LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/DUMPANL+0000000 OBJECT /READ

SYMBOL/DUMPANL

```
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE                    *    00000001
         * FILE ID: SYMBOL/DUMPANL  TAPE ID: SYMBOL2/FILE000             *    00000002
         * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION         *    00000003
         * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED               *    00000004
         * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON             *    00000005
         * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF               *    00000006
         * BURROUGHS CORPORATION, DETROIT, MICHIGAN 46232.               *    00000007
         *                                                               *    00000008
         * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION                *    00000009
         *  AA220206  AA332366  AA386657                                 *;   00000010
                   DUMP/ANALYZE                                               00000012
                   LAST PATCHED 11/21/73                                      00000013
     SEVERAL OPTIONS ARE AVAILABLE FOR CONTROLLING THE ANALYSIS               00000015
OF A MEMORY DUMP.  THEY MAY BE CLASSIFIED INTO  FOUR CATEGORIES:              00000020
                                                                              00000025
I.  THOSE WHICH SPECIFY THE FORMAT OF THE PRINTOUT OF MEMORY:                 00000030
SPLASH DUMP,OCTAL DUMP,ALPHA/OCTAL DUMP,OR ALPHA DUMP;                        00000040
EITHER SINGLY OR DOUBLY SPACED.                                               00000050
                                                                              00000055
II.  THOSE WHICH INCLUDE OR EXCLUDE AREAS OF MEMORY NOT OTHERWISE             00000060
INCLUDED OR EXCLUDED:                                                         00000070
1.  INCLUSION OF AVAILABLE AREAS.                                             00000080
2.  EXCLUSION OF:                                                             00000110
A.  MCP CODE AREAS.                                                           00000120
B.  NORMAL STATE CODE AREAS.                                                  00000130
C.  NORMAL STATE AREAS OTHER THAN THOSE RELATING TO ONE                       00000140
SPECIFIC MIX INDEX WHERE THE MIX INDEX IS SPECIFIED BY THE USER.              00000150
D.  ENTIRE CONTENTS OF MEMORY                                                 00000155
III.  THOSE WHICH CAUSE OMISSION OF CERTAIN SECTIONS OF ANALYSIS:             00000160
1.  OMISSION OF THE PRINTOUT OF MCP PRT IDENTIFIERS, SORTED                   00000170
ALPHABETICALLY AND BY PRT LOCATION.                                           00000180
                                                                              00000205
IV.  THOSE WHICH CONTROL THE DUMPING OF STACKS:                               00000210
1.  EXCLUSION OF ALL NORMAL STATE STACKS EXCEPTING THE ONE                    00000220
BELONGING TO THE SAME MIX INDEX AS THE ONE SPECIFIED BY THE USER.             00000230
2.  INCLUSION OF A USER SELECTED "STACK" AREA WHICH THE                       00000240
ANALYZER WOULD BE UNABLE TO LOCATE WITHOUT THE USER"S ASSISTANCE.             00000250
                                                                              00000255
    THE MAJORITY OF OPTIONS MAY BE SET USING A COMMON CONTROL                 00000260
CARD; A FEW MAY BE SET ONLY VIA THE "IN" MESSAGE TO ALTER THE                 00000270
CONTENTS OF CERTAIN PRT CELLS WHICH HAVE BEEN RESERVED  FOR                   00000280
CONTROL FUNCTIONS.                                                            00000290
                                                                              00000295
    THE COMMON CONTROL CARD IS ASSUMED TO CONTAIN AN EIGHT DIGIT              00000300
NUMBER--IF LESS THAN EIGHT APPEAR, LEADING ZEROES ARE                         00000310
AUTOMATICALLY SUPPLIED. THE ANALYZER SEPARATES THE 8 DIGITS INTO              00000320
TWO GROUPS:                                                                   00000330
                                                                              00000335
A.  THE 5 LEFT-MOST DIGITS ARE EXPECTED TO BE *OCTAL* DIGITS                  00000340
WHICH REPRESENT THE TOP-OF-STACK ADDRESS OF THE USER SPECIFIED                00000350
"STACK" AREA.  IF ANY DIGIT EXCEEDS "7" OR IF ALL FIVE ARE "0",               00000360
ONLY STACKS LOCATED BY THE ANALYZER WILL BE DUMPED.  OTHERWISE,               00000370
ONE ADDITIONAL "STACK", PROVIDING IT ALREADY HAS NOT BEEN DUMPED,             00000380
WILL BE DUMPED STARTING AT THIS ADDRESS AND CONTINUING UNTIL                  00000390
200(DECIMAL) WORDS BELOW IT HAVE BEEN PRINTED( THE STACK IS                   00000400
ASSUMED TO BE A CONTROL STATE STACK FOR PURPOSES OF STACK                     00000410
ANALYSIS).  SHOULD MORE OR LESS STACK BE DESIRED, THE AMOUNT                  00000420
( IN DECIMAL) OF STACK TO ANALYZE CAN BE ENTERED INTO PRT CELL                00000430
27(OCTAL) USING THE "IN" MESSAGE.                                             00000435
IN MOST CASES, THE TOP-OF-STACK VALUE USED MAY BE THE SETTING                 00000440
```

```
OF THE "S" REGISTER TAKEN FROM THE DISPLAY PANEL AT THE TIME                00000445
OF THE HANG.  IF NO TOP-OF-STACK IS SPECIFIED THE ANALYZER WILL OBTAIN      00000450
ONE FROM THE CONTROL STATE MSCW LOCATED IN CELL 7.  THE STACK               00000451
OBTAINED IN THIS MANNER IS GENERALLY OF INTEREST, ESPECIALLY WHEN           00000452
LINKS HAVE BEEN CLOBBERED.                                                  00000453
                                                                            00000455
B.  THE RIGHT-MOST 3 DIGITS ARE INTERPRETED AS DECIMAL DIGITS               00000460
AND SERVE TO SET THE MAJORITY OF ANALYZER OPTIONS.  GENERALLY               00000470
SPEAKING, THESE OPTIONS ARE INTEGRAL POWERS OF 2, ARE INDEPENDENT           00000480
OF ONE ANOTHER AND MAY THEREFORE BE USED ADDITIVELY.  THAT IS,              00000490
THE COMBINED FUNCTIONS OF COMMON VALUES 2,16,32, AND 64                     00000500
COULD BE INVOKED BY "COMMON=114"(2+16+32+64). THE EXCEPTIONS                00000510
TO THIS RULE WILL BE ELABORATED AFTER EACH COMMON OPTION IS                 00000520
DESCRIBED IN DETAIL.                                                        00000530
                        COMMON CARD OPTIONS                                 00000540
                                                                            00000550
NOTE: IN THE DESCRIPTION OF EACH OPTION, IT IS ASSUMED THAT A               00000560
STANDARD MEMORY DUMP ANALYSIS IS OBTAINED *EXCEPT* FOR THE                  00000570
DIFFERENCES GENERATED BY THAT PARTICULAR OPTION.  A "STANDARD"              00000580
ANALYSIS IS ONE OBTAINED WITH NO COMMON VALUE(I. E. COMMON=0).              00000590
                                                                            00000600
COMMON=0.                                                                   00000610
    YIELDS A STANDARD ANALYSIS.  AN MCP/PRT FILE IS REQUIRED.               00000620
                                                                            00000625
COMMON=1.                                                                   00000630
    YIELDS A "SPLASH",UNANALYZED DUMP OF MEMORY.  NO MCP/PRT IS             00000640
REQUIRED.  A "COMMENT" ENTERED WHEN THE DUMP TAPE WAS CREATED               00000650
IS ALSO PRINTED.                                                            00000660
                                                                            00000665
COMMON=2.                                                                   00000670
    DUMPS ALL AVAILABLE AREAS.                                              00000680
                                                                            00000765
COMMON=4.                                                                   00000770
    EXCLUDES ALL NORMAL STATE OBJECT CODE.                                  00000780
THIS INCLUDES ALL PROGRAM SEGMENTS ASSOCIATED WITH A                        00000790
MIX INDEX AND ALL INTRINSIC SEGMENTS.                                       00000800
                                                                            00000815
COMMON=8.                                                                   00000820
    EXCLUDES ALL MCP OBJECT CODE.                                           00000830
                                                                            00000845
COMMON=16.                                                                  00000850
    CAUSES CERTAIN ARRAYS SELECTED BY THE USER TO BE DUMPED.                00000860
SEE THE COMMENT AT SEQUENCE #00807480 FOR DETAILS.                          00000870
                                                                            00000875
COMMON=32.                                                                  00000880
    INHIBITS THE DUMP OF THE MCP PRT IDENTIFIERS SORTED                     00000890
ALPHABETICALLY AND BY PRT LOCATION.                                         00000900
                                                                            00000905
COMMON=64.                                                                  00000910
    YIELDS AN OCTAL-ONLY DUMP OF MEMORY FORMATTED SIX WORDS/LINE.           00000920
                                                                            00000945
COMMON=128.                                                                 00000950
    YIELDS AN ALPHA/OCTAL DUMP OF MEMORY.  EACH PRINTED LINE                00000960
CONTAINS FOUR WORDS OF OCTAL FOLLOWED, ON THE SAME LINE, BY                 00000970
THEIR ALPHA EQUIVALENT.                                                     00000980
                                                                            00000985
COMMON=256.                                                                 00000990
                        LAST PATCHED 12/11/73                               00000999
    YIELDS AN ALPHA DUMP OF MEMORY FORMATTED TWELVE WORDS                   00001000
OF ALPHA PER PRINTED LINE.                                                  00001010
```

```
COMMON=384.                                                          00001015
                                                                     00001020
    COMPLETELY INHIBITS THE PRINTOUT OF THE CONTENTS OF MEMORY.      00001030
THIS DOES NOT IMPLY THAT MEMORY IS NOT ANALYZED: VERIFICATION        00001040
OF MEMORY LINKS IS MADE, THE BED AND SLATE ARE CHECKED,              00001050
STACKS ARE LOCATED,ETC.                                              00001060
                                                                     00001070
                                                                     00001072
COMMON=512                                                           00001073
    CAUSES THE CONTENTS OF THE "ARGH" ARRAY TO BE PRINTED UP TO THE  00001074
LAST VALID WORD POINTED TO BY "YECH".  THIS INFORMATION IS PRESENT    00001075
ONLY WHEN AN MCP PATCH,THE "ROTO" PATCH, HAS BEEN COMPILED INTO THE  00001076
MCP.                                                                 00001077
                                                                     00001078
            PRT CELLS WHICH EXERCISE CONTROL FUNCTIONS                00001080
                                                                     00001085
NOTE: PRT LOCATIONS SPECIFIED REFER TO OCTAL PRT LOCATIONS FOR       00001090
USE IN AN "IN" MESSAGE.                                              00001100
                                                                     00001110
PRT 25.                                                              00001120
    THIS CELL IS NORMALLY SET FROM THE RIGHT-MOST 3 DIGITS WHICH     00001130
APPEAR ON A COMMON CARD; HOWEVER, THEY MAY BE EXPLICITLY SET         00001140
AND/OR MODIFIED DURING EXECUTION OF THE ANALYZER VIA THE "IN"        00001150
MESSAGE.                                                             00001160
                                                                     00001165
PRT 26.                                                              00001170
    THIS CELL CAN BE USED TO ENABLE A DUMP-BY-MIX-INDEX.  IF SET     00001180
TO A NON-ZERO DECIMAL NUMBER EQUAL TO THE MIX INDEX OF A JOB IN      00001190
THE MIX AT THE TIME THE DUMP IS TAKEN, IT WILL CAUSE MEMORY AREAS    00001200
BELONGING TO OTHER MIX INDEXES TO BE EXCLUDED FROM THE ANALYSIS.     00001210
NOTE THAT IF THE CONTENTS OF THIS CELL DOES NOT CORRESPOND TO A      00001220
VALID MIX INDEX, *NO* NORMAL STATE STACKS WILL BE DUMPED NOR         00001230
WILL ANY NORMAL STATE AREAS ASSOCIATED WITH ANY MIX INDEX BE         00001240
PRINTED.                                                             00001250
                                                                     00001255
PRT 27.                                                              00001260
    THIS CELL MAY BE USED ONLY IN CONJUNCTION WITH THE               00001270
PREVIOUSLY DESCRIBED COMMON CARD CONVENTION OF SPECIFYING AN         00001280
OCTAL STACK ADDRESS AS THE FIRST 5 DIGITS OF THE COMMON VALUE.       00001290
IF MADE NON-ZERO, THEN THE DECIMAL VALUE ENTERED IN CELL 27          00001300
WILL DETERMINE THE AMOUNT OF STACK DUMPED BELOW THE TOP-OF-STACK     00001310
VALUE.  IF LEFT ZERO, THE DEFAULT AMOUNT OF STACK DUMPED(AND         00001320
ANALYZED) WILL BE 200(DECIMAL) WORDS.                                00001330
                                                                     00001335
PRT 30.                                                              00001340
    THIS CELL, IF SET TO 1, WILL CAUSE THE PRINTOUT OF MEMORY        00001350
TO BE DOUBLE-SPACED.  OTHERWISE, THE PRINTOUT IS SINGLE-SPACED.      00001360
                                                                     00001370
        PRECEDENCE OF OPTIONS AND SUGGESTIONS FOR COMBINING THEM     00001380
                                                                     00001390
    AS STATED EARLIER, COMMON OPTIONS MAY BE USED SEPARATELY         00001400
OR IN COMBINATION.  ALTHOUGH MOST COMBINATIONS ARE ALLOWED, THERE    00001410
ARE IMPORTANT EXCEPTIONS.  THE RULES ARE:                            00001420
1.  A COMMON VALUE OF 1 PRECLUDES THE USE OF ALL OTHERS.             00001430
2.  A COMMON VALUE OF 512 PRECLUDES THE USE OF ALL OTHERS BUT 1.     00001435
3.  A COMMON VALUE OF 384 PRECLUDES THE USE OF ALL REMAINING         00001440
COMMON VALUES EXCEPT FOR 16 AND 32.  NOTE THAT 384 IS THE            00001450
SUM OF 128+256.                                                      00001460
4.  COMMON VALUES 2,4,8,16, AND 32 ARE TOTALLY INDEPENDENT AND       00001530
ADDITIVE IN ALL POSSIBLE COMBINATIONS.                               00001540
```

```
                SOME SPECIFIC EXAMPLES OF COMMON OPTIONS                        00001550
                                                                                00001560
                                                                                00001565
    A.  "COMMON=35721432".                                                      00001570
        THIS CAUSES A "STACK" AT 35721 TO BE DUMPED ALONG WITH                  00001580
    OTHER STACKS; NONE OF MEMORY IS PRINTED(384), THE MCP PRT                   00001590
    IDENTIFIERS ARE NOT PRINTED(32), AND SELECTED ARRAYS ARE                    00001600
    DUMPED(16).  THIS COMBINATION MIGHT BE USED TO PROVIDE A                    00001610
    MINIMUM OF OUTPUT AFTER A MEMORY DUMP HAS BEEN ONCE ANALYZED BUT            00001620
    AFTERWARDS FOUND TO HAVE MISSED A STACK(IN THIS CASE,                       00001630
    LOCATED NEAR ADDRESS 35721).                                                00001640
    B.  "COMMON=12".                                                            00001650
        THIS ELIMINATES ALL OBJECT CODE FROM THE PRINTOUT OF                    00001660
    MEMORY.                                                                     00001670
    C.  "COMMON=66".                                                            00001690
        THIS ENSURES THAT  AVAILABLE AREAS ARE ALWAYS PRINTED(2)                00001700
    AND PROVIDES THAT MEMORY IS PRINTED IN OCTAL FORMAT ONLY(64).               00001710
    D.  "COMMON=8".                                                             00001730
        THIS TOGETHER WITH PRT 26 SET TO A VALID MIX INDEX REDUCES              00001740
    THE AMOUNT OF OUTPUT OBTAINED DURING A MIX-ONLY DUMP.  NO                   00001750
    MCP CODE IS PRINTED LEAVING ONLY NON-MCP CODE AREAS BESIDES                 00001760
    THOSE  RELATING TO THE MIX INDEX BEING DUMPED.  FINALLY, THE ONLY NORMAL    00001770
    STATE STACK DUMPED IS THE ONE BELONGING TO THE PARTICULAR MIX               00001790
    INDEX.                                                                      00001800
                                                                                00001900
                PROCESSING MULTIFILE DUMP TAPES                                 00003000
                                                                                00003005
        THE ANALYZER WILL AUTOMATICALLY PROCESS, IN A GIVEN RUN, ALL FILES      00003010
    ON A "DPMT" TAPE IF LABEL EQUATION IS MADE TO ANY FILE BEYOND THE           00003020
    FIRST ONE.  THE ORDER  OF ANALYSIS IS OPPOSITE THAT OF CREATION.  FOR       00003030
    EXAMPLE, IF LABEL EQUATION IS MADE TO THE FOURTH FILE, "FILE MDUMP=         00003040
    MEMORY/DUMP004", THE FOURTH FILE IS FIRST ANALYZED, THE THIRD NEXT,         00003050
