODESEGMENT(L,CF, SEG, ADR, R,[40:2], ALINE[*]); %104-
IF NOT RCW % SO FAR ANYWAY                                      %104-
    AND NORMAL THEN % CHECK FOR OVERLAID NORMAL STATE CODE.     %104-
  IF OPERAND(I,X) AND X.[1:1] = 1 AND X.[3:1]=0 THEN            %104-
    IF RCW := ((Y := X.FF) GEQ BOS AND Y < I AND (SEG := X.CF) ≠ 0
        AND SEG < 1023 AND CONTROLDESC(Y,R)) THEN               %104-
      BEGIN                                                     %104-
        FORMATCODENAME(ALINE[*],3,SEG,R,CF,X,[10:2],X,X);       %104-
        FUNNYMSCW := Y;                                         %104-
      END;                                                      %104-
IF RCW THEN % FORMAT THE LINE.                                  %104-
    FORMATRCW(LINE[*], H, L, ALINE[*])                          %104-
ELSE                                                            %104-
```

```
01419000 T  0070:0251:2
01420000 T  0070:0252:0
01421000 T  0070:0255:2
01422000 T  0070:0256:1
01423000 T  0070:0257:3
01424000 T  0070:0261:2
01435000 T  0070:0265:2
01436000 T  0070:0265:2
01437000 T  0070:0265:2
01438000 T  0070:0265:3
01439000 T  0070:0273:2
01440000 T  0070:0278:0
01441000 T  0070:0283:2
01442000 T  0070:0287:3
01443000 T  0070:0295:2
01444000 P  0070:0297:2
01445000 T  0070:0300:0
01446000 T  0070:0300:1
01447000 T  0070:0315:2
01448000 T  0070:0328:0
01449000 T  0070:0330:1
01450000 T  0070:0332:1
01451000 T  0070:0337:2
01452000 T  0070:0337:2
01453000 T  0070:0341:3
01454000 T  0070:0341:3
01454100 T  0070:0344:1
01455000 T  0070:0348:0
01456000 T  0070:0353:3
01457000 T  0070:0354:0
01457100 T  0070:0357:2
01457110 T  0070:0357:2
01457120 T  0070:0358:1
01457130 T  0070:0359:3
01457140 T  0070:0361:2
01457150 T  0070:0362:0
01458000 T  0070:0363:3
01459000 T  0070:0365:2
01463000 T  0070:0365:2
01465100 P  0070:0366:0
01465200 C  0070:0366:1
01465300 C  0070:0378:0
01465400 C  0070:0386:0
01465500 C  0070:0390:0
01465600 C  0070:0392:1
01465700 C  0070:0397:2
01465800 C  0070:0397:2
01465900 C  0070:0398:0
01466000 P  0070:0402:1
01466100 C  0070:0407:2
01466200 C  0070:0410:1
01466300 C  0070:0411:2
01466400 C  0070:0414:1
01466500 C  0070:0415:3
01466600 C  0070:0415:3
01466700 C  0070:0415:3
01467600 C  0070:0418:0
```

```
IF CONTROLDESC(I,X) AND X.[2:4] = 0 AND (X.CF = 0 OR                    %104-
    I=FUNNYMSCW) THEN                                                   %104-
  FORMATMSCW(LINE[*], X.[6:9], X.[16:1], X.[17:1],                      %104-
    H)                                                                  %104-
ELSE                                                                    %104-
  IF DESCRIPTOR(I,X,Y) AND Y LEQ "50" THEN                              %104-
    FORMATDESC(LINE[*], Y.[41:1], X.[8:10], X.CF, X.FF, IF X.FF = I     
      THEN 1 ELSE IF X.FF ≠ 0 THEN IF DESCRIPTOR(X.FF,H,Y) AND (Y =     
        "40" OR (Y = "50" AND H.CF = X.CF)) THEN 2 ELSE 0 ELSE 0)       
  ELSE                                                                  %104-
    IF OPERAND(I,X) AND X≠0 THEN                                        %104-
      BEGIN                                                             %104-
        IF X.[3:6] ≠ 0 THEN                                             %104-
          WRITE(ALINE[*], <R20,10>, X)                                  %104-
        ELSE                                                            %104-
          WRITE(ALINE[*], <I15>, X);                                    %104-
        FORMATOP(LINE[*], X, ALINE[*]);                                 %104-
      END                                                               %104-
    ELSE                                                                %104-
      IF DESCRIPTOR(I,X,Y) AND B(Y.[40:1]) AND B(Y.[45:1]) THEN%104-
        BEGIN                                                           %104-
          IF B(Y.[41:1]) THEN % PRESENT PROGRAM DESC,                   %104-
            IF NOT WITHINCODESEGMENT(X,CF,SEG,ADR,0,ALINE[*]) THEN      
              ALINE[0] := REAL(NOT FALSE);                              %104-
          FORMATPROGDESC(LINE[*],Y.[46:2],Y.[41:1],ALINE[*]);           %104-
        END;                                                            %104-
WRITE(P,17,LINE[*]);                                                    %104-
      END;
  ERROWT:
  END DUMPING A STACK;

  INTEGER PROCEDURE SPREAD(AT,F,C);
      VALUE AT;
      INTEGER AT,F,C;
  BEGIN
      SPREAD:=CHRS(M[AT,ROW,AT,COL],0,3);
      C:=(F:=CHRS(M[AT,ROW,AT,COL],3,5)).CF;
      F:=F.FF;
  END SPREAD;
  PROCEDURE DUMPROGRAMS;
  BEGIN
      INTEGER MIX,TYP,PRTH,PRTF,H,F,C,FPB,SD,AIT,S;

  INTEGER L; ALPHA STARBLANK;
      INTEGER A;
      REAL JAR00;
  REAL STARS,JAR0;
      FORMAT HD(A5," = ",A6,2(X1,A5),X2,A1,X1,*A6),

       H1(X8,"PROGRAM:",X8,"/",X7,", MIX =",I2);

      INTEGER PCOL,SIZ,R0,CL,RP;
      REAL SEGS,SGM;
      REAL SEG,ADR;
      ARRAY LSTD[0:3];
```

```
        01467700 C  0070:0418:1
        01467800 C  0070:0422:1
        01467900 C  0070:0424:0
        01468000 P  0070:0427:3
        01468100 C  0070:0427:3
        01468200 C  0070:0428:0
        01468300 C  0070:0430:1
        01468400 C  0070:0435:3
        01468450 C  0070:0441:2
        01468500 C  0070:0447:2
        01468600 C  0070:0447:3
        01468700 C  0070:0450:0
        01468800 C  0070:0450:1
        01468900 C  0070:0451:3
72 IS     4 LONG, NEXT SEG   70
        01469000 P  0070:0461:2
        01469100 C  0070:0461:2
73 IS     4 LONG, NEXT SEG   70
        01469200 C  0070:0470:0
        01469300 C  0070:0472:0
        01469400 C  0070:0472:0
        01469500 C  0070:0472:0
        01469600 C  0070:0475:3
        01469700 C  0070:0476:0
        01469800 C  0070:0477:2
        01469900 C  0070:0480:1
        01470000 P  0070:0482:1
        01470100 C  0070:0485:3
        01470200 C  0070:0485:3
        01509000 T  0070:0489:3
        01510000 T  0070:0492:0
        01511000 T  0070:0492:0
70 IS   499 LONG, NEXT SEG   10
        01512000 T  0010:0266:1
        01513000 T  0010:0266:1
        01514000 T  0010:0266:1
        01515000 T  0010:0266:1
        01516000 T  0010:0266:1
        01517000 T  0010:0271:2
        01518000 T  0010:0277:2
        01519000 T  0010:0278:1
        01520000 T  0010:0280:0
        01521000 T  0010:0280:0
        01522000 T  0010:0280:0
START OF SEGMENT **********   74
        01523000 T  0074:0000:0
        01524000 T  0074:0000:0
        01525000 T  0074:0000:0
%103-   01526000 P  0074:0000:0
        01527000 T  0074:0000:0
START OF SEGMENT **********   75
        01528000 T  0075:0000:0
75 IS    26 LONG, NEXT SEG   74
        01529000 T  0074:0000:0
        01530000 T  0074:0000:0
        01531000 T  0074:0000:0
        01532000 T  0074:0000:0
```

```
FORMAT SEGH(X8,"PRESENT PROGRAM SEGMENTS"//
            "SEGMENT SIZE  AT    THRU"//),
BADSIZE(8("*")," INVALID SEGMENT DICTIONARY SIZE ",8("*")),
BADSDESC(8("*")," BAD SEGMENT DICTIONARY DESCRIPTOR ",8("*")),
    SEGMENT(X3,2(A4,X1),2(X1,A5),X2,A3,A1),
        PD(A5," = ",A6,X1,2(A5,X1),", R+",A4,", +",A4);

STARS:="****"; STARS,[6:18]:=STARS,[24:18];
IF DESCRIPTOR(JAR,JAR0,TYP) AND TYP="50" THEN
JAR0:=JAR0,CF;
IF (PRTOK:=PDATADESC(PRT,PRT0)) THEN
PRTH:=SPREAD(PRT,PRTF,PRT0);
IF PRT0>0 THEN
FOR MIX:=1 STEP 1 UNTIL MIXMAX DO
IF ONEMIX=0 OR ONEMIX=MIX THEN
IF PDATADESC(PRT0+MIX,R) THEN
BEGIN
NEXTPAGE;
    WRITE(LINE[*],H1,MIX);
    IF
        IF JAR0=0 THEN FALSE
        ELSE
        DESCRIPTOR(JAR0+MIX,JAR00,TYP) AND TYP="50"
        AND (JAR00:=JAR00,CF) GTR 0          THEN
    BEGIN
    MOVC(M[(A:=JAR00),ROW,A,COL],1,
         LINE[2],1,0,7);
    MOVC(M[(A:=A+1),ROW,A,COL],1,
         LINE[3],1,0,7);
END
ELSE
BEGIN
    MOVC(STARS,1,LINE[2],1,0,7);
    MOVC(STARS,1,LINE[3],1,0,7);
END;
WRITE(P[DBL],15,LINE[*]);
NEXTITEM;
WRITE(P,HD,
    OCTAL(PRT),
    OCTAL(PRTH),OCTAL(PRTF),OCTAL(PRT0)," ",
    2,"PRT[*,","*]    ",
    OCTAL(PRT0+MIX),
    OCTAL(SPREAD(PRT0+MIX,F,R)),OCTAL(F),OCTAL(R),
    IF F=0 THEN " " ELSE "*",
    3,"PRT[MI","X,*], ","CF=R  ",
    OCTAL(R+3),
    OCTAL(H:=SPREAD(R+3,F,FPB)),OCTAL(F),OCTAL(FPB),
    IF OCTAL(H,[30:6])="50" THEN " " ELSE "*",
    2,"R+3, F","PB    ",
    OCTAL(R+4),
    OCTAL(H:=SPREAD(R+4,F,SD)),OCTAL(F),OCTAL(SD),
    IF OCTAL(H,[30:6])="50" THEN " " ELSE "*",
    4,"R+4, S","EGMENT","," DICTI","ONARY ",
    OCTAL(R+6),
    OCTAL(H:=SPREAD(R+6,F,AIT)),OCTAL(F),OCTAL(AIT),
    IF BOOLEAN(H,[30:1]) THEN " " ELSE "*",
```

```
                01533000 T  0074:0001:3
START OF SEGMENT **********   76
                01534000 T  0076:0001:3
                01535000 T  0076:0001:3
                01536000 T  0076:0001:3
                01537000 T  0076:0001:3
                01538000 T  0076:0001:3
76 IS   76 LONG, NEXT SEG   74
                01539000 T  0074:0001:3
                01540000 T  0074:0004:1
                01541000 T  0074:0007:2
                01542000 T  0074:0009:2
                01543000 T  0074:0010:1
                01544000 T  0074:0013:2
                01545000 T  0074:0014:0
                01545100 T  0074:0017:2
                01546000 T  0074:0018:1
                01550000 T  0074:0020:1
                01551000 T  0074:0021:2
                01552000 T  0074:0025:2
                01553000 T  0074:0034:0
                01554000 T  0074:0034:0
                01555000 T  0074:0035:3
                01556000 T  0074:0036:0
                01557000 T  0074:0038:0
                01558000 T  0074:0041:2
                01559000 T  0074:0041:3
                01560000 T  0074:0044:1
                01561000 T  0074:0046:0
                01562000 T  0074:0050:0
                01563000 T  0074:0051:3
                01564000 T  0074:0051:3
                01565000 T  0074:0051:3
                01566000 T  0074:0052:0
                01567000 T  0074:0054:0
                01568000 T  0074:0056:1
                01569000 T  0074:0056:1
                01570000 T  0074:0060:1
                01571000 T  0074:0061:2
                01572000 T  0074:0063:3
                01573000 T  0074:0068:0
                01574000 T  0074:0077:3
                01575000 T  0074:0086:0
                01576000 T  0074:0089:2
                01577000 T  0074:0098:0
                01578000 T  0074:0102:0
                01579000 T  0074:0113:2
                01580000 T  0074:0116:0
                01581000 T  0074:0125:3
                01582000 T  0074:0130:1
                01583000 T  0074:0139:2
                01584000 T  0074:0142:0
                01585000 T  0074:0151:3
                01586000 T  0074:0156:1
                01587000 T  0074:0171:2
                01588000 T  0074:0174:0
                01589000 T  0074:0183:3
```

```
        2,"R+6, A","IT    ",
        OCTAL(R+7),
        OCTAL(H:=SPREAD(R+7,F,C)),OCTAL(F),OCTAL(C),
        IF H,[30:2]=3 AND F<R AND C=0 THEN " " ELSE "*",
        7,"R+7, L","AST MS","CW FOR"," WHICH"," MSFF ",
          "WAS FA","LSE   ",
        OCTAL(R+8),
        OCTAL(H:=SPREAD(R+8,F,C)),OCTAL(F),OCTAL(C),
        " ",
        2,"R+10, ","INCW  ",
        OCTAL(R+9),
        OCTAL(H:=SPREAD(R+9,F,C)),OCTAL(F),OCTAL(C),
        IF H=0 AND F=0 THEN " " ELSE "*",
        9,"R+11, ","LITERA","L FOR ","LAST C","OMMUNI",
          "CATE OR"," PROGR","AM REL","EASE  ",
        OCTAL(R+10),
        OCTAL(H:=SPREAD(R+10,S,C)),OCTAL(S),OCTAL(C),
        IF OCTAL(H,[30:6])="50" AND 0<S AND S<R AND R=C
        THEN " " ELSE "*",
        5,"R+12, ","FF = B","OTTOM ","OF THE"," STACK");
    IF CONTROLDESC(R+8,H) AND (LOCIRCW:=H,CF) GTR 1024
    AND H,FF GTR 1024 THEN ELSE LOCIRCW:=-1;
    NEXTITEM;
    PROGS[PROWS:=0,255]:=PCOL:=-1;
    IF DESCRIPTOR(R+4,SD,TYP) AND TYP="50" THEN
    IF OPERAND(SD:=SD,CF,SEGS) AND SEGS,[1:37]=0 THEN
    BEGIN
        WRITE(P,SEGH);
        FOR SEG:=1 STEP 1 UNTIL SEGS DO
        IF OPERAND(SD+SEG,SGM) AND
            (ADR:=SGM,FF)>1023 THEN
        BEGIN
            SIZ:=
    IF (OPERAND(ADR-1,SIZ)  AND
        SIZ,CF=SEG)                          OR
        (OPERAND(ADR-1,SIZ)      AND
        OPERAND(ADR-2,H)      AND
        H,[3:6]=7       AND
        SIZ,[8:10]=SGM,CF)       THEN
                SIZ,FF ELSE 0;
            IF BOOLEAN(SGM,[2:1]) THEN
        BEGIN
            L:=SGM,CF;
            STARBLANK:="*";
            FOR I:=0 STEP 1 UNTIL INTSPMAX DO
            IF INTSP[I],[8:10]=L THEN
            BEGIN
                    IF L=17 THEN SIZ:=4;
                STARBLANK:=" ";
                I:=INTSPMAX+2;
            END;
            WRITE(LINE[*],SEGMENT,OCTAL(SEG),OCTAL(SIZ),
                OCTAL(ADR),IF SIZ=0 THEN STARS ELSE
                OCTAL(ADR+SIZ-1),OCTAL(L),STARBLANK);
            IF L LEQ INTMAX THEN
            MOVE(INAMS[INAME[L],CF],LINE[4],INAME[L],FF);
            WRITE(P,7,LINE[*]);
```

```
01590000 T  0074:0187:3
01591000 T  0074:0196:0
01592000 T  0074:0199:2
01593000 T  0074:0208:1
01594000 T  0074:0215:2
01595000 T  0074:0232:0
01596000 T  0074:0238:0
01597000 T  0074:0241:2
01598000 T  0074:0250:1
01599000 T  0074:0252:1
01600000 T  0074:0261:2
01601000 T  0074:0264:0
01602000 T  0074:0273:3
01603000 T  0074:0278:1
01604000 T  0074:0296:0
01605000 T  0074:0308:0
01606000 T  0074:0311:2
01607000 T  0074:0320:1
01608000 T  0074:0325:2
01609000 T  0074:0328:1
01609100 T  0074:0356:0
01609200 T  0074:0359:2
01610000 T  0074:0365:2
01611000 T  0074:0365:3
01612000 T  0074:0368:1
01613000 T  0074:0371:3
01614000 T  0074:0376:0
01615000 T  0074:0376:1
01616000 T  0074:0379:3
01617000 T  0074:0381:2
01618000 T  0074:0382:1
01619000 T  0074:0384:1
01620000 T  0074:0385:2
01621000 T  0074:0385:2
01623000 T  0074:0386:1
01624000 T  0074:0388:0
01625000 T  0074:0389:3
01626000 T  0074:0391:2
01627000 T  0074:0392:1
01628000 T  0074:0395:2
01629000 T  0074:0397:3
01630000 T  0074:0398:0
01631000 T  0074:0398:1
01632000 T  0074:0400:0
01633000 T  0074:0400:1
01634000 T  0074:0402:0
01635000 T  0074:0403:3
01635100 T  0074:0404:0
01636000 T  0074:0406:0
01637000 T  0074:0406:1
01638000 T  0074:0408:0
01639000 T  0074:0410:0
01640000 T  0074:0420:1
01641000 T  0074:0425:3
01642000 T  0074:0435:2
01643000 T  0074:0436:0
01644000 T  0074:0440:0
```

```
END
      ELSE
      BEGIN
          IF (PCOL:=PCOL+1)=255 THEN
          BEGIN
              PROGS[PROWS,255]:=254;
              PROWS:=PROWS+1;
              PCOL:=0;
          END;
          PROGS[PROWS,PCOL]:=
              ADR&
              (IF SIZ=0 THEN 0 ELSE ADR+SIZ-1)[18:33:15]&
              SEG[8:38:10];
      END;
  END;
  IF (PROGS[PROWS,255]:=PCOL)≥0 THEN
  BEGIN
      FOR R0:=0 STEP 1 UNTIL PROWS DO
      SEQUENCE(PROGS[R0,*],PROGS[R0,255]);
      SEGS:=PROWS+1;
      FOR I:=0 STEP 1 UNTIL PROWS DO
      BEGIN
          SEGS:=SEGS+PROGS[I,255];
          LSTD[I]:=0;
      END;
      FOR SEG:=1 STEP 1 UNTIL SEGS DO
      BEGIN
          R0:=0;
          IF PROWS>0 THEN
          FOR I:=1 STEP 1 UNTIL PROWS DO
          IF LSTD[R0]=255 OR
              (LSTD[I]<255 AND
              PROGS[I,LSTD[I]].CF<PROGS[R0,LSTD[R0]].CF)
                                                    THEN
          R0:=I;
          WRITE(P,SEGMENT,
                  OCTAL((SGM:=PROGS[R0,LSTD[R0]]).[8:10]),
                  IF SGM.FF=0 THEN STARS ELSE
                      OCTAL(SGM.FF-SGM.CF+1),
                  OCTAL(SGM.CF),IF SGM.FF=0 THEN
                  STARS ELSE OCTAL(SGM.FF));
          LSTD[R0]:=LSTD[R0]+1;
      END;
  END;
END
    ELSE WRITE(P[DBL],BADSIZE) ELSE WRITE(P[DBL],BADSDESC);
     IF (AIT:=MIXSTK[MIX]) = 0 THEN
        STAX[0]:=
            (R-1)&
        (S-1)[18:33:15]&
          1[7:47:1] ELSE
    BEGIN
     IF STAX[AIT].FF=0 THEN STAX[AIT].FF:=S-1;
    MIXSTK[MIX]:=0;
    END;
     DUMPSTACK(AIT);
IF ONEMIX NEQ 0 AND ONEMIX=MIX THEN MIX:=MIXMAX+1; % XIT
```

```
01645000 T  0074:044:0
01646000 T  0074:044:0
01647000 T  0074:044:0
01648000 T  0074:044:1
01649000 T  0074:046:1
01650000 T  0074:047:2
01651000 T  0074:049:2
01652000 T  0074:050:0
01653000 T  0074:051:2
01654000 T  0074:051:2
01655000 T  0074:052:0
01656000 T  0074:052:1
01657000 T  0074:056:0
01658000 T  0074:057:3
01659000 T  0074:057:3
01660000 T  0074:059:3
01661000 T  0074:062:0
01662000 T  0074:062:1
01663000 T  0074:064:0
01664000 T  0074:069:2
01665000 T  0074:070:0
01666000 T  0074:071:2
01667000 T  0074:071:2
01668000 T  0074:073:2
01669000 T  0074:074:1
01670000 T  0074:076:1
01671000 T  0074:078:0
01672000 T  0074:078:0
01673000 T  0074:078:1
01674000 T  0074:079:3
01675000 T  0074:081:2
01676000 T  0074:082:0
01677000 T  0074:083:2
01678000 T  0074:086:0
01679000 T  0074:087:3
01680000 T  0074:091:2
01681000 T  0074:093:3
01682000 T  0074:099:2
01683000 T  0074:103:3
01684000 T  0074:107:2
01684100 T  0074:112:0
01685000 T  0074:118:0
01686000 T  0074:120:0
01687000 T  0074:122:1
01688000 T  0074:122:1
01689000 T  0074:122:1
01690000 T  0074:129:3
01691000 T  0074:131:2
01692000 T  0074:132:0
01693000 T  0074:132:1
01694000 T  0074:133:3
01694100 T  0074:135:2
01695000 T  0074:135:3
01695100 T  0074:140:1
01695200 T  0074:141:3
01696000 T  0074:141:3
01697000 T  0074:142:1
```

```
    END;
    IF PRTOK THEN
    FOR MIX:=1 STEP 1 UNTIL MIXMAX DO
    IF(AIT:=MIXSTK[MIX]) NEQ 0 THEN STAX[AIT].[7:1]:=0;
    COMMENT THE ABOVE WILL CAUSE ANY STACKS ASLEEP, ASSOCIATED WITH
    A MIX INDEX, BUT NOT DUMPED IN THIS PROCEDURE,TO BE DUMPED
    LATER AS CONTROL STATE STATES,  ;
END DUMPING NORMAL STATE PROGRAM INFO;

PROCEDURE DUMPCONTROLSTACKS;
BEGIN
    FORMAT H(X8,"CONTROL STATE STACKS");



    REAL V,R,A;
     INTEGER INX,I;
    NEXTPAGE;
    WRITE(P[DBL],H);
     STAX[0]:=127&64[18:33:15];
    DUMPSTACK(0);
    NEXTPAGE;
     I:=-1;
    IF DESCRIPTOR(ISTACK,A,R) AND R="50" THEN
    IF (V:=A.[8:10])>0 AND (A:=A.CF)>0 THEN
     I:=A+V-1;
 IF BEDSTK GTR 0 THEN
   FOR INX:=1 STEP 1 UNTIL BEDSTK DO
      IF BOOLEAN(STAX[INX].[7:1]) THEN ELSE
   BEGIN
        R:=STAX[INX].CF;
        IF R GTR A AND R LEQ I THEN
        BEGIN % ISTACK ASLEEP
             STAX[INX].FF:=A;
             NEXTITEM;
             DISPLAY(ISTACK,TRUE);
             I:=0;% SO DONT DUMP ISTACK TWICE
        END SPECIAL ISTACK STUFF;
        DUMPSTACK(INX);
   NEXTPAGE;
   END OF DUMPING BEDDED CONTROL STATE STACKS ;
   IF I GTR 0 THEN % ISTACK AWAKE
   BEGIN
        STAX[0]:=I&A[18:33:15];
        NEXTITEM;
        DISPLAY(ISTACK,TRUE);
        DUMPSTACK(0);
      NEXTPAGE;
     END;
IF MAXSTK GTR BEDSTK THEN %MORE C.S. STACKS TO DP
FOR INX:=BEDSTK+1 STEP 1 UNTIL MAXSTK DO
BEGIN
  DUMPSTACK(INX);
  IF INX LSS MAXSTK THEN NEXTPAGE;
END;
IF NOT MYSTACKDUMPED THEN
IF MYSTACKADR GTR 0 THEN
```

```
01698000 T  0074:0546:0
01698100 T  0074:0548:0
01698200 T  0074:0549:2
01698300 T  0074:0551:2
01698400 T  0074:0557:3
01698500 T  0074:0557:3
01698600 T  0074:0557:3
01699000 T  0074:0557:3
  74 IS  572 LONG, NEXT SEG   10
01700000 T  0010:0280:0
01701000 T  0010:0280:0
01702000 T  0010:0280:0
START OF SEGMENT **********   77
START OF SEGMENT **********   78
  78 IS    8 LONG, NEXT SEG   77
01703000 T  0077:0000:0
01704000 T  0077:0000:0
01705000 T  0077:0000:0
01706000 T  0077:0004:0
01707000 T  0077:0007:2
01708000 T  0077:0008:1
01709000 T  0077:0009:3
01710000 T  0077:0013:3
01711000 T  0077:0014:1
01712000 T  0077:0017:2
01713000 T  0077:0021:2
01714000 T  0077:0023:3
01715000 T  0077:0024:0
01716000 T  0077:0026:0
01717000 T  0077:0027:3
01718000 T  0077:0028:0
01719000 T  0077:0029:3
01720000 T  0077:0031:2
01721000 T  0077:0031:3
01722000 T  0077:0033:3
01723000 T  0077:0034:0
01724000 T  0077:0035:3
01725000 T  0077:0036:1
01726000 T  0077:0036:1
01727000 T  0077:0037:2
01728000 T  0077:0041:2
01729000 T  0077:0043:3
01730000 T  0077:0044:0
01731000 T  0077:0044:1
01732000 T  0077:0046:1
01733000 T  0077:0047:2
01734000 T  0077:0048:1
01735000 T  0077:0049:2
01736000 T  0077:0053:2
01737000 T  0077:0053:2
01738000 T  0077:0054:0
01739000 T  0077:0058:1
01740000 T  0077:0058:1
01741000 T  0077:0059:3
01742000 T  0077:0064:1
01742100 T  0077:0065:2
01742110 T  0077:0065:3
```

```
BEGIN % SPECIAL DUMP OF USER SELECTED "STACK" AREA
  STAX[0]:=MYSTACKADR & MAX(0,MYSTACKADR+1 -
            (IF MYSTACKSIZE NEQ 0 THEN MYSTACKSIZE ELSE 200)) CTF &
            1[4:47:1];
  DUMPSTACK(0);
  MYSTACKDUMPED:=TRUE;
END;
  END DUMPING CONTROL STATE STACKS;

  DEFINE MAXMESSAGES=100#;
      STREAM PROCEDURE MOVD(S,D,W);
          VALUE W;
      BEGIN
          LABEL EXIT;
          SI:=S; DI:=D;
          W(8(IF SC="." THEN JUMP OUT 2 TO EXIT; DS:=CHR));
          SI:=SI-1; DI:=DI-1;
          EXIT: DS:=CHR;
      END MOVD;
      PROCEDURE PRINTQUEUE(PROC,CODE); VALUE PROC,CODE;
                INTEGER PROC,CODE;
      BEGIN
          BOOLEAN B;

          INTEGER C;
         REAL A,R,S,T,J,USERCODE;
          FORMAT Q1(A5," = ",2(O,X1),I2,"/",I2,X89),

                 Q2(A5," = ",2(O,X1),I2,"/",I2,X1,"USER=",A1,A6,X76)
                ,Q3(A5,X3,2(2(O,X1),X1),X2,X72)
                 ;

          NEXTITEM;
          C:=CODE.FF;
          CODE:=0 & CODE CTC;
          IF CODE=1 THEN
          BEGIN
              DISPLAY(MIXMASK,FALSE);
              DISPLAY(INFOMASK1,FALSE);
              DISPLAY(INFOMASK2,FALSE);
              DISPLAY(CCMASK1,FALSE);
              DISPLAY(CCMASK2,FALSE);
          END;
         DISPLAY(PROC,FALSE);
          I:=0;
          IF OPERAND(PROC,A) AND A.[9:9]=511 AND (R:=A.CF) NEQ PROC
     THEN
          DO BEGIN
              IF NOT OPERAND(R,A) THEN I:=MAXMESSAGES+1 ELSE
              CASE CODE OF
              BEGIN
                 BEGIN % 0(E,G, STATIONMESSAGEHOLDER)
                 WRITE(LINE[*],Q1, OCTAL(R),OCTAL(HIHALF(R)),
                       OCTAL(LOHALF(R)),A.[9:4],A.[14:4]);
MOVD(M[(R+1 +C),ROW,(R+1 +C),COL],LINE[4],
                       MIN(11,MAXCOR-11));
                 WRITE(P,15,LINE[*]);
```

```
                                    01742120 T  0077:0067:2
                                    01742130 T  0077:0067:3
                                    01742140 T  0077:0069:3
                                    01742150 T  0077:0074:0
                                    01742160 T  0077:0075:3
                                    01742165 T  0077:0076:0
                                    01742170 T  0077:0077:2
                                    01743000 T  0077:0077:2
                                77 IS   81 LONG, NEXT SEG   10
                                    01747000 T  0010:0280:0
                                    01750000 T  0010:0280:0
                                    01751000 T  0010:0280:0
                                    01752000 T  0010:0280:0
                                    01753000 T  0010:0281:2
                                    01754000 T  0010:0281:2
                                    01755000 T  0010:0281:3
                                    01756000 T  0010:0284:1
                                    01757000 T  0010:0285:2
                                    01757100 T  0010:0286:0
                                    01757300 T  0010:0286:1
                                    01757310 T  0010:0286:1
                                    01757320 T  0010:0286:1
                                    01757330 T  0010:0286:1
                                START OF SEGMENT **********   79
                                    01757332 T  0079:0000:0
                                    01757335 T  0079:0000:0
                                    01757340 T  0079:0000:0
                                START OF SEGMENT **********   80
                                    01757350 T  0080:0000:0
                                    01757352 T  0080:0000:0
                                    01757355 T  0080:0000:0
                                 80 IS   44 LONG, NEXT SEG   79
                                    01757400 T  0079:0000:0
                                    01757410 T  0079:0000:1
                                    01757415 T  0079:0001:3
                                    01757420 T  0079:0003:2
                                    01757425 T  0079:0003:3
                                    01757434 T  0079:0004:0
                                    01757436 T  0079:0005:3
                                    01757440 T  0079:0007:2
                                    01757450 T  0079:0008:1
                                    01757455 T  0079:0010:0
                                    01757460 T  0079:0011:3
                                    01757470 T  0079:0011:3
                                    01757500 T  0079:0013:2
                                    01757505 T  0079:0013:3
                                    01757510 T  0079:0017:3
                                    01757520 T  0079:0018:0
                                    01757530 T  0079:0019:2
                                    01757550 T  0079:0022:0
                                    01757560 T  0079:0022:1
                                    01757570 T  0079:0023:2
                                    01757580 T  0079:0023:2
                                    01757590 T  0079:0034:0
                                    01757600 T  0079:0045:2
                                    01757610 T  0079:0050:0
                                    01757620 T  0079:0053:3
```

```
        END; % 0
        BEGIN % 1(LOOKQ)
        B:=OPERAND(R+1,USERCODE);
        WRITE(LINE[*],Q2,OCTAL(R),OCTAL(HIHALF(R)),
              OCTAL(LOHALF(R)),A.[9:4],A.[14:4],
              IF B THEN USERCODE.[6:6] ELSE " ",
              IF B THEN USERCODE.[12:36] ELSE "      ");
        WRITE(P,15,LINE[*]);
        PRINT(R+REAL(B),R+9);
        END; % 1
        BEGIN % 2(E.G. DC19Q)
        WRITE(LINE[*],Q1,OCTAL(R),OCTAL(HIHALF(R)),
              OCTAL(LOHALF(R)),A.[9:4],A.[14:4]);
        WRITE(P,15,LINE[*]);
        PRINT(R+1,R+5);
        IF OPERAND(R+1,S) AND(S:=S,CF) NEQ R+1 THEN
        DO BEGIN
             WRITE(LINE[*],Q3,OCTAL(S),OCTAL(HIHALF(S)),
                   OCTAL(LOHALF(S)),OCTAL(HIHALF(S+1)),
                   OCTAL(LOHALF(S+1)));
             IF OPERAND(S+1,T) AND T.CF NEQ S+1 THEN
             MOVD(M[(S+2),ROW,(S+2),COL],LINE[6],
                  MIN(9,MAXCOR-9));
             WRITE(P,15,LINE[*]);
             END UNTIL NOT OPERAND(S,T) OR
                       (S:=T.CF)=R+1 OR
                       (J:=J+1) GTR MAXMESSAGES; J:=0;
          END; % 2
        END CASE;


        END UNTIL I:=I+1 GTR MAXMESSAGES OR
                  (R:=A.CF)=PROC OR (A.[9:4]=0);
    END PRINTQUEUE;

PROCEDURE DUMPMCPINFO;
BEGIN
    REAL R,A,N,L,TA,TS,MA,MS,LA,LS,RA,RS,PA,PS;

    INTEGER TYP,S,C;
    BOOLEAN TINUOK;
    BOOLEAN COMSAVE;
    FORMAT SE(A3," = ",2(O,X1),X24,"(",O,X1,O,")"),

NULLMESSAGES("### NO SPO MESSAGES QUEUED"),
NULLSLATE("### NO INDEPENDENT-RUNNERS TO BE INITIATED"),
          NOMEMXI(  ,"TOGLE.[15:6]",X2," = ",A2,", NOMEM"),
          TFXI(X8,"BIT ",I2," = ",I1,", ",X28),
            TF(X8,"BIT ",I2," = ",I1,", ",*A6/);

    BOOLEAN STREAM PROCEDURE BITON(W,B);
        VALUE B;
    BEGIN
        S:=W; SKIP B SB;
        IF SB THEN TALLY:=1;
        BITON:=TALLY;
    END;
```

```
01757630 T  0079:0057:3
01757700 T  0079:0058:0
01757710 T  0079:0058:0
01757720 T  0079:0060:0
01757730 T  0079:0071:2
01757740 T  0079:0079:2
01757750 T  0079:0083:2
01757760 T  0079:0091:2
01757770 T  0079:0095:3
01757780 T  0079:0099:3
01757800 T  0079:0100:0
01757810 T  0079:0100:0
01757820 T  0079:0111:2
01757830 T  0079:0122:0
01757840 T  0079:0126:1
01757845 T  0079:0130:1
01757850 T  0079:0134:1
01757855 T  0079:0136:0
01757860 T  0079:0146:0
01757865 T  0079:0152:1
01757870 T  0079:0159:2
01757875 T  0079:0162:1
01757880 T  0079:0167:2
01757885 T  0079:0170:1
01757890 T  0079:0174:1
01757895 T  0079:0176:0
01757897 T  0079:0178:1
01757900 T  0079:0181:3
01757920 T  0079:0182:0
START OF SEGMENT **********   81
 81 IS     4 LONG, NEXT SEG   79
01757950 T  0079:0182:1
01757960 T  0079:0184:1
01757980 T  0079:0188:1
 79 IS   192 LONG, NEXT SEG   10
01758000 T  0010:0286:1
01758100 T  0010:0286:1
01758200 T  0010:0286:1
START OF SEGMENT **********   82
01758300 T  0082:0000:0
01758400 T  0082:0000:0
01758500 T  0082:0000:0
01759000 T  0082:0000:0
START OF SEGMENT **********   83
01760000 T  0083:0000:0
01761000 T  0083:0000:0
01762000 T  0083:0000:0
01763000 T  0083:0000:0
01764000 T  0083:0000:0
 83 IS    64 LONG, NEXT SEG   82
01765000 T  0082:0000:0
01766000 T  0082:0000:0
01767000 T  0082:0000:0
01768000 T  0082:0000:0
01769000 T  0082:0000:1
01770000 T  0082:0001:3
01771000 T  0082:0001:3
```

```
ARRAY TB[0:143];                                                                01772000 T  0082:0002:1
    REAL UA,US,IA,IS,FA,FS;                                                     01773000 T  0082:0004:1
    FORMAT LUN(A3),                                                             01774000 T  0082:0004:1
                                                                      START OF SEGMENT *********   84
       LQUEHDR("EL",X2,"IOQUE=INDEX", X2,"DISK=ADDRESS"),                       01774100 T  0084:0004:1
       BADLQUENTRY(A2,"  # BAD ENTRY IS  ",2(O,X1)),                            01774110 T  0084:0004:1
       BADLQUE("## THE LQE IS INCORRECT"),                                      01774120 T  0084:0004:1
       FLQUE(A2,X6,A3,X8,A1,A6),                                                01774130 T  0084:0004:1
       NULLQUE("### NO ENTRIES IN THE LQUE"),                                   01774140 T  0084:0004:1
        FT(A1,X1,A2,X1,A1,X1,A3,X1,A2,X1,A1,X1,A2,X1,A3),                       01775000 T  0084:0004:1
        RT(A3,X1,A2,X1,A4,X1,A6,X1,A3),                                         01776000 T  0084:0004:1
        PT(A1,X1,A1,X1,A5,X1,A5,X1,A6),                                         01777000 T  0084:0004:1
        IOATH(X8,"FIELDS OF WORDS IN THE I/O ASSIGNMENT TABLES:"//              01778000 T  0084:0004:1
        X8,"TINU"/                                                              01779000 T  0084:0004:1
        X12,"  [0:3]"/                                                          01780000 T  0084:0004:1
        X12,"1 [3:5]   HARDWARE UNIT NUMBER"/                                   01781000 T  0084:0004:1
        X12,"  [8:3]"/                                                          01782000 T  0084:0004:1
        X12,"2 [11:7]  POWER OF 2"/                                             01783000 T  0084:0004:1
        X12,"  [18:6]"/                                                         01784000 T  0084:0004:1
        X12,"3 [24:1]  IN=0, OUT=1"/                                            01785000 T  0084:0004:1
        X12,"  [25:5]"/                                                         01786000 T  0084:0004:1
        X12,"4 [30:18] UNIT MNEMONIC"//                                         01787000 T  0084:0004:1
        X8,"RDCTABLE"/                                                          01788000 T  0084:0004:1
        X12,"  [0:8]"/                                                          01789000 T  0084:0004:1
        X12,"1 [8:6]   MIX INDEX IF ASSIGNED"/                                  01790000 T  0084:0004:1
        X12,"2 [14:10] REEL NUMBER"/                                            01791000 T  0084:0004:1
        X12,"3 [24:17] CREATION DATE"/                                          01792000 T  0084:0004:1
        X12,"4 [41:7]  CYCLE"//                                                 01793000 T  0084:0004:1
        X8,"PRNTABLE"/                                                          01794000 T  0084:0004:1
        X12,"  [0:1]"/                                                          01795000 T  0084:0004:1
        X12,"1 [1:1]   IF WRITE RING PRESENT"/                                  01796000 T  0084:0004:1
        X12,"  [2:13]"/                                                         01797000 T  0084:0004:1
        X12,"2 [15:15] ADDRESS OF TOP I/O DESCRIPTOR"/                          01798000 T  0084:0004:1
        X12,"3 [30:18] PHYSICAL REEL NUMBER"//                                  01799000 T  0084:0004:1
        "LUN",X6,                                                               01800000 T  0084:0004:1
        "TINU",X24,                                                             01801000 T  0084:0004:1
        "MULTITABLE",X4,                                                        01802000 T  0084:0004:1
        "LABELTABLE",X4,                                                        01803000 T  0084:0004:1
        "RDCTABLE",X24,                                                         01804000 T  0084:0004:1
        "PRNTABLE"/                                                             01805000 T  0084:0004:1
        X9,                                                                     01806000 T  0084:0004:1
        X2,"1",X4,"2",X6,"3",X4,"4",X8,                                         01807000 T  0084:0004:1
        X14,X14,                                                                01808000 T  0084:0004:1
        X4,"1",X2,"2",X4,"3",X6,"4",X8,                                         01809000 T  0084:0004:1
        X2,"1",X7,"2",X5,"3",X5);                                               01810000 T  0084:0004:1
                                                                      84 IS  265 LONG, NEXT SEG   82
    BOOLEAN PROCEDURE VARIFY(WHAT,A,S,B);                                       01811000 T  0082:0004:1
        VALUE WHAT,B;                                                           01812000 T  0082:0004:1
    INTEGER WHAT;                                                               01813000 T  0082:0004:1
    REAL A,S;                                                                   01814000 T  0082:0004:1
    BOOLEAN B;                                                                  01814500 T  0082:0004:1
BEGIN                                                                           01815000 T  0082:0004:1
        DISPLAY(WHAT,B);                                                        01816000 T  0082:0004:1
    VARIFY:=                                                                    01817000 T  0082:0007:2
        DESCRIPTOR(WHAT,A,S) AND                                                01818000 T  0082:0007:2
        S="50" AND                                                              01819000 T  0082:0009:2
        (S:=A,[8:10])>0 AND                                                     01820000 T  0082:0010:0
```

```
                (A:=A,CF)>0 AND
                (A+S-1)<MAXCOR;
        END VARIFY;
        DEFINE VERIFY(VERIFY1,VERIFY2,VERIFY3)=
                VARIFY(VERIFY1,VERIFY2,VERIFY3,FALSE)#;
        FORMAT !OQSH("FIELDS OF WORDS IN THE I/O-QUEUE TABLES"//

            X8,"UNIT"/
            X12,"  [0:1]"/
            X12,"1 [1:4]   UNIT TYPE"/
            X12,"2 [5:8]   ERROR FIELD"/
            X12,"3 [13:1]  UNIT NOT READY"/
            X12,"4 [14:1]  ERROR FLAG"/
            X12,"5 [15:1]  WAITING FOR AN I/O CHANNEL"/
            X12,"6 [16:2]  I/O IN PROCESS"/
            X12,"7 [18:15] INDEX OF FIRST I/O REQUEST"/
            X12,"8 [33:15] INDEX OF LAST I/O REQUEST"//
            X8,"LOCATQUE"/
            X12,"  [0:3]   = 5, DESCRIPTOR BITS"/
            X12,"1 [3:5]   MIX INDEX"/
            X12,"  [8:4]"/
            X12,"2 [12:6]  LOGICAL UNIT NUMBER"/
            X12,"3 [18:15] INDEX OF NEXT I/O REQUEST"/
            X12,"4 [33:15] ADDRESS OF I/O DESCRIPTOR"//
    "LUN/  TINU  ",
            "UNIT",X31,
            "IOQUE",X19,
            "LOCATQUE",X21,
            "FINALQUE",X11/
            "INDEX",X7,
            X2,"1",X2,"2",X3,"3",X1,"4",X1,"5",
                X1,"6",X1,"7",X5,"8",X11,X24,
            X2,"1",X5,"2",X2,"3",X5,"4"//
                                                ),
    IFO(A5,X2,A3),
            UFO(A1,X1,A2,X1,A3,4(X1,A1),2(X1,A5)),
            WFO(0,X1,0),
            LFO(A1,3(X1,A2),2(X1,A5));

        NEXTPAGE;
        COMSAVE:=COMMON;
        COMMON:=BOOLEAN(64); % OCTAL ONLY WHEN CALL PRINT
        DISPLAY(TOGLE,FALSE);
        IF OPERAND(TOGLE,R) THEN
        BEGIN
    WRITE(P[DBL],NOMEMXI,OCTAL(R,[15:6]));
    FOR I:=1 STEP 1 UNTIL 11 DO
        IF BITON(R,I) THEN WRITF(P,TF,I,1,-1);
        FILL TB[*] WITH
        "          ","          ","          ",%00%BIT 00(NEVER USED)

        "          ","          ","          ",%03%BIT 01
        "          ","          ","          ",%06%BIT 02
        "          ","          ","          ",%09%BIT 03
        "          ","          ","          ",%12%BIT 04
        "          ","          ","          ",%15%BIT 05
        "          ","          ","          ",%18%BIT 06
```

```
01821000 T  0082:0012:0
01822000 T  0082:0014:1
01823000 T  0082:0017:2
01823100 T  0082:0019:2
01823300 T  0082:0019:2
01824000 T  0082:0019:2
START OF SEGMENT *********   85
01825000 T  0085:0019:2
01826000 T  0085:0019:2
01827000 T  0085:0019:2
01828000 T  0085:0019:2
01829000 T  0085:0019:2
01830000 T  0085:0019:2
01831000 T  0085:0019:2
01832000 T  0085:0019:2
01833000 T  0085:0019:2
01834000 T  0085:0019:2
01835000 T  0085:0019:2
01836000 T  0085:0019:2
01837000 T  0085:0019:2
01838000 T  0085:0019:2
01839000 T  0085:0019:2
01840000 T  0085:0019:2
01841000 T  0085:0019:2
01842000 T  0085:0019:2
01843000 T  0085:0019:2
01844000 T  0085:0019:2
01845000 T  0085:0019:2
01846000 T  0085:0019:2
01847000 T  0085:0019:2
01848000 T  0085:0019:2
01849000 T  0085:0019:2
01850000 T  0085:0019:2
01851000 T  0085:0019:2
01852000 T  0085:0019:2
01853000 T  0085:0019:2
01854000 T  0085:0019:2
01855000 T  0085:0019:2
 85 IS  202 LONG, NEXT SEG   82
01856000 T  0082:0019:2
01856100 T  0082:0024:0
01856200 T  0082:0024:1
01857000 T  0082:0025:3
01858000 T  0082:0027:2
01859000 T  0082:0028:0
01860000 T  0082:0028:1
01861000 T  0082:0038:0
01862000 T  0082:0039:2
01863000 T  0082:0055:3
01864000 T  0082:0056:0
START OF SEGMENT *********   86
01865000 T  0082:0057:2
01866000 T  0082:0057:2
01867000 T  0082:0057:2
01868000 T  0082:0057:2
01869000 T  0082:0057:2
01870000 T  0082:0057:2
```

```
"        ","        ","        ",%21%BIT 07
"        ","        ","        ",%24%BIT 08
"        ","        ","        ",%27%BIT 09
"        ","        ","        ",%30%BIT 10
"        ","        ","        ",%33%BIT 11
"DIRECTOR","YTOG    ","        ",%36%BIT 12
"SEPTICTA","NKING   ","        ",%39%BIT 13
"BREAKTOG","        ","        ",%42%BIT 14
"        ","        ","        ",%45%BIT 15
"        ","        ","        ",%48%BIT 16
"        ","        ","        ",%51%BIT 17
"        ","        ","        ",%54%BIT 18
"        ","        ","        ",%57%BIT 19
"        ","        ","        ",%60%BIT 20
"CDFREE  ","        ","        ",%63%BIT 21
"FINDINGA","DDRESS  ","        ",%66%BIT 22
"SCRATCHD","IRECTORY","READY   ",%69%BIT 23
"MCPFREE ","        ","        ",%72%BIT 24
"INQPTSTO","PPED    ","        ",%75%BIT 25
"DCQPTSTO","PPED    ","        ",%78%BIT 26
"DCWAITIN","G       ","        ",%81%BIT 27
"SMWSTOPP","ED      ","        ",%84%BIT 28
"NINETEEN","NOTREADI","NG      ",%87%BIT 29
"STARTOG ","        ","        ",%90%BIT 30
"EGGSELEC","TSTOPPED","        ",%93%BIT 31
"REMOTELO","GFREE   ","        ",%96%BIT 32
"SPOEDNUL","LOG     ","        ",%99%BIT 33
"INTFREE ","        ","        ",%102%BIT 34
"QTRDY   ","        ","        ",%105%BIT 35
"NOBACKTA","LK      ","        ",%108%BIT 36
"KEYBOARD","READY   ","        ",%111%BIT 37
"BUMPTUTI","TIME    ","        ",%114%BIT 38
"ABORTABL","E       ","        ",%117%BIT 39
"NSECONDR","EADY    ","        ",%120%BIT 40
"HOLDFREE","        ","        ",%123%BIT 41
"USERDISK","READY   ","        ",%126%BIT 42
"STOREDY ","        ","        ",%129%%BIT 43
"STACKUSE","        ","        ",%132%BIT 44
"SHEETFRE","E       ","        ",%135%BIT 45
"STATUSBI","T       ","        ",%138%BIT 46
"HP2TOG  ","        ","        ";%141%BIT 47

FOR I:=12,13,14,21 STEP 1 UNTIL 47 DO
BEGIN
    WRITE(LINE[*],TFXI,I,REAL(BITON(R,I)));
    MOVC(TB[3*I],0,LINE[2],4,3,0);
    WRITE(P[DBL],6,LINE[*]);
END;
    END;
    DISPLAY(NOPROCESSTOG,FALSE);
    NEXTITEM;
    DUMPARRAY(TAR);
    DISPLAY(TOGLE,FALSE); % FOR COMPARISION
    NEXTITEM;
    NEXTPAGE;
    DISPLAY(OPTION,FALSE);
    IF OPERAND(OPTION,R) THEN
    BEGIN
```