    THE SECOND AFTER THAT ONE AND THE FIRST FILE LAST.  AS PROCESSING OF A      00003060
    NEW FILE IS BEGUN, A MESSAGE IS WRITTEN TO THE CONSOLE SO THAT THE RUN      00003070
    MAY BE DISCONTINUED IF NO FURTHER FILES ARE TO BE ANALYZED.                 00003080
                                                                                00003090
        A COMMON VALUE MAY BE INCLUDED IN THE COMMENT PORTION OF THE            00003100
    REMARKS ENTERED AFTER A "DPMT" COMMAND OR AFTER KEYING IN THE UNIT          00003110
    WHEN USING THE MEMORY DUMP DECK.  THE SYNTAX IS: "COMMON="<COMMON VALUE>    00003120
    , WHERE "COMMON" MAY BE ABBREVIATED AS "COM" AND MUST BEGIN NO LATER        00003130
    THAN THE 140"TH CHARACTER OF THE MESSAGE.  A COMMON VALUE ENTERED           00003140
    IN THIS MANNER OVERRIDES ANY WHICH MAY BE SUPPLIED BY LABEL EQUATION.       00003150
    ONE EXAMPLE IS: "DPMT COMMON=36; ALL JOBS APPEAR ASLEEP."                   00003160
                                                                                00003170
    BEGIN                                                                       00003800
    BOOLEAN COMMON;                                                             00003900
    INTEGER PRT26; DEFINE ONEMIX=PRT26#; % FOR DUMPING ONLY ONE JOB             00004000
    INTEGER PRT27; DEFINE MYSTACKSIZE=PRT27#; % IF ≠0, AMT BELOW MYSTACKADR     00004100
    BOOLEAN PRT30; DEFINE DOUBLESPACE=PRT30#;% DBL-SPACE DUMP OF MEMORY         00004200
    BOOLEAN PRT31;% RFE                                                         00004300
    BOOLEAN PRT32;% USED FOR DEBUGGING THE ANALYZER                             00004400
    INTEGER MYSTACKADR;% IF COMMON GTR MAXCOMMONVALUE, IT PNTS TO A STACK       00004500
    DEFINE DUMPAVAIL=COMMON.[46:1]#; % COMMON=2 DUMPS AVAILABLE AREAS           00005000
    DEFINE NONORMALCODE=COMMON.[45:1]#;% COMMON=4 DONT DUMP TYPE 1,7,13 CODE    00006000
    DEFINE NOMCPCODE=COMMON.[44:1]#;% 8=DONT DUMP MCP CODE(TYPE=1,MIX=0)        00007000
    DEFINE DUMPTABLES=COMMON.[43:1]#;% 16=DUMP SELECTED ARRAYS                  00008000
    DEFINE DONTDUMPRT=COMMON.[42:1]#;%COMMON=32 STOPS DP OF PRT                 00009000
    DEFINE DUMPOCTALONLY=COMMON.[41:1]#;% 64=UNCONDITIONAL OCTAL ONLY DUMP      00010000
    DEFINE DUMPALPHAOCTAL=COMMON.[40:1]#;% 128=DUMP MEMORY IN ALPHA/OCTAL       00011000
    DEFINE DUMPALPHAONLY=COMMON.[39:1]#;% 256=DUMP MEMORY IN ALPHA ONLY         00012000
```

```
DEFINE NODUMP=REAL(COMMON).[39:2]=3#;% 384=OMIT DUMP OF MEMORY                    00012500
DEFINE DUMPCESSPOOLONLY=COMMON.[38:1]#;% 512=DUMP ONLY THE "ARGH" ARRAY          00013000
DEFINE MAXCOMMONVALUE=1023#;%                                                    00016000
FILE IN DISK DISK SERIAL "MCP" "PRT"(2,30);                                      00016500
FILE IN MDUMP 2(2,533); % DISK=533, TAPE=513                                     00016510
FILE SPO 11(1,18);                                                               00016520
FILE P 4(3,15);                                                                  00016530
PROCEDURE PRINTCOMMONVALUES;                                                     00017000
BEGIN                                                                            00017010
    INTEGER I;                                                                   00017020
    SWITCH FORMAT COMINFO:=                                                      00017030
    ("///"THE COMMON VALUES AVAILABLE ARE:"/),                                   00017040
    ("COMMON=0 YIELDS A STANDARD MEMORY DUMP AND ANALYSIS."),                    00017050
    ("COMMON=1 YIELDS A SPLASH DUMP WITH NO ANALYSIS."),                         00017060
    ("COMMON=2 PRINTS AVAILABLE AREAS."),                                        00017070
    ("COMMON=4 OMITS NORMAL STATE CODE SEGMENTS."),                              00017080
    ("COMMON=8 OMITS MCP CODE SEGMENTS."),                                       00017100
    ("COMMON=16 CAUSES CERTAIN USER DEFINED ARRAYS TO BE PRINTED."),             00017110
    ("COMMON=32 OMITS THE PRINTING OF THE SORTED MCP PRT IDENTIFIERS."),         00017120
    ("COMMON=64 CAUSES MEMORY TO BE PRINTED ENTIRELY IN OCTAL."),                00017130
    ("COMMON=128 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA/OCTAL."),         00017140
    ("COMMON=256 CAUSES MEMORY TO BE PRINTED ENTIRELY IN ALPHA."),               00017150
    ("COMMON=384 CAUSES NONE OF THE CONTENTS OF MEMORY TO BE PRINTED."),         00017160
    ("COMMON=512 DISPLAYS THE CONTENTS OF THE ARGH ARRAY."/),                    00017170
    ("FOR ADDITIONAL INFORMATION, CONSULT THE FIRST SEVERAL ",                   00017180
    "PAGES OF THE FILE:SYMBOL/DUMPANL.");                                        00017190
    FOR I:=0 STEP 1 UNTIL 13 DO WRITE(P,COMINFO[I]);                             00017200
END PRINTCOMMONVALUES;                                                           00017210
BOOLEAN DONTPRINTLINKS,MYSTACKDUMPED;                                            00018000
FORMAT FINI(49("*"),"  END OF DUMP ANALYZE  ",49("*"));                          00023000
INTEGER PRTMAX,INTMAX,INFOMAX;                                                   00024000
DEFINE PRTBASE=129#;%FIRST PRT CELL ALLOCATED BY ESPOL                           00025000
DEFINE ACTUALPRTBASE=112#;% FIRST PRT CELL AS PER MCP DEFINE                     00026000
DEFINE PRTSMAX=100#; % UPPER LIMIT OF PRTS ARRAY                                 00027000
REAL LEVEL,SUBLEVEL,VERSION;                                                     00028000
INTEGER KLASS;% IDENTIFIER CLASS,AS DETERMINED BY ESPOL                          00029000
INTEGER NAMESIZE,NAMSSIZE;                                                       00030000
DEFINE NSNAME[NSNAME1]=NAME[NSNAME1].[8:10]#;                                    00031000
INTEGER INAMESIZE,INAMSSIZE;                                                     00032000
DEFINE ISNAME[ISNAME1]=INAME[ISNAME1].[8:10]#;                                   00033000
ARRAY SEGZERO[0:29];                                                             00034000
BOOLEAN  MCP;% FILL/PRT DETERMINED WHAT MODULES WERE INCLUDED                    00035000
STREAM PROCEDURE MOVE(S,D,W); VALUE W;                                           00036000
BEGIN SI:=S; DI:=D; DS:= W WDS ; END MOVE;                                       00037000
PROCEDURE BUSTCOMMON;                                                            00040100
BEGIN                                                                            00040110
    REAL MY,D;                                                                   00040120
    ARRAY T[0:0];                                                                00040130
    FORMAT F("%",I5,",",X1);                                                     00040140
    REAL STREAM PROCEDURE OCTDEC(C); VALUE C;                                    00040150
        BEGIN SI←LOC C; DI←LOC OCTDEC; DS←8DEC END;                              00040160
    BOOLEAN STREAM PROCEDURE NOTSOCTADES(C); VALUE C;                            00040170
        BEGIN SI←LOC C; S(IF SC GTR "7" THEN BEGIN TALLY←1; JUMP OUT             00040180
        END ELSE SI←SI+1); NOTSOCTADES←TALLY END;                                00040190
    STREAM PROCEDURE DECOCT(D,D1,D2); VALUE D;                                   00040200
        BEGIN SI←LOC D; DI←D1;DS←5OCT;DI←D2;DS←3OCT END;                         00040210
%                                                                                00040220
    IF NOTSOCTADES(D←OCTDEC(COMMON))THEN D.[1:29]←0;                             00040230
    DECOCT(D,MY,COMMON);                                                         00040240
    IF MY=0 THEN MYSTACKADR←-1  ELSE                                             00040250
```

```
        BEGIN % CONVERT FIRST 5 DIGITS OF COM AS IF THEY WERE OCTAL DIGITS  00040260
            WRITE(T[*],F,MY);                                              00040270
            READ(T[*],/,MYSTACKADR);                                       00040280
        END;                                                               00040290
    END BUSTCOMMON;                                                        00040300
    PROCEDURE READARRAY(WORDSIZE,ANAME,BASE);                              00041000
    VALUE WORDSIZE,BASE;                                                   00042000
    INTEGER WORDSIZE,BASE;                                                 00043000
    ARRAY ANAME[*];                                                        00044000
    BEGIN                                                                  00045000
        INTEGER I,SEGS,N;                                                  00046000
        ARRAY BUF[0:29];                                                   00047000
        N:=WORDSIZE MOD 30;                                                00048000
        SEGS:=(WORDSIZE+29) DIV 30;                                        00049000
        SEGS:=SEGS-1;                                                      00050000
        FOR I:=0 STEP 1 UNTIL SEGS DO                                      00051000
        BEGIN                                                              00052000
            READ(DISK,30,BUF[*]);                                          00053000
            MOVE(BUF,ANAME[BASE+30*I],IF I #SEGS OR N=0 THEN 30 ELSE N);   00054000
        END;                                                               00055000
        IF PRT32 THEN BEGIN WRITE(P[PAGE]); WRITE(P[DBL],<"BASE=",         00056000
        I5," SIZE=",I5>,BASE,WORDSIZE); FOR I:=BASE STEP 1 UNTIL           00057000
        BASE+WORDSIZE -1 DO BEGIN MOVE(ANAME[ I],BUF[0],1);                00058000
        WRITE(P,1,BUF[*]) END END;                                         00059000
    END OF READARRAY;                                                      00060000
    BUSTCOMMON;                                                            00061000
    COMMENT:TO GET OPTIONS,USE"(MIx)IN25" AND/OR COMMON CONTROL CARD;      00063000
    IF COMMON THEN NAMESIZE:=NAMSSIZE:=INAMESIZE:=INAMSSIZE:=1 ELSE        00064000
    BEGIN                                                                  00065000
    READARRAY(30,SEGZERO[*],0);                                            00066000
    CLOSE(DISK);                                                           00067000
    LEVEL:="2"; % 1 DIGIT                                                  00068000
    SUBLEVEL:="00"; % 2 DIGITS                                             00068100
    VERSION:="B"; %                                                        00068200
    COMMENT EACH TIME ANALYZER IS CHANGED INCREMENT ALPHABETICALLY;        00068300
    MCP:=BOOLEAN(SEGZERO[1]);% COMPILE-TIME OPTIONS                        00069000
    NAMESIZE:=SEGZERO[2]+1;% # OF MCP PRT CELLS(ABOVE @200)                00070000
    NAMSSIZE:=SEGZERO[3];% # OF WORDS THE MCP IDENTIFIERS OCCUPY           00071000
    INAMESIZE:=SEGZERO[4]+1;% # OF INTRINSICS                             00072000
    INAMSSIZE:=SEGZERO[5];% # OF WORDS THE INTRINSIC IDENTIFIERS OCCUPY    00073000
    END SEGZEROSTUFF;                                                      00074000
    INFOMAX:=4*PRTSMAX+3;                                                  00078000
    % % % % % % % % % % % % INNER BLOCK BEGINS % % % % % % % % % % % %    00079000
    BEGIN                                                                  00080000
        LABEL EOPROG;                                                      00080400
        LABEL RESTART,XX;                                                  00080500
        INTEGER I;                                                         00081000
        REAL R;                                                            00082000
        REAL AUXTYPE;                                                      00082050
        BOOLEAN SGLTOG;                                                    00082100
        BOOLEAN NODUMPTOG;                                                 00082110
        BOOLEAN NOMO,MULTI,REPEATING,RELOAD;                               00082150
        ARRAY PRTNAME[0:4];%                                               00082200
        ARRAY TPNAME[0:2];                                                 00082250
        DEFINE TDMFID=TPNAME[0]#,                                          00082300
               TDFID=TPNAME[1]#,                                           00082310
               FILETYPE=TPNAME[2]#;                                        00082330
        ARRAY PRTS[0:PRTSMAX];                                             00083000
        ARRAY INFO[0:INFOMAX+1];                                           00084000
        ARRAY NAME[PRTBASE:PRTBASE+NAMESIZE -1];                           00085000
```

```
        ARRAY NAMS[0:NAMSSIZE-1];                                                00086000
        ARRAY INAME[0:INAMESIZE-1];                                              00087000
        ARRAY INAMS[0:INAMSSIZE-1];                                              00088000
        ARRAY XNAME[ACTUALPRTBASE:201];                                          00089000
        ARRAY XNAMS[0:18];                                                       00090000
        DEFINE XSNAME[XSNAME1]=XNAME[XSNAME1].[8:10]#;                           00091000
        INTEGER MIXMAX;%MIXMAX OBTAINED IN "GETPRTENTRIES"                       00092000
        REAL VJOBNUM,VBED;                                                       00093000
        DEFINE STAXMAX=80#;%                                                     00094000
        ARRAY STAX[0:STAXMAX-1]; INTEGER MAXSTK,BEDSTK;                          00095000
    COMMENT:THE DEFINES BELOW MUST CORRESPOND TO THOSE IN FILL/PRT;              00096000
        DEFINE BREAKOUT      = MCP.[47:1]#,                                      00097000
               CHECKLINK     = MCP.[46:1]#,                                      00098000
               DATACOM       = MCP.[45:1]#,                                      00099000
               DCLOG         = MCP.[44:1]#,                                      00100000
               DCSPO         = MCP.[43:1]#,                                      00101000
               DEBUGGING     = MCP.[42:1]#,                                      00102000
               DFX           = MCP.[41:1]#,                                      00103000
               DISKLOG       = MCP.[40:1]#,                                      00104000
               DUMPP         = MCP.[39:1]#,                                      00105000
               INQUIRY       = MCP.[38:1]#,                                      00106000
               SAVERESULTS   = MCP.[37:1]#,                                      00107000
               SHAREDISK     = MCP.[36:1]#,                                      00108000
               STATISTICS    = MCP.[35:1]#,                                      00109000
               AUXMEMM       = MCP.[34:1]#,                                      00110000
               RJE           = MCP.[33:1]#,                                      00110100
               B6500LOAD     = MCP.[32:1]#,                                      00110200
               SEPTICTANK    = MCP.[31:1]#,                                      00110300
               PACKETS       = MCP.[30:1]#,                                      00110400
               MONITORR      = MCP.[29:1]#,                                      00110500
               MAXOPT        = 19#; % UPDATE AS OPTIONS ARE ADDED                00111000
        ARRAY MEMORY[0:63,0:511];                                                00112000
        DEFINE TYPMAX=63#;%                                                      00113000
        DEFINE M=MEMORY#, %                                                      00114000
           FF=[18:15]#,%                                                         00115000
           CF=[33:15]#,%                                                         00116000
          CTF=[18:33:15]#,%                                                      00117000
          CTC=[33:33:15]#,%                                                      00117100
              ROW=[33:6]#, %                                                     00118000
              COL=[39:9]#; %                                                     00119000
        DEFINE DEFINEDMIXMAX=9#;%MIXMAX NOW OBTAINED FM PRT MOTHER               00120000
        DEFINE SLATE         = PRTS[00]#,                                        00121000
               NSLATE        = PRTS[01]#,                                        00122000
               LSLATE        = PRTS[02]#,                                        00123000
               ESPBIT        = PRTS[03]#,                                        00124000
               AVAIL         = PRTS[04]#,                                        00125000
               MSTART        = PRTS[05]#,                                        00126000
               MEND          = PRTS[06]#,                                        00127000
               TOGLE         = PRTS[07]#,                                        00128000
               BED           = PRTS[08]#,                                        00129000
               PRT           = PRTS[12]#,                                        00130000
               JAR           = PRTS[13]#,                                        00131000
               INTRNSC       = PRTS[14]#,                                        00132000
               SHEET         = PRTS[15]#,                                        00133000
               JOBNUM        = PRTS[16]#,                                        00134000