```
01871000 T  0082:0057:2
01872000 T  0082:0057:2
01873000 T  0082:0057:2
01874000 T  0082:0057:2
01875000 T  0082:0057:2
01876000 T  0082:0057:2
01877000 T  0082:0057:2
01878000 T  0082:0057:2
01879000 T  0082:0057:2
01880000 T  0082:0057:2
01881000 T  0082:0057:2
01882000 T  0082:0057:2
01883000 T  0082:0057:2
01884000 T  0082:0057:2
01885000 T  0082:0057:2
01886000 T  0082:0057:2
01887000 T  0082:0057:2
01888000 T  0082:0057:2
01889000 T  0082:0057:2
01890000 T  0082:0057:2
01891000 T  0082:0057:2
01892000 T  0082:0057:2
01893000 T  0082:0057:2
01894000 T  0082:0057:2
01895000 T  0082:0057:2
01896000 T  0082:0057:2
01897000 T  0082:0057:2
01898000 T  0082:0057:2
01899000 T  0082:0057:2
01900000 T  0082:0057:2
01901000 T  0082:0057:2
01902000 T  0082:0057:2
01903000 T  0082:0057:2
01904000 T  0082:0057:2
01905000 T  0082:0057:2
01906000 T  0082:0057:2
01907000 T  0082:0057:2
01908000 T  0082:0057:2
01909000 T  0082:0057:2
01910000 T  0082:0057:2
01911000 T  0082:0057:2
86 IS  144 LONG, NEXT SEG   82
01912000 T  0082:0057:2
01913000 T  0082:0067:2
01914000 T  0082:0067:2
01915000 T  0082:0079:2
01916000 T  0082:0082:0
01917000 T  0082:0086:1
01918000 T  0082:0087:2
01918200 T  0082:0087:2
01919000 T  0082:0088:1
01919100 T  0082:0089:2
01919200 T  0082:0090:0
01919300 T  0082:0091:3
01919400 T  0082:0092:0
01920000 T  0082:0096:0
01921000 T  0082:0097:3
01922000 T  0082:0098:1
```

```
IF BITON(R,1) THEN
WRITE(P,TF,I,1,-1);
FILL TB[*] WITH
    "USEDRA","      ",

    "USEDRB","      ",
    "BOJMES","S     ",
    "EOJMES","S     ",
    "OPNMES","S     ",
    "TERMGO","      ",
    "GIVEDA","TE    ",
    "GIVETI","ME    ",
    "SAMEBR","EAKTAP","E     ",
    "AUTOPR","INT   ","      ",
    "CLEARW","RS, ST","OPUNT ",
    "DISCON","DC, NO","TIFYOP",
    "COPNME","SS    ",
    "CLOSEM","ESS   ",
    "ERRORM","SG    ",
    "RETMSG","      ",
    "LIBMSG","      ",
    "SCHEDM","SG    ",
    "SECMSG","      ",
    "DSKTOG","      ",
    "RELTOG","      ",
    "PBDREL","      ",
    "CHECKL","INK   ",
    "DISKMS","G     ",
    "NOT US","ED    ",
    "LIBERR","      ", % BIT 22
    "USEPBD","      ",
    "SVPBT ","      ",
    "RSTOG ","      ",
    "AUTOUN","LD    ",
    "RNALL ","      ",
    "CODEOL","AY    ",
    "COREST","      ",
    "DATAOL","AY    ",
    "HALT  ","      ", % BIT 13
    "NOT US","ED    ",
    "NOT US","ED    ",
    "NOT US","ED    ",
    "NOT US","ED    ",
    "STOPTE","ST    ",
    "PUNCHL","CK    ",
    "CDONLY","      ",
    "PKTONL","Y     ",
    "SEPARA","TE    ",
    "NOT US","ED    ", % BIT 3
    "MOD310","S     ", % BIT 2
    "AUTOME","SS    "; % BIT 1

%

    FOR I := 1 STEP 1 UNTIL 35 DO
WRITE(P,TF,I,REAL(BITON(R,I)),2,
    TB[A:=28+2*(35-I)],TB[A+1]);
    FOR I:=36 STEP 1 UNTIL 39 DO
```

```
                                01924000 T  0082:0099:2
                                01925000 T  0082:0100:1
                                01926000 T  0082:0114:0
                                01927000 T  0082:0114:1
                          START OF SEGMENT *********   87
                                01928000 T  0082:0116:0
                                01929000 T  0082:0116:0
                                01930000 T  0082:0116:0
                                01931000 T  0082:0116:0
                                01932000 T  0082:0116:0
                                01933000 T  0082:0116:0
                                01934000 T  0082:0116:0
                                01935000 T  0082:0116:0
                                01936000 T  0082:0116:0
                                01937000 T  0082:0116:0
                                01938000 T  0082:0116:0
                                01939000 T  0082:0116:0
                                01940000 T  0082:0116:0
                                01941000 T  0082:0116:0
                                01942000 T  0082:0116:0
                                01943000 T  0082:0116:0
                                01944000 T  0082:0116:0
                                01945000 T  0082:0116:0
                                01946000 T  0082:0116:0
                                01947000 T  0082:0116:0
                                01948000 T  0082:0116:0
                                01949000 T  0082:0116:0
                                01950000 T  0082:0116:0
                                01950100 T  0082:0116:0
%108-                           01950200 P  0082:0116:0
                                01951000 T  0082:0116:0
                                01952000 T  0082:0116:0
                                01953000 T  0082:0116:0
                                01954000 T  0082:0116:0
                                01955000 T  0082:0116:0
                                01955100 T  0082:0116:0
                                01955200 T  0082:0116:0
                                01955300 T  0082:0116:0
%108-                           01955305 P  0082:0116:0
                                01955310 T  0082:0116:0
                                01955320 T  0082:0116:0
                                01955330 T  0082:0116:0
                                01955340 T  0082:0116:0
                                01955350 T  0082:0116:0
                                01955360 T  0082:0116:0
                                01955370 T  0082:0116:0
                                01955380 T  0082:0116:0
                                01955390 T  0082:0116:0
%108-                           01955400 P  0082:0116:0
%108-                           01955500 C  0082:0116:0
%108-                           01955600 C  0082:0116:0
                          87 IS   98 LONG, NEXT SEG   82
%108-                           01956000 P  0082:0116:0
%108-                           01960000 P  0082:0116:0
                                01961000 T  0082:0117:2
                                01962000 T  0082:0127:2
                                01963000 T  0082:0137:3
```

```
            WRITE(P,TF,I,REAL(BITON(R,I)),3,                                    01964000 T  0082:0139:2
                TB[A:=16+3×(39-I)],TB[A+1],TB[A+2]);                            01965000 T  0082:0149:2
            FOR I:=40 STEP 1 UNTIL 47 DO                                        01966000 T  0082:0161:3
            WRITE(P,TF,I,REAL(BITON(R,I)),2,                                    01967000 T  0082:0163:2
                TB[A:=2×(47-I)],TB[A+1]);                                       01968000 T  0082:0173:2
        END;                                                                    01969000 T  0082:0183:3
        NEXTITEM;                                                               01970000 T  0082:0183:3
        DISPLAY(MESSAGEHOLDER,FALSE);                                           01971000 T  0082:0184:0
        I:=0;                                                                   01972000 T  0082:0185:3
        IF OPERAND(MESSAGEHOLDER,R) AND (R:=R,CF)≠0 THEN                        01973000 T  0082:0186:0
        DO                                                                      01974000 T  0082:0189:3
        BEGIN                                                                   01975000 T  0082:0190:0
            WRITE(LINE[*],ITEM,OCTAL(R),                                        01976000 T  0082:0190:0
        OCTAL(HIHALF(R)),OCTAL(LOHALF(R)),"     ");                             01977000 T  0082:0197:2
            MOVD(M[(R+1),ROW,(R+1),COL],LINE[4],MIN(9,MAXCOR-R));               01978000 T  0082:0209:2
            WRITE(P,15,LINE[*]);                                                01979000 T  0082:0216:1
        END                                                                     01980000 T  0082:0220:1
        UNTIL                                                                   01981000 T  0082:0220:1
        (I:=I+1) GEQ MAXMESSAGES OR                                             01982000 T  0082:0220:1
            NOT OPERAND(R,R) OR                                                 01983000 T  0082:0222:1
    (R:=R,FF)=0                                                                 01984000 T  0082:0224:0
    ELSE WRITE(P,NULLMESSAGES);                                                 01985000 T  0082:0225:2
        IF DATACOM THEN                                                         01985100 T  0082:0230:0
        BEGIN                                                                   01985105 T  0082:0230:1
            PRINTQUEUE(STATIONMESSAGEHOLDER,0);                                 01985120 T  0082:0231:2
            PRINTQUEUE(LOOKQ,1);                                                01985150 T  0082:0232:1
            PRINTQUEUE(DC19Q,2);                                                01985200 T  0082:0233:3
            PRINTQUEUE(PINGQ,0);                                                01985250 T  0082:0235:2
            PRINTQUEUE(ILL,(0&3 CTF));                                          01985300 T  0082:0236:0
            IF RJE THEN                                                         01985350 T  0082:0238:0
            PRINTQUEUE(RJEWAITQ,0);                                             01985375 T  0082:0238:1
            NEXTPAGE;                                                           01985380 T  0082:0240:1
            PRINTARRAY(STATION);                                                01985385 T  0082:0244:1
            FOR I:=0 STEP 1 UNTIL 15 DO PAROW(STATION,I);                       01985390 T  0082:0246:0
            NEXTPAGE;                                                           01986000 T  0082:0250:1
    END ELSE NEXTITEM;                                                          01986100 T  0082:0254:1
    DISPLAY(NSLATE,FALSE);                                                      01987000 T  0082:0255:3
    DISPLAY(LSLATE,FALSE);                                                      01988000 T  0082:0257:2
    IF VARIFY(SLATE,A,S,TRUE) AND                                               01989000 T  0082:0258:1
        S.[47:1]=0 AND                                                          01990000 T  0082:0260:0
        OPERAND(NSLATE,N) AND                                                   01991000 T  0082:0261:3
        N.[1:8]=0 AND                                                           01992000 T  0082:0263:2
        N<S AND                                                                 01993000 T  0082:0264:1
        N.[47:1]=0 AND                                                          01994000 T  0082:0265:3
        OPERAND(LSLATE,L) AND                                                   01995000 T  0082:0267:2
        L.[1:8]=0 AND                                                           01996000 T  0082:0268:1
        L<S AND                                                                 01997000 T  0082:0270:0
        L.[47:1]=0 AND                                                          01998000 T  0082:0271:2
        L≠N THEN                                                                01999000 T  0082:0272:1
    DO                                                                          02000000 T  0082:0273:3
    BEGIN                                                                       02001000 T  0082:0275:2
    N:=REAL(BOOLEAN(N+2) AND BOOLEAN(S-1));                                     02002000 T  0082:0275:2
        WRITE(LINE[*],SE,OCTAL(N),                                              02003000 T  0082:0277:2
            OCTAL(HIHALF(A+N)),OCTAL(LOHALF(A+N)),                              02004000 T  0082:0285:2
            OCTAL(HIHALF(A+N+1)),OCTAL(R:=LOHALF (A+N+1)));                     02005000 T  0082:0292:0
        IF 129≤(R:=R,CF) AND R≤PRTMAX THEN                                      02006000 T  0082:0303:2
        MOVE(NAMS[NAME[R].CF],LINE[3],                                          02007000 T  0082:0306:0
```

```
                        NAME[R],FF);                                                02008000 T  0082:0309:2
                    WRITE(P,15,LINE[*]);                                            02009000 T  0082:0311:2
                END UNTIL N=L                                                       02010000 T  0082:0315:2
                ELSE WRITE(P,NULLSLATE);                                            02011000 T  0082:0315:3
            NEXTITEM; NEXTPAGE;                                                     02012000 T  0082:0320:0
            SGLTOG:=TRUE;                                                           02012100 T  0082:0324:1
    IF TINUOK:=VERIFY(TINU,TA,TS) AND                                               02013000 T  0082:0325:2
               VERIFY(MULTITABLE,MA,MS) AND                                         02014000 T  0082:0327:2
               VERIFY(LABELTABLE,LA,LS) AND                                         02015000 T  0082:0329:2
               VERIFY(RDCTABLE,RA,RS) AND                                           02016000 T  0082:0331:2
               VERIFY(PRNTABLE,PA,PS) THEN                                          02017000 T  0082:0333:2
            BEGIN                                                                   02018000 T  0082:0335:3
                S:=MAX(TS,MS,LS,RS,PS)-1;                                           02019000 T  0082:0336:0
                WRITE(P,10ATH);                                                     02020000 T  0082:0346:0
                FOR I:=0 STEP 1 UNTIL S DO                                          02021000 T  0082:0349:2
                BEGIN                                                               02022000 T  0082:0350:0
                WRITE(TB[*],LUN,OCTAL(I));                                          02023000 T  0082:0350:0
                IF I<TS THEN                                                        02024000 T  0082:0360:0
                BEGIN                                                               02025000 T  0082:0361:2
                    WRITE(LINE[*],FT,                                               02026000 T  0082:0361:3
                        (A:=HIHALF(TA+I)),[24:3],                                   02027000 T  0082:0365:2
                        OCTAL(A,[27:5]),                                            02028000 T  0082:0370:1
                        A,[32:3],                                                   02029000 T  0082:0373:3
                        OCTAL(A,[35:7]),                                            02030000 T  0082:0376:0
                        OCTAL(A,[42:6]),                                            02031000 T  0082:0379:2
                        (A:=LOHALF(TA+I)),[24:1],                                   02032000 T  0082:0382:0
                        OCTAL(A,[25:5]),                                            02033000 T  0082:0386:0
                        A,[30:18]);                                                 02034000 T  0082:0389:2
                    MOVC(LINE[0],0,TB[1],1,2,6);                                    02035000 T  0082:0394:0
                END;                                                                02036000 T  0082:0396:1
                IF I<MS THEN                                                        02037000 T  0082:0396:1
                    MOVC(M[(MA+I),ROW,(MA+I),COL],0,TB[4],5,1,0);                   02038000 T  0082:0397:3
                IF I<LS THEN                                                        02039000 T  0082:0403:2
                    MOVC(M[(LA+I),ROW,(LA+I),COL],0,TB[6],3,1,0);                   02040000 T  0082:0404:0
                    IF I<RS THEN                                                    02041000 T  0082:0409:3
                    BEGIN                                                           02042000 T  0082:0410:1
                        WRITE(LINE[*],RT,                                           02043000 T  0082:0411:2
                        OCTAL((A:=HIHALF(RA+I)),[24:8]),                            02044000 T  0082:0414:1
                        OCTAL(A,[32:6]),                                            02045000 T  0082:0420:0
                        OCTAL(A,[38:10]),                                           02046000 T  0082:0423:2
                        OCTAL((A:=LOHALF(RA+I)),[24:17]),                           02047000 T  0082:0426:0
                        OCTAL(A,[41:7]));                                           02048000 T  0082:0430:1
                        MOVC(LINE[0],0,TB[8],1,2,6);                                02049000 T  0082:0436:0
                    END;                                                            02050000 T  0082:0438:1
                    IF I<PS THEN                                                    02051000 T  0082:0438:1
                    BEGIN                                                           02052000 T  0082:0439:3
                        WRITE(LINE[*],PT,                                           02053000 T  0082:0440:0
                        (A:=HIHALF(PA+I)),[24:1],                                   02054000 T  0082:0443:3
                        A,[25:1],                                                   02055000 T  0082:0448:1
                        OCTAL(A,[26:13]),                                           02056000 T  0082:0451:2
                        OCTAL((A:=LOHALF(PA+I)&A[15:39:9]),[15:15]),                02057000 T  0082:0454:0
                        OCTAL(A,[30:18]));                                          02058000 T  0082:0459:3
                        MOVC(LINE[0],0,TB[11],5,2,6);                               02059000 T  0082:0465:2
                    END;                                                            02060000 T  0082:0467:3
                    WRITE(P,15,TB[*]);                                              02061000 T  0082:0467:3
                END;                                                                02062000 T  0082:0472:0
            END;                                                                    02063000 T  0082:0474:0
```

```
NEXTPAGE;                                                                  02064000 T  0082:0474:0
DISPLAY(IOQUESLOTS,FALSE);                                                 02064500 T  0082:0478:0
DISPLAY(IOQUEAVAIL,FALSE);                                                 02065000 T  0082:0479:3
IF  VERIFY(UNIT,UA,US) AND                                                 02066000 T  0082:0481:2
    VERIFY(IOQUE,IA,IS) AND                                                02067000 T  0082:0483:2
    VERIFY(LOCATQUE,LA,LS) AND                                             02068000 T  0082:0485:2
    VERIFY(FINALQUE,FA,FS) THEN                                            02069000 T  0082:0487:2
BEGIN                                                                      02070000 T  0082:0489:2
    S:=MAX(US,IS,LS,FS)-1;                                                 02071000 T  0082:0489:3
    WRITE(P,IOQSH);                                                        02072000 T  0082:0497:3
    FOR I:=0 STEP 1 UNTIL S DO                                             02073000 T  0082:0500:1
    BEGIN                                                                  02074000 T  0082:0502:0
 WRITE(TB[*],IFO,OCTAL(I),IF TINUOK THEN                                   02075000 T  0082:0502:0
      LOHALF(TA+I),[30:18] ELSE "***");                                    02076000 T  0082:0510:0
        IF I<US THEN                                                       02077000 T  0082:0518:0
        BEGIN                                                              02078000 T  0082:0519:2
            WRITE(LINE[*],UFO,                                             02079000 T  0082:0519:3
            (A:=HIHALF(UA+I)),[24:1],                                      02080000 T  0082:0523:2
            OCTAL(A,[25:4]),                                               02081000 T  0082:0528:1
            OCTAL(A,[29:8]),                                               02082000 T  0082:0531:3
            A,[37:1],                                                      02083000 T  0082:0534:1
            A,[38:1],                                                      02084000 T  0082:0537:2
            A,[39:1],                                                      02085000 T  0082:0539:3
            A,[40:2],                                                      02086000 T  0082:0542:0
            OCTAL((A:=LOHALF(UA+I)&A[18:42:6]),[18:15]),                   02087000 T  0082:0544:1
            OCTAL(A,[33:15]));                                             02088000 T  0082:0550:0
            MOVC(LINE[0],0,TB[1],4,3,4);                                   02089000 T  0082:0556:0
        END;                                                               02090000 T  0082:0558:1
        IF I<IS THEN                                                       02091000 T  0082:0558:1
        BEGIN                                                              02092000 T  0082:0559:3
            WRITE(LINE[*],WFO,                                             02093000 T  0082:0560:0
            OCTAL(HIHALF(IA+I)),                                           02094000 T  0082:0563:3
            OCTAL(LOHALF(IA+I)));                                          02095000 T  0082:0568:0
            MOVC(LINE[0],0,TB[5],7,2,1);                                   02096000 T  0082:0574:0
        END;                                                               02097000 T  0082:0576:1
        IF I<LS THEN                                                       02098000 T  0082:0576:1
        BEGIN                                                              02099000 T  0082:0577:3
            WRITE(LINE[*],LFO,                                             02100000 T  0082:0578:0
            (A:=HIHALF(LA+I)),[24:3],                                      02101000 T  0082:0581:3
            OCTAL(A,[27:5]),                                               02102000 T  0082:0586:1
            OCTAL(A,[32:4]),                                               02103000 T  0082:0589:3
            OCTAL(A,[36:6]),                                               02104000 T  0082:0592:1
            OCTAL((A:=LOHALF(LA+I)&A[18:42:6]),[18:15]),                   02105000 T  0082:0595:3
            OCTAL(A,[33:15]));                                             02106000 T  0082:0601:2
            MOVC(LINE[0],0,TB[8],7,2,6);                                   02107000 T  0082:0607:2
        END;                                                               02108000 T  0082:0609:3
        IF I<FS THEN                                                       02109000 T  0082:0609:3
        BEGIN                                                              02110000 T  0082:0610:1
            WRITE(LINE[*],WFO,                                             02111000 T  0082:0611:2
            OCTAL(HIHALF(FA+I)),                                           02112000 T  0082:0614:1
            OCTAL(LOHALF(FA+I)));                                          02113000 T  0082:0619:2
            MOVC(LINE[0],0,TB[12],4,2,1);                                  02114000 T  0082:0625:2
        END;                                                               02115000 T  0082:0627:3
        WRITE(P,15,TB[*]);                                                 02116000 T  0082:0627:3
    END;                                                                   02117000 T  0082:0632:0
NEXTITEM;                                                                  02117050 T  0082:0634:0
PRINTARRAY(CHANNEL);                                                       02117100 T  0082:0634:1
```

```
NEXTITEM;                                                                    02117125 T  0082:0636:0
DISPLAY(DISKOUNT,FALSE);                                                     02117150 T  0082:0636:1
IF DFX THEN BEGIN                                                            02117155 T  0082:0638:0
DISPLAY(EUW,FALSE);                                                          02117160 T  0082:0639:3
PRINTARRAY(EUQ);                                                             02117200 T  0082:0641:2
END;                                                                         02117205 T  0082:0642:1
IF SHAREDISK THEN                                                            02117400 T  0082:0642:1
BEGIN                                                                        02117410 T  0082:0643:2
    NEXTITEM;                                                                02117420 T  0082:0643:3
    DISPLAY(LQAVAIL,FALSE);                                                  02117430 T  0082:0644:0
    IF VARIFY(LQUE,UA,US,TRUE) AND                                           02117440 T  0082:0645:3
    OPERAND(LQAVAIL,FA) AND FA GEQ 0 AND FA LEQ US THEN                      02117450 T  0082:0647:3
    IF FA=0 THEN WRITE(P,NULLQUE) ELSE                                       02117460 T  0082:0651:2
    BEGIN                                                                    02117470 T  0082:0655:3
        WRITE(P[DBL],LQUEHDR);                                               02117480 T  0082:0656:0
        S:=MIN(2^0,FA-1);                                                    02117490 T  0082:0659:2
        FOR C:=0 STEP 1 UNTIL S DO                                           02117500 T  0082:0662:1
        IF NOT OPERAND(UA+C,L) OR (LA:=L.[1:7]) GEQ 64 THEN                  02117510 T  0082:0664:0
        WRITE(P,BADLQUENTRY,OCTAL(C),OCTAL(HIHALF(L)),                       02117520 T  0082:0667:3
              OCTAL(LOHALF(L))) ELSE                                         02117530 T  0082:0678:0
        WRITE(P,FLQUE,OCTAL(C),OCTAL(LA),L.[8:4],L.[12:36])                  02117540 T  0082:0684:0
    END ELSE WRITE(P,BADLQUE);                                               02117550 T  0082:0696:1
END LQUE;                                                                    02117560 T  0082:0707:2
END;                                                                         02118000 T  0082:0707:2
COMMON:=COMSAVE;                                                             02118100 T  0082:0707:2
SGLTOG:=FALSE;                                                               02118200 T  0082:0707:3
END DUMPING MCP INFO;                                                        02119000 T  0082:0708:1
                                                                           82 IS  719 LONG, NEXT SEG   10
PROCEDURE DUMPAUXMEM;                                                        02119100 T  0010:0286:1
BEGIN                                                                        02119110 T  0010:0286:1
    FORMAT AUX("DR",A1," MEMORY DUMP"),                                      02119120 T  0010:0286:1
                                                                       START OF SEGMENT **********   88
                                                                       START OF SEGMENT **********   89
          AUXMESS("# PRINTING CONTENTS OF DR",A1,",,,");                     02119130 T  0089:0000:0
                                                                           89 IS   17 LONG, NEXT SEG   88
    LABEL TRYANOTHER,EXIT;                                                   02119140 T  0088:0000:0
    BOOLEAN COMSAVE;                                                         02119142 T  0088:0000:0
    RELOAD:=TRUE;                                                            02119145 T  0088:0000:0
    TRYANOTHER:%                                                             02119150 T  0088:0000:1
    IF NOMO THEN GO EXIT;                                                    02119160 T  0088:0001:2
    NEXTPAGE;                                                                02119170 T  0088:0002:0
    LOAD;                                                                    02119180 T  0088:0006:0
    IF AUXTYPE NEQ 0 THEN % AUXMEM PRESENT                                   02119190 T  0088:0006:1
    BEGIN                                                                    02119200 T  0088:0007:2
        NEXTITEM ;                                                           02119210 T  0088:0007:3
        WRITE(P,AUX,16+AUXTYPE DIV 4);                                       02119220 T  0088:0008:0
        WRITE(SPO,AUXMESS,16+AUXTYPE DIV 4);                                 02119230 T  0088:0018:0
        NEXTITEM;                                                            02119240 T  0088:0028:0
        COMSAVE:=COMMON;                                                     02119244 T  0088:0028:1
        COMMON:=BOOLEAN(5,12); % ALPHA/OCTAL                                 02119245 T  0088:0029:3
        PRINT(0,32768);                                                      02119250 T  0088:0030:0
        COMMON:=COMSAVE;                                                     02119255 T  0088:0033:3
        GO TRYANOTHER;                                                       02119260 T  0088:0034:0
    END;                                                                     02119270 T  0088:0036:0
    EXIT:RELOAD:=FALSE;                                                      02119280 T  0088:0036:0
END DUMPAUXMEM;                                                              02119290 T  0088:0036:1
                                                                           88 IS   39 LONG, NEXT SEG   10
```

```
PROCEDURE FIXFID;                                                                    02119300 T  0010:0286:1
BEGIN                                                                                02119310 T  0010:0286:1
      REAL A;                                                                        02119320 T  0010:0286:1
                                                                                   START OF SEGMENT **********   90
      FORMAT FNEWFILE("#NEXT FILE TO BE ANALYZED IS MEMORY/",A1,A6);               02119330 T  0090:0000:0
                                                                                   START OF SEGMENT **********   91
                                                                                     91 IS   11 LONG, NEXT SEG   90
      STREAM PROCEDURE MINUS1(N,A); VALUE N;                                       02119340 T  0090:0000:0
      BEGIN SI:=LOC N; SI:=SI+5; DI:=A; DI:=DI+5; DS:=3 SUB END;                   02119350 T  0090:0000:0
      MINUS1(1,TDFID);                                                             02119360 T  0090:0001:3
      FILL MDUMP WITH *,TDFID;                                                     02119370 T  0090:0003:2
      WRITE(SPO,FNEWFILE,TDFID,[6:6],TDFID);                                       02119380 T  0090:0007:3
END;                                                                               02119390 T  0090:0018:0
                                                                                     90 IS   21 LONG, NEXT SEG   10
FORMAT COLHDR("B5500 COLLATING SEQUENCE"//X8,"01234567"/),                         02119500 T  0010:0286:1
                                                                                   START OF SEGMENT **********   92
       FPUNT("HANG CAUSED BY: ",X24),                                              02119505 T  0092:0286:1
       COLINE(X5,I1,X2,X8);                                                        02119510 T  0092:0286:1
                                                                                     92 IS   27 LONG, NEXT SEG   10
ARRAY COLLATE[0:7];                                                                02119600 T  0010:0286:1
INTEGER DUMPD;                                                                     02120000 T  0010:0289:3
PROCEDURE INITIALIZE;                                                              02120100 T  0010:0289:3
COMMENT:ZERO VARIABLES & ARRAYS IN THE EVENT MORE THAN ONE FILE IS                 02120110 T  0010:0289:3
PROCESSED IN A GIVEN RUN;                                                          02120115 T  0010:0289:3
BEGIN                                                                              02120120 T  0010:0289:3
INTEGER K,J;                                                                       02120125 T  0010:0289:3
                                                                                   START OF SEGMENT **********   93
LABEL EXIT;                                                                          02120130 T  0093:0000:0
IF NOT RELOAD THEN GO EXIT;                                                          02120135 T  0093:0000:0
COMMENT:ZERO REALS & INTEGERS;                                                       02120140 T  0093:0000:1
I:=                                                                                  02120145 T  0093:0000:1
R:=                                                                                  02120150 T  0093:0000:1
VJOBNUM:=                                                                            02120155 T  0093:0000:1
VBED:=                                                                               02120160 T  0093:0000:1
TABLESLOC:=                                                                          02120165 T  0093:0000:1
0;                                                                                   02120170 T  0093:0000:1
MINLNK:=                                                                             02120175 T  0093:0003:3
MINBAD:=                                                                             02120180 T  0093:0003:3
MAXBAD:=                                                                             02120185 T  0093:0003:3
MAXLNK:=                                                                             02120190 T  0093:0003:3
PRTCODE:=                                                                            02120195 T  0093:0003:3
0;                                                                                   02120200 T  0093:0003:3
MAXMCPROG:=                                                                          02120205 T  0093:0006:0
ESP:=                                                                                02120210 T  0093:0006:0
BADSAVEPRTLOC:=                                                                      02120215 T  0093:0006:0
INTSPMAX:=                                                                           02120220 T  0093:0006:0
PROWS:=                                                                              02120225 T  0093:0006:0
0;                                                                                   02120230 T  0093:0006:0
MAXSTK:=                                                                             02120235 T  0093:0009:2
BEDSTK:=                                                                             02120240 T  0093:0009:2
DUMPD:=                                                                              02120245 T  0093:0009:2
0;                                                                                   02120300 T  0093:0009:2
COMMENT:MAKE BOOLEANS FALSE;                                                         02120400 T  0093:0010:1
LNKSOK:=                                                                             02120405 T  0093:0010:1
AVALNKOK:=                                                                           02120410 T  0093:0010:1
SOMOKF:=                                                                             02120415 T  0093:0010:1
SOMOKB:=                                                                             02120420 T, 0093:0010:1
```

```
    NEEDCHECKAVAILNKS:=
    FALSE;
    BADCOMMENT:=
    FALSE;
    COMMENT:ZERO ARRAYS;
    FOR K:=0 STEP 1 UNTIL MIXMAX DO
        MIXSTK[K]:=
        0;
    FOR K:=0 STEP 1 UNTIL INAMESIZE-1 DO
        INTSP[K]:=
        0;
    EXIT;
END INITIALIZE;

RESTART:%%
    INITIALIZE;
    FILLAREATYPE;
    LOAD;
    DUMPMCPSPRT;
    IF NOT COMMON THEN
    IF DUMPCESSPOOLONLY THEN
    BEGIN
        DUMPCESSPOOL;
        GO EOPROG;
    END;
    IF NOT COMMON THEN
    BEGIN
        CHECKMEMORYLINKS;
        GETSTACKSFROMTHEBED;
        GETANDSORTMCPROG;
END;
    IF COMNT THEN BEGIN
        WRITE(P[DBL],STARS);
    IF NOT COMMON THEN
    IF OPERAND(107,PROWS) AND PROWS NEQ "EEEE"&"EEEE"[1:25:23] THEN
    IF PDATADESC(PUNTER,R) AND R.CF LSS 4096 THEN
    BEGIN
        I:=R.[8:10];
        R:=R.CF;
        IF PROWS GEQ R AND PROWS LEQ R+I THEN
        BEGIN
            WRITE(LINE[*],FPUNT);
            MOVD(M[PROWS,ROW,PROWS,COL],LINE[2],3);
            WRITE(P[DBL],15,LINE[*]);
        END;
    END;
        IF BADCOMMENT THEN WRITE(P[DBL],COMNTPAR) ELSE
        BEGIN
        WRITE(P,10,COMMT[*]);
        MOVE(COMMT[10],COMMT[0],10);
        WRITE(P,10,COMMT[*]);
        END;
        WRITE(P[PAGE],STARS);
    END;
     IF NOT COMMON THEN
     BEGIN
     IF NODUMP THEN ELSE
```

```
                02120425 T  0093:0010:1
                02120430 T  0093:0010:1
                02120435 T  0093:0013:3
                02120500 T  0093:0013:3
                02120600 T  0093:0014:0
                02120605 T  0093:0014:0
                02120610 T  0093:0015:2
                02120615 T  0093:0015:3
                02120650 T  0093:0018:1
                02120655 T  0093:0022:1
                02120660 T  0093:0023:2
                02120700 T  0093:0024:1
                02120705 T  0093:0025:2
          93 IS   28 LONG, NEXT SEG   10
                02120900 T  0010:0289:3
                02120910 T  0010:0290:0
%106=           02120920 C  0010:0290:1
                02121000 T  0010:0291:2
                02122000 T  0010:0291:3
                02122050 T  0010:0292:0
                02122100 T  0010:0292:1
                02122200 T  0010:0293:3
                02122300 T  0010:0294:0
                02122400 T  0010:0294:1
                02122500 T  0010:0295:2
                02123000 T  0010:0295:2
                02124000 T  0010:0295:3
                02125000 T  0010:0296:0
                02126000 T  0010:0296:1
                02126500 T  0010:0297:2
                02127000 T  0010:0298:0
                02128000 T  0010:0298:0
                02129000 T  0010:0298:1
                02129050 T  0010:0301:3
                02129100 T  0010:0302:0
                02129110 T  0010:0305:3
                02129120 T  0010:0309:2
                02129130 T  0010:0309:3
                02129140 T  0010:0310:1
                02129150 T  0010:0312:0
                02129160 T  0010:0314:0
                02129170 T  0010:0314:1
                02129180 T  0010:0318:1
                02129190 T  0010:0322:0
                02129200 T  0010:0326:1
                02129210 T  0010:0326:1
                02130000 T  0010:0326:1
                02131000 T  0010:0330:0
                02132000 T  0010:0333:2
                02133000 T  0010:0337:2
                02134000 T  0010:0339:2
                02135000 T  0010:0343:2
                02136000 T  0010:0343:2
                02137000 T  0010:0346:0
                02137100 T  0010:0346:0
                02137200 T  0010:0346:1
                02138000 T  0010:0347:2
```

```
      IF OPERAND(16,R) AND R GEQ 12 AND R LEQ 15                                    02139000 T  0010:0349:2
           THEN BEGIN                                                               02140000 T  0010:0351:3
      NEEDCHECKAVAILNKS:=NOT AVALNKOK;                                              02141000 T  0010:0353:2
PRINT(0,R);                                                                         02142000 T  0010:0354:0
IF OPERAND(107,PROWS) AND PROWS="EEEE"&"EEEE"[1:25:23]                              02142100 T  0010:0357:2
THEN IF OPERAND(R+96,PROWS) AND PROWS NEQ "EEEE"&"EEEE"[1:25:23]                    02142120 T  0010:0359:3
THEN WRITE(P[DBL],ITEM,OCTAL(R),OCTAL(HIHALF(PROWS:=R+96)),                         02142200 T  0010:0363:3
           OCTAL(LOHALF(PROWS)));                                                   02142250 T  0010:0376:0
DONTPRINTLINKS:=TRUE;                                                               02142260 T  0010:0382:0
PRINT(R+1,16);                                                                      02142300 T  0010:0383:2
      END ELSE PRINT(0,36);                                                         02143000 T  0010:0386:1
  DONTPRINTLINKS:=TRUE;                                                             02143100 T  0010:0390:1
  IF NODUMP THEN ELSE                                                               02144000 T  0010:0391:2
  IF MINLNK GTR DUMPD:=36 THEN                                                      02145000 T  0010:0393:2
  BEGIN                                                                             02146000 T  0010:0394:1
      DUMPD:=MINLNK;                                                                02147000 T  0010:0395:2
      WRITE(P[DBL],<X48,"MCP INTERRUPT LOCATIONS">);                %107-           02147500 C  0010:0396:0
                                                                          94 IS    8 LONG, NEXT SEG   10
      PRINT(36,64); % INTERRUPT CODE                                                02148000 T  0010:0399:2
      WRITE(P[DBL],<X50,"MCP INTERRUPT STACK">);                    %107-           02148500 C  0010:0402:0
                                                                          95 IS    8 LONG, NEXT SEG   10
      PRINT(64,112); % INTERRUPT STACK                                              02149000 T  0010:0405:2
      WRITE(P[DBL],<X56,"MCP PRT">);                                %107-           02149050 C  0010:0408:1
                                                                          96 IS    6 LONG, NEXT SEG   10
      PRINT(112,PRTMAX+1); % PRT                                                    02149100 T  0010:0411:3
      IF NOT NOMCPCODE THEN                                                         02149200 T  0010:0415:2
      IF (R:=OUTERBLOCK[0])=0 THEN PRINT(36,MINLNK) ELSE                            02149300 T  0010:0416:0
      BEGIN                                                                         02149350 T  0010:0422:0
          WRITE(P[DBL],<X50,"MCP OUTER BLOCK CODE">);               %107-           02149360 C  0010:0422:1
                                                                          97 IS    8 LONG, NEXT SEG   10
          PRINT(R,CF,R,FF+ 1); % OUTERBLOCK                                         02149400 T  0010:0425:3
      END;                                                                          02150000 T  0010:0430:0
      DUMPMCPSAVECODE(ESP,MINLNK); % ANALYZE SAVE CODE              %106-           02150010 C  0010:0430:0
END;                                                                                02150050 T  0010:0431:2
DONTPRINTLINKS:=FALSE;                                                              02150075 T  0010:0431:2
IF LNKSOK THEN                                                                      02151000 T  0010:0432:0
    DUMPMEMORYANDNOTESTACKS(MINLNK,DUMPD:=MAXLNK)                                   02152000 T  0010:0432:0
ELSE                                                                                02153000 T  0010:0433:2
BEGIN                                                                               02154000 T  0010:0434:0
    IF SOMOKF THEN                                                                  02155000 T  0010:0434:1
        DUMPMEMORYANDNOTESTACKS(DUMPD,DUMPD:=MINBAD);                               02156000 T  0010:0435:2
    IF SOMOKB THEN                                                                  02157000 T  0010:0437:2
    BEGIN                                                                           02158000 T  0010:0437:2
        PRINT(DUMPD,MAXBAD);                                                        02160000 T  0010:0437:3
        DUMPMEMORYANDNOTESTACKS(MAXBAD,DUMPD+MAXLNK);                               02161000 T  0010:0441:2
    END;                                                                            02162000 T  0010:0442:1
END;                                                                                02163000 T  0010:0442:1
      END; % IF NOT COMMON                                                          02163100 T  0010:0442:1
PRINT(DUMPD,MAXCOR+1);                                                              02165000 T  0010:0442:1
IF NODUMP THEN                                                                      02165100 T  0010:0446:0
BEGIN                                                                               02165110 T  0010:0447:3
     NODUMPTOG:=TRUE;                                                               02165120 T  0010:0448:0
     COMMON:=COMMON AND NOT BOOLEAN(384);                                           02165130 T  0010:0448:1
END;                                                                                02165140 T  0010:0450:0
IF NOT COMMON THEN                                                                  02166000 T  0010:0450:0
BEGIN                                                                               02167000 T  0010:0450:1
    LISTMCPROG;                                                                     02168000 T  0010:0451:2
    DUMPMCPINFO;                                                                    02169000 T  0010:0452:0
```

```
        GETSORTANDLISTINTRINSICS;
        DUMPROGRAMS;
        DUMPCONTROLSTACKS;
        PRINTSELECTEDARRAYS;
        DUMPAUXMEM;
    END;
EOPROG:%
    FILL COLLATE[*] WITH
    OCT0001020304050607,

    OCT1011121314151617,
    OCT2021222324252627,
    OCT3031323334353637,
    OCT4041424344454647,
    OCT5051525354555657,
    OCT6061626364656667,
    OCT7071727374757677;

    WRITE(P[PAGE]);
    WRITE(P,COLHDR);
    FOR I:=0 STEP 1 UNTIL 7 DO
    BEGIN
        WRITE(LINE[*],COLINE,I);
        MOVE(COLLATE[I],LINE[1],1);
        WRITE(P,2,LINE[*]);
    END;
    IF MULTI AND TDFID.[30:18] GTR 1 THEN
    BEGIN
        IF NOT REPEATING THEN
        BEGIN
            REPEATING:=TRUE;
            SPACE(MDUMP,-1)[XX:XX];
            XX:CLOSE(MDUMP,*);
        END;
        FIXFID;
        RELOAD:=TRUE;
        PRINTCOMMONVALUES;
        WRITE(P[DBL]); WRITE(P[DBL]);
        WRITE(P,FINI);
        NEXTPAGE;
        IF NODUMPTOG THEN BEGIN
        NODUMPTOG:=FALSE;
        COMMON:=COMMON OR BOOLEAN(384) END;
        MYSTACKADR:=-1;
        GO RESTART;
    END;
    END;% % % % % % % % % % % END INNER BLOCK % % % % % % % % % % %

PRINTCOMMONVALUES;
WRITE(P[DBL]);WRITE(P[DBL]);
WRITE(P,FINI);
END.

NUMBER OF ERRORS DETECTED =   1.  COMPILATION TIME =  202 SECONDS.

PRT SIZE = 354; TOTAL SEGMENT SIZE =  7971 WORDS; DISK SIZE = 360 SEGS; NO. PGM. SEGS = 107
```

```
        02170000 T  0010:0452:1
        02171000 T  0010:0453:2
        02172000 T  0010:0453:3
        02172100 T  0010:0454:0
        02172200 T  0010:0454:1
        02173000 T  0010:0455:2
        02173400 T  0010:0455:2
        02173500 T  0010:0455:2
        02173510 T  0010:0455:3
START OF SEGMENT *********   98
        02173520 T  0010:0456:1
        02173530 T  0010:0456:1
        02173540 T  0010:0456:1
        02173550 T  0010:0456:1
        02173560 T  0010:0456:1
        02173570 T  0010:0456:1
        02173580 T  0010:0456:1
  98 IS    8 LONG, NEXT SEG   10
        02173590 T  0010:0456:1
        02173600 T  0010:0460:1
        02173610 T  0010:0463:3
        02173620 T  0010:0465:2
        02173630 T  0010:0465:2
        02173640 T  0010:0473:2
        02173650 T  0010:0475:2
        02173660 T  0010:0479:2
        02173700 T  0010:0481:3
        02173710 T  0010:0483:3
        02173720 T  0010:0484:0
        02173730 T  0010:0484:1
        02173740 T  0010:0485:2
        02173750 T  0010:0485:3
        02173760 T  0010:0491:2
        02173770 T  0010:0493:3
        02173780 T  0010:0493:3
        02173790 T  0010:0494:0
        02173791 T  0010:0495:2
        02173792 T  0010:0495:3
        02173793 T  0010:0503:3
        02173795 T  0010:0506:1
        02173796 T  0010:0510:1
        02173797 T  0010:0511:2
        02173798 T  0010:0512:0
        02173799 T  0010:0513:2
        02173800 T  0010:0514:0
        02173810 T  0010:0514:1
        02174000 T  0010:0514:1
  10 IS   518 LONG, NEXT SEG    2
        02174500 T  0002:0038:0
        02175000 T  0002:0038:1
        02176000 T  0002:0046:1
        02178000 T  0002:0049:3
   2 IS    53 LONG, NEXT SEG    1
   1 IS     2 LONG, NEXT SEG    0
 107 IS    69 LONG, NEXT SEG    0
00597100
```

ESTIMATED CORE STORAGE REQUIRED = 11551 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED = 0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED = 3679.

DUMP /ANALYZE

SOURCE FILE: SYMBOL /DUMPANL

```
A -- STREAM VARIABLE -- NAMF PARAMETER -- DECLARED IN SEGMENT 19 AT 00319000
      00320000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 20 AT 00352000
      00353000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 29 AT 00586000
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 41 AT 00763170
      00763175
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 41 AT 00763190
      00763195 *00763205*
A -- REAL -- VALUE PARAMETER -- DECLARED IN SEGMENT 41 AT 00763290
      00763330 *00763335*
A -- STREAM VARIABLE -- NAMF PARAMETER -- DECLARED IN SEGMENT 44 AT 00763300
      00763305
A -- STREAM VARIABLE -- NAMF PARAMETER -- DECLARED IN SEGMENT 41 AT 00763355
      00763360
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 45 AT 00763385
      00763390
A -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 00765000
      00766000
A -- STREAM VARIABLE -- DECLARED IN SEGMENT 47 AT 00807170
      *00807200* 00807230
A -- REAL ARRAY -- DECLARED IN SEGMENT 50 AT 00817000
      00824110 00826990 00827000 00828000 00829000 00832000 00832100 00833000 00834100 00835100
A -- INTEGER -- DECLARED IN SEGMENT 74 AT 01524000
      *01559000**01561000*
A -- REAL -- DECLARED IN SEGMENT 77 AT 01703000
      01711000 *01712000* 01713000 01719000 01721000 01731000
A -- REAL -- DECLARED IN SEGMENT 79 AT 01757335
      01757505 01757530 01757590 01757730 01757820 01757960
A -- REAL -- DECLARED IN SEGMENT 82 AT 01758200
      *01962000**01965000**01968000* 01989000 02004000 02005000 *02027000* 02028000 02029000 02030000 02031000
      *02032000* 02033000 02034000 *02044000* 02045000 02046000 *02047000* 02048000 *02054000* 02055000 02056000
      *02057000* 02058000 *02080000* 02081000 02082000 02083000 02084000 02085000 02086000 *02087000* 02088000
      *02101000* 02102000 02103000 02104000 *02105000* 02106000
A -- REAL -- NAME PARAMETER -- DECLARED IN SEGMENT 82 AT 01814000
      01811000 01818000 01820000 *01821000* 01822000
A -- REAL -- DECLARED IN SEGMENT 90 AT 02119320
A -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 90 AT 02119340
      02119350
ACCIDENTRY -- STREAM LABEL -- DECLARED IN SEGMENT 70 AT 01371135 -- OCCURS AT 01371280
      01371230
ACTUALPRTBASE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00026000
      00089000 00270000 00292000 00293000 00913000 01392000
ADATE -- ALPHA -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 00463000
      00461000 00462000 00479000
ADDRESS -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 01349520
```

```
        01349500  01349510  01349560  01349600  01349610  01349630  01349690
ADDRESS  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370530
        01370550  01370680
ADR  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00655000
        00653000  00654000  00658500
ADR  --  REAL  --  DECLARED IN SEGMENT 36 AT 00734000
       *00748000* 00753000
ADR  --  REAL  --  DECLARED IN SEGMENT 60 AT 01104000
        01135000 *01136000* 01168000  01168900
ADR  --  REAL  --  DECLARED IN SEGMENT 66 AT 01287000
        01292000  01294000 *01295000* 01295100  01298000
ADR  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349120
        01349130  01349260
ADR  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349520
        01349500  01349560  01349570  01349600 *01349610* 01349630  01349640  01349690  01349700
ADR  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
        01444000  01447000  01465600  01469800
ADR  --  REAL  --  DECLARED IN SEGMENT 74 AT 01531000
       *01618000* 01621000  01624000  01625000  01640000  01641000  01655000  01656000
ADRR  --  REAL  --  DECLARED IN SEGMENT 36 AT 00735000
AGAIN  --  LABEL  --  DECLARED IN SEGMENT 47 AT 00807150  --  OCCURS AT 00807320
        00807410
AIT  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
        01588000 *01690000* 01695000  01696000 *01698300*
AL  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00583000
AL  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00589000
       *00590000**00594100**00594300* 00594400 *00595000* 00642000  00644000
ALINE  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00559000
        00640000  00643000  00644400  00645000  00646000  01465600  01466300  01466700  01468900  01469100  01469200
        01469800 *01469900* 01470000
ALINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 29 AT 00586100
        00586120
AMAX  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01335000
        01332000  01333000  01339000  01340000
ANAME  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 2 AT 00044000
        00041000  00054000  00058000
AO  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00579000
AO  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00589000
       *00591000**00594200* 00594300 *00594400**00595000* 00642000  00644000
ARAY  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01226000
        01224000  01242000  01243000  01247000 *01248000*
ARAY  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01334000
        01332000  01342000  01344000  01345000
ARFATYPE  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00295110
       *00295280* 00593000
ARGH  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164290
        00763445  00763470
ARRAYSFILLED  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00295115
        00295275 *00295990*
ARROW  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00113150
        01118100  01349480  01370230  01371420
AT  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00536000
        00540000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00651200
        00651100
```

```
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00671000
        00669000  00670000  00675000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00685000
        00683000  00684000  00689000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00697000
        00695000  00696000  00701000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00707000
        00705000  00706000  00711000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01352000
        01351000
AT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01514000
        01512000  01513000  01516000  01517000
ATP  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00295110
       *00295700* 01169400
ATTACHED  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164230
AUX  --  FORMAT  --  DECLARED IN SEGMENT 89 AT 02119120
        02119220
AUXCODE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164120
AUXDATA  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164110
AUXMEMM  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00110000
AUXMESS  --  FORMAT  --  DECLARED IN SEGMENT 89 AT 02119130
        02119230
AUXTYPE  --  REAL  --  DECLARED IN SEGMENT 10 AT 00082050
       *00834100* 02119190  02119220  02119230
AVAIL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00125000
        00967000  01026000
AVAIL  --  STREAM LABEL  --  DECLARED IN SEGMENT 60 AT 01115800  --  OCCURS AT 01118600
        01116700
AVAILM  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00949000
       *00997000* 01035000  01055000
AVAILN  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00949000
       *00995000* 01033000  01053000
AVAILT  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00949000
       *00996000* 01034000  01054000
AVALNK  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00559100
        00594000  00644400 *01151800**01152100*
AVALNKOK  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00498000
       *00978000* 00989000 *00991000**00992000**01026000* 01032000 *01037000* 01051000 *01052000**02120410* 02141000
AVL  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00960000
        01057000
AVLBAD  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00959000
        01056000
AVTABLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164150
B  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00113160
        01151000  01151500  01167000  01469500  01469700
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 19 AT 00319000
        00320000
B  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00395000
        00396000  00398000
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 29 AT 00586000
B  --  REAL  --  DECLARED IN SEGMENT 41 AT 00763140
       *00763505* 00763510 *00763525* 00763530 *00763535* 00763540 *00763595* 00763600  00763605 *00763610* 00763615
B  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 41 AT 00763190
        00763200
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 44 AT 00763300
        00763305
B  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 41 AT 00763375
```