               PRYOR         = PRTS[17]#,                                        00135000
               NFO           = PRTS[18]#,                                        00136000
               ISTACK        = PRTS[19]#,                                        00137000
               PROCTIME      = PRTS[20]#,                                        00138000
               IOTIME        = PRTS[21]#,                                        00139000
```

```
CHANNEL          = PRTS[22]#,                                    00140000
FINALQUE         = PRTS[23]#,                                    00141000
LOCATQUE         = PRTS[24]#,                                    00142000
IOQUEAVAIL       = PRTS[25]#,                                    00143000
IOQUE            = PRTS[26]#,                                    00144000
UNIT             = PRTS[27]#,                                    00145000
TINU             = PRTS[28]#,                                    00146000
WAITQUE          = PRTS[29]#,                                    00147000
NEXTWAIT         = PRTS[30]#,                                    00148000
FIRSTWAIT        = PRTS[31]#,                                    00149000
LABELTABLE       = PRTS[32]#,                                    00150000
MULTITABLE       = PRTS[33]#,                                    00151000
RDCTABLE         = PRTS[34]#,                                    00152000
OPTION           = PRTS[35]#,                                    00153000
MESSAGEHOLDER    = PRTS[36]#,                                    00154000
PRNTABLE         = PRTS[37]#,                                    00155000
INITIALIZE       = PRTS[38]#,                                    00156000
P1MIX            = PRTS[39]#,                                    00157000
P2MIX            = PRTS[40]#,                                    00158000
NOTHINGTODO      = PRTS[41]#,                                    00159000
STACKOVERFLOW    = PRTS[42]#,                                    00160000
RETURN           = PRTS[43]#,                                    00161000
DIRECTORYBUILDER = PRTS[44]#,                                    00162000
@CQPTSTACK       = PRTS[45]#,                                    00163000
SAVERESULT        = PRTS[46]#,                                   00164100
AUXDATA           = PRTS[47]#,                                   00164110
AUXCODE          = PRTS[48]#,                                    00164120
EUIO             = PRTS[49]#,                                    00164130
PEOIO            = PRTS[50]#,                                    00164140
AVTABLE          = PRTS[51]#,                                    00164150
REPLY            = PRTS[52]#,                                    00164160
TRANSACTION      = PRTS[53]#,                                    00164170
DALOC            = PRTS[54]#,                                    00164180
MEMASK           = PRTS[55]#,                                    00164190
CTABLE           = PRTS[56]#,                                    00164200
FS               = PRTS[57]#,                                    00164210
DBARRAY          = PRTS[58]#,                                    00164220
ATTACHED         = PRTS[59]#,                                    00164230
STATION          = PRTS[60]#,                                    00164240
DCQARA           = PRTS[61]#,                                    00164250
USERSTA          = PRTS[62]#,                                    00164260
TUSTABYMIX       = PRTS[63]#,                                    00164270
QTIMES           = PRTS[64]#,                                    00164275
EUQ              = PRTS[65]#,                                    00164280
CIDROW           = PRTS[66]#,                                    00164285
ARGH             = PRTS[67]#,                                    00164290
YECH             = PRTS[68]#,                                    00164295
DIRECTORYFREE    = PRTS[69]#,                                    00164300
SYSNO            = PRTS[70]#,                                    00164305
STATIONMESSAGEHOLDER                                             00164310
                 = PRTS[71]#,                                    00164311
LOOKQ            = PRTS[72]#,                                    00164315
PINGQ            = PRTS[73]#,                                    00164320
DC19Q            = PRTS[74]#,                                    00164325
ILL              = PRTS[75]#,                                    00164330
RJEWAITQ         = PRTS[76]#,                                    00164335
NOPROCESSTOG     = PRTS[77]#,                                    00164340
MIXMASK          = PRTS[78]#,                                    00164345
INFOMASK1        = PRTS[79]#,                                    00164350
INFOMASK2        = PRTS[80]#,                                    00164355
```

```
                CCMASK1         = PRTS[81]#,                                    00164360
                CCMASK2         = PRTS[82]#,                                    00164365
                CORE            = PRTS[83]#,                                    00164370
                DISKOUNT        = PRTS[84]#,                                    00164380
                EUW             = PRTS[85]#,                                    00164390
                PUNTER          = PRTS[86]#,                                    00164400
                LQUE            = PRTS[87]#,                                    00164410
                LQAVAIL         = PRTS[88]#,                                    00164420
                IAR             = PRTS[89]#,                                    00164430
                IOQUESLOTS      = PRTS[90]#,                                    00164440
                DUMMY           = DUMMY#;%                                      00164999
        PROCEDURE FILLINFO;                                                     00165000
        FILL INFO[*] WITH %                                                     00166000
        "SLATE    ",0,0,26,                 % 00                                00167000
        "NSLATE   ",0,0,22,                 % 01                                00168000
        "LSLATE   ",0,0,22,                 % 02                                00169000
        "ESPBIT   ",0,0,10,                 % 03                                00170000
        "AVAIL    ",0,0,23,                 % 04                                00171000
        "MSTART   ",0,0,23,                 % 05                                00172000
        "MEND     ",0,0,23,                 % 06                                00173000
        "TOGLE    ",0,0,22,                 % 07                                00174000
        "BED      ",0,0,26,                 % 08                                00175000
        0,0,0,0,                            % 09                                00176000
        0,0,0,0,                            % 10                                00177000
        0,0,0,0,                            % 11                                00178000
        "PRT      ",0,0,26,                 % 12                                00179000
        "JAR      ",0,0,26,                 % 13                                00180000
        "INTRNSC  ",0,0,26,                 % 14                                00181000
        "SHEET    ",0,0,26,                 % 15                                00182000
        "JOBNUM   ",0,0,22,                 % 16                                00183000
        "PRYOR    ",0,0,26,                 % 17                                00184000
        "NFO      ",0,0,26,                 % 18                                00185000
        "ISTACK   ",0,0,26,                 % 19                                00186000
        "PROCTIME",0,0,26,                  % 20                                00187000
        "IOTIME   ",0,0,26,                 % 21                                00188000
        "CHANNEL  ",0,0,26,                 % 22                                00189000
        "FINALQUE",0,0,26,                  % 23                                00190000
        "LOCATQUE",0,0,26,                  % 24                                00191000
        "IOQUEAVA","IL      ",0,22,         % 25                                00192000
        "IOQUE    ",0,0,26,                 % 26                                00193000
        "UNIT     ",0,0,26,                 % 27                                00194000
        "TINU     ",0,0,26,                 % 28                                00195000
        "WAITQUE  ",0,0,26,                 % 29                                00196000
        "NEXTWAIT",0,0,22,                  % 30                                00197000
        "FIRSTWAI","T       ",0,22,         % 31                                00198000
        "LABELTAB","LE      ",0,26,         % 32                                00199000
        "MULTITAB","LE      ",0,26,         % 33                                00200000
        "RDCTABLE",0,0,26,                  % 34                                00201000
        "OPTION   ",0,0,22,                 % 35                                00202000
        "MESSAGEH","OLDER   ",0,22,         % 36                                00203000
        "PRNTABLE",0,0,26,                  % 37                                00204000
        "INITIALI","ZE      ",0,10,         % 38                                00205000
        "P1MIX    ",0,0,22,                 % 39                                00206000
        "P2MIX    ",0,0,22,                 % 40                                00207000
        "NOTHINGT","ODO     ",0,32,         % 41                                00208000
        "STACKOVE","RFLOW   ",0,32,         % 42                                00209000
        "RETURN   ",0,0,32,                 % 43                                00210000
        "DIRECTOR","YBUILDER",0,10,         % 44                                00211000
        "DCOPTSTA","CK      ",0,22,         % 45                                00212000
        "SAVERESU","LT      ",0,26,           % 46                              00212100
```

```
"AUXDATA ",0,0,26,                          % 47                 00212200
"AUXCODE ",0,0,26,                          % 48                 00212300
"EUIO    ",0,0,26,                          % 49                 00212400
"PEUIO   ",0,0,26,                          % 50                 00212450
"AVTABLE ",0,0,26,                          % 51                 00212500
"REPLY   ",0,0,26,                          % 52                 00212550
"TRANSACT","ION     ",0,26,                 % 53                 00212600
"DALOC   ",0,0,26,                          % 54                 00212625
"MEMASK  ",0,0,26,                          % 55                 00212650
"CTABLE  ",0,0,26,                          % 56                 00212675
"FS      ",0,0,26,                          % 57                 00212700
"DBARRAY ",0,0,26,                          % 58                 00212725
"ATTACHED",0,0,26,                          % 59                 00212750
"STATION ",0,0,26,                          % 60                 00212775
"DCQARA  ",0,0,26,                          % 61                 00212800
"USERSTA ",0,0,26,                          % 62                 00212825
"TUSTABYM","IX      ",0,26,                 % 63                 00212850
"QTIMES  ",0,0,26,                          % 64                 00212875
"EUQ     ",0,0,26,                          % 65                 00212880
"CIDROW  ",0,0,26,                          % 66                 00212885
"ARGH    ",0,0,26,                          % 67                 00212890
"YECH    ",0,0,22,                          % 68                 00212895
"DIRECTOR","YFREE   ",0,22,                 % 69                 00212900
"SYSNO   ",0,0,22,                          % 70                 00212905
"STATIONM","ESSAGEHO","LDER    ",22,        % 71                 00212910
"LUOKQ   ",0,0,22,                          % 72                 00212915
"PINGQ   ",0,0,22,                          % 73                 00212920
"DC19Q   ",0,0,22,                          % 74                 00212925
"ILL     ",0,0,22,                          % 74                 00212930
"RJEWAITQ",0,0,22,                          % 76                 00212935
"NUPROCES","STOG    ",0,22,                 % 77                 00212940
"MIXMASK ",0,0,22,                          % 78                 00212945
"INFOMASK","1       ",0,22,                 % 79                 00212950
"INFOMASK","2       ",0,22,                 % 80                 00212955
"CCMASK1 ",0,0,22,                          % 81                 00212960
"CCMASK2 ",0,0,22,                          % 82                 00212965
"CORE    ",0,0,22,                          % 83                 00212970
"DISKOUNT",0,0,23,                          % 84                 00212980
"EUW     ",0,0,22,                          % 85                 00212990
"PUNTER  ",0,0,26,                          % 86                 00212992
"LQUE    ",0,0,26,                          % '87                00212994
"LQAVAIL ",0,0,22,                          % 88                 00212996
"TAR     ",0,0,26,                          % 89                 00212998
"IUQUESLO","TS      ",0,22,                 % 90                 00212999
0;%                                                                 00213000
COMMENT**************************************************************** 00214000
    FOR EACH IDENTIFIER DEFINED IN THE ARRAY "PRTS", THERE MUST BE  00215000
    A CORRESPONDING 4 WORD ENTRY IN "INFO".  THE FIRST THREE WORDS  00216000
    ARE RESERVED FOR THE IDENTIFIER TEXT.  THE FOURTH MUST CONTAIN  00217000
    THE CLASS OF THE IDENTIFIER AS DETERMINED BY THE ESPOL COMPILER. 00218000
    THIS CLASS APPEARS IN THE FIRST FOUR COLUMNS OF THE STUFF CARD  00219000
    FOR THIS IDENTIFIER.  IF THE CLASS IS NOT KNOWN, A ZERO MAY BE  00220000
    USED IN THE "INFO" ENTRY FOR IT. HOWEVER USING A CLASS OF ZERO WILL 00221000
    INCREASE THE TIME NEEDED TO LOCATE THE PRT ADDRESS OF AN        00222000
    IDENTIFIER.  NOTE THAT ALTHOUGH ARRAY "PRTS" MAY CONTAIN GAPS   00223000
    "INFO" MUST CONTAIN A DUMMY ENTRY COMPRISED OF ALL ZEROES(SEE,FOR 00224000
    INSTANCE, PRTS[9], PRTS[10], AND PRTS[111]).                    00225000
    FOR CONVENIENCE, THE CLASS NUMBERS THAT THE ESPOL COMPILER MAY  00226000
    ASSIGN AND WHICH MAY APPEAR ON A STUFF CARD ARE LISTED BELOW:   00227000
        PROCID              =10#,                                   00228000
```

```
            STRPROCID           =12#,                                   00229000
            BOOSTRPROCID        =13#,                                   00230000
            REALSTRPROCID       =14#,                                   00231000
            INTSTRPROCID        =15#,                                   00232000
            BOOPROCID           =17#,                                   00233000
            REALPROCID          =18#,                                   00234000
            INTPROCID           =19#,                                   00235000
            BOOID               =21#,                                   00236000
            REALID              =22#,                                   00237000
            INTID               =23#,                                   00238000
            BOOARRAYID          =25#,                                   00239000
            REALARRAYID         =26#,                                   00240000
            INTARRAYID          =27#,                                   00241000
            NAMEID              =30#,                                   00242000
            ININAMEID           =31#,                                   00243000
            LABELID             =32#,                                   00244000
      ************************************************************************; 00245000
      PROCEDURE SETUPXNAMEANDXNAMS;                                       00246000
      BEGIN                                                               00247000
          STREAM PROCEDURE FILLXNAMS(XNAMS);                              00248000
          BEGIN                                                           00249000
              DI:=XNAMS;                                                  00250000
              DS:= 8LIT"NT1     ";                                        00251000
              DS:= 8LIT"NT2     ";                                        00252000
              DS:= 8LIT"NT3     ";                                        00253000
              DS:= 8LIT"NT4     ";                                        00254000
              DS:= 8LIT"NT5     ";                                        00255000
              DS:= 8LIT"NT6     ";                                        00256000
              DS:= 8LIT"NT7     ";                                        00257000
              DS:= 8LIT"DATE    ";                                        00258000
              DS:= 8LIT"CLOCK   ";                                        00259000
              DS:= 8LIT"XCLOCK  ";                                        00260000
              DS:= 8LIT"READY   ";                                        00261000
              DS:= 8LIT"--------";                                        00262000
              DS:= 8LIT"KLUMP   ";                                        00263000
              DS:=16LIT"FIRSTDECK      ";                                00264000
              DS:= 8LIT"LASTDECK";                                        00265000
              DS:= 8LIT"DIRDSK  ";                                        00266000
              DS:= 8LIT"MEMORY  ";                                        00267000
              DS:= 8LIT"RRRMECH ";                                        00268000