```
        00763410  00763420
B  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 45 AT 00763385
        00763390
B  --  BOOLEAN  --  DECLARED IN SEGMENT 46 AT 00771000
        *00786000* 00787000
B  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 47 AT 00807170
        *00807210* 00807230
B  --  BOOLEAN  --  DECLARED IN SEGMENT 58 AT 01068000
        *01082000* 01095000
B  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01252000
        01259500
B  --  BOOLEAN  --  DECLARED IN SEGMENT 79 AT 01757330
        *01757710* 01757740  01757750  01757770
B  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 82 AT 01765000
        01766000  01768000
B  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 82 AT 01814500
        01811000  01812000  01816000
BAD  --  STREAM LABEL  --  DECLARED IN SEGMENT 46 AT 00771105  --  OCCURS AT 00771140
        00771130
BAD  --  LABEL  --  DECLARED IN SEGMENT 52 AT 00875000  --  OCCURS AT 00894000
        00885000  00887000  00889000  00892000
BAD  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00957000
        01051000  01060000
BADBED  --  FORMAT  --  DECLARED IN SEGMENT 23 AT 00446000
        00773000
BADCLOCK  --  FORMAT  --  DECLARED IN SEGMENT 24 AT 00449000
        00797000
BADCOMMENT  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00808000
       *00824000**00824220**00836000**02120435* 02130000
BADDATE  --  FORMAT  --  DECLARED IN SEGMENT 24 AT 00447000
        00775000
BADEOF  --  FORMAT  --  DECLARED IN SEGMENT 51 AT 00814000
        00866000
BADINTO  --  FORMAT  --  DECLARED IN SEGMENT 67 AT 01288100
        01310100
BADLOC  --  FORMAT  --  DECLARED IN SEGMENT 64 AT 01257100
        01280300
BADLQUE  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774120
        02117550
BADLQUENTRY  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774110
        02117520
BADPRIMARY  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00958000
        00999100  01007000  01022000
BADPRT  --  FORMAT  --  DECLARED IN SEGMENT 38 AT 00741000
        00747000
BADPRT  --  FORMAT  --  DECLARED IN SEGMENT 53 AT 00879000
        00894000
BADSAVEPRTLOC  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 01251500
        01280300 *02120215*
BADSDESC  --  FORMAT  --  DECLARED IN SEGMENT 76 AT 01536000
        01689000
BADSIZE  --  FORMAT  --  DECLARED IN SEGMENT 76 AT 01535000
        01689000
BADTM  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00813000  --  OCCURS AT 00866000
        00824210  00826990  00827000
BADXCLOCK  --  FORMAT  --  DECLARED IN SEGMENT 24 AT 00448000
        00781000
```

```
BASE -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 2 AT 00043000
        00041000  00042000  00054000  00057000  00058000
BED -- DEFINE -- DECLARED IN SEGMENT 10 AT 00129000
        00720000  01076000
BEDDED -- BOOLEAN -- DECLARED IN SEGMENT 60 AT 01101000
       *01194000* 01195000 *01196000* 01199000
BEDSTK -- INTEGER -- DECLARED IN SEGMENT 10 AT 00095000
       *01087000* 01088000  01089000  01195000  01383000  01714000  01715000  01737000  01738000 *02120240*
BITON -- BOOLEAN STREAM PROCEDURE -- DECLARED IN SEGMENT 10 AT 00395000
       *00400000* 00401000  00411000
BITON -- BOOLEAN STREAM PROCEDURE -- DECLARED IN SEGMENT 82 AT 01765000
       *01770000* 01862000  01914000  01924000  01961000  01964000  01967000
BLINE -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 00559050
        00644400  00644500
BLINE -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 29 AT 00586100
        00586110
BOS -- INTEGER -- DECLARED IN SEGMENT 70 AT 01357000
       *01377000* 01378000 *01379000* 01391000  01401000  01404000  01405000  01412000  01418500 *01454000**01455000*
        01456000 *01457000* 01457140  01458000  01465400  01466000
BREAKOUT -- DEFINE -- DECLARED IN SEGMENT 10 AT 00097000
BUF -- REAL ARRAY -- DECLARED IN SEGMENT 8 AT 00047000
        00053000  00054000  00058000  00059000
BUILD -- STREAM PROCEDURE -- DECLARED IN SEGMENT 29 AT 00576000
        00585000  00640000  00643000
BUILDHEADER -- STREAM PROCEDURE -- DECLARED IN SEGMENT 60 AT 01115300
        01121900  01151400  01168600  01169100  01169400
BUMPI -- DEFINE -- DECLARED IN SEGMENT 41 AT 00763150
        00763415  00763515  00763560
BUSTCOMMON -- PROCEDURE -- DECLARED IN SEGMENT 2 AT 00040100
        00040300  00061000
B6500LOAD -- DEFINE -- DECLARED IN SEGMENT 10 AT 00110200
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 6 AT 00040150
        00040160
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 6 AT 00040170
        00040180
C -- INTEGER -- DECLARED IN SEGMENT 32 AT 00674000
       *00675000* 00679000
C -- INTEGER -- DECLARED IN SEGMENT 33 AT 00688000
       *00689000* 00692000
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 41 AT 00763190
C -- REAL -- DECLARED IN SEGMENT 45 AT 00763380
       *00763410* 00763415
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 46 AT 00771100
        00771115
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 01325000
        01326000  01330000
C -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 01352000
        01351000
C -- FORMAT -- DECLARED IN SEGMENT 71 AT 01370010
        01446000
C -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 70 AT 01370020
        01370030  01370170
C -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 01514000
        01512000 *01517000*
C -- INTEGER -- DECLARED IN SEGMENT 74 AT 01522000
        01592000  01593000  01597000  01601000  01606000  01607000
C -- INTEGER -- DECLARED IN SEGMENT 79 AT 01757332
```

```
      *01757410* 01757600
C  --  INTEGER  --  DECLARED IN SEGMENT 82 AT 01758300
      *02117500* 02117510  02117520  02117540
CC  --  REAL ARRAY  --  DECLARED IN SEGMENT 41 AT 00763160
       00763515
CC  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 41 AT 00763290
      *00763330* 00763345
CCMASK1  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164360
       01757450
CCMASK2  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164365
       01757455
CD  --  DEFINE  --  DECLARED IN SEGMENT 70 AT 01363000
       01406000  01413000
CF  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00116000
       00329000  00662000  00665000  00744000  00763036  00763450  00807350  00807370  00863120  00918000  00919000
       00928000  00929000  00967140  00967150  00972000  00972400  00976100  00976300  00984000  00985000  00988000
       01014000  01036000  01044000  01056000  01079000  01127000  01134000  01136000  01149000  01164100  01167200
       01167700  01168600  01168700  01169000  01169200  01169500  01169550  01197000  01242000  01243000  01262000
       01270000  01272000  01273000  01281620  01281640  01281650  01281660  01281760  01295000  01298000  01299100
       01300000  01301000  01304000  01316000  01319000  01342000  01345000  01349570  01349640  01375000  01396000
       01439000  01444000  01448000  01465400  01465600  01466000  01466300  01467700  01468300  01468450  01469800
       01517000  01541000  01557000  01609100  01613000  01623000  01627000  01631000  01643000  01677000  01683000
       01684000  01712000  01718000  01757505  01757845  01757870  01757895  01757960  01821000  01973000  02006000
       02007000  02129110  02129140  02149400
CHANNEL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00140000
       02117100
CHECKFORLABELDESCRIPTOR  --  DEFINE  --  DECLARED IN SEGMENT 52 AT 00876000
       00885000  00887000  00889000  00892000
CHECKFORPROGRAMDESCRIPTOR  --  DEFINE  --  DECLARED IN SEGMENT 52 AT 00878000
       00892000
CHECKLINK  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00098000
CHECKMEMORYLINKS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00944000
       02125000
CHECKPRTFILE  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00870000
       00895000  00907000
CHRS  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00536000
       00541000  00543000  00547000  00551000  00689000  00692000  01516000  01517000
CIDROW  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164285
CL  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01529000
CLOCK  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00415000
       00796000  00799000
CLOCX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00432000
       00799000  00800000  00801000  00803000
CODE  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01757310
       01757300  01757410 *01757415* 01757420  01757550
COL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00119000
       00547000  00551000  00606000  00608000  00615000  00617000  00640000  00641000  00643000  00675000  00689000
       00763595  00774000  00777000  00780000  00783000  00796000  00799000  01149000  01516000  01517000  01559000
       01561000  01757600  01757875  01978000  02038000  02040000  02129180
COLHDR  --  FORMAT  --  DECLARED IN SEGMENT 92 AT 02119500
       02173600
COLINE  --  FORMAT  --  DECLARED IN SEGMENT 92 AT 02119510
       02173630
COLLATE  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 02119600
      *02173500* 02173640
COMINFO  --  SWITCH FORMAT  --  DECLARED IN SEGMENT 4 AT 00017030
       00017200
COMMON  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00003900
```

```
          00040230  00040240  00064000  00590000  00591000  00592000  00772000  00785700  00794000  00795000  00807280
         *00807290* 00807350 *00807430* 00807520 *00837055* 00839000  00854000 *00904500* 00905000  00908000  00910000
          00920100  00930100  00937500  00938000  01151100  01151900  01152000  01164900  01210000  01211000  01213300
          01281720  01281780  01281835  01281855  01757770  01757840  01856100 *01856200**02118100* 02119244 *02119245*
          02119250 *02119255* 02122050  02122100  02123000  02129050  02137100  02138000  02142000  02142300  02143000
          02144000  02148000  02149000  02149100  02149200  02149300  02149400  02160000  02165000  02165100 *02165130*
          02166000 *02173798*
COMMT  -- REAL ARRAY  -- DECLARED IN SEGMENT 10 AT 00717000
          00824210  00835000  00835100  00837050  02132000  02133000  02134000
COMMT  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 50 AT 00822100
          00822120
COMNT  -- BOOLEAN  -- DECLARED IN SEGMENT 10 AT 00717000
         *00824000**00837000* 02128000
COMNTPAR  -- FORMAT  -- DECLARED IN SEGMENT 49 AT 00809000
          02130000
COMPAREWORDS  -- BOOLEAN STREAM PROCEDURE  -- DECLARED IN SEGMENT 10 AT 00560000
         *00567000* 00619000
COMSAVE  -- BOOLEAN  -- DECLARED IN SEGMENT 47 AT 00807140
         *00807280* 00807430
COMSAVE  -- BOOLEAN  -- DECLARED IN SEGMENT 82 AT 01758500
         *01856100* 02118100
COMSAVE  -- BOOLEAN  -- DECLARED IN SEGMENT 88 AT 02119142
         *02119244* 02119255
CONTROLDESC  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 10 AT 00705000
         *00711000* 00714000  00863110  01164300  01164500  01166300  01166400  01402000  01406000  01465400  01466100
          01467700  01609100
CONV  -- STREAM PROCEDURE  -- DECLARED IN SEGMENT 27 AT 00469000
          00479000
CONVERTIT  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 01349010
          01349230  01349270  01349300
CORE  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00164370
          00967100  00967120  00967130  00967180
CTABLE  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00164200
CTC  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00117100
          01757415
CTF  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00117000
          01266000  01299100  01742140  01985300
CUTBACK  -- LABEL  -- DECLARED IN SEGMENT 70 AT 01362000  -- OCCURS AT 01404000
          01414000
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 2 AT 00036000
          00037000
D  -- REAL  -- DECLARED IN SEGMENT 6 AT 00040120
         *00040230* 00040240
D  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 6 AT 00040200
          00040210
D  -- REAL  -- DECLARED IN SEGMENT 27 AT 00466000
          00479000  00482000 *00486000* 00489000 *00492000* 00493000
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 00560000
          00564000
D  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 41 AT 00763190
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 50 AT 00818000
          00820000
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 62 AT 01231000
          01234000
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 01325000
          01329000
D  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 01750000
```

```
        01754000
DALOC  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164180
DASHES  --  FORMAT  --  DECLARED IN SEGMENT 55 AT 00901000
        00922000
DAT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 27 AT 00469000
        00470000  00472000
DATACOM  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00099000
        01123000  01985100
DATE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00414000
        00495000  00774000  00777000
DATEANALYZED  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00421000
        00791000  00793000
DATES  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00461000
        00778000  00791000
DATETAKEN  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00423000
        00778000  00788000
DATIME  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00770000
        00908000  00937000
DATX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00430000
        00777000  00778000
DAY  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00464000
        00461000 *00493000*
DAY  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 27 AT 00469000
        00475000
DAYS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00428000
        00778000  00788000  00791000  00793000
DAYTABLE  --  REAL ARRAY  --  DECLARED IN SEGMENT 27 AT 00468000
        *00477000* 00489000  00493000
DBARRAY  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164220
DBLSAVE  --  BOOLEAN  --  DECLARED IN SEGMENT 47 AT 00807140
        *00807295* 00807435
DC  --  DEFINE  --  DECLARED IN SEGMENT 60 AT 01110000
        01125000
DCHIT  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01109000
        *01146000**01164200* 01168300  01202000
DCLOG  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00100000
DCQARA  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164250
DCQPTSTACK  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00163000
DCQPTSTACK  --  REAL  --  DECLARED IN SEGMENT 60 AT 01108000
        01125000  01131000
DCQPTSTACKV  --  REAL  --  DECLARED IN SEGMENT 60 AT 01108000
        01125000  01126000 *01127000* 01128000  01129000  01164200
DCSPO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00101000
DCSTACK  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01109000
        *01125000* 01168300  01202000
DCV  --  DEFINE  --  DECLARED IN SEGMENT 60 AT 01110000
        01125000  01126000  01127000  01128000  01129000  01164200
DC19Q  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164325
        01985200
DEBUGGING  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00102000
DECOCT  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 6 AT 00040200
        00040240
DEFINEDMIXMAX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00120000
        00746000
DELTA  --  DEFINE  --  DECLARED IN SEGMENT 70 AT 01364000
        01408000
DESCRIPTOR  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00683000
```

```
       *00691000* 00694000  00701000  00711000  00885000  00887000  00889000  00892000  01081000  01260000  01268000
        01281610  01292000  01468200  01468400  01469500  01540000  01556000  01612000  01711000  01818000
DFX -- DEFINE -- DECLARED IN SEGMENT 10 AT 00103000
        00807550  02117155
DIRECTORYBUILDER -- DEFINE -- DECLARED IN SEGMENT 10 AT 00162000
        00891000
DIRECTORYFREE -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164300
DISK -- FILE -- DECLARED IN SEGMENT 2 AT 00016500
        00053000  00067000  00763145  00844000  00849000  00849100
DISKLOG -- DEFINE -- DECLARED IN SEGMENT 10 AT 00104000
DISKOUNT -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164380
        02117150
DISPLAY -- DEFINE -- DECLARED IN SEGMENT 10 AT 00668100
        00720000  00721000  00722000  00723000  00724000  00725000  00726000  00727000  00728000  00729000  00730000
        00763028  00763470  00807270  00964000  00965000  00966000  00967000  00967180  01279000  01281850  01310000
        01723000  01733000  01757434  01757436  01757440  01757450  01757455  01757470  01816000  01857000  01918200
        01919200  01920000  01971000  01987000  01988000  02064500  02065000  02117150  02117160  02117430
DISPLAYIT -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 00653000
        00668000  00720000  00721000  00722000  00723000  00724000  00725000  00726000  00727000  00728000  00729000
        00730000  00763028  00763470  00807270  00807370  00964000  00965000  00966000  00967000  00967180  01279000
        01281850  01310000  01723000  01733000  01757434  01757436  01757440  01757450  01757455  01757470  01816000
        01857000  01918200  01919200  01920000  01971000  01987000  01988000  02064500  02065000  02117150  02117160
        02117430
DONTDUMPRT -- DEFINE -- DECLARED IN SEGMENT 2 AT 00009000
        00904500  00908000  00910000  00920100  00930100
DONTPRINTLINKS -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00018000
        00644310 *02142260**02143100**02150075*
DOUBLESPACE -- DEFINE -- DECLARED IN SEGMENT 2 AT 00004200
        00622000  00645000  00807295  00807300  00807435
DP -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 01325000
        01326000  01329000
DUMMY -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164999
DUMPALPHAOCTAL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00011000
        00591000
DUMPALPHAONLY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00012000
        00590000
DUMPARRAY -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 00763020
        00763048  01919100
DUMPAUXMEM -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 02119100
        02119290  02172200
DUMPAVAIL -- DEFINE -- DECLARED IN SEGMENT 2 AT 00005000
        01151900
DUMPCESSPOOL -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 00763100
        00807899  02122300
DUMPCESSPOOLONLY -- DEFINE -- DECLARED IN SEGMENT 2 AT 00013000
        00904500  00937500  02122100
DUMPCONTROLSTACKS -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 01700000
        02172000
DUMPD -- INTEGER -- DECLARED IN SEGMENT 10 AT 02120000
       *02120245**02145000**02147000**02152000**02156000* 02160000 *02161000* 02165000
DUMPIT -- LABEL -- DECLARED IN SEGMENT 70 AT 01360100 -- OCCURS AT 01457100
        01381100
DUMPMCPINFO -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 01758000
        02169000
DUMPMCPSAVECODE -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 01281500 -- FORWARD AT 01096500
        01213100  01281910  02150010
DUMPMCPSPRT -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 00896000
```

```
        02122000
DUMPMEMORYANDNOTESTACKS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01097000
        02152000  02156000  02161000
DUMPOCTALONLY  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00010000
        00592000
DUMPP  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00105000
DUMPROGRAMS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01520000
        02171000
DUMPSTACK  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01353000
        01696000  01708000  01726000  01734000  01740000  01742160
DUMPTABLES  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00008000
        00807520
DUPBED  --  FORMAT  --  DECLARED IN SEGMENT 59 AT 01069000
        01084000
D32  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 62 AT 01231000
        01232000  01235000
E  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 41 AT 00763190
        00763195
EDITLINE  --  LABEL  --  DECLARED IN SEGMENT 29 AT 00589100  --  OCCURS AT 00637100
        00610100
EF  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00812010
EOJ  --  LABEL  --  DECLARED IN SEGMENT 41 AT 00763165  --  OCCURS AT 00763570
        00763455  00763480  00763495  00763505
EOPROG  --  LABEL  --  DECLARED IN SEGMENT 10 AT 00080400  --  OCCURS AT 02173400
        00866000  00867000  00868000  02122400
EOT  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00812005  --  OCCURS AT 00838000
        00835000  00837100
ERRMESS  --  DEFINE  --  DECLARED IN SEGMENT 52 AT 00881000
        00893000  00894000
ERROWT  --  LABEL  --  DECLARED IN SEGMENT 70 AT 01360000  --  OCCURS AT 01510000
        01381000
ESP  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 01222000
       *01262000* 01263000  01266000  01270000  01281620 *02120210* 02150010
ESPBIT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00124000
        01260000  01264000  01281620
EUIO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164130
EUQ  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164280
        00807550  02117200
EUW  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164390
        02117160
EXIT  --  LABEL  --  DECLARED IN SEGMENT 54 AT 00904100  --  OCCURS AT 00931100
        00920100  00930100
EXIT  --  LABEL  --  DECLARED IN SEGMENT 68 AT 01338000  --  OCCURS AT 01348000
        01346000
EXIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 01349160  --  OCCURS AT 01349470
        01349370  01349374  01349460
EXIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 01753000  --  OCCURS AT 01757000
        01755000
EXIT  --  LABEL  --  DECLARED IN SEGMENT 88 AT 02119140  --  OCCURS AT 02119280
        02119160
EXIT  --  LABEL  --  DECLARED IN SEGMENT 93 AT 02120130  --  OCCURS AT 02120700
        02120135
EXPAND  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 29 AT 00586100
        00586140  00644400
F  --  FORMAT  --  DECLARED IN SEGMENT 7 AT 00040140
        00040270
F  --  FORMAT  --  DECLARED IN SEGMENT 40 AT 00763026
```

```
          00763042
F  --  FORMAT  --  DECLARED IN SEGMENT 55 AT 00902000
          00935000
F  --  FORMAT  --  DECLARED IN SEGMENT 64 AT 01258000
          01277000
F  --  FORMAT  --  DECLARED IN SEGMENT 67 AT 01289000
          01311000
F  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01352000
          01351000
F  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370020
          01370030  01370120
F  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370260
          01370270  01370350
F  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01514000
          01512000 *01517000**01518000*
F  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
          01576000  01577000  01580000  01584000  01588000  01592000  01593000  01597000  01601000  01602000
FA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
          02069000  02112000  02113000  02117450  02117460  02117490
FANAL  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00444400
          00785550
FCOMVAL  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00444300
          00785700
FCORE  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00956100
          00967120
FF  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00115000
          00327000  00329000  00666000  00863120  00918000  00919000  00928000  00929000  00967150  00971000  00972400
          00982000  00996000  00997000  01015000  01018000  01030000  01047000  01048000  01059000  01090000  01135000
          01151500  01164000  01164400  01164600  01166300  01166500  01168600  01169100  01198000  01281660  01281855
          01281860  01302000  01317000  01320000  01342000  01349580  01349650  01377000  01396000  01404000  01449000
          01465400  01465500  01466000  01468300  01468400  01518000  01609200  01618000  01628000  01643000  01682000
          01683000  01684000  01684100  01695000  01721000  01757410  01984000  02008000  02149400
FF  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 44 AT 00763300
          00763320  00763345
FILETYPE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00082330
          00785600  00834070  00834080  00834200
FILLAREATYPE  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00295120
          02120920
FILLINFO  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00165000
          00357000
FILLXNAMS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 12 AT 00248000
          00269000  00291000
FINALQUE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00141000
          02069000
FINI  --  FORMAT  --  DECLARED IN SEGMENT 5 AT 00023000
          02173793  02176000
FIRST  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00587000
         *00597000* 00604000 *00611000* 00644100 *00644300*
FIRSTWAIT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00149000
FIX  --  STREAM LABEL  --  DECLARED IN SEGMENT 46 AT 00771105  --  OCCURS AT 00771145
          00771135  00771140
FIXDEFINES  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00341000
          00862000
FIXFID  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 02119300
          02173780
FIXLINE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 47 AT 00807160
          00807250  00807380
```

```
FLAG  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00813000  --  OCCURS AT 00868000
        00828000
FLAG  --  BOOLEAN  --  DECLARED IN SEGMENT 69 AT 01349555
        01349670 *01349680*
FLAGERR  --  FORMAT  --  DECLARED IN SEGMENT 51 AT 00816000
        00868000
FLGBIT  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00552000
       *00556000* 00557000  00675000  00774000  00780000  00796000  00828000
FLL  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 41 AT 00763190
        00763285  00763510
FLQUE  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774130
        02117540
FMXX  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00436000
        00793000
FMX2  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00444000
        00802000
FMX3  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00445000
        00804000
FNEWFILE  --  FORMAT  --  DECLARED IN SEGMENT 91 AT 02119330
        02119380
FORI  --  DEFINE  --  DECLARED IN SEGMENT 29 AT 00573000
        00608000  00617000
FORMATCODENAME  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01349120
        01349490  01349570  01349610  01349640  01349700  01466300
FORMATDESC  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 70 AT 01370530
        01370930  01468300
FORMATMSCW  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 70 AT 01370260
        01467900
FORMATOP  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 70 AT 01370940
        01469200
FORMATPROGDESC  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 70 AT 01371110
        01371430  01470000
FORMATRCW  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 70 AT 01370020
        01370250  01466700
FOUND  --  BOOLEAN  --  DECLARED IN SEGMENT 68 AT 01337000
       *01341000* 01348000
FPB  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
        01580000
FPUNT  --  FORMAT  --  DECLARED IN SEGMENT 92 AT 02119505
        02129170
FROM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00571000
        00569000  00570000  00597100 *00598000* 00599000  00602000  00606000  00608000  00615000  00617000  00638000
        00639000  00640000  00641000  00643000 *00648000*
FROM  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00578000
FROM  --  REAL  --  DECLARED IN SEGMENT 41 AT 00763140
        00763445 *00763450* 00763580  00763585 *00763590* 00763595  00763600 *00763605* 00763615
FROM  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01096500
FROM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01099000
        01097000  01098000  01143000  01149000  01151300  01151400  01152000  01163000  01164200  01164300  01164500
        01166200  01166300  01166400  01167000  01167100  01167700  01168000  01168600  01168700  01169100  01169200
        01169400  01169500  01169550  01197000  01198000  01203000  01213100  01213300 *01218000*
FROM  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 60 AT 01115300
        01115500  01116200
FROM  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01281520
        01281500  01281510  01281640  01281690  01281720 *01281730* 01281760  01281780 *01281790* 01281820 *01281860*
FS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164210
```