          END OF FILLXNAMS;                                               00269000
          ARRAY T[ACTUALPRIBASE:213];                                     00270000
          INTEGER I,J;                                                    00271000
          FILL T[*] WITH % xNAMES                                         00272000
              123,1,00,                                                   00273000
              120,1,01,                                                   00274000
              119,1,02,                                                   00275000
              127,1,03,                                                   00276000
              125,1,04,                                                   00277000
              124,1,05,                                                   00278000
              126,1,06,                                                   00279000
              128,1,07,                                                   00280000
              112,1,08,                                                   00281000
              113,1,09,                                                   00282000
              114,1,10,                                                   00283000
              115,1,11,                                                   00284000
              116,1,12,                                                   00285000
              117,2,13,                                                   00286000
              118,1,15,                                                   00287000
              122,1,16,                                                   00288000
```

```
        129,1,17,                                                                00289000
        121,1,18);%                                                              00290000
    FILLXNAMS(XNAMS);                                                            00291000
    J:=ACTUALPRTBASE-1;                                                          00292000
    FOR I:=ACTUALPRTBASE STEP 3 UNTIL 211 DO                                     00293000
    XNAME[J:=J+1]:=0&T[I][8:38:10]&T[I+1][18:33:15]&T[I+2][33:33:15];            00294000
END SETTING UP XNAMEANDXNAMS;                                                    00295000
REAL LINKTYPE;                                                                   00295100
ARRAY AREATYPE[0:TYPMAX+1];                                                      00295110
PROCEDURE FILLAREATYPE;                                                          00295120
COMMENT***AREATYPE[LINKTYPE], WHERE LINKTYPE IS THE TYPE OF MEMORY               00295130
LINK TO BE PRINTED, CONTAINS A CODE WHICH DETERMINES THE FORMAT FOR              00295140
PRINTING THE AREA : 1=OCTAL, 2=ALPHA/OCTAL, 3=ALPHA ONLY.                        00295150
DUMPMEMORYANDNOTESTACKS DETERMINES LINKTYPE FROM [3:6] OF A PRIMARY              00295160
LINK.  PRINTCORE THEN OBTAINS AREATYPE[LINKTYPE] AND PRINTS THE AREA             00295170
IN THE APPROPRIATE FORMAT.  ELEMENTS 0 THRU TYPMAX CORRESPOND TO                 00295180
POSSIBLE LINK TYPES.  ELEMENT TYPMAX+1 IS USED TO CONTAIN  THE DEFAULT           00295190
FORMAT CODE USED IN PRINTING AN AREA WHICH DOES NOT BEGIN WITH A LINK            00295200
OR CONTAINS A CLOBBERED LINK.  THE CONTENTS OF AREATYPE ARE NOT                  00295210
USED IF DUMPOCTALONLY, DUMPALPHOCTAL OR DUMPALPHAONLY ARE SET OR                 00295220
IF AN AVAILABLE AREA IS BEING PRINTED WHOSE CORRESPONDING FORMAT                 00295230
CODE IS NOT NEGATIVE.  FOR EXAMPLE, A NEGATIVE CODE SUCH AS -2 CAUSES           00295240
BOTH IN-USE AND AVAILABLE AREAS TO BE FORMATTED AS ALPHA/OCTAL.  A               00295250
CODE OF +2 WOULD CAUSE ONLY IN-USE AREAS TO BE PRINTED IN ALPHA/OCTAL-           00295260
AVAILABLE AREAS, IF PRINTED, WOULD BE PRINTED IN OCTAL ONLY.****;                00295270
FILL  AREATYPE[*] WITH % 1-OCTAL, 2=ALPHA/OCTAL,3=ALPHA                          00295280
2,% UNKNOWN(0)                                                                   00295290
1,% CODE(1)                                                                      00295300
2,% DATA(2)                                                                      00295310
2,% IOBUF(3)                                                                     00295320
1,% ALGFIB(4)                                                                    00295330
2,% INQBUF(5)                                                                    00295340
2,% COBFIB(6)                                                                    00295350
1,% INTSEG(7)                                                                    00295360
1,% HEADER(8)                                                                    00295370
1,% (9)                                                                          00295380
1,% (10)                                                                         00295390
1,% (11)                                                                         00295400
2,% STACK(12)                                                                    00295410
1,% (13)                                                                         00295420
1,% (14)                                                                         00295430
1,% (15)                                                                         00295440
1,% (16)                                                                         00295450
1,% (17)                                                                         00295460
1,% (18)                                                                         00295470
1,% (19)                                                                         00295480
2,% CIDROW(20)                                                                   00295490
1,% (21)                                                                         00295500
1,% (22)                                                                         00295510
1;%***DEFAULT IF UNABLE TO DETRMINE LINK TYPE(TYPMAX+1)                          00295520
    DEFINE NEXTPAGE=WRITE(P[PAGE])#;                                             00296000
    FORMAT STARS(20("******")),                                                  00297000
           STARZ(*("*"));                                                        00298000
FORMAT MCPHDR("DCMCP.XV.",A1,".",A2," COMPILE-TIME OPTIONS:");                   00299000
SWITCH FORMAT MCPOPT:=(          "BREAKOUT")                                     00300000
                      ,(    "CHECKLINK")                                         00301000
                      ,(    "DATACOM")                                           00302000
                      ,(    "DCLOG")                                             00303000
                      ,(    "DCSPO")                                             00304000
                      ,(    "DEBUGGING")                                         00305000
```

```
                        ,(      "DFX")                                          00306000
                        ,(      "DISKLOG")                                      00307000
                        ,(      "DUMP")                                         00308000
                        ,(      "INQUIRY")                                      00309000
                        ,(      "SAVERESULTS")                                  00310000
                        ,(      "SHAREDISK")                                    00311000
                        ,(      "STATISTICS")                                   00312000
                        ,(      "AUXMEM")                                       00313000
                        ,(      "RJE")                                          00313100
                        ,(      "B6500LOAD")                                    00313200
                        ,(      "SEPTICTANK")                                   00313300
                        ,(      "PACKETS")                                      00313400
                        ,(      "MONITOR")                                      00313500
                                        ;                                       00314000
    ALPHA ARRAY ID[0:2];%USED TO CONTAIN AN IDENT WHOSE PRT LOC IS SOUGHT       00315000
    PROCEDURE TABLELOOKUP(ID,LOC);ARRAY ID[0];INTEGER LOC;                      00316000
    BEGIN                                                                       00317000
    INTEGER I,N;                                                                00318000
    BOOLEAN STREAM PROCEDURE GOTIDENT(A,N,B);VALUE N;                           00319000
    BEGIN SI:=A;DI:=B;TALLY:=0;                                                 00320000
        IF N SC = DC THEN TALLY:=1;                                             00321000
        GOTIDENT:=TALLY;                                                        00322000
    END GOTIDENT;                                                               00323000
    KLASS:=KLASS=1;                                                             00324000
    FOR I:= 129 STEP 1 UNTIL PRTMAX DO                                          00325000
    IF N:=NAME[I].[18:30] NEQ 0 THEN                                            00326000
    IF N.FF=LOC OR LOC=0 THEN                                                   00327000
    IF KLASS LEQ 0 OR KLASS=NAME[I].[3:5] THEN                                  00328000
    IF GOTIDENT(NAMS[N.CF],N:=8×N.FF,ID[0]) THEN                                00329000
    BEGIN                                                                       00330000
            LOC:=I;                                                             00331000
            I:=PRTMAX+2;%FORCE FALL THRU                                        00332000
    END;                                                                        00333000
     IF I GEQ PRTMAX +2 THEN ELSE LOC:=-1;                                      00334000
    END OF TABLELOOKUP;                                                         00335000
    DEFINE GETIDLOC(GETIDLOC1,GETIDLOC2,GETIDLOC3,GETIDLOC4)=                   00336000
            BEGIN                                                               00337000
                FILL ID[*] WITH GETIDLOC1,GETIDLOC2,GETIDLOC3;                  00338000
                TABLELOOKUP(ID,GETIDLOC4);                                      00339000
            END#;%                                                              00340000
    PROCEDURE FIXDEFINES;                                                       00341000
    BEGIN COMMENT**************************************************************00342000
        THIS PROCEDURE IS RESPONSIBLE FOR FINDING THE PRT LOCATION OF AN        00343000
        IDENTIFIER DEFINED IN "PRTS" FOR WHICH THERE IS A CORRESPONDING         00344000
        ENTRY IN "INFO".  IT DOES THIS BY CALLING "TABLELOOKUP" TO              00345000
        PERFORM A LINEAR SEARCH OF "NAMES" TO LOCATE AN IDENTIFIER IN           00346000
        "NAMS" AND COMPARE IT WITH THE ONE GIVEN IN "INFO".  IF AN              00347000
        IDENTIFIER CAN NOT BE LOCATED IN "NAMS", THE "PRTS" ELEMENT IS          00348000
        LEFT UNCHANGED--OTHERWISE, THE ADDRESS OBTAINED IS STORED               00349000
        IN "PRTS".*************************************************************; 00350000
        INTEGER I,LOC;                                                          00351000
        INTEGER STREAM PROCEDURE IDLENGTH(A);                                   00352000
            BEGIN LOCAL L; SI:=LOC L; DI:=A;                                    00353000
              3(IF BSC=DC THEN JUMP OUT ELSE BEGIN TALLY:=TALLY+1;              00354000
                SI:=LOC L END);   IDLENGTH:=TALLY;                              00355000
            END IDLENGHTH;                                                      00356000
        FILLINFO;                                                               00357000
        FOR I:=0 STEP 4 UNTIL INFOMAX DO                                        00358000
        IF (LOC:=IDLENGTH(INFO[I]))=0 THEN ELSE BEGIN                           00359000
            MOVE(INFO[I],ID[0],3); KLASS:=INFO[I+3];                            00360000
```

```
            TABLELOOKUP(ID,LOC);                                                00361000
            PRTS[I/4]:=IF LOC GTR 0 THEN LOC ELSE PRTS[I/4] END;                00362000
        KLASS:=0;                                                               00363000
        IF PRT32.[46:1] THEN FOR I:=0 STEP 1 UNTIL PRTSMAX DO                   00364000
        WRITE(P,<"PRTS[",I3,"]= ",I4>,I,PRTS[I]);                               00365000
    END FIXING DEFINES;                                                         00366000
    BOOLEAN STREAM PROCEDURE BITON(W,B);                                        00395000
        VALUE B;                                                                00396000
    BEGIN                                                                       00397000
        SI:=W;SKIP B SB;                                                        00398000
        IF SB THEN TALLY:=1;                                                    00399000
        BITON:=TALLY;                                                           00400000
    END OF BITON;                                                               00401000
    PROCEDURE NEXTITEM;                                                         00402000
        BEGIN                                                                   00403000
            WRITE(P);                                                           00404000
            WRITE(P[DBL],STARS);                                                00405000
        END;                                                                    00406000
    PROCEDURE PRINTMCPOPTIONS;                                                  00407000
    BEGIN                                                                       00408000
        WRITE(P[DBL],MCPHDR,LEVEL,SUBLEVEL);                                   00409000
        FOR I:=13,0,15,1,2,3,4,5,6,7,8,9,18,17,14,10,16,11,12 DO                00410000
        IF BITON(MCP,47=I) THEN WRITE(P,MCPOPT[I]);                             00411000
        WRITE(P[DBL]);                                                          00412000
    END OF PRINTMCPOPTIONS;                                                     00413000
    DEFINE   DATE       =(119)#,                                                00414000
             CLOCK      =(120)#,                                                00415000
             XCLOCK     =(121)#;                                                00416000
        BOOLEAN ARRAY MODON[0:7];                                               00417000
        INTEGER ARRAY ITD[0:9];                                                 00418000
        ARRAY RTD[0:5];                                                         00419000
        DEFINE TIMEANALYZED             = ITD[0]#,                              00420000
               DATEANALYZED             = ITD[1]#,                              00421000
               TIMETAKEN                = ITD[2]#,                              00422000
               DATETAKEN                = ITD[3]#,                              00423000
               TIMELASTHL               = ITD[4]#,                              00424000
               SINCSLASTHL              = ITD[5]#,                              00425000
               MINUTES                  = ITD[6]#,                              00426000
               SECONDS                  = ITD[7]#,                              00427000
               DAYS                     = ITD[8]#,                              00428000
               YEARS                    = ITD[9]#;                              00429000
        DEFINE DATX                     = RTD[0]#,                              00430000
               XCLOCX                   = RTD[1]#,                              00431000
               CLOCX                    = RTD[2]#,                              00432000
               TEMP                     = RTD[3]#,                              00433000
               KINDS                    = RTD[4]#,                              00434000
               MCPVERSION               = RTD[5]#;                              00435000
    FORMAT OUT FMXX("DATE ANALYZED ",I2,"/",I2,"/",I2/                          00436000
    "TIME ANALYZED ",I2,":",I2,":",I2),                                         00437000
    X1("MCP VERSION NUMBER ",I2/                                                00438000
    "DATE TAKEN    ",I2,"/",I2,"/",I2/                                          00439000
    "TIME TAKEN    ",I2,":",I2,":",I2),                                         00440000
    X1MARKXI("MCP VERSION NUMBER ",A1,A*/                                       00441000
             "DATE TAKEN    ",I2,"/",I2,"/",I2 /                                00442000
             "TIME TAKEN    ",I2,":",I2,":",I2),                                00443000
    FMX2("TIME OF THE LAST HALT-LOAD ",I2,":",I2,":",I2),                       00444000
    SYSID("DUMP TAKEN ON SYSTEM ",A1),                                          00444100
    FTAPDSK(A4," FILE CONTAINING DUMP IS ",A1,A6,"/",A1,A6),                    00444200
    FCOMVAL("COMMON VALUE USED IS ",I5),                                        00444300
     FANAL("ANALYZER VERSION=",A1),                                             00444400
```

```
      FMX3("TIME SINCE LAST HALT-LOAD  ",I2,":",I2,":",I2);                    00445000
          FORMAT BADBED("********* BAD BED ENTRY *********");                   00446000