```
FS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
        02069000  02071000  02109000
FT  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01775000
        02096000
FTAPDSK  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00444200
        00785600
FUNNYMSCW  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357100
        01465500 *01466400* 01467800
F1  --  FORMAT  --  DECLARED IN SEGMENT 37 AT 00740000
        00762000
GETANDSORTMCPROG  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 01281400
        02126500
GETCOMMON  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 50 AT 00822100
        00822260  00837050
GETIDLOC  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00336000
GETNUM  --  STREAM LABEL  --  DECLARED IN SEGMENT 50 AT 00822110  --  OCCURS AT 00822175
        00822170
GETPRTENTRIES  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00732000
        00763010  00941000
GETSORTANDLISTINTRINSICS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01284000
        02170000
GETSORTANDLISTMCPROG  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01252000
        02126500  02168000
GETSTACKSFROMTHEBED  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01064000
        01096000  02126000
GLUNK  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 41 AT 00763355
        *00763365* 00763370  00763560
GOTIDFNT  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 19 AT 00319000
        *00322000* 00323000  00329000
H  --  INTEGER  --  DECLARED IN SEGMENT 31 AT 00658000
        00658600 *00660000* 00662000
H  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
        01387000  01388000  01394000  01395000  01418575  01418580  01439000  01440000  01441000  01442000  01465200
        01465300  01466700  01468000  01468400  01468450
H  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
        *01580000* 01581000 *01584000* 01585000 *01588000* 01589000 *01592000* 01593000 *01597000**01601000* 01602000
        *01606000* 01607000  01609100  01609200  01625000  01626000
H  --  FORMAT  --  DECLARED IN SEGMENT 78 AT 01702000
        01706000
HD  --  FORMAT  --  DECLARED IN SEGMENT 71 AT 01367000
        01374000
HD  --  FORMAT  --  DECLARED IN SEGMENT 75 AT 01527000
        01571000
HDR  --  FORMAT  --  DECLARED IN SEGMENT 42 AT 00763110
        00763475
HDR  --  FORMAT  --  DECLARED IN SEGMENT 55 AT 00898000
        00912000
HIHALF  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00544000
        *00547000* 00660000  00763042  00915000  00917000  00925000  00927000  00967120  01757580  01757720  01757810
        01757855  01757860  01977000  02004000  02005000  02027000  02044000  02054000  02080000  02094000  02101000
        02112000  02117520  02142200
HIT  --  STREAM LABEL  --  DECLARED IN SEGMENT 50 AT 00822110  --  OCCURS AT 00822170
        00822150
HRS  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00452000
        *00455000**00457000*
H1  --  FORMAT  --  DECLARED IN SEGMENT 75 AT 01528000
        01552000
```

```
I  --  INTEGER  --  DECLARED IN SEGMENT 3 AT 00017020
       *00017200*
I  --  INTEGER  --  DECLARED IN SEGMENT 8 AT 00046000
       *00051000* 00054000 *00057000* 00058000
I  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00081000
       *00410000* 00411000 *01241000* 01243000 *01244000* 01245000  01248000 *01267000* 01268000  01274000 *01278000*
        01279000 *01297000* 01298000  01299100  01305000 *01313000* 01315000  01316000  01317000 *01340000* 01342000
        01344000  01345000 *01633000* 01634000 *01637000**01665000* 01667000  01668000 *01674000* 01676000  01677000
        01679000 *01757500**01757530**01757950**01861000* 01862000 *01912000* 01914000  01915000  01925000 *01960000*
        01961000  01962000 *01963000* 01964000  01965000 *01966000* 01967000  01968000 *01972000**01982000**01985390*
       *02021000* 02023000  02024000  02027000  02032000  02037000  02038000  02039000  02040000  02041000  02044000
        02047000  02051000  02054000  02057000 *02073000* 02075000  02076000  02077000  02080000  02087000  02091000
        02094000  02095000  02098000  02101000  02105000  02109000  02112000  02113000 *02120145**02129130* 02129150
       *02173610* 02173630  02173640
I  --  INTEGER  --  DECLARED IN SEGMENT 12 AT 00271000
       *00293000* 00294000
I  --  INTEGER  --  DECLARED IN SEGMENT 19 AT 00318000
       *00325000* 00326000  00328000  00331000 *00332000* 00334000
I  --  INTEGER  --  DECLARED IN SEGMENT 20 AT 00351000
       *00358000* 00359000  00360000  00362000 *00364000* 00365000
I  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 00588000
       *00593000**00594100* 00594200 *00608000* 00609000 *00617000* 00618000
I  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 00736000
       *00752000* 00753000  00757000 *00762000*
I  --  INTEGER  --  DECLARED IN SEGMENT 39 AT 00763024
       *00763040* 00763042
I  --  REAL  --  DECLARED IN SEGMENT 41 AT 00763140
        00763440  00763450  00763465  00763485 *00763495* 00763505 *00763515* 00763535  00763540  00763555 *00763560*
        00763580  00763585  00763590 *00763595**00763605* 00763615 *00763620*
I  --  INTEGER  --  DECLARED IN SEGMENT 44 AT 00763295
       *00763325* 00763330
I  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 41 AT 00763375
       *00763415*
I  --  INTEGER  --  DECLARED IN SEGMENT 46 AT 00771000
       *00786000* 00787000  00788000  00790110  00790120
I  --  INTEGER  --  DECLARED IN SEGMENT 48 AT 00807515
       *00807580*
I  --  INTEGER  --  DECLARED IN SEGMENT 50 AT 00823000
       *00824110**00825000**00833000**00858000* 00859000 *00860000* 00861000  00863110 *00863120* 00863130  00866000
        00867000  00868000
I  --  REAL  --  DECLARED IN SEGMENT 52 AT 00873000
        00885000  00887000  00889000  00892000
I  --  INTEGER  --  DECLARED IN SEGMENT 58 AT 01066000
       *01080000* 01081000  01084000  01091000
I  --  INTEGER  --  DECLARED IN SEGMENT 65 AT 01281560
       *01281600* 01281610  01281620  01281660 *01281680* 01281700  01281760  01281835  01281850  01281855  01281860
       *01281865**01281890*
I  --  INTEGER  --  DECLARED IN SEGMENT 69 AT 01349550
       *01349556**01349670* 01349680
I  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
       *01418530**01418540* 01418560 *01458000* 01465200  01465400  01465500  01465900  01466000  01467700  01467800
        01468200  01468300  01468600  01469500
I  --  INTEGER  --  DECLARED IN SEGMENT 77 AT 01704000
       *01710000**01713000* 01719000 *01724000* 01729000  01731000
IA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
        02067000  02094000  02095000
ID  --  ALPHA ARRAY  --  DECLARED IN SEGMENT 10 AT 00315000
        00360000  00361000
```

```
ID -- REAL ARRAY -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 00316000
      00329000
IDLENGTH -- INTEGER STREAM PROCEDURE -- DECLARED IN SEGMENT 20 AT 00352000
     *00355000* 00359000
IFO -- FORMAT -- DECLARED IN SEGMENT 85 AT 01852000
      02075000
IGNOREPAR -- LABEL -- DECLARED IN SEGMENT 50 AT 00813000 -- OCCURS AT 00836000
      00835000
IGPAR -- LABEL -- DECLARED IN SEGMENT 50 AT 00812010 -- OCCURS AT 00824220
      00824210
ILL -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164330
      01985300
IMAX -- INTEGER -- DECLARED IN SEGMENT 66 AT 01286000
     *01294000**01295200* 01296000  01297000
INAME -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 00087000
      00847000 *00860000* 00861000  01169100  01169200  01319000  01320000  01349640  01349650  01643000
INAMESIZE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00032000
     *00064000**00072000* 00087000  00847000  00857000  01282000  02120650
INAMS -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 00088000
      00848000  01169200  01319000  01349640  01643000
INAMSSIZE -- INTEGER -- DECLARED IN SEGMENT 2 AT 00032000
     *00064000**00073000* 00088000  00848000
INCOMPAT -- FORMAT -- DECLARED IN SEGMENT 53 AT 00880000
      00893000
INFO -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 00084000
     *00166000* 00359000  00360000  00763415  00763495  00763505  00763515  00763560  00763600  00763615  01699000
      02119000
INFOMASK1 -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164350
      01757436
INFOMASK2 -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164355
      01757440
INFOMAX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00024000
     *00078000* 00084000  00358000
INIT -- STREAM PROCEDURE -- DECLARED IN SEGMENT 41 AT 00763170
      00763185  00763435
INITIALIZE -- DEFINE -- DECLARED IN SEGMENT 10 AT 00156000
INITIALIZE -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 02120100
      02120705  02120910
INQUIRY -- DEFINE -- DECLARED IN SEGMENT 10 AT 00106000
INTCODE -- STREAM LABEL -- DECLARED IN SEGMENT 60 AT 01115800 -- OCCURS AT 01119400
      01116900
INTMAX -- INTEGER -- DECLARED IN SEGMENT 2 AT 00024000
     *00857000* 00860000  01169000  01318000  01642000
INTR -- BOOLEAN -- DECLARED IN SEGMENT 60 AT 01104000
     *01135000* 01168900
INTRINSIC -- STREAM LABEL -- DECLARED IN SEGMENT 10 AT 01349160 -- OCCURS AT 01349410
      01349350
INTRNSC -- DEFINE -- DECLARED IN SEGMENT 10 AT 00132000
      00723000  01135000  01292000  01310000
INTSP -- DEFINE -- DECLARED IN SEGMENT 41 AT 00763155
      00763415  00763495  00763505  00763515  00763560  00763600  00763615
INTSP -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 01282000
     *01299000* 01312000  01315000  01316000  01317000  01349630  01634000 *02120655*
INTSPMAX -- INTEGER -- DECLARED IN SEGMENT 10 AT 01283000
     *01291000**01299000* 01307000  01312000  01313000  01349630  01633000  01637000 *02120220*
INX -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 00807110
      00807100  00807260 *00807275* 00807370  00807380
```

```
INX  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 47 AT 00807160
         00807210
INX  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01355000
         01353000  01354000  01375000  01377000  01381100  01382000  01383000  01399000  01438000
INX  --  INTEGER  --  DECLARED IN SEGMENT 77 AT 01704000
        *01715000* 01716000  01718000  01721000  01726000 *01738000* 01740000  01741000
IOATH  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01778000
         02020000
IOQSH  --  FORMAT  --  DECLARED IN SEGMENT 85 AT 01824000
         02072000
IOQUE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00144000
         02067000
IOQUEAVAIL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00143000
         02065000
IOQUESLOTS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164440
         02064500
IOTIME  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00139000
IRSTACK  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01106000
        *01145000**01164300* 01168400  01199000
IS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
         02067000  02071000  02091000
ISNAME  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00033000
         00860000  00861000
ISTACK  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00137000
         00727000  01132000  01711000  01723000  01725000  01733000
ISTACKOK  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01106000
        *01132000* 01168400
ISTACKV  --  REAL  --  DECLARED IN SEGMENT 60 AT 01106000
         01132000  01133000 *01134000* 01164400  01164600
ITD  --  INTEGER ARRAY  --  DECLARED IN SEGMENT 10 AT 00418000
         00778000  00784000  00788000  00789000  00791000  00792000  00793000  00801000  00802000  00803000  00804000
ITEM  --  FORMAT  --  DECLARED IN SEGMENT 30 AT 00651000
         00659000  01976000  02142200
ITEM  --  FORMAT  --  DECLARED IN SEGMENT 61 AT 01103000
ITEM  --  FORMAT  --  DECLARED IN SEGMENT 67 AT 01288000
         01315000
ITEM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01335000
         01332000  01333000  01342000  01345000
J  --  INTEGER  --  DECLARED IN SEGMENT 12 AT 00271000
        *00292000**00294000*
J  --  INTEGER  --  DECLARED IN SEGMENT 44 AT 00763295
        *00763330*
J  --  INTEGER  --  DECLARED IN SEGMENT 65 AT 01281560
         01281610  01281620  01281630  01281640  01281650  01281660 *01281760* 01281780  01281790  01281820  01281855
J  --  REAL  --  DECLARED IN SEGMENT 79 AT 01757335
        *01757897*
J  --  INTEGER  --  DECLARED IN SEGMENT 93 AT 02120125
JAR  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00131000
         00722000  01540000
JARO  --  REAL  --  DECLARED IN SEGMENT 74 AT 01526000
         01540000 *01541000* 01554000  01556000
JAROO  --  REAL  --  DECLARED IN SEGMENT 74 AT 01525000
         01556000 *01557000* 01559000
JORNUM  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00134000
         00725000  01073000
K  --  INTEGER  --  DECLARED IN SEGMENT 36 AT 00736000
        *00750000**00756000* 00757000 *00758000* 00762000
```

```
K  --  INTEGER  --  DECLARED IN SEGMENT 65 AT 01281560
        01281610  01281620  01281630  01281660
K  --  INTEGER  --  DECLARED IN SEGMENT 93 AT 02120125
       *02120605* 02120610 *02120650* 02120655
KIND  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00686000
        00683000 *00689000*
KIND  --  INTEGER  --  DECLARED IN SEGMENT 34 AT 00700000
        00701000  00702000
KINDS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00434000
KLASS  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00029000
       *00324000* 00328000 *00360000**00363000*
L  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 20 AT 00353000
        00355000
L  --  INTEGER  --  DECLARED IN SEGMENT 31 AT 00658000
       *00661000* 00662000
L  --  REAL  --  DECLARED IN SEGMENT 36 AT 00734000
        00744000  00745000  00748000  00749000
L  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 46 AT 00771100
        00771110
L  --  REAL  --  DECLARED IN SEGMENT 60 AT 01102000
       *01149000**01149100* 01149200  01151000  01151600  01164000  01167000  01168300  01168700  01169200  01169400
        01169550  01211000
L  --  INTEGER  --  DECLARED IN SEGMENT 62 AT 01229000
       *01242000* 01243000 *01245000* 01247000
L  --  REAL  --  DECLARED IN SEGMENT 66 AT 01287000
        01301000 *01302000* 01303000  01304000 *01315000* 01318000  01319000  01320000
L  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
        01387000  01388000  01394000  01395000  01418575  01418580  01439000  01440000  01441000  01442000  01444000
        01465200  01465300  01465600  01466700
L  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01523000
       *01631000* 01634000  01635100  01641000  01642000  01643000
L  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        01995000  01996000  01997000  01998000  01999000  02010000  02117510  02117520  02117530  02117540
LA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        02015000  02040000  02068000  02101000  02105000 *02117510* 02117540
LABELDESC  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01371135  --  OCCURS AT 01371256
        01371250
LABELTABLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00150000
        02015000
LAST  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00587000
       *00602000* 00634000
LASTROW  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00558000
        00606000  00609000  00619000  00630000
LENGTH  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370530
        01370550  01370660  01370720  01370730  01370770
LEVEL  --  REAL  --  DECLARED IN SEGMENT 2 AT 00028000
       *00068000* 00409000
LFO  --  FORMAT  --  DECLARED IN SEGMENT 85 AT 01855000
        02100000
LIM  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01227000
        01224000  01225000  01238000  01241000
LINE  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00578000
LINE  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00652000
        00659000  00665000  00666500  00667000  00763345  00763415  00763420  00763510  00763520  00785400  00785500
        00807380  00807390  00915000  00918000  00919000  00920000  00925000  00928000  00929000  00930000  00967120
        00967160  01151400  01151700  01168600  01169100  01169400  01169600  01315000  01319000  01321000  01394000
```

```
          01396000  01397000  01446000  01448000  01450000  01465200  01466700  01467900  01468300  01469200  01470000
          01470200  01552000  01560000  01562000  01566000  01567000  01569000  01639000  01643000  01644000  01757580
          01757600  01757620  01757720  01757760  01757810  01757830  01757855  01757875  01757885  01914000  01915000
          01916000  01976000  01978000  01979000  02003000  02007000  02009000  02026000  02035000  02043000  02049000
          02053000  02059000  02079000  02089000  02093000  02096000  02100000  02107000  02111000  02114000  02129170
          02129180  02129190  02173630  02173640  02173650
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 47 AT 00807160
          00807180
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 60 AT 01115300
          01115900  01116100
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01370020
          01370060  01370090
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01370260
          01370290  01370320
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01370530
          01370580  01370610
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01370940
          01370970  01371000
LINE  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01371110
          01371140  01371170
LINE2 -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 70 AT 01370940
          01371030
LINK  -- REAL -- DECLARED IN SEGMENT 56 AT 00948000
         *00976100* 00980000  00983000  00985000  00987000 *00988000* 01007000  01009000 *01011000* 01012000  01015000
          01017000 *01018000* 01022000 *01036000* 01036100  01037000  01040000  01042000  01044000  01049000  01051000
LINKOK  -- BOOLEAN PROCEDURE -- DECLARED IN SEGMENT 56 AT 00950000
         *00953000* 00972400  00981000  01013000  01038000
LINKTYPE  -- REAL -- DECLARED IN SEGMENT 10 AT 00295100
          00593000 *01149100* 01166100 *01166600* 01166900 *01167300* 01167700 *01167800**01168100* 01168300 *01168500*
          01168550  01168900  01169400  01200000  01210000  01213000 *01281835*
LINK1  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 60 AT 01115400
          01115500  01121300
LINK2  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 60 AT 01115400
          01115500  01120950  01121500
LINK3  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 60 AT 01115400
          01115500  01121700
LISTMCPROG  -- DEFINE -- DECLARED IN SEGMENT 10 AT 01281420
          02168000
LL  -- REAL -- DECLARED IN SEGMENT 60 AT 01104000
         *01168900* 01169000  01169100  01169200 *01197000*
LNKSOK  -- BOOLEAN -- DECLARED IN SEGMENT 10 AT 00498000
         *00976000* 00979000 *00980000* 01000000  01033000  01034000  01035000  01051100  01052100  01058000 *02120405*
          02151000
LO  -- BOOLEAN -- DECLARED IN SEGMENT 41 AT 00763140
         *00763450* 00763490  00763570 *00763575*
LOAD  -- PROCEDURE -- DECLARED IN SEGMENT 10 AT 00810000
          00869000  02119180  02121000
LOC  -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 00316000
          00327000 *00331000**00334000*
LOC  -- INTEGER -- DECLARED IN SEGMENT 20 AT 00351000
         *00359000* 00361000  00362000
LOC  -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 00546000
          00544000  00545000  00547000
LOC  -- INTEGER -- VALUE PARAMETER -- DECLARED IN SEGMENT 10 AT 00550000
          00548000  00549000  00551000
LOC  -- INTEGER -- DECLARED IN SEGMENT 39 AT 00763024
          00763032  00763034 *00763036* 00763042
```

```
LOC  --  INTEGER  --  DECLARED IN SEGMENT 47 AT 00807130
      00807330  00807340 *00807350* 00807370
LOC  --  REAL  --  DECLARED IN SEGMENT 52 AT 00873000
     *00884000* 00885000 *00886000* 00887000 *00888000* 00889000 *00891000* 00892000
LOC  --  INTEGER  --  DECLARED IN SEGMENT 54 AT 00904000
     *00913000* 00915000  00916000  00918000 *00923000* 00925000  00926000  00928000
LOCATQUE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00142000
      02068000
LOCIRCW  --  REAL  --  DECLARED IN SEGMENT 10 AT 01352500
     *01609100**01609200*
LOHALF  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00548000
     *00551000* 00661000  00763042  00916000  00917000  00926000  00927000  00967130  01757590  01757730  01757820
      01757860  01757865  01977000  02004000  02005000  02032000  02047000  02057000  02076000  02087000  02095000
      02105000  02113000  02117530  02142250
LOOKQ  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164315
      01985150
LOOP  --  LABEL  --  DECLARED IN SEGMENT 41 AT 00763165  --  OCCURS AT 00763500
      00763560  00763620
LQAVAIL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164420
      02117430  02117450
LQUE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164410
      02117440  02117560
LQUEHDR  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774100
      02117480
LS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
      02015000  02019000  02039000  02068000  02071000  02098000
LSLATE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00123000
      01988000  01995000
LSTD  --  REAL ARRAY  --  DECLARED IN SEGMENT 74 AT 01532000
     *01668000* 01675000  01676000  01677000  01681000 *01685000*
LUN  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774000
      02023000
L1  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01370570  --  OCCURS AT 01370650
      01370630
L1  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01371135  --  OCCURS AT 01371210
      01371190
M  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00114000
      00547000  00551000  00606000  00608000  00615000  00617000  00640000  00641000  00643000  00644000  00675000
      00679000  00689000  00692000  00763600  00763615  00774000  00777000  00780000  00783000  00796000  00799000
      00824110  00832000  00832100  01149000  01516000  01517000  01559000  01561000  01757600  01757875  01978000
      02038000  02040000  02129180
M  --  REAL  --  DECLARED IN SEGMENT 27 AT 00466000
     *00484000**00488000* 00489000  00491000  00492000  00493000
M  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00963000
      01035000 *01048000* 01055000  01057000
M  --  INTEGER  --  DECLARED IN SEGMENT 58 AT 01068000
     *01083000*
MA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
      02014000  02038000
MANY  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00536000
      00537000  00541000  00542000
MATCH  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00587000
     *00619000* 00627000  00636000
MAXBAD  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00715000
     *01020000**01021000**02120185* 02160000  02161000
MAXCOMMONVALUE  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00016000
MAXCOR  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00496000
```

```
        00807350 *00853000* 00971000  01020000  01036000  01757610  01757880  01822000  01978000  02165000
MAXLNK  -- INTEGER -- DECLARED IN SEGMENT 10 AT 00715000
        *00971000* 00979000  00983000  00991000  01001000  01011000  01021000  01027000  01056000  01058000  01060000
        *02120190* 02152000  02161000
MAXMCPDATA  -- INTEGER -- DECLARED IN SEGMENT 65 AT 01281560
        *01281590**01281650* 01281670  01281700
MAXMCPROG  -- INTEGER -- DECLARED IN SEGMENT 10 AT 01222000
        *01261000**01265000* 01265900 *01271000* 01275000  01278000  01280000  01349560  01444000 *02120205*
MAXMESSAGES  -- DEFINE -- DECLARED IN SEGMENT 10 AT 01747000
        01757530  01757897  01757950  01982000
MAXMOD  -- INTEGER -- DECLARED IN SEGMENT 10 AT 00496000
        *00851000**00852000* 00853000
MAXOPT  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00111000
MAXSTACKSIZE  -- DEFINE -- DECLARED IN SEGMENT 70 AT 01365000
        01454000  01456000  01457000
MAXSTK  -- INTEGER -- DECLARED IN SEGMENT 10 AT 00095000
        *01087000**01203000* 01737000  01738000  01741000 *02120235*
MAYBE  -- STREAM LABEL -- DECLARED IN SEGMENT 46 AT 00771105  -- OCCURS AT 00771135
        00771125
MAYBE  -- LABEL -- DECLARED IN SEGMENT 52 AT 00875000  -- OCCURS AT 00893000
        00885000  00887000  00889000  00892000
MCP  -- BOOLEAN -- DECLARED IN SEGMENT 2 AT 00035000
        *00069000* 00411000  00790100  00807550  01123000  01281000  01985100  01985350  02117155  02117400  02119000
MCPCNT  -- INTEGER STREAM PROCEDURE  -- DECLARED IN SEGMENT 10 AT 00765000
        *00768000* 00769000  00786000
MCPCODE  -- STREAM LABEL  -- DECLARED IN SEGMENT 60 AT 01115800  -- OCCURS AT 01119100
        01116800
MCPCODE  -- STREAM LABEL  -- DECLARED IN SEGMENT 10 AT 01349160  -- OCCURS AT 01349380
        01349330
MCPDATA  -- REAL ARRAY  -- DECLARED IN SEGMENT 65 AT 01281540
        *01281650* 01281670  01281760  01281835  01281850  01281855  01281860
MCPENTRIES  -- PROCEDURE  -- DECLARED IN SEGMENT 10 AT 00718000
        00940000
MCPHDR  -- FORMAT  -- DECLARED IN SEGMENT 17 AT 00299000
        00409000
MCPOPT  -- SWITCH FORMAT  -- DECLARED IN SEGMENT 18 AT 00300000
        00411000
MCPOUTERBLOCK  -- STREAM LABEL  -- DECLARED IN SEGMENT 10 AT 01349160  -- OCCURS AT 01349371
        01349340
MCPROG  -- REAL ARRAY  -- DECLARED IN SEGMENT 10 AT 01221000
        *01261000**01271000* 01275000  01279000  01349560  01444000
MCPSAVE  -- BOOLEAN  -- DECLARED IN SEGMENT 60 AT 01107100
        *01164000* 01168550  01208100  01211000 *01215100*
MCPVERSION  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00435000
        00773000  00786000  00787000  00788000
MDUMP  -- FILE  -- DECLARED IN SEGMENT 2 AT 00016510
        00824210  00826990  00827000  00834050  00834060  00834070  00834095  00834097  00835000  00837100  00838000
        02119370  02173750  02173760
MEMASK  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00164190
        00807560
MEMORY  -- REAL ARRAY  -- DECLARED IN SEGMENT 10 AT 00112000
        00547000  00551000  00606000  00608000  00615000  00617000  00640000  00641000  00643000  00644000  00675000
        00679000  00689000  00692000  00763600  00763615  00774000  00777000  00780000  00783000  00796000  00799000
        00824110  00832000  00832100  01149000  01220000  01516000  01517000  01559000  01561000  01757600  01757875
        01978000  02038000  02040000  02129180
MEND  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00127000
        00966000
```