          FORMAT BADDATE ("BAD DATE TAKEN ....."),                             00447000
                 BADXCLOCK ("BAD TIME TAKEN ....."),                           00448000
                 BADCLOCK("BAD TIME OF H/L");                                  00449000
             FORMAT BADCELL3("WORD 3 HAS THE FLAG BIT ON.....");               00450000
          PROCEDURE TIMES (WHEN,HRS,MIN,SEC);                                  00451000
          REAL WHEN;  INTEGER SEC,MIN,HRS;                                     00452000
          BEGIN                                                                00453000
          INTEGER T;                                                           00454000
      IF WHEN LSS 0 OR WHEN GTR 5184000 THEN HRS:=MIN:=SEC:=100 %OVERFLOW      00455000
                 ELSE BEGIN T:=WHEN;                                           00456000
                 HRS:=T DIV 216000;                                            00457000
                 MIN:=T DIV 3600 MOD 60;                                       00458000
                 SEC:=T DIV 60 MOD 60;        END;                             00459000
          END OF TIMES PROCEDURE;                                              00460000
          PROCEDURE DATES(ADATE,MONTH,DAY,YEAR);                               00461000
          VALUE ADATE;                                                         00462000
          ALPHA ADATE;                                                         00463000
          INTEGER MONTH,DAY,YEAR ;                                             00464000
             BEGIN                                                             00465000
               REAL Y,D,M;                                                     00466000
               LABEL ON;                                                       00467000
               ARRAY DAYTABLE [0:11];                                          00468000
               STREAM PROCEDURE CONV (YEAR,DAY,DAT );                          00469000
                VALUE DAT ;                                                    00470000
                BEGIN                                                          00471000
          SI:= LOC DAT;                                                        00472000
          SI := SI +3;                                                         00473000
          DI := YEAR; DS := 2 OCT;                                             00474000
          DI := DAY; DS := 3 OCT;                                              00475000
                END;                                                           00476000
               FILL DAYTABLE [*] WITH                                          00477000
                0,31,59,90,120,151,181,212,243,273,304,334;                    00478000
               CONV (Y,D,ADATE);                                               00479000
               IF ((Y MOD 4)=0) AND (Y#0) THEN                                 00480000
                BEGIN                                                          00481000
                  IF D =60 THEN                                                00482000
                    BEGIN                                                      00483000
          M := 1; GO TO ON;                                                    00484000
                    END;                                                       00485000
                  IF D > 60 THEN D:=D-1;                                       00486000
                END;                                                           00487000
          FOR M := 0 STEP 1 UNTIL 11 DO                                        00488000
                IF DAYTABLE [M] GEQ D THEN GO TO ON;                           00489000
             ON:                                                               00490000
          MONTH := M;                                                          00491000
           IF M=0 THEN D:=0 ELSE %GO AHEAD                                     00492000
          DAY := D - DAYTABLE[M-1];                                            00493000
          YEAR :=Y;                                                            00494000
             END OF PROCEDURE DATE;  %RCC                                      00495000
      INTEGER MAXMOD,MAXCOR;                                                   00496000
      INTEGER TABLESLOC;                                                       00497000
      BOOLEAN LNKSOK,AVALNKOK,SOMOKF,SOMOKB;                                   00498000
      % UTILITY PROCEDURES                                                     00528000
      REAL PROCEDURE OCTAL(N);%                                                00529000
          VALUE N; %                                                           00530000
          INTEGER N;%                                                          00531000
          % N.[1:23]=0 SO THAT IF N CONTAINS AT MOST A HALF-WORD THEN          00532000
          % OCTAL IF PRINTED USING O FORMAT, OR A FORMAT FOR FEWER OCTADES    00533000
```

```
        % WILL BE THE OCTAL REPRESENTATION OF N                                  00534000
      OCTAL:=N.[45:3]&(IF N>7 THEN OCTAL(N.[24:21]) ELSE 0)[3:9:39];             00535000
      REAL STREAM PROCEDURE CHRS(AT,SKIPPING,MANY);                             00536000
1         VALUE SKIPPING,MANY; %                                                00537000
2         % RETURNING THE 7 OR LESS CHRS REQUIRED                               00538000
3     BEGIN                                                                     00539000
4         SI:=AT; SI:=SI+SKIPPING;                                              00540000
5         DI:=LOC CHRS; DS:=6 LIT"0"; DI:=DI-MANY;                              00541000
6         DS:=MANY CHR;                                                         00542000
7     END CHRS;                                                                 00543000
8     INTEGER PROCEDURE HIHALF(LOC);                                            00544000
9         VALUE LOC;                                                            00545000
10        INTEGER LOC; %                                                        00546000
11    HIHALF:=CHRS(M[LOC.ROW,LOC.COL],0,4); %                                   00547000
12    INTEGER PROCEDURE LOHALF(LOC); %                                          00548000
13        VALUE LOC;                                                            00549000
14        INTEGER LOC; %                                                        00550000
15    LOHALF:=CHRS(M[LOC.ROW,LOC.COL],4,4); %                                   00551000
16    BOOLEAN STREAM PROCEDURE FLGBIT(WORD);                                    00552000
17    BEGIN                                                                     00553000
18        SI:=WORD; %                                                           00554000
19        IF SB THEN TALLY:=1; %                                                00555000
20        FLGBIT:=TALLY; %                                                      00556000
21    END FLGBIT; %                                                             00557000
22  ARRAY THISROW,LASTROW[0:11]; %                                              00558000
23  SAVE ARRAY ALINE[0:18];                                                     00559000
24  ARRAY BLINE[0:18];                                                          00559050
25  BOOLEAN NOTPRINTCALL,AVALNK;                                                00559100
26  BOOLEAN STREAM PROCEDURE COMPAREWORDS(S,D,W);                               00560000
27      VALUE W;                                                                00561000
28  BEGIN                                                                       00562000
29      LABEL XIT;                                                              00563000
30      SI:=S;DI:=D;                                                            00564000
31      W(IF 8 SC NEQ DC THEN JUMP OUT TO XIT);                                 00565000
32      TALLY:=1;                                                               00566000
33      XIT:COMPAREWORDS:=TALLY;                                                00567000
34  END COMPARING WORDS;                                                        00568000
35  %                                                                           00568100
36  DEFINE PRINT(PRINT1,PRINT2)=IF NODUMP THEN ELSE                             00568200
37          PRINTCORE(PRINT1,PRINT2)#;                                          00568300
38  %                                                                           00568400
39      PROCEDURE PRINTCORE(FROM,TOO);                                          00569000
40          VALUE FROM,TOO; %                                                   00570000
41          INTEGER FROM,TOO; %                                                 00571000
42      BEGIN %                                                                 00572000
43          DEFINE FORI = FOR I := 0 STEP 1 UNTIL 21 DO#; %                     00573000
44      DEFINE OCTADE = (DS:=3 RESET;3(IF SB THEN DS:=SET ELSE                  00574000
45                      DS:=RESET;SKIP SB))# ;                                  00575000
46  STREAM PROCEDURE BUILD(FROM,LINE,XA,NA,XB,NB,NC,AL,AO); %                   00576000
47  VALUE FROM,NA,NB,NC,AL,AO; %                                                00577000
48          BEGIN  DI:=LINE;SI:=LOC FROM;SI:=SI+5;SKIP 3SB;                     00578000
49          5 OCTADE; DS := LIT " "; AO( SI := XA; %                            00579000
50                  NA(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));SI:=XB;               00580000
51                  NB(DS:=LIT" ";2(DS:=LIT" ";8 OCTADE));                      00581000
52          NC(DS:=19LIT" "); DS:=LIT" "); %                                    00582000
53        AL(SI:=XA;NA(DS:=LIT" ";DS:=8 CHR); %                                 00583000
54          SI:=XB;NB(DS:=LIT" ";DS:=8 CHR); %                                  00584000
55          NC(DS:=9LIT" "); DS:=6 LIT " "); END BUILD; %                       00585000
56  STREAM PROCEDURE MW(A,B); BEGIN SI:=A;DI:=B;DS:= WDS END MV;                00586000
57  STREAM PROCEDURE EXPAND(ALINE,BLINE,N); VALUE N;                            00586100
```

```
 BEGIN DI:=BLINE; 18(DS:="LIT" "); DI:=BLINE; DI:=DI+7;                00586110
       SI:=ALINE; SI:=SI+7;                                            00586120
       N(2(8 OCTADE; DI:=DI+1); SI:=SI+1);                             00586130
 END EXPAND;                                                           00586140
BOOLEAN MATCH,FIRST,LAST,STARD;                                        00587000
INTEGER I,R,STARCOUNT;                                                 00588000
 BOOLEAN AL,AO;      INTEGER Z,Z1; %                                   00589000
LABEL EDITLINE;                                                        00589100
 IF DUMPALPHAONLY THEN AL:=TRUE ELSE                                   00590000
 IF DUMPALPHAOCTAL THEN AO:=TRUE ELSE                                  00591000
 IF NOT DUMPOCTALONLY THEN                                             00592000
 IF(I:=AREATYPE[IF NOTPRINTCALL THEN LINKTYPE ELSE                     00593000
       TYPMAX+1]) GTR 0 AND AVALNK THEN ELSE                           00594000
 IF I:=ABS(I)=3 THEN AL:=TRUE ELSE                                     00594100
 IF I=2 THEN AO:=TRUE;                                                 00594200
Z1:=(Z:=IF AL THEN 12 ELSE IF AL:=AO THEN 4 ELSE 6)=1;                 00594300
 IF NOT (AL OR AO) THEN AO:=TRUE;                                      00594400
AO:=AO AND TRUE; AL:=AL AND TRUE; %                                    00595000
STARCOUNT:=IF Z=12 THEN 114 ELSE IF Z=6 THEN 120 ELSE 119;             00596000
FIRST:=TRUE;                                                           00597000
TOO:=TOO.[32:16]; FROM:=FROM.[32:16];                                  00598000
IF TOO>FROM THEN %                                                     00599000
DO %                                                                   00600000
BEGIN %                                                                00601000
 IF NOT LAST := (TOO - FROM LEQ Z) THEN % NOT LAST LINE                00602000
    BEGIN                                                              00603000
         IF FIRST THEN                                                 00604000
         BEGIN                                                         00605000
IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],LASTROW,Z)     00606000
ELSE %                                                                 00607000
         FORI MV(M[(FROM+I).ROW,(FROM+1).COL],                         00608000
           LASTROW[I]);                                                00609000
         STARC:=FALSE;                                                 00610000
      IF Z=12 THEN GO EDITLINE;                                        00610100
         FIRST:=FALSE;                                                 00611000
         END                                                           00612000
         ELSE                                                          00613000
         BEGIN                                                         00614000
IF FROM.ROW=(FROM+Z).ROW THEN MOVE(M[FROM.ROW,FROM.COL],THISROW,Z)     00615000
ELSE %                                                                 00616000
         FORI MV(M[(FROM+I).ROW,(FROM+I).COL],                         00617000
                THISROW[I]);                                           00618000
   IF (MATCH:=COMPAREWORDS(THISROW[0],LASTROW[0],Z)) %                 00619000
                AND NOT STARD THEN                                     00620000
         BEGIN                                                         00621000
              IF DOUBLESPACE THEN                                      00622000
              WRITE(P[DBL],STARZ,STARCOUNT) ELSE                       00623000
              WRITE(P,STARZ,STARCOUNT);                                00624000
             STARD:=TRUE;                                              00625000
         END;                                                          00626000
         IF NOT MATCH THEN                                             00627000
         BEGIN                                                         00628000
              STARD:=FALSE;                                            00629000
           MOVE(THISROW,LASTROW,Z); %                                  00630000
         END;                                                          00631000
         END;                                                          00632000
    END;% IF NOT LAST                                                  00633000
    IF LAST    OR                                                      00634000
         NOT STARD OR                                                  00635000
         NOT MATCH THEN                                                00636000
```

```
                BEGIN                                                                   00637000
EDITLINE:                                                                               00637100
          R := MIN(Z,TOO-FROM); %                                                       00638000
          IF(FROM+R).ROW NEQ FROM.ROW THEN % CROSS ROW ROUND                            00639000
          BUILD(FROM,ALINE[*],M[FROM.ROW,FROM.COL],512-FROM.COL,                        00640000
               M[(FROM+R).ROW,0],(FROM+R).COL,IF R LSS Z THEN %                         00641000
         Z-R ELSE 0,AL,AO) ELSE %  STILL IN SAME ROW OF M ARRAY                        00642000
          BUILD(FROM ,ALINE[*],M[FROM.ROW,FROM.COL],R,                                  00643000
               M[0,0],C,IF R LSS Z THEN Z-R ELSE 0,AL,AO); %                            00644000
          IF Z=12 THEN IF FIRST THEN                                                    00644100
          BEGIN                                                                         00644200
               FIRST:=FALSE;                                                            00644300
               IF DONTPRINTLINKS THEN ELSE                                              00644310
               BEGIN                                                                    00644320
               EXPAND(ALINE,BLINE,2+REAL(AVALNK));                                      00644400
               WRITE(P[DBL],18,BLINE[*]);                                               00644500
               END;                                                                     00644510
          END;                                                                          00644600
          IF DOUBLESPACE THEN WRITE(P[DBL],18,ALINE[*]) ELSE                            00645000
          WRITE(P,18,ALINE[*]);                                                         00646000
             END; %                                                                     00647000
        END UNTIL (FROM := FROM + Z) GEQ TOO; %                                         00648000
        WRITE(P); %                                                                     00649000
    END PRINT; %                                                                        00650000
FORMAT ITEM(A5," = ",2(O,X1),A5,X89);                                                   00651000
    BOOLEAN PROCEDURE PDATADESC(AT,WHAT); VALUE AT;                                     00651100
                                    INTEGER AT; REAL WHAT;FORWARD;                      00651200
    ARRAY LINE[0:14];                                                                   00652000