```
MESSAGEHOLDER  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00154000
        01971000  01973000
MIN  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00452000
        00451000 *00455000**00458000*
MINBAD  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00715000
       *01009000**02120180* 02156000
MINLNK  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00715000
       *00976100* 01000000  01003000  01005000 *02120175* 02145000  02147000  02149300  02150010  02152000
MINUS1  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 90 AT 02119340
        02119360
MINUTES  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00426000
        00784000  00789000  00792000  00793000  00801000  00802000  00803000  00804000
MIX  --  REAL ARRAY  --  DECLARED IN SEGMENT 36 AT 00737000
       *00757000**00758000* 00762000
MIX  --  INTEGER  --  DECLARED IN SEGMENT 58 AT 01067000
       *01082000* 01083000  01084000  01088000  01092000
MIX  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 60 AT 01115300
        01115500  01118300
MIX  --  REAL  --  DECLARED IN SEGMENT 66 AT 01287100
       *01295200* 01310100
MIX  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
       *01545000* 01545100  01546000  01552000  01556000  01575000  01576000  01690000  01695100 *01697000**01698200*
        01698300
MIXMASK  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164345
        01757434
MIXMAX  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00092000
       *00746000**00751000* 00954000  01545000  01697000  01698200  02120605
MIXSTK  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 01063000
        01083000 *01088000* 01690000 *01695100* 01698300 *02120610*
MMM  --  PROCEDURE  --  DECLARED IN SEGMENT 41 AT 00763375
        00763535  00763540
MODE  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01371135  --  OCCURS AT 01371360
        01371254
MODON  --  BOOLEAN ARRAY  --  DECLARED IN SEGMENT 10 AT 00417000
       *00829000* 00852000  00936000
MOM  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 60 AT 01115400
        01115500  01120900
MOMADDRESS  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370540
        01370550  01370850
MOMFLG  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370540
        01370550  01370810
MONITORR  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00110500
MONTH  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00464000
        00461000 *00491000*
MOVADR  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 01349160  --  OCCURS AT 01349250
        01349190  01349200  01349210
MOVC  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01325000
        01331000  01559000  01561000  01566000  01567000  01915000  02035000  02038000  02040000  02049000  02059000
        02089000  02096000  02107000  02114000
MOVD  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01750000
        01757100  01757600  01757875  01978000  02129180
MOVE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00036000
        00037000  00054000  00058000  00360000  00606000  00615000  00630000  00665000  00763600  00763615  00835100
        00918000  00919000  00928000  00929000  01319000  01396000  01448000  01643000  02007000  02133000  02173640
MOVF  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 45 AT 00763385
        00763405  00763415
MOVE  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 62 AT 01231000
```

```
        01237000  01247000
MOVER  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 50 AT 00818000
        00822000  00832000
MOVNAM  --  STREAM LABEL  --  DECLARED IN SEGMENT 60 AT 01115850  --  OCCURS AT 01119600
        01119300
MOVNAM  --  STREAM LABEL  --  DECLARED IN SEGMENT 10 AT 01349160  --  OCCURS AT 01349430
        01349400
MOVSIZ  --  STREAM LABEL  --  DECLARED IN SEGMENT 60 AT 01115850  --  OCCURS AT 01120000
        01118500  01119000  01119900
MOVTYP  --  STREAM LABEL  --  DECLARED IN SEGMENT 60 AT 01115850  --  OCCURS AT 01117500
        01117200
MS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        02014000  02019000  02037000
MSFF  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370260
        01370270  01370400
MSG  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349120
        01349170 *01349230**01349270**01349300*
MSG  --  REAL ARRAY  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349530
        01349500  01349570  01349610  01349640  01349700
MSTART  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00126000
        00965000  00972500  00973000
MSTARTOK  --  BOOLEAN  --  DECLARED IN SEGMENT 56 AT 00947000
       *00972500**00973000* 00976000  00976100  00976200
MULTI  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00082150
        00824200  00826990  00834075 *00834085* 00837095  02173700
MULTITABLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00151000
        02014000
MV  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 29 AT 00586000
        00608000  00617000
MX  --  REAL  --  DECLARED IN SEGMENT 60 AT 01113000
       *01149200* 01169400  01209000  01210000
MXNOTO  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01107100
       *01149200* 01166000  01168400  01168550
MXO  --  DEFINE  --  DECLARED IN SEGMENT 60 AT 01107200
        01168400  01168550
MY  --  REAL  --  DECLARED IN SEGMENT 6 AT 00040120
        00040240  00040250  00040270
MYSTACKADR  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00004500
       *00040250* 00040280  00863100 *00863130* 01457110  01457130  01457140  01742110  01742130 *02173799*
MYSTACKDUMPED  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00018000
        01457120 *01457150* 01742100 *01742165*
MYSTACKSIZE  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00004100
        01457110  01742140
MYTYPE  --  REAL  --  DECLARED IN SEGMENT 52 AT 00874000
       *00883000* 00885000  00887000  00889000 *00890000* 00892000
M32  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 62 AT 01231000
        01232000  01236000
N  --  INTEGER  --  DECLARED IN SEGMENT 8 AT 00046000
       *00048000* 00054000
N  --  INTEGER  --  DECLARED IN SEGMENT 19 AT 00318000
       *00326000* 00327000 *00329000*
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 19 AT 00319000
        00321000
N  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00531000
        00529000  00530000  00535000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00586100
        00586130
```

```
N  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 50 AT 00822100
        00822120  00822240
N  --  INTEGER  --  DECLARED IN SEGMENT 54 AT 00904000
       *00916000* 00917000  00919000 *00926000* 00927000  00929000 *00935000* 00936000
N  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00963000
       *00967140**00967150**00978100* 01033000 *01046000* 01053000  01057000
N  --  INTEGER  --  DECLARED IN SEGMENT 60 AT 01112000
       *01149000* 01152000  01167700  01168700  01169200  01169500  01197000  01203000  01213100  01213300  01218000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349120
        01349130  01349460
N  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        01991000  01992000  01993000  01994000  01999000 *02002000* 02003000  02004000  02005000  02010000
N  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 90 AT 02119340
        02119350
NA  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00580000  00583000
NAM  --  REAL ARRAY  --  DECLARED IN SEGMENT 41 AT 00763160
        00763435  00763510
NAME  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00085000
        00326000  00328000  00665000  00666000  00845000 *00858000* 00859000  00926000  00928000  00929000  01168600
        01168700  01349570  01349580  01448000  01449000  02007000  02008000
NAME  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 60 AT 01115300
        01118000  01119800
NAME  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349120
        01349450
NAME  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370020
        01370220
NAME  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 70 AT 01371110
        01371410
NAMESIZE  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00030000
       *00064000**00070000* 00085000  00845000  00856000
NAMS  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00086000
        00329000  00665000  00846000  00928000  00929000  01168700  01349570  01448000  02007000
NAMSSIZE  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00030000
       *00064000**00071000* 00086000  00846000
NB  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00581000  00584000
NC  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 29 AT 00576000
        00577000  00582000  00585000
NEEDCHECKAVAILNKS  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00716000
        01151900 *02120425**02141000*
NEXT  --  LABEL  --  DECLARED IN SEGMENT 36 AT 00743000  --  OCCURS AT 00759000
        00755000
NEXTITEM  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00402000
        01373000  01570000  01610000  01722000  01732000  01757400  01919000  01919300  01970000  01986100  02012000
        02117050  02117125  02117420  02119210  02119240
NEXTPAGE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00296000
        00932000  00941000  01276000  01309000  01551000  01705000  01709000  01727000  01735000  01741000  01856000
        01919400  01985380  01986000  02012000  02064000  02119170  02173795
NEXTWAIT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00148000
NFO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00136000
        00726000
NODUMP  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00012500
        00807350  01151100  01152000  01164900  01213300  01281720  01281780  01281855  01757770  01757840  02119250
        02138000  02142000  02142300  02143000  02144000  02148000  02149000  02149100  02149300  02149400  02160000
        02165000  02165100
NODUMPTOG  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00082110
```

```
      *02165120* 02173796 *02173797*
NOMCPCODE  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00007000
       01211000  01281835  02149200
NOMEMXI  --  FORMAT  --  DECLARED IN SEGMENT 83 AT 01762000
       01860000
NOMO  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00082150
      *00837095**00838010* 02119160
NOMOM  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01370570  --  OCCURS AT 01370920
       01370820  01370890
NONORMALCODE  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00006000
       01210000
NOPROCESSTOG  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164340
       01918200
NORMAL  --  BOOLEAN  --  DECLARED IN SEGMENT 70 AT 01366000
      *01382000* 01385000  01391000  01420000  01422000  01454000  01465800
NOTHINGTODO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00159000
       00884000
NOTPRINTCALL  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00559100
       00593000 *01122100**01208100**01219100**01281685**01281905*
NOT50CTADES  --  BOOLEAN STREAM PROCEDURE  --  DECLARED IN SEGMENT 6 AT 00040170
      *00040190* 00040230
NOWRAPAROUND  --  BOOLEAN  --  DECLARED IN SEGMENT 56 AT 00947100
      *01025000* 01032100 *01036000*
NSLATE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00122000
       01987000  01991000
NSNAME  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00031000
       00858000  00859000  00926000
NULLMESSAGES  --  FORMAT  --  DECLARED IN SEGMENT 83 AT 01760000
       01985000
NULLQUE  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01774140
       02117460
NULLSLATE  --  FORMAT  --  DECLARED IN SEGMENT 83 AT 01761000
       02011000
N1  --  ALPHA  --  DECLARED IN SEGMENT 50 AT 00823100
       00837050 *00837055**00849100* 00849200
N2  --  ALPHA  --  DECLARED IN SEGMENT 50 AT 00823100
      *00849100* 00849200
OCTADF  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00112500
       00579000  00580000  00581000  00586130  01116200  01120500  01121100  01121400  01121600  01121800  01370150
       01370200  01370380  01370510  01370710  01370750  01370880
OCTAL  --  REAL PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00529000
      *00535000* 00659000  00660000  00661000  00662000  00689000  00762000  00763042  00915000  00916000  00917000
       00925000  00926000  00927000  00967120  00967130  00967140  00967150  01000000  01001000  01005000  01006000
       01007000  01021000  01022000  01051000  01057000  01060000  01084000  01315000  01316000  01317000  01376000
       01378000  01387000  01388000  01394000  01395000  01418570  01418575  01418580  01438000  01439000  01440000
       01441000  01442000  01443000  01447000  01465200  01465300  01572000  01573000  01575000  01576000  01579000
       01580000  01581000  01583000  01584000  01585000  01587000  01588000  01591000  01592000  01596000  01597000
       01600000  01601000  01605000  01606000  01607000  01639000  01640000  01641000  01681000  01683000  01684000
       01684100  01757580  01757590  01757720  01757730  01757810  01757820  01757855  01757860  01757865  01860000
       01976000  01977000  02003000  02004000  02005000  02023000  02028000  02030000  02031000  02033000  02044000
       02045000  02046000  02047000  02048000  02056000  02057000  02058000  02075000  02081000  02082000  02087000
       02088000  02094000  02095000  02102000  02103000  02104000  02105000  02106000  02112000  02113000  02117520
       02117530  02117540  02142200  02142250
OCTALWORD  --  DEFINE  --  DECLARED IN SEGMENT 60 AT 01115100
       01121400  01121600  01121800
OCTDEC  --  REAL STREAM PROCEDURE  --  DECLARED IN SEGMENT 6 AT 00040150
       00040160  00040230
```

```
OLAY  --  STREAM LABEL  --  DECLARED IN SEGMENT 60 AT 01115800  --  OCCURS AT 01117300
        01116500
ON  --  LABEL  --  DECLARED IN SEGMENT 27 AT 00467000  --  OCCURS AT 00490000
        00484000  00489000
ONEMIX  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00004000
        01209000  01545100  01697000
OPERAND  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00669000
       *00676000**00680000* 00682000  00763440  00763450  00763580  00763585  00773000  00790110  00967100  00968000
        00972400  00973000  00980000  00992000  01012000  01026000  01028000  01037000  01040000  01042000  01058000
        01073000  01125000  01128000  01129000  01151300  01151400  01163000  01166200  01166400  01167000  01167100
        01168900  01295100  01298000  01301000  01418510  01418540  01419000  01423000  01465900  01468600  01613000
        01617000  01621000  01624000  01625000  01757505  01757530  01757710  01757845  01757870  01757890  01858000
        01921000  01973000  01983000  01991000  01995000  02117450  02117510  02129100  02139000  02142100  02142120
OPTION  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00153000
        01920000  01921000
OUTERBLOCK  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 01223000
       *01266000* 01349600  02149300
01  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 6 AT 00040200
        00040210
02  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 6 AT 00040200
        00040210
P  --  FILE  --  DECLARED IN SEGMENT 2 AT 00016530
        00017200  00056000  00059000  00365000  00404000  00405000  00409000  00411000  00412000  00623000  00624000
        00644500  00645000  00646000  00649000  00666500  00667000  00747000  00761000  00762000  00763042  00763044
        00763420  00763460  00763465  00763475  00763520  00763550  00773000  00775000  00781000  00785500  00785550
        00785600  00785700  00786000  00790120  00793000  00794000  00797000  00802000  00804000  00807390  00807540
        00893000  00894000  00908000  00912000  00920000  00922000  00930000  00932000  00935000  00941000  00967160
        00999100  01000000  01005000  01007000  01021000  01022000  01051000  01056000  01057000  01060000  01061000
        01084000  01095000  01151700  01169600  01276000  01277000  01280000  01280300  01309000  01310100  01311000
        01321000  01374000  01387000  01397000  01418570  01438000  01450000  01452000  01470200  01551000  01569000
        01571000  01615000  01644000  01680000  01689000  01705000  01706000  01709000  01727000  01735000  01741000
        01757620  01757760  01757830  01757885  01856000  01860000  01862000  01916000  01919400  01925000  01961000
        01964000  01967000  01979000  01985000  01985380  01986000  02009000  02011000  02012000  02020000  02061000
        02064000  02072000  02116000  02117460  02117480  02117520  02117540  02117550  02119170  02119220  02129000
        02129190  02130000  02132000  02134000  02136000  02142200  02147500  02148500  02149050  02149360  02173590
        02173600  02173650  02173792  02173793  02173795  02175000  02176000
P  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 46 AT 00771100
        00771110
PA  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00807470
        00807550  00807560  00807570  00807575
PA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        02017000  02054000  02057000
PACKETS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00110400
PAR  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00813000  --  OCCURS AT 00867000
        00826990  00827000  00837100
PARERR  --  FORMAT  --  DECLARED IN SEGMENT 51 AT 00815000
        00867000
PAROW  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00807470
        00807580  01985390
PBIT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01371110
        01371120  01371190  01371400
PCOL  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01529000
       *01611000**01648000**01652000* 01654000  01660000
PD  --  FORMAT  --  DECLARED IN SEGMENT 76 AT 01538000
PDATADESC  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00695000  --  FORWARD AT 00651100
       -00658600 *00703000* 00744000  00753000  00763032  00763445  00807330  00942100  01076000  01132000  01135000
        01169500  01542000  01546000  02129110
```

```
PEOI0  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164140
PINGO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164320
        01985250
PLUS  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 01335000
        01332000 *01345000*
PP  --  BOOLEAN  --  DECLARED IN SEGMENT 31 AT 00658100
       *00658500* 00658600  00666900
PR  --  LABEL  --  DECLARED IN SEGMENT 50 AT 00812010
PRESENCEBIT  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370530
        01370550  01370630
PREVLINK  --  REAL  --  DECLARED IN SEGMENT 56 AT 00948000
        00979000  00982000 *00987000* 00991000  00992000  01003000  01006000 *01010000* 01014000 *01017000* 01020000
        01021000  01023000  01026000  01027000  01028000  01041000 *01049000*
PRINT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00568200
        00650000  00807350  01152000  01213300  01281720  01281780  01281855  01757770  01757840  02119250  02142000
        02142300  02143000  02148000  02149000  02149100  02149300  02149400  02160000  02165000
PRINTARRAY  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00807450
        00807550  00807560  00807570  00807575  01985385  02117100  02117200
PRINTARRAYROW  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00807100
        00807440  00807550  00807560  00807570  00807575  00807580  01985385  01985390  02117100  02117200
PRINTCOMMONVALUES  --  PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00017000
        00017210  02173791  02174500
PRINTCORE  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00569000
        00807350  01152000  01213300  01281720  01281780  01281855  01757770  01757840  02119250  02142000  02142300
        02143000  02148000  02149000  02149100  02149300  02149400  02160000  02165000
PRINTMCPOPTIONS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00407000
        00413000  00909000
PRINTQUEUE  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01757300
        01757980  01985120  01985150  01985200  01985250  01985300  01985375
PRINTSELECTEDARRAYS  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00807500
        00807900  02172100
PRNTABLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00155000
        02017000
PRO  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
       *01463000*
PROC  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01757310
        01757300  01757470  01757505  01757960
PROCTIME  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00138000
PROGS  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 01349005
        01349680 *01611000**01650000**01654000**01660000* 01663000  01667000  01677000  01681000
PROWS  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 01349005
        01349670 *01611000* 01650000 *01651000* 01654000  01660000  01662000  01664000  01665000  01673000  01674000
       *02120225* 02129100  02129150  02129180  02142100  02142120 *02142200* 02142250
PRT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00130000
        00721000  00744000  00942100  00943000  01542000  01543000  01572000
PRTBASE  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00025000
        00085000  00845000  00856000  00858000  00859000  00861000  00913000  00923000  01281600
PRTCONE  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00769900
        00785400 *00893000**00894000**02120195*
PRTF  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
        01543000  01573000
PRTFILE  --  FORMAT  --  DECLARED IN SEGMENT 51 AT 00814100
        00849200
PRTFILEMESS  --  STREAM PROCEDURE  --  DECLARED IN SEGMENT 46 AT 00771100
        00771155  00785400
PRTH  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
       *01543000* 01573000
```

```
PRTITEM  --  FORMAT  --  DECLARED IN SEGMENT 55 AT 00903000
        00915000  00925000
PRTLOC  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00763020
        00763028  00763030  00763032
PRTLOC  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00807110
        00807100  00807270  00807310  00807330 *00807370*
PRTMAX  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00024000
        00325000  00332000  00334000  00664000  00763030  00807310 *00856000* 00858000  00923000  01164100  01266000
        01267000  01281600  02006000  02149100
PRTNAME  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00082200
        00785400  00849200
PRTOK  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00092100
       *00942100**01542000* 01698100
PRTOUT  --  FORMAT  --  DECLARED IN SEGMENT 37 AT 00738000
        00761000
PRTROW  --  REAL  --  DECLARED IN SEGMENT 36 AT 00734000
       *00753000* 00754000  00758000
PRTS  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00083000
       *00362000* 00365000  00720000  00721000  00722000  00723000  00724000  00725000  00726000  00727000  00728000
        00729000  00730000  00744000  00763440  00763445  00763470  00790110  00807550  00807560  00807570  00807575
        00967130  00967180  00972500  00973000  01026000  01073000  01076000  01132000  01135000  01260000  01264000
        01281620  01292000  01310000  01540000  01542000  01543000  01572000  01711000  01723000  01733000  01757434
        01757436  01757440  01757450  01757455  01857000  01858000  01918200  01919100  01919200  01920000  01921000
        01971000  01973000  01985120  01985150  01985200  01985250  01985300  01985375  01985385  01985390  01987000
        01988000  01989000  01991000  01995000  02013000  02014000  02015000  02016000  02017000  02064500  02065000
        02066000  02067000  02068000  02069000  02117100  02117150  02117160  02117200  02117430  02117440  02117450
        02129110
PRTSMAX  --  DEFINE  --  DECLARED IN SEGMENT 2 AT 00027000
        00078000  00083000  00364000
PRT0  --  REAL  --  DECLARED IN SEGMENT 10 AT 00092100
        00942100  01167700  01542000  01543000  01544000  01546000  01573000  01575000  01576000
PRT26  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00004000
        01209000  01545100  01697000
PRT27  --  INTEGER  --  DECLARED IN SEGMENT 2 AT 00004100
        01457110  01742140
PRT30  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00004200
        00622000  00645000  00807295 *00807300**00807435*
PRT31  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00004300
PRT32  --  BOOLEAN  --  DECLARED IN SEGMENT 2 AT 00004400
        00056000  00364000  00843000
PRYOR  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00135000
PS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        02017000  02019000  02051000
PT  --  FORMAT  --  DECLARED IN SEGMENT 84 AT 01777000
        02053000
PUNTER  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164400
        02129110
P1MIX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00157000
        00729000
P2MIX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00158000
        00730000
Q  --  REAL  --  DECLARED IN SEGMENT 60 AT 01104000
        01129000  01130000  01151300  01151600  01163000  01164000  01164100  01168600  01168700  01168900  01169200
        01169500  01169550
QT  --  BOOLEAN  --  DECLARED IN SEGMENT 60 AT 01104000
       *01151300**01163000* 01164000  01168550  01168900
```

```
QTIMES  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164275
        00807570
Q1  --  FORMAT  --  DECLARED IN SEGMENT 80 AT 01757340
        01757580  01757810
Q2  --  FORMAT  --  DECLARED IN SEGMENT 80 AT 01757350
        01757720
Q3  --  FORMAT  --  DECLARED IN SEGMENT 80 AT 01757352
        01757855
R   --  REAL  --  DECLARED IN SEGMENT 10 AT 00082000
        *00763595* 00763600 *00763605* 00763615  01546000  01576000  01579000  01580000  01583000  01584000  01587000
        01588000  01591000  01592000  01593000  01596000  01597000  01600000  01601000  01605000  01606000  01607000
        01609100  01612000  01692000 *02120150* 02129110  02129130 *02129140* 02129150  02139000  02142000  02142120
        02142200  02142300 *02149300* 02149400
R   --  INTEGER  --  DECLARED IN SEGMENT 29 AT 00588000
        *00638000* 00639000  00641000  00642000  00643000  00644000
R   --  INTEGER  --  DECLARED IN SEGMENT 32 AT 00674000
        *00675000* 00679000
R   --  INTEGER  --  DECLARED IN SEGMENT 33 AT 00688000
        *00689000* 00692000
R   --  REAL  --  DECLARED IN SEGMENT 60 AT 01104000
        01128000  01164300  01164400  01164500  01164600  01168900 *01169000*
R   --  REAL  --  DECLARED IN SEGMENT 63 AT 01254000
        01260000  01262000  01263000  01268000  01270000  01272000  01273000
R   --  REAL  --  DECLARED IN SEGMENT 66 AT 01287000
        01295100  01295200  01298000  01299100  01300000  01301000  01304000
R   --  REAL  --  DECLARED IN SEGMENT 70 AT 01358000
        *01418575* 01423000 *01441000* 01443000  01447000 *01465200* 01465600  01466100  01466300
R   --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370260
        01370270  01370480
R   --  REAL  --  DECLARED IN SEGMENT 77 AT 01703000
        01711000 *01718000* 01719000
R   --  REAL  --  DECLARED IN SEGMENT 79 AT 01757335
        *01757505* 01757530  01757580  01757590  01757600  01757710  01757720  01757730  01757770  01757810  01757820
        01757840  01757845  01757895 *01757960*
R   --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        01858000  01860000  01862000  01914000  01921000  01924000  01961000  01964000  01967000 *01973000* 01976000
        01977000  01978000  01983000 *01984000**02005000**02006000* 02007000  02008000
RA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
        02016000  02044000  02047000
RANGE  --  BOOLEAN  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00656000
        00653000  00654000 *00658700* 00662000
RANGE  --  FORMAT  --  DECLARED IN SEGMENT 57 AT 00956000
        01000000  01005000  01021000
RCW  --  BOOLEAN  --  DECLARED IN SEGMENT 70 AT 01366000
       *01465100**01465600* 01465700 *01466000* 01466600
RDCTABLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00152000
        02016000
READARRAY  --  PROCEDURE  --  DECLARED IN SEGMENT 2 AT 00041000
        00060000  00066000  00843000  00845000  00846000  00847000  00848000
RELOAD  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00082150
        00824100  00839000  00850100 *02119145**02119280* 02120135 *02173790*
REPEATING  --  BOOLEAN  --  DECLARED IN SEGMENT 10 AT 00082150
        00834095  02173720 *02173740*
REPLY  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164160
REST  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01371135  --  OCCURS AT 01371380
        01371270  01371300
RESTART  --  LABEL  --  DECLARED IN SEGMENT 10 AT 00080500  --  OCCURS AT 02120900
```

```
        02173800
RESULT  -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 01335000
        01332000 *01344000*
RETURN  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00161000
        00888000
RJE  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00110100
        01985350
RJEWAITQ  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164335
        01985375
RO  -- INTEGER -- DECLARED IN SEGMENT 74 AT 01529000
       *01662000* 01663000 *01672000* 01675000  01677000 *01679000* 01681000  01685000
ROW  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00118000
        00547000  00551000  00606000  00608000  00615000  00617000  00639000  00640000  00641000  00643000  00675000
        00689000  00763600  00763615  00774000  00777000  00780000  00783000  00796000  00799000  00824110  00832000
        00832100  01149000  01516000  01517000  01559000  01561000  01757600  01757875  01978000  02038000  02040000
        02129180
RP  -- INTEGER -- DECLARED IN SEGMENT 74 AT 01529000
RS  -- REAL -- DECLARED IN SEGMENT 82 AT 01758200
        02016000  02019000  02041000
RT  -- FORMAT -- DECLARED IN SEGMENT 84 AT 01776000
        02043000
RTD  -- REAL ARRAY -- DECLARED IN SEGMENT 10 AT 00419000
        00773000 *00777000* 00778000 *00783000* 00784000  00786000  00787000  00788000 *00799000**00800000* 00801000
        00803000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 2 AT 00036000
        00037000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 00560000
        00564000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 50 AT 00818000
        00820000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 62 AT 01231000
        01234000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 01325000
        01328000
S  -- FORMAT -- DECLARED IN SEGMENT 71 AT 01369000
        01438000
S  -- INTEGER -- DECLARED IN SEGMENT 74 AT 01522000
        01606000  01607000  01693000  01695000
S  -- STREAM VARIABLE -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 01750000
        01754000
S  -- REAL -- DECLARED IN SEGMENT 79 AT 01757335
       *01757845* 01757855  01757860  01757865  01757870  01757875  01757890 *01757895*
S  -- INTEGER -- DECLARED IN SEGMENT 82 AT 01758300
        01989000  01990000  01993000  01997000  02002000 *02019000* 02021000 *02071000* 02073000 *02117490* 02117500
S  -- REAL -- NAME PARAMETER -- DECLARED IN SEGMENT 82 AT 01814000
        01811000  01818000  01819000 *01820000* 01822000
SALF  -- STREAM VARIABLE -- VALUE PARAMETER -- DECLARED IN SEGMENT 70 AT 01370260
        01370270  01370440
SAVEAREA  -- STREAM LABEL -- DECLARED IN SEGMENT 60 AT 01115800  -- OCCURS AT 01117000
        01116600
SAVERESULT  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00164100
SAVERESULTS  -- DEFINE -- DECLARED IN SEGMENT 10 AT 00107000
SD  -- INTEGER -- DECLARED IN SEGMENT 74 AT 01522000
        01584000  01612000 *01613000* 01617000
SE  -- FORMAT -- DECLARED IN SEGMENT 83 AT 01759000
        02003000
SEC  -- INTEGER -- NAME PARAMETER -- DECLARED IN SEGMENT 10 AT 00452000
```

```
        00451000 *00455000**00459000*
SECONDS  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00427000
        00784000  00789000  00792000  00793000  00801000  00802000  00803000  00804000
SEG  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 10 AT 01349120
        01349130  01349220
SEG  -- INTEGER  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 01349520
        01349500  01349560  01349570  01349580  01349600  01349610  01349630  01349640  01349650  01349690  01349700
SEG  -- INTEGER  -- DECLARED IN SEGMENT 70 AT 01357000
        01444000  01447000  01448000  01449000  01465600 *01466000* 01466100  01466300  01469800
SEG  -- REAL  -- DECLARED IN SEGMENT 74 AT 01531000
       *01616000* 01617000  01623000  01639000  01657000 *01670000*
SEGH  -- FORMAT  -- DECLARED IN SEGMENT 76 AT 01533000
        01615000
SEGMENT  -- FORMAT  -- DECLARED IN SEGMENT 76 AT 01537000
        01639000  01680000
SEGS  -- INTEGER  -- DECLARED IN SEGMENT 8 AT 00046000
       *00049000**00050000* 00051000  00054000
SEGS  -- REAL  -- DECLARED IN SEGMENT 74 AT 01530000
        01613000  01616000 *01664000**01667000* 01670000
SEGZERO  -- REAL ARRAY  -- DECLARED IN SEGMENT 2 AT 00034000
        00066000  00068000  00069000  00070000  00071000  00072000  00073000  00843000
SEPTIC  -- FILE  -- DECLARED IN SEGMENT 41 AT 00763145
        00763485  00763495  00763575
SEPTICTANK  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00110300
SEQUENCE  -- PROCEDURE  -- DECLARED IN SEGMENT 10 AT 01224000
        01251000  01275000  01281670  01312000  01663000
SETUPXNAMEANDXNAMS  -- PROCEDURE  -- DECLARED IN SEGMENT 10 AT 00246000
        00863000
SGLTOG  -- BOOLEAN  -- DECLARED IN SEGMENT 10 AT 00082100
        00666500 *00719100**00730100**01277100**01279100**01281580**01281900**02012100**02118200*
SGM  -- REAL  -- DECLARED IN SEGMENT 74 AT 01530000
        01617000  01618000  01627000  01629000  01631000 *01681000* 01682000  01683000  01684000  01684100
SHAREDISK  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00108000
        00790100  02117400
SHEET  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00133000
        00724000
SINCSLASTHL  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00425000
        00803000  00804000
SIZ  -- INTEGER  -- DECLARED IN SEGMENT 74 AT 01529000
       *01620000* 01621000  01623000  01624000  01627000  01628000 *01635100* 01639000  01640000  01641000  01656000
SIZE  -- INTEGER  -- DECLARED IN SEGMENT 39 AT 00763024
       *00763034* 00763040
SIZE  -- INTEGER  -- DECLARED IN SEGMENT 47 AT 00807130
       *00807340* 00807350
SIZE  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 60 AT 01115400
        01115500  01120200  01120700
SIZEF  -- DEFINE  -- DECLARED IN SEGMENT 65 AT 01281570
        01281630  01281640  01281660  01281850
SKIPEFE  -- LABEL  -- DECLARED IN SEGMENT 70 AT 01361000  -- OCCURS AT 01418600
SKIPPING  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 10 AT 00536000
        00537000  00540000
SLATE  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00121000
        01989000
SOMOKB  -- BOOLEAN  -- DECLARED IN SEGMENT 10 AT 00498000
       *01023000**02120420* 02157000
SOMOKF  -- BOOLEAN  -- DECLARED IN SEGMENT 10 AT 00498000
       *01003000**02120415* 02155000
```

```
SP  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01325000
        01326000  01328000
SPEC  --  DEFINE  --  DECLARED IN SEGMENT 70 AT 01365100
        01381100
SPO  --  FILE  --  DECLARED IN SEGMENT 2 AT 00016520
        00866000  00867000  00868000  00893000  00894000  02119230  02119380
SPREAD  --  INTEGER PROCEDURE  --  DECLARED IN SEGMENT 10 AT 01512000  --  FORWARD AT 01351000
        01387000  01394000  01418575  01439000  01441000  01465200 *01516000* 01519000  01543000  01576000  01580000
        01584000  01588000  01592000  01597000  01601000  01606000
SQUEEZE  --  LABEL  --  DECLARED IN SEGMENT 70 AT 01362000  --  OCCURS AT 01401000
        01415000
STACKOVERFLOW  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00160000
        00886000
STARBLANK  --  ALPHA  --  DECLARED IN SEGMENT 74 AT 01523000
       *01632000**01636000* 01641000
STARCOUNT  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 00588000
       *00596000* 00623000  00624000
STARD  --  BOOLEAN  --  DECLARED IN SEGMENT 29 AT 00587000
       *00610000* 00620000 *00625000**00629000* 00635000
STARS  --  FORMAT  --  DECLARED IN SEGMENT 16 AT 00297000
        00405000  02129000  02136000
STARS  --  REAL  --  DECLARED IN SEGMENT 74 AT 01526000
       *01539000* 01566000  01567000  01640000  01682000  01684100
STARZ  --  FORMAT  --  DECLARED IN SEGMENT 16 AT 00298000
        00623000  00624000
STATION  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164240
        00807575  00807580  01985385  01985390
STATIONMESSAGEHOLDER  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164310
        01985120
STATISTICS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00109000
STAX  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00095000
       *01089000* 01197000 *01198000**01203000* 01375000  01377000  01381100  01382000  01399000  01438000 *01691000*
       *01695000**01698300**01707000* 01716000  01718000 *01721000**01731000**01742130*
STAXMAX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00094000
        00095000
STK  --  INTEGER  --  DECLARED IN SEGMENT 60 AT 01112000
       *01194000**01195000* 01197000  01198000
SUBLEVEL  --  REAL  --  DECLARED IN SEGMENT 2 AT 00028000
       *00068100* 00409000
SW  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 60 AT 01115700
       *01118900* 01121700
SYL  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349120
        01349130  01349290
SYL  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01349520
        01349500  01349510  01349570  01349610  01349640  01349700
SYSID  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00444100
        00790120
SYSNO  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164305
        00790110
T  --  REAL ARRAY  --  DECLARED IN SEGMENT 6 AT 00040130
        00040270  00040280
T  --  REAL ARRAY  --  DECLARED IN SEGMENT 12 AT 00270000
       *00272000* 00294000
T  --  INTEGER  --  DECLARED IN SEGMENT 26 AT 00454000
       *00456000* 00457000  00458000  00459000
T  --  INTEGER  --  DECLARED IN SEGMENT 31 AT 00658000
       *00658500* 00658600  00659000  00660000  00661000
```

```
T  --  INTEGER  --  DECLARED IN SEGMENT 56 AT 00963000
       00967100  00967130  00967140  00967150  00972400 *00978100* 01034000 *01047000* 01054000  01057000  01058000
       01059000
T  --  DEFINE  --  DECLARED IN SEGMENT 60 AT 01101100
       01149100  01166100  01166600  01166900  01167300  01167700  01167800  01168100  01168300  01168500  01168550
       01168900  01169400  01200000  01210000  01213000
T  --  INTEGER  --  DECLARED IN SEGMENT 62 AT 01229000
       *01238000**01239000* 01242000  01245000  01247000
T  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 10 AT 01349150
       01349230  01349270  01349300
T  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 70 AT 01370050
T  --  REAL  --  DECLARED IN SEGMENT 79 AT 01757335
       01757870  01757890  01757895
TA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
       02013000  02027000  02032000  02076000
TABLELOOKUP  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00316000
       00335000  00361000
TABLESLOC  --  INTEGER  --  DECLARED IN SEGMENT 10 AT 00497000
      *01058000* 01167700 *02120165*
TAR  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164430
       01919100
TB  --  REAL ARRAY  --  DECLARED IN SEGMENT 82 AT 01772000
      *01863000* 01915000 *01926000* 01962000  01965000  01968000  02023000  02035000  02038000  02040000  02049000
       02059000  02061000  02075000  02089000  02096000  02107000  02114000  02116000
TDFID  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00082310
       00785610  00834060  00834085  00834095  02119360  02119370  02119380  02173700
TDMFID  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00082300
       00785600  00834050  00834080
TEMP  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00433000
TF  --  FORMAT  --  DECLARED IN SEGMENT 83 AT 01764000
       01862000  01925000  01961000  01964000  01967000
TFXI  --  FORMAT  --  DECLARED IN SEGMENT 83 AT 01763000
       01914000
THISISAMOM  --  STREAM LABEL  --  DECLARED IN SEGMENT 70 AT 01370570  --  OCCURS AT 01370900
       01370830
THISROW  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00558000
       00615000  00618000  00619000  00630000
TIM  --  PROCEDURE  --  DECLARED IN SEGMENT 41 AT 00763290
       00763350  00763515
TIMEANALYZED  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00420000
       00792000  00793000
TIMELASTHL  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00424000
       00801000  00802000
TIMES  --  PROCEDURE  --  DECLARED IN SEGMENT 10 AT 00451000
       00460000  00784000  00792000  00801000  00803000
TIMETAKEN  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00422000
       00784000  00789000
TINU  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00146000
       02013000
TINUOK  --  BOOLEAN  --  DECLARED IN SEGMENT 82 AT 01758400
      *02013000* 02075000
TOG  --  BOOLEAN  --  DECLARED IN SEGMENT 47 AT 00807140
      *00807260* 00807350 *00807400*
TOGLE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00128000
       00763440  01857000  01858000  01919200
TOO  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00571000
       00569000  00570000 *00598000* 00599000  00602000  00638000  00648000
```

```
TOO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01096500
TOO  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01099000
       01097000  01098000  01143000
TOO  --  REAL  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01281520
       01281500  01281510  01281640  01281690  01281720  01281730  01281855
TOS  --  INTEGER  --  DECLARED IN SEGMENT 70 AT 01357000
      *01375000* 01376000 *01380000* 01381000  01387000 *01389000**01392000* 01394000  01396000  01400000 *01401000*
       01402000 *01403000**01408000* 01412000 *01418560* 01418570  01418575  01419000  01423000 *01424000* 01454000
       01455000  01456000  01457000  01457130  01458000  01465500
TOSORIG  --  REAL  --  DECLARED IN SEGMENT 70 AT 01359000
      *01400000* 01408000
TOTALPROCS  --  FORMAT  --  DECLARED IN SEGMENT 64 AT 01256000
       01280000
TPNAME  --  REAL ARRAY  --  DECLARED IN SEGMENT 10 AT 00082250
       00785600  00785610 *00834050**00834060**00834070* 00834080  00834085  00834095  00834200  02119360  02119370
       02119380  02173700
TRANSACTION  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164170
TRITEM  --  FORMAT  --  DECLARED IN SEGMENT 71 AT 01368000
       01387000  01394000  01418570  01465200
TRYANOTHER  --  LABEL  --  DECLARED IN SEGMENT 88 AT 02119140  --  OCCURS AT 02119150
       02119260
TS  --  REAL  --  DECLARED IN SEGMENT 82 AT 01758200
       02013000  02019000  02024000
TUSTABYMIX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164270
TYP  --  INTEGER  --  DECLARED IN SEGMENT 35 AT 00710000
       00711000  00712000  00713000
TYP  --  REAL  --  DECLARED IN SEGMENT 52 AT 00873000
       00885000  00887000  00889000  00892000
TYP  --  REAL  --  DECLARED IN SEGMENT 58 AT 01072000
       01081000  01082000
TYP  --  INTEGER  --  DECLARED IN SEGMENT 63 AT 01255000
       01260000  01268000  01269000
TYP  --  INTEGER  --  DECLARED IN SEGMENT 66 AT 01286000
       01292000
TYP  --  INTEGER  --  DECLARED IN SEGMENT 74 AT 01522000
       01540000  01556000  01612000
TYP  --  INTEGER  --  DECLARED IN SEGMENT 82 AT 01758300
TYPE  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 60 AT 01115300
       01115500  01117700  01119900
TYPMAX  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00113000
       00295110  00594000  00953000  01149100
UA  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
       02066000  02080000  02087000  02117440  02117510
UFO  --  FORMAT  --  DECLARED IN SEGMENT 85 AT 01853000
       02079000
UNIT  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00145000
       02066000
US  --  REAL  --  DECLARED IN SEGMENT 82 AT 01773000
       02066000  02071000  02077000  02117440  02117450
USERCODE  --  REAL  --  DECLARED IN SEGMENT 79 AT 01757335
       01757710  01757740  01757750
USERSTA  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164260
V  --  REAL  --  DECLARED IN SEGMENT 58 AT 01072000
       01081000  01082000  01090000
V  --  REAL  --  DECLARED IN SEGMENT 62 AT 01230000
      *01242000* 01248000
V  --  REAL  --  DECLARED IN SEGMENT 70 AT 01358000
```

```
        01418510  01418540  01419000  01420000  01421000  01423000 *01438000**01439000* 01441000
V  --  REAL  --  DECLARED IN SEGMENT 77 AT 01703000
       *01712000* 01713000
VAL  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 70 AT 01370940
        01370950  01371070
VARIFY  --  BOOLEAN PROCEDURE  --  DECLARED IN SEGMENT 82 AT 01811000
       *01817000* 01823000  01989000  02013000  02014000  02015000  02016000  02017000  02066000  02067000  02068000
        02069000  02117440
VBFD  --  REAL  --  DECLARED IN SEGMENT 10 AT 00093000
        01076000  01077000 *01079000* 01081000  01084000  01439000 *02120160*
VERIFY  --  DEFINE  --  DECLARED IN SEGMENT 82 AT 01823100
        02013000  02014000  02015000  02016000  02017000  02066000  02067000  02068000  02069000
VERSION  --  REAL  --  DECLARED IN SEGMENT 2 AT 00028000
       *00068200* 00785550
VJOBNUM  --  REAL  --  DECLARED IN SEGMENT 10 AT 00093000
        01073000  01074000  01075000  01077000  01080000 *02120155*
VLINK  --  REAL  --  DECLARED IN SEGMENT 56 AT 00948000
        00980000  00981000  00982000  00984000  00985000  00988000  00990000  00992000  00993000  00996000  00997000
        01012000  01013000  01014000  01015000  01018000  01028000  01029000  01030000  01036000  01037000  01038000
        01039000  01040000  01041000  01042000  01043000  01044000  01047000  01048000  01056000
VMSTART  --  REAL  --  DECLARED IN SEGMENT 56 AT 00947000
        00973000  00974000  00975000  00976100  00976200
VO  --  REAL  --  DECLARED IN SEGMENT 56 AT 00946000
        00968000  00969000  00970000  00971000  00972000  00972400  00976100  00976300
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 2 AT 00036000
        00037000
W  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00395000
        00398000
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00560000
        00561000  00565000
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01325000
        01326000  01330000
W  --  STREAM VARIABLE  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 01750000
        01751000  01755000
W  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 82 AT 01765000
        01768000
WAITQUE  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00147000
        00728000
WC  --  REAL  --  DECLARED IN SEGMENT 36 AT 00734000
       *00749000* 00751000  00752000
WFO  --  FORMAT  --  DECLARED IN SEGMENT 85 AT 01854000
        02093000  02111000
WHAT  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00651200
        00651100
WHAT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 10 AT 00655000
        00653000  00654000  00658500  00664000  00665000  00666000
WHAT  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00672000
        00669000 *00679000*
WHAT  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00686000
        00683000 *00692000*
WHAT  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00698000
        00695000  00701000
WHAT  --  REAL  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00708000
        00705000  00711000
WHAT  --  INTEGER  --  VALUE PARAMETER  --  DECLARED IN SEGMENT 82 AT 01813000
        01811000  01812000  01816000  01818000
WHATFILE  --  FORMAT  --  DECLARED IN SEGMENT 43 AT 00763135
```

```
        00763465
WHEN  -- REAL  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 00452000
        00451000  00455000  00456000
WHICH  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 60 AT 01115300
        01115500  01116400
WHICH  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 10 AT 01349120
        01349130  01349180  01349320
WHICH  -- INTEGER  -- DECLARED IN SEGMENT 70 AT 01357100
WHICH  -- STREAM VARIABLE  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 70 AT 01371110
        01371120  01371220
WITHIN  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 10 AT 01332000
       *01348000* 01349000  01349560  01349600  01349630  01349680  01444000
WITHINCODESEGMENT  -- BOOLEAN PROCEDURE  -- DECLARED IN SEGMENT 10 AT 01349500
       *01349560**01349600**01349630**01349680* 01349710  01465600  01469800
WOOPS  -- BOOLEAN  -- DECLARED IN SEGMENT 70 AT 01359000
       *01401000* 01403000 *01406000* 01413000
WORD  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 10 AT 00552000
        00554000
WORD  -- REAL  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 56 AT 00952000
        00950000  00951000  00953000  00954000  00955000
WORDMODE  -- STREAM LABEL  -- DECLARED IN SEGMENT 70 AT 01371135  -- OCCURS AT 01371310
        01371240
WORDSIZE  -- INTEGER  -- VALUE PARAMETER  -- DECLARED IN SEGMENT 2 AT 00043000
        00041000  00042000  00048000  00049000  00057000  00058000
X  -- STREAM VARIABLE  -- DECLARED IN SEGMENT 50 AT 00822110
        00822180  00822230
X  -- REAL  -- DECLARED IN SEGMENT 60 AT 01105000
        01151400  01151500  01151600  01166200  01166300  01166400  01166500  01167000  01167100  01167200  01169500
        01169550
X  -- REAL  -- DECLARED IN SEGMENT 70 AT 01359000
        01402000  01404000  01406000  01465400  01465500  01465900  01466000  01466300  01467700  01467900  01468200
        01468300  01468400  01468450  01468600  01468800  01468900  01469100  01469200  01469500  01469800
XA  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 29 AT 00576000
        00579000  00583000
XB  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 29 AT 00576000
        00580000  00584000
XCLOCK  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00416000
        00780000  00783000
XCLOCX  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00431000
        00783000  00784000  00800000  00801000
XIT  -- STREAM LABEL  -- DECLARED IN SEGMENT 10 AT 00563000  -- OCCURS AT 00567000
        00565000
XIT  -- LABEL  -- DECLARED IN SEGMENT 36 AT 00742000  -- OCCURS AT 00763000
        00747000
XIT  -- STREAM LABEL  -- DECLARED IN SEGMENT 46 AT 00771105  -- OCCURS AT 00771150
        00771120
XIT  -- STREAM LABEL  -- DECLARED IN SEGMENT 50 AT 00822110  -- OCCURS AT 00822260
        00822160  00822170
XIT  -- LABEL  -- DECLARED IN SEGMENT 70 AT 01362000  -- OCCURS AT 01416000
        01403000  01409000
XNAME  -- REAL ARRAY  -- DECLARED IN SEGMENT 10 AT 00089000
       *00294000* 00916000  00918000  00919000  01396000
XNAMS  -- REAL ARRAY  -- DECLARED IN SEGMENT 10 AT 00090000
        00291000  00918000  00919000  01396000
XNAMS  -- STREAM VARIABLE  -- NAME PARAMETER  -- DECLARED IN SEGMENT 12 AT 00248000
        00250000
XSNAME  -- DEFINE  -- DECLARED IN SEGMENT 10 AT 00091000
```

```
        00916000
XX  --  LABEL  --  DECLARED IN SEGMENT 10 AT 00080500  --  OCCURS AT 02173760
        02173750
X1  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00438000
X1MARKX1  --  FORMAT  --  DECLARED IN SEGMENT 22 AT 00441000
        00786000
Y  --  REAL  --  DECLARED IN SEGMENT 27 AT 00466000
        00479000  00480000  00494000
Y  --  STREAM VARIABLE  --  DECLARED IN SEGMENT 50 AT 00822110
        *00822220* 00822250
Y  --  REAL  --  DECLARED IN SEGMENT 70 AT 01359000
        *01404000* 01405000  01406000  01412000 *01418510* 01418540 *01466000* 01466100  01466400  01468200  01468300
        01468400  01468450  01469500  01469700  01470000
YEAR  --  INTEGER  --  NAME PARAMETER  --  DECLARED IN SEGMENT 10 AT 00464000
        00461000 *00494000*
YEAR  --  STREAM VARIABLE  --  NAME PARAMETER  --  DECLARED IN SEGMENT 27 AT 00469000
        00474000
YEARS  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00429000
        00778000  00788000  00791000  00793000
YECH  --  DEFINE  --  DECLARED IN SEGMENT 10 AT 00164295
Z  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 00589000
        *00594300* 00596000  00602000  00606000  00610100  00615000  00619000  00630000  00638000  00641000  00642000
        00644000  00644100  00648000
ZEROK  --  BOOLEAN  --  DECLARED IN SEGMENT 56 AT 00946000
        *00968000* 00976000  00976100  00976300  00999100  01000000  01005000
Z1  --  INTEGER  --  DECLARED IN SEGMENT 29 AT 00589000
        *00594300* 00608000  00617000
```

```
CROSS REFERENCE STATISTICS
----- --------- ----------

PHASE ONE - SORT    963 IDENTIFIERS
        0:36 ELAPSED TIME (MIN:SEC)
        0:18 PROCESSOR TIME
        0:31 I/O TIME

PHASE TWO - SORT    6233 REFERENCES
        0:46 ELAPSED TIME (MIN:SEC)
        0:44 PROCESSOR TIME
        0:08 I/O TIME

PHASE THREE - PRINT CROSS REFERENCE (    2013 LINES)
        0:44 ELAPSED TIME (MIN:SEC)
        0:32 PROCESSOR TIME
        0:22 I/O TIME


 LABEL  000000000LINE    00177311? COMPILE DUMP/ANALYZE ALGOL LIBRARY                    ALGOL  /DUMP
```