    PROCEDURE DISPLAYIT(WHAT,RANGE,ADR);                                                00653000
        VALUE WHAT,RANGE,ADR;                                                           00654000
        INTEGER WHAT,ADR;                                                               00655000
        BOOLEAN RANGE;                                                                  00656000
    BEGIN                                                                               00657000
        INTEGER H,L,T;                                                                  00658000
        BOOLEAN PP;                                                                     00658100
        T:=IF PP:=(ADR=0) THEN WHAT ELSE ADR;                                           00658500
        IF NOT PP THEN IF PDATADESC(T,H) AND H.[8:10] NEQ 0 THEN ELSE                   00658600
        RANGE:=FALSE;                                                                   00658700
        WRITE(LINE[*],ITEM,OCTAL(T),                                                    00659000
                OCTAL(H:=HIHALF(T)),                                                    00660000
                OCTAL(L:=LOHALF(T)),                                                    00661000
                IF RANGE THEN OCTAL(H.[32:10]+L.CF-1)                                   00662000
                ELSE "      ");                                                         00663000
        IF 129≤WHAT AND WHAT≤PRTMAX THEN                                                00664000
        MOVE(NAMS[NAME[WHAT].CF],LINE[4],                                               00665000
                 NAME[WHAT].FF);                                                        00666000
    IF SGLTOG THEN WRITE(P,17,LINE[*]) ELSE                                             00666500
        IF PP THEN                                                                      00666900
    WRITE(P[DBL],17,LINE[*]);                                                           00667000
    END DISPLAYIT;                                                                      00668000
    DEFINE DISPLAY(DISPLAY1,DISPLAY2)=                                                  00668100
         DISPLAYIT(DISPLAY1,DISPLAY2,0)#;                                               00668200
    BOOLEAN PROCEDURE OPERAND(AT,WHAT); %                                               00669000
        VALUE AT; %                                                                     00670000
        INTEGER AT; %                                                                   00671000
        REAL WHAT; %                                                                    00672000
    BEGIN %                                                                             00673000
        INTEGER R,C; %                                                                  00674000
        IF FLGBIT(M[R:=AT.ROW,C:=AT.COL]) THEN %                                        00675000
        OPERAND:=FALSE %                                                                00676000
```

```
            ELSE %                                                                  00677000
            BEGIN %                                                                 00678000
                WHAT:=M[R,C]; %                                                     00679000
                OPERAND:=TRUE; %                                                    00680000
            END; %                                                                  00681000
        END OPERAND; %                                                              00682000
        BOOLEAN PROCEDURE DESCRIPTOR(AT,WHAT,KIND);                                 00683000
            VALUE AT;                                                               00684000
            INTEGER AT;                                                             00685000
            REAL WHAT,KIND;                                                         00686000
        BEGIN                                                                       00687000
            INTEGER R,C;                                                            00688000
            IF (KIND:=OCTAL(CHRS(M[R:=AT.ROW,C:=AT.COL],0,1)))≥"40" THEN           00689000
            BEGIN                                                                   00690000
                DESCRIPTOR:=TRUE;                                                   00691000
                WHAT:=CHRS(M[R,C],1,7);                                             00692000
            END;                                                                    00693000
        END DESCRIPTOR;                                                             00694000
        BOOLEAN PROCEDURE PDATADESC(AT,WHAT);                                       00695000
            VALUE AT;                                                               00696000
            INTEGER AT;                                                             00697000
            REAL WHAT;                                                              00698000
        BEGIN                                                                       00699000
            INTEGER KIND;                                                           00700000
            IF DESCRIPTOR(AT,WHAT,KIND) AND                                         00701000
               KIND="50"      THEN                                                 00702000
            PDATADESC:=TRUE;                                                        00703000
        END;                                                                        00704000
    BOOLEAN PROCEDURE CONTROLDESC(AT,WHAT);                                         00705000
        VALUE AT;                                                                   00706000
        INTEGER AT;                                                                 00707000
        REAL WHAT;                                                                  00708000
    BEGIN                                                                           00709000
        INTEGER TYP;                                                                00710000
        CONTROLDESC:=DESCRIPTOR(AT,WHAT,TYP) AND                                    00711000
                     TYP.[40:1]=1  AND                                              00712000
                     TYP.[45:1]=0;                                                  00713000
    END CONTROLDESC;                                                                00714000
        INTEGER MINLNK,MINBAC,MAXBAD,MAXLNK; %                                     00715000
        BOOLEAN NEEDCHECKAVAILNKS;%FALSE IF DPMT DUMP                               00716000
         ARRAY COMMT[0:19]; BOOLEAN COMNT;                                          00717000
             PROCEDURE MCPENTRIES;                                                  00718000
               BEGIN                                                                00719000
               SGLTOG:=TRUE;                                                        00719100
               DISPLAY(BED,FALSE);                                                  00720000
               DISPLAY(PRT,FALSE);                                                  00721000
               DISPLAY(JAR,FALSE);                                                  00722000
               DISPLAY(INTRNSC,FALSE);                                              00723000
               DISPLAY(SHEET,FALSE);                                                00724000
               DISPLAY(JOBNUM,FALSE);                                               00725000
               DISPLAY(NFO,FALSE);                                                  00726000
               DISPLAY(ISTACK,FALSE);                                               00727000
               DISPLAY(WAITQUE,FALSE);                                              00728000
               DISPLAY(P1MIX,FALSE);                                                00729000
               DISPLAY(P2MIX,FALSE);                                                00730000
               SGLTOG:=FALSE;                                                       00730100
               END;                                                                 00731000
          PROCEDURE GETPRTENTRIES;                                                  00732000
           BEGIN                                                                    00733000
              REAL ADR,WC,PRTROW,L;                                                 00734000
```

```
          REAL ADRR;                                                          00735000
          INTEGER I,K;                                                        00736000
          ARRAY MIX[0:40];                                                    00737000
          FORMAT PRTOUT(X10,"PRT LOCATIONS:"/                                 00738000
                X20,"MIX",X20,"PRT"/),                                        00739000
            F1( X20, A2,X20, A5);                                             00740000
          FORMAT BADPRT("********BAD PRT DESCRIPTOR*******");                 00741000
          LABEL XIT;                                                          00742000
          LABEL NEXT;                                                         00743000
IF NOT(PDATADESC(PRT,L) AND L.CF NEQ 0 AND L.[8:10] GTR 0                     00744000
       AND L.[8:10] LSS 41) THEN BEGIN                                        00745000
       MIXMAX:=DEFINEDMIXMAX;%                              FOR LINK CK       00746000
          WRITE (P,BADPRT); GO XIT END;                                       00747000
      ADR:= L.[33:15];                                                        00748000
      WC:= L.[8:10];                                                          00749000
      K:=-1;                                                                  00750000
      MIXMAX:=WC-1;                                                           00751000
     FOR I:= 1 STEP 1 UNTIL (WC-1) DO BEGIN                                   00752000
      IF PDATADESC(ADR+I,PRTROW) THEN ELSE PRTROW:=0;                         00753000
         IF PRTROW =0 THEN                                                    00754000
            GO TO NEXT;                                                       00755000
          K:=K+1;                                                             00756000
          MIX[K]:= I; % SAVE MIX NUMBER                                       00757000
          MIX[K:=K+1]:= PRTROW;                                               00758000
          NEXT:                                                               00759000
          END; % OF FINDING PRTS & VALID MIXES;                               00760000
       WRITE(P,PRTOUT);                                                       00761000
       WRITE(P,F1,FOR I:=0 STEP 1 UNTIL K DO OCTAL(MIX[I],[33:15]));          00762000
          XIT:                                                                00763000
        END OF GETPRTENTRIES;                                                 00763010
    PROCEDURE DUMPARRAY(PRTLOC); VALUE PRTLOC; INTEGER PRTLOC;                00763020
    BEGIN                                                                     00763022
        INTEGER LOC,SIZE,I;                                                   00763024
        FORMAT F(X2,I4,X2,2(O,X1));                                           00763026
        DISPLAY(PRTLOC,TRUE);                                                 00763028
        IF PRTLOC GEQ 129 AND PRTLOC LEQ PRTMAX THEN                          00763030
        IF PDATADESC(PRTLOC,LOC) THEN                                         00763032
        IF SIZE:=LOC.[8:10] NEQ 0 THEN                                        00763034
        IF LOC:=LOC.CF GTR 0 THEN                                             00763036
        BEGIN                                                                 00763038
            FOR I:=0 STEP 1 UNTIL SIZE-1 DO                                   00763040
            WRITE(P,F,I,OCTAL(HIHALF(LOC+I)),OCTAL(LOHALF(LOC+I)));           00763042
            WRITE(P[DBL]);                                                    00763044
        END;                                                                  00763046
    END DUMPARRAY;                                                            00763048
    PROCEDURE DUMPCESSPOOL;                                                   00763100
    BEGIN                                                                     00763105
        FORMAT HDR("D25-ABN=ABNORMAL"/"D24-RR =READ READY"/                   00763110
                   "D23-BFL=BUFFER FINAL LOCATION"/                           00763115
                   "D21-WR =WRITE READY"/"D20-BSY=BUSY"/                      00763120
                   "D18-NTR=DTCU NOT READY"/                                  00763125
                   "NOTE:BSY AND NTR=ADAPTER/DTTU NOT READY"///);             00763130
        FORMAT WHATFILE("CORE PORTION OF SEPTIC /",A1,A6,///);                00763135
        REAL I,B,FROM; BOOLEAN LO;                                            00763140
        FILE SEPTIC DISK SERIAL (2,60,MYUSE=INPUT);                           00763145
        DEFINE BUMPI = I:=I+1#;                                               00763150
        DEFINE INTSP=INFO#;                                                   00763155
        ARRAY CC,NAM[0:3];                                                    00763160
        LABEL LOOP,EOJ;                                                       00763165
        STREAM PROCEDURE INIT(A);                                             00763170
```

```
BEGIN DI:=A; DS:=32 LIT                                                  00763175
    "PASS INTACT. INTWRITE   READ    ";                                  00763180
END INIT;                                                                00763185
    STREAM PROCEDURE FLL(A,B,C,D,E); VALUE B,C,D;                        00763190
    BEGIN DI:=E; 15(DS:=8LIT" "); DI:=E; SI:=A;                          00763195
        DS:=WDS; SI:=LOC B; DI:=DI+1;                                    00763200
        2(DS:=2DEC; A:=DI;DI:=DI-2;DS:=FILL;DI:=A;DS:=LIT"/");           00763205
        DI:=DI-1; DS:=4LIT" ";                                           00763210
        SI:=SI+6; SKIP 2 SB; % START AT D25                              00763215
        IF SB THEN DS:=3LIT"ABN" ELSE DI:=DI+3; SKIP SB; %D25            00763220
        DI:=DI+3;                                                        00763225
        IF SB THEN DS:=3LIT"RR " ELSE DI:=DI+3; SKIP SB; %D24            00763230
        DI:=DI+3;                                                        00763235
        IF SB THEN DS:=3LIT"BFL" ELSE DI:=DI+3; SKIP SB; %D23            00763240
        DI:=DI+3;                                                        00763245
        SKIP SB;%IGNORE D22                                              00763250
        IF SB THEN DS:=3LIT"WR " ELSE DI:=DI+3; SKIP SB; %D21            00763255
        DI:=DI+3;                                                        00763260
        IF SB THEN DS:=3LIT"BSY" ELSE DI:=DI+3; SKIP SB; %D20            00763265
        DI:=DI+3;                                                        00763270
        SKIP SB; %IGNORE D19                                             00763275
        IF SB THEN DS:=3LIT"NTR" ELSE DI:=DI+3; %D18                     00763280
    END FLL;                                                             00763285
PROCEDURE TIM(A,CC); VALUE A; REAL A; ARRAY CC[0];                       00763290
BEGIN INTEGER I,J;                                                       00763295
    STREAM PROCEDURE FF(A,B);                                            00763300
    BEGIN SI:=A; DI:=B;                                                  00763305
        4(DS:=2 DEC; DS:=LIT "!");                                       00763310
        DI:=DI-1; DS:=5 LIT " ";                                         00763315
    END FF;                                                              00763320
    FOR I:=3 STEP -1 UNTIL 0 DO                                          00763325
    BEGIN CC[I]:=J:=A MOD 60;                                            00763330
        A:=A DIV 60;                                                     00763335
    END;                                                                 00763340
    FF(CC,LINE[7]);                                                      00763345
END TIM;                                                                 00763350
BOOLEAN STREAM PROCEDURE GLUNK(A);                                       00763355
BEGIN SI:=A; IF SC=" " THEN TALLY:=1;                                    00763360
    GLUNK:=TALLY;                                                        00763365
END GLUNK;                                                               00763370
PROCEDURE MMM(B,I); VALUE B; REAL I,B;                                   00763375
BEGIN REAL C;                                                            00763380
    STREAM PROCEDURE MOVE(A,B);                                          00763385
    BEGIN SI:=A; DI:=B;                                                  00763390
        8(IF TOGGLE THEN DS:=LIT " " ELSE                                00763395
        IF SC="+" THEN DS:=CHR ELSE DS:=CHR);                            00763400
    END MOVE;                                                            00763405
    FOR C:=0 STEP 1 UNTIL B-1 DO                                         00763410
    MOVE(INTSP[BUMPI],LINE[C]);                                          00763415
    WRITE(P,B,LINE[*]);                                                  00763420
END;                                                                     00763425
%%%%%%%%%%%%%%%%%%%%START                                                00763430
INIT(NAM);                                                               00763435
IF OPERAND(TOGLE,I) AND BOOLEAN(I.[13:1]) THEN                           00763440
IF PDATADESC(ARGH,FROM) AND FROM.[8:10]=128 THEN                         00763445
IF OPERAND((FROM:=FROM.CF)+127,I) THEN GO:=TRUE                          00763450
ELSE GO TO EOJ ELSE GO TO EOJ ELSE GO TO EOJ;                            00763455
WRITE(P[PAGE]);                                                          00763460
WRITE(P,WHATFILE,I.[6:6],I);                                             00763465
DISPLAY(ARGH,TRUE);                                                      00763470
```

```
        WRITE(P,HDR);                                                    00763475
        GO EOJ; % VOID THIS CARD TO DUMP SEPTIC DISK FILE                00763480
        FILL SEPTIC WITH "SEPTIC ",I;                                    00763485
        WHILE LO DO                                                      00763490
        BEGIN READ(SEPTIC,60,INTSP[*])[EOJ]; I:=0;                       00763495
LOOP:                                                                    00763500
             IF (B:=INTSP[I]).[9:9]=0 THEN GO TO EOJ;                    00763505
             FLL(NAM[B.[7:2]],B.[9:4],B.[14:4],B.[21:12],LINE);          00763510
             TIM(INTSP[BUMPI],[18:30],CC);                               00763515
             WRITE(P,10,LINE[*]);                                        00763520
             IF (B:=B.[33:15])≠0 THEN                                    00763525
             BEGIN WHILE B GTR 14 DO                                     00763530
                  BEGIN MMM(14,I); B:=B-14; END;                         00763535
                  MMM(B,I);                                              00763540
             END;                                                        00763545
             WRITE(P);                                                   00763550
             IF I≠59 THEN                                                00763555
             IF GLUNK(INTSP[BUMPI]) THEN ELSE GO TO LOOP;                00763560
        END;                                                             00763565
EOJ:    IF LO THEN                                                       00763570
        BEGIN CLOSE(SEPTIC); LO:=FALSE;                                  00763575
             IF OPERAND(FROM+125,I) AND 1 NEQ 0 THEN                     00763580
             IF OPERAND(FROM+62,I) AND (I=0 OR I=64) THEN                00763585
             BEGIN FROM:=FROM+I;                                         00763590
                  IF (B:=512-(R:=FROM.COL)) LSS (I:=60) THEN             00763595
                  BEGIN MOVE(M[FROM.ROW,R],INTSP,B);                     00763600
                       R:=0; I:=60-B; FROM:=FROM+60;                     00763605
                  END ELSE B:=0;                                         00763610
                  MOVE(M[FROM.ROW,R],INTSP[B],I);                        00763615
                  I:=0; GO TO LOOP;                                      00763620
        END; END; END;                                                   00763625
        INTEGER STREAM PROCEDURE MCPCNT(A); VALUE A;                     00765000
        BEGIN SI:=LOC A; SI:=SI+1;                                       00766000
            7(IF TOGGLE THEN TALLY:=TALLY+1 ELSE IF SC NEQ "0" THEN      00767000
              TALLY:=TALLY+1;SI:=SI+1);    MCPCNT:=TALLY;                00768000
        END OF MCPCNT;%                                                  00769000
        INTEGER PRTCODE; % SET TO 0,1 OR 2 BY CHECKPRTFILE               00769900
        PROCEDURE DATIME;                                                00770000
        BEGIN INTEGER I; BOOLEAN B;                                      00771000
            STREAM PROCEDURE PRTFILEMESS(P,L,C);    VALUE C;             00771100
            BEGIN LABEL MAYBE,BAD,FIX,XIT;                               00771105
                SI:=P; DI:=L; DS:=4 WDS;                                 00771110
                CI:=CI+C;                                                00771115
                GO XIT; % 0                                              00771120
                GO MAYBE; % 1                                            00771125
                GO BAD; % 2                                              00771130
                MAYBE:DS:=9LIT"(APPEARS "; GO FIX;                       00771135
                BAD:  DS:=12LIT"(DEFINITELY "; GO FIX;                   00771140
                FIX   :DS:=10LIT"INCORRECT)";                            00771145
                XIT   :DS:=82LIT" ";                                     00771150
            END PRTFILEMESS;                                             00771155
              IF NOT COMMON THEN BEGIN                                   00772000
                IF NOT OPERAND(3,MCPVERSION) THEN WRITE(P,BADCELL3);     00773000
                 IF FLGBIT(M[DATE.ROW,DATE.COL]) THEN                    00774000
                   WRITE(P,BADDATE) ELSE                                 00775000
                     BEGIN                                               00776000
                       DATX := M[DATE.ROW,DATE.COL];                     00777000
                       DATES(DATX,DATETAKEN,DAYS,YEARS);                 00778000
                   END;  % OF DATE                                       00779000
                IF FLGBIT(M[XCLOCK.ROW,XCLOCK.COL]) THEN                 00780000
```

```
        WRITE (P,BADXCLOCK) ELSE                                                00781000
              BEGIN                                                             00782000
                 XCLOCX := M[XCLOCK,ROW,xCLOCK,COL];                            00783000
                 TIMES(XCLOCX,TIMETAKEN,MINUTES,SECONDS);                       00784000
              END;    %     OF XCLOCK                                           00785000
        PRTFILEMESS(PRTNAME,LINE,PRTCODE);                                      00785400
        WRITE(P,8,LINE[*]),                                                     00785500
         WRITE(P,FANAL,VERSION);                                                00785550
        WRITE(P,FTARDSK,FILETYPE,TDMFID,[6:6],TLMFID,                           00785600
                                  TDFID,[6:6],TDFID);                           00785610
        WRITE(P,FCOMVAL,REAL(COMMON));                                          00785700
      WRITE(P,X1MARKXI,IF B:=(I:=MCPCNT(MCPVERSION) GTR 6)                      00786000
           THEN MCPVERSION.[6:6] ELSE " ", IF B THEN 6 ELSE IF I=0              00787000
           THEN 1 ELSE I,MCPVERSION,DATETAKEN,DAYS,YEARS,                       00788000
           TIMETAKEN,MINUTES,SECONDS);                                          00789000
       END;                                                                     00790000
       IF SHAREDISK THEN                                                        00790100
       IF OPERAND(SYSNO,I) AND I GEQ 0 AND I LEQ 3 THEN                         00790110
       WRITE(P,SYSID,I+17);                                                     00790120
        DATES(TIME(0),DATEANALYZED,DAYS,YEARS);                                 00791000
        TIMES(TIME(1),TIMEANALYZED,MINUTES,SECONDS);                            00792000
     WRITE(P,FMXX,DATEANALYZED,DAYS,YEARS,TIMEANALYZED,MINUTES,SECONDS);        00793000
       IF COMMON THEN WRITE(P);                                                 00794000
       IF NOT COMMON THEN BEGIN                                                 00795000
       IF FLGBIT(M[CLOCK,ROW,CLOCK,COL]) THEN                                   00796000
         WRITE(P,BADCLOCK) ELSE                                                 00797000
           BEGIN                                                                00798000
             CLOCX := M[CLOCK,ROW,CLOCK,COL];                                   00799000
           IF XCLOCX LSS CLOCX THEN XCLOCX:=XCLOCX+5184000;                     00800000
           TIMES(XCLOCX-CLOCX,TIMELASTHL,MINUTES,SECONDS);                      00801000
             WRITE(P,FMX2,TIMELASTHL,MINUTES,SECONDS);                          00802000
             TIMES(CLOCX,SINCSLASTHL,MINUTES,SECONDS);                          00803000
             WRITE(P[DBL],FMX3,SINCSLASTHL,MINUTES,SECONDS);                    00804000
           END ; %OF CLOCK                                                      00805000
       END;                                                                     00806000
END;% OF DATIME                                                                 00807000
PROCEDURE PRINTARRAYRCW(PRTLOC,INX); VALUE PRTLOC,INX;                          00807100
                                     INTEGER PRTLOC,INX;                        00807110
BEGIN                                                                           00807120
    INTEGER LOC,SIZE;                                                           00807130
    BOOLEAN TOG,COMSAVE,DBLSAVE;                                                00807140
    LABEL AGAIN;                                                                00807150
    STREAM PROCEDURE FIXLINE(LINE,INX); VALUE INX;                              00807160
    BEGIN LOCAL A,B;                                                            00807170
        SI:=LINE;4(SI:=SI+B);                                                   00807180
        32(IF SC=" " THEN JUMP OUT ELSE SI:=SI+1);                              00807190
        A:=SI; DI:=A; DS:="LIT"["; A:=DI;                                       00807200
        SI:=LOC INX; DS:=4 DEC; B:=DI;                                          00807210
        DI:=A; DS:=3 FILL; DI:=B;                                               00807230
        DS:=3"LIT",*]";                                                         00807240
    END FIXLINE;                                                                00807250
    IF TOG:=BOOLEAN(INX,[1:1]) THEN                                             00807260
    DISPLAY(PRTLOC,TRUE);                                                       00807270
    INX:=ABS(INX);                                                              00807275
    COMSAVE:=COMMON;                                                            00807280
    COMMON:=FALSE;                                                              00807290
    DBLSAVE:=DOUBLESPACE;                                                       00807295
    DOUBLESPACE:=TRUE;                                                          00807300
    IF PRTLOC GEQ 129 AND PRTLOC LEQ PRTMAX THEN                                00807310
AGAIN:                                                                          00807320
```

```
        IF PDATADESC(PRTLOC,LOC) THEN                                   00807330
        IF (SIZE:=LOC.[8:10]) NEQ 0 THEN                                00807340
        IF TOG THEN PRINT(LOC:=LOC.CF,MIN(MAXCOR,LOC+SIZE)) ELSE        00807350
        BEGIN                                                           00807360
             DISPLAYIT(PRTLOC,TRUE,PRTLOC:=LOC.CF+INX);                 00807370
             FIXLINE(LINE,INX);                                         00807380
             WRITE(P[DBL],15,LINE[*]);                                  00807390
             TOG:=TRUE;                                                 00807400
             GO AGAIN;                                                  00807410
        END;                                                            00807420
        COMMON:=COMSAVE;                                                00807430
        DOUBLESPACE:=DBLSAVE;                                           00807435
    END PRINTARRAYROW;                                                  00807440
    DEFINE PRINTARRAY(PRINTARRAY1)=                                     00807450
        PRINTARRAYROW(PRINTARRAY1,-1)#;                                 00807460
    DEFINE PA=PRINTARRAY#,PAROW=PRINTARRAYROW#;                         00807470
    COMMENT: THE FOLLOWING PROCEDURE IS INTENDED TO BE MODIFIED TO SUIT 00807480
    INDIVIDUAL USER DEBUGGING REQUIREMENTS.  IT ALLOWS AN ARRAY,        00807482
    EITHER 1 OR 2 DIMENSIONAL, TO BE DISPLAYED WITH A MINIMUM OF EFFORT. 00807484
    IN GENERAL, A 3 CARD PATCH IS NECESSARY TO THE ANALYZER:  ONE CARD TO 00807486
    DEFINE AN ELEMENT IN "PRTS", SEQUENCE #00121000,ANOTHER TO MAKE AN ENTRY00807488
    IN "FILLINFO",SEQUENCE #00165000, AND A THIRD TO CALL BELOW ON      00807490
    "PRINTARRAY"(OR "PA") FOR 1 DIMENSIONAL ARRAYS OR ON                00807492
    "PRINTARRAYROW"(OR "PAROW") FOR 2 DIMENSIONAL ARRAYS.               00807494
    NOTE THAT THE PROCEDURE "PRINTSELECTEDARRAYS" IS CALLED ONLY WHEN   00807496
    COMMON=16;                                                          00807498
    PROCEDURE PRINTSELECTEDARRAYS;                                      00807500
    BEGIN                                                               00807510
        INTEGER I;                                                      00807515
        IF DUMPTABLES THEN                                              00807520
        BEGIN                                                           00807530
             WRITE(P[PAGE]);                                            00807540
        IF DFX THEN PA(EUQ);                                            00807550
        PA(MEMASK);                                                     00807560
        PA(QTIMES);                                                     00807570
        PA(STATION);                                                    00807575
        FOR I:=0 STEP 1 UNTIL 15 DO PAROW(STATION,I);                   00807580
        END;                                                            00807890
        DUMPCESSPOOL;                                                   00807899
    END PRINTSELECTEDARRAYS;                                            00807900
    BOOLEAN BADCOMMENT;                                                 00808000
    FORMAT COMNTPAR("---PARITY ERROR OCCURRED IN COMMENTS BLOCK..."); 00809000
        PROCEDURE LOAD;                                                 00810000
        BEGIN                                                           00811000
            LABEL EOT;                                                  00812005
            LABEL EF,PR,IGPAR;                                          00812010
            LABEL FLAG,PAR,BADTM,IGNOREPAR;%                            00813000
            FORMAT BADEOF("---INVALID EOF AFTER BLOCK # ",I2),          00814000
                 PRTFILE("PRT FILE USED IS ",A1,A6,"/",A1,A6,X8),       00814100
                 PARERR ("---IRRECOVERABLE PARITY IN BLOCK # ",I2),     00815000
                 FLAGERR("---FLAG BIT IN WORD ZERO OF BLOCK # ",I2);    00816000
            ARRAY A[0:%32];                                             00817000
            STREAM PROCEDURE MOVER(S,D);                                00818000
            BEGIN                                                       00819000
                SI:=S; DI:=D;                                           00820000
                16(DS:=32 WDS);                                         00821000
            END MOVER;                                                  00822000
            STREAM PROCEDURE GETCOMMON(COMMT,N);                        00822100
            BEGIN LOCAL X,Y; LABEL XIT,HIT,GETNUM;                      00822110
            DI:=N; DS:=8LIT"+0000001"; SI:=COMMT;                       00822120
```

```
7(20(IF SC="C" THEN BEGIN SI:=SI+1;                                    00822130
     IF SC="C" THEN BEGIN SI:=SI+1;                                    00822140
     IF SC="N" THEN JUMP OUT 2 TO HIT ELSE SI:=SI+1                    00822150
     END ELSE SI:=SI+1 END ELSE SI:=SI+1));                            00822155
GO XIT;                                                                00822160
HIT:15(IF SC="=" THEN JUMP OUT TO GETNUM ELSE SI:=SI+1); GO XIT;       00822170
GETNUM:SI:=SI+1;                                                       00822175
DI:=LOC X;                                                             00822180
10(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);                             00822190
5(IF SC GEQ "0" THEN IF SC LEQ "9" THEN                                00822200
BEGIN TALLY:=TALLY+1; DS:=CHR END ELSE JUMP OUT ELSE JUMP OUT);        00822210
Y:=TALLY;                                                              00822220
SI:=LOC X;                                                             00822230
DI:=N;                                                                 00822240
DS:=Y OCT;                                                             00822250
XIT: END GETCOMMON;                                                    00822260
INTEGER I;                                                             00823000
ALPHA N1,N2;                                                           00823100
COMNT:=BADCOMMENT:=FALSE;                                              00824000
IF RELOAD THEN                                                         00824100
FOR I:=0 STEP 1 UNTIL 63 DO UNLOCK(M[A[0],ROW,*]);                     00824110
IF MULTI THEN                                                          00824200
READ REVERSE(MDUMP,20,COMMT[*])[BADTM:IGPAR];                          00824210
IF FALSE THEN IGPAR:BADCOMMENT:=TRUE;                                  00824220
FOR I:=0 STEP 1 UNTIL 63 DO                                            00825000
  BEGIN                                                                00826000
    IF MULTI THEN READ REVERSE(MDUMP,533,A[*])[BADTM:PAR] ELSE         00826990
    READ(MDUMP,533,A[*])[BADTM:PAR];%                                  00827000
    IF FLGBIT(A) THEN GO FLAG ELSE                                     00828000
    IF MODON[A[0],[33:3]]:=NOT BOOLEAN(A[0],[1:1]) THEN                00829000
    BEGIN                                                              00830000
      MOVER(A[1],M[A[0],ROW,0]);                                       00832000
      LOCK(M[A[0],ROW,*]);                                             00832100
    END ELSE IF A[0].[3:5]=0 THEN I:=I+7; %                            00833000
    COMMENT ONE BLOCK IS WRITTEN PER ABSENT MOD;                       00833010
  END;                                                                 00834000
  TDMFID:=MDUMP.MFID;                                                  00834050
  TDFID:=MDUMP.FID;                                                    00834060
  FILETYPE:=IF MDUMP.TYPE=2 THEN "TAPE" ELSE "DISK";                   00834070
  IF NOT MULTI THEN                                                    00834075
  IF FILETYPE="TAPE" AND TDMFID NEQ 0 THEN                             00834080
  IF TDFID.[30:18]=1 THEN ELSE MULTI:=TRUE;                            00834085
  IF REPEATING THEN IF TDFID.[30:18]=1 THEN CLOSE(MDUMP)               00834095
  ELSE CLOSE(MDUMP,*) ELSE                                             00834097
  IF (AUXTYPE:=A[0].[3:5])=0 THEN                                      00834100
  IF FILETYPE="TAPE" THEN                                              00834200
  READ(MDUMP,20,COMMT[*])[EOT:IGNOREPAR] ELSE % DISK                   00835000
  MOVE(A[513],COMMT,20);                                               00835100
 IF FALSE THEN IGNOREPAR:BADCOMMENT:=TRUE;                             00836000
COMNT := TRUE;                                                         00837000
 GETCOMMON(COMMT,N1);                                                  00837050
 IF N1 GEQ 0 THEN COMMON:=BOOLEAN(N1); N1:=0;                          00837055
  IF MULTI THEN NOMO:=TRUE ELSE                                        00837095
  READ(MDUMP[NO])[EOT:PAR];                                            00837100
  IF FALSE THEN                                                        00837200
  BEGIN                                                                00837300
  EOT: CLOSE(MDUMP);                                                   00838000
  NOMO:=TRUE;                                                          00838010
  END;                                                                 00838100
  IF COMMON OR RELOAD THEN                                             00839000
```

```
        ELSE                                                                    00840000
        BEGIN                                                                   00841000
        IF PRT32 THEN READARRAY(30,SEGZERO[*],0) ELSE                           00843000
         SPACE(DISK,1);                                                         00844000
         READARRAY(NAMESIZE,NAME[*],PRTBASE);                                   00845000
         READARRAY(NAMSSIZE,NAMS[*],0);                                         00846000
        READARRAY(INAMESIZE,INAME[*],0);                                        00847000
         READARRAY(INAMSSIZE,INAMS[*],0);                                       00848000
        CLOSE(DISK);                                                            00849000
        N1:=DISK.MFID; N2:=DISK.FID;                                            00849100
        WRITE(PRTNAME[*],PRTFILE,N1.[6:6],N1,N2.[6:6],N2);                      00849200
        END;                                                                    00850000
            IF RELOAD THEN ELSE                                                 00850100
            BEGIN                                                               00850200
        MAXMOD:=7; %                                                            00851000
        WHILE NOT MOCON[MAXMOD] DO MAXMOD:=MAXMOD-1;                            00852000
        MAXCOR:=4096*(MAXMOD+1)-1; %                                            00853000
    IF COMMON THEN ELSE                                                         00854000
    BEGIN                                                                       00855000
        PRTMAX:=PRTBASE+NAMESIZE-1;                                             00856000
        INTMAX:=INAMESIZE-1;                                                    00857000
        FOR I:=PRTBASE STEP 1 UNTIL PRTMAX DO NSNAME[I]:=                       00858000
        NSNAME[I]+PRTBASE;                                                      00859000
        FOR I:=0 STEP 1 UNTIL INTMAX DO ISNAME[I]:=                             00860000
        ISNAME[I]+PRTBASE;                                                      00861000
        FIXDEFINES;                                                             00862000
        SETUPXNAMEANDXNAMS;                                                     00863000
        IF MYSTACKADR LSS 0 THEN % LETS USE IT                                  00863100
        IF CONTROLDESC(7,I) THEN % CHECK FOR MSCW IN CELL 7                     00863110
        IF I.CF=0 AND I!=I.FF GEQ 0 THEN % ASSUME VALID                         00863120
        MYSTACKADR:=I+30; % MAY NEED MORE THAN 30                               00863130
    END;                                                                        00864000
            END;                                                                00864100
     IF FALSE THEN BEGIN                                                        00865000
     BADTM:WRITE(SPO,BADEOF,I); GO EOPROG;                                      00866000
     PAR:   WRITE(SPO,PARERR,I+1); GO EOPROG;                                   00867000
     FLAG:  WRITE(SPO,FLAGERR,I+1); GO EOPROG;  END;                            00868000
    END LOAD;                                                                   00869000
PROCEDURE CHECKPRTFILE;%                                                        00870000
%SIGNALS USER IF MCP/PRT FILE *MIGHT* BE THE WRONG ONE                          00871000
BEGIN                                                                           00872000
REAL LOC,TYP,I;                                                                 00873000
REAL MYTYPE;                                                                    00874000
LABEL BAD,MAYBE;                                                                00875000
DEFINE CHECKFORLABELDESCRIPTOR=IF LOC LSS 0 THEN GO BAD ELSE                    00876000
       IF DESCRIPTOR(LOC,I,TYP) AND TYP=MYTYPE THEN ELSE GO MAYBE#;             00877000
DEFINE CHECKFORPROGRAMDESCRIPTOR=CHECKFORLABELDESCRIPTOR#;                      00878000
FORMAT BADPRT("---YOUR MCP/PRT FILE IS WRONG..."),                              00879000
INCOMPAT("---MCP/PRT FILE NOT COMPATIBLE WITH MCPS PRT IN MEMORY...");          00880000
DEFINE ERRMESS(ERRMESS1)=BEGIN WRITE(SPO,ERRMESS1);                             00881000
            WRITE(P[PAGE],ERRMESS1) END#;%                                      00882000
MYTYPE:="76";                                                                   00883000
IF (LOC:=NOTHINGTODO) NEQ 0 THEN                                                00884000
CHECKFORLABELDESCRIPTOR;                                                        00885000
IF (LOC:=STACKOVERFLOW) NEQ 0 THEN                                              00886000
CHECKFORLABELDESCRIPTOR;                                                        00887000
IF (LOC:=RETURN) NEQ 0 THEN                                                     00888000
CHECKFORLABELDESCRIPTOR;                                                        00889000
MYTYPE:="75";                                                                   00890000
IF (LOC:=DIRECTORYBUILDER)=0 THEN LOC:=-1;                                      00891000
```

```
      CHECKFORPROGRAMDESCRIPTOR;                                                00892000
      IF FALSE THEN MAYBE:BEGIN ERRMESS(INCOMPAT); PRTCODE:=1 END;              00893000
      IF FALSE THEN BAD:BEGIN ERRMESS(BADPRT); PRTCODE:=2 END;                  00894000
      END OF CHECKPRTFILE;                                                      00895000
      PROCEDURE DUMPMCPSPRT;                                                    00896000
      BEGIN                                                                     00897000
      FORMAT HDR(X42,"D C M C P S  P R T"/                                      00898000
                 X42,"= = = = = =  = = ="///,                                   00899000
                 2("PRT",X5,"CONTENTS",X16,"NAME",X28)//),                      00900000
             DASHES(120("=")),                                                  00901000
             F(X8,"MEMORY MODS: ",8I1),                                         00902000
             PRTITEM(2(A5," = ",B,X1,O,X39));                                   00903000
      INTEGER LOC,N;                                                            00904000
      LABEL EXIT;                                                               00904100
      IF DUMPCESSPOOLONLY THEN DONTDUMPRT:=TRUE;                                00904500
      IF NOT COMMON THEN                                                        00905000
      BEGIN                                                                     00906000
      CHECKPRTFILE;                                                             00907000
      IF NOT DONTDUMPRT THEN BEGIN DATIME; WRITE(P[DBL]) END;                   00908000
      PRINTMCPOPTIONS;                                                          00909000
      IF DONTDUMPRT THEN ELSE                                                   00910000
      BEGIN                                                                     00911000
         WRITE(P[DBL],HDR);                                                     00912000
         FOR LOC:=ACTUALPRTBASE STEP 1 UNTIL PRTBASE DO                         00913000
         BEGIN                                                                  00914000
             WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),               00915000
                OCTAL(LOHALF(LOC)),OCTAL(N:=XSNAME[LOC]),                       00916000
                OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));                             00917000
             MOVE(XNAMS[XNAME[LOC],CF],LINE[4],XNAME[LOC],FF);                  00918000
             MOVE(XNAMS[XNAME[N],CF],LINE[12],XNAME[N],FF);                     00919000
             WRITE(P,15,LINE[*]);                                               00920000
             IF DONTDUMPRT THEN GO EXIT;                                        00920100
         END;                                                                   00921000
         WRITE(P,DASHES);                                                       00922000
         FOR LOC:=PRTBASE+1 STEP 1 UNTIL PRTMAX DO                              00923000
         BEGIN                                                                  00924000
             WRITE(LINE[*],PRTITEM,OCTAL(LOC),OCTAL(HIHALF(LOC)),               00925000
                OCTAL(LOHALF(LOC)),OCTAL(N:=NSNAME[LOC]),                       00926000
                OCTAL(HIHALF(N)),OCTAL(LOHALF(N)));                             00927000
             MOVE(NAMS[NAME[LOC],CF],LINE[4],NAME[LOC],FF);                     00928000
             MOVE(NAMS[NAME[N],CF],LINE[12],NAME[N],FF);                        00929000
             WRITE(P,15,LINE[*]);                                               00930000
             IF DONTDUMPRT THEN GO EXIT;                                        00930100
         END;                                                                   00931000
      EXIT:                                                                     00931100
         NEXTPAGE;                                                              00932000
      END;                                                                      00933000
      END;                                                                      00934000
      WRITE(P[DBL],F,FOR N:=0 STEP 1 UNTIL 7 DO                                 00935000
           [IF MODON[N] THEN N ELSE 10]);                                       00936000
      DATIME;                                                                   00937000
      IF DUMPCESSPOOLONLY THEN ELSE                                             00937500
      IF NOT COMMON THEN                                                        00938000
      BEGIN                                                                     00939000
         MCPENTRIES;                                                            00940000
         GETPRTENTRIES; NEXTPAGE;                                               00941000
      END;                                                                      00942000
      END OF DUMP OF MCPS PRT;                                                  00943000
         PROCEDURE CHECKMEMORYLINKS;                                            00944000
         BEGIN %                                                                00945000
```

```
BOOLEAN ZEROK; REAL V0; %                                                    00946000
BOOLEAN MSTARTOK; REAL VMSTART; %                                            00947000
BOOLEAN NOWRAPAROUND;                                                        00947100
REAL LINK,VLINK,PREVLINK;                                                    00948000
INTEGER AVAILN,AVAILT,AVAILM;                                                00949000
BOOLEAN PROCEDURE LINKOK(WORD); %                                            00950000
    VALUE WORD; %                                                            00951000
    REAL WORD; %                                                             00952000
LINKOK:=WORD.[3:6]≤TYPMAX AND %                                              00953000
    WORD.[9:6]≤MIXMAX     AND                                                00954000
    WORD.[15:3]=0; %                                                         00955000
FORMAT RANGE(X8,"PRIMARY LINKS FROM ",A5," TO ",A5," ARE OK"),               00956000
       FCORE(A5," = ",2(O,X1),X6,"CORE",X8,"FACTOR = ",F4.2,                 00956100
       ", MAX CORE = ",A5,"(",I5,")",",USING ",A5,"(",I5,")"),               00956110
        BAD(X8,A6," LINK AT ",A5," IS NOT OK"),                              00957000
BADPRIMARY(X8,"PRIMARY LINK AT ",A5," IS NOT OK"),                           00958000
       AVLBAD(X8,"AVAILABLE STORAGE TOTALS DO NOT CROSS CHECK"),             00959000
       AVL(X8,"AVAILABLE LINKS ARE OK, TOTALING",                            00960000
           I4,"(DECIMAL) AREAS OF ",A5,"(",I5,                               00961000
           ") WORDS UP TO ",A5,"(",I5,") WORDS EACH");                       00962000
INTEGER N,T,M; %                                                             00963000
DISPLAY(O,FALSE);                                                            00964000
DISPLAY(MSTART,FALSE);                                                       00965000
DISPLAY(MEND,FALSE);                                                         00966000
DISPLAY(AVAIL,FALSE);                                                        00967000
IF OPERAND(CORE,T) AND T.[1:3]=0 THEN                                        00967100
BEGIN                                                                        00967110
    WRITE(LINE[*],FCORE,OCTAL(CORE),OCTAL(HIHALF(CORE)),                     00967120
         OCTAL(LOHALF(CORE)),T.[4:14]/100,                                   00967130
         OCTAL((N:=T.CF×64).CF),N,                                           00967140
         OCTAL((N:=T.FF×64).CF),N);                                          00967150
    WRITE(P[DBL],15,LINE[*]);                                                00967160
END ELSE                                                                     00967170
    DISPLAY(CORE,FALSE);                                                     00967180
    ZEROK:=OPERAND(0,V0)          AND                                        00968000
        V0.[1:2]=1                AND                                        00969000
V0.[03:15]=0                  AND                                            00970000
        V0.FF=(MAXLNK:=MAXCOR-2) AND                                         00971000
        V0.CF GTR 1023 AND V0.CF LSS 4096 AND                                00972000
        OPERAND(V0.CF,T) AND LINKOK(T) AND T.FF=0;                           00972400
    MSTARTOK:=OPERAND(MSTART,VMSTART) AND                                    00973000
        VMSTART.[1:35]=0          AND %                                      00974000
        VMSTART GTR 1023 AND VMSTART LSS 4096;                               00975000
    IF LNKSOK:=MSTARTOK OR ZEROK THEN                                        00976000
    MINLNK:=LINK:=IF MSTARTOK AND ZEROK THEN MAX(V0.CF,VMSTART)              00976100
    ELSE IF MSTARTOK THEN VMSTART ELSE                                       00976200
    IF ZEROK THEN V0.CF ELSE 0;                                              00976300
    IF LNKSOK:=MSTARTOK THEN MINLNK:=LINK:=VMSTART; %                        00977000
    AVALNKOK:=TRUE;                                                          00978000
     N:=T:=0;                                                                00978100
    WHILE LNKSOK AND MAXLNK>PREVLINK DO %                                    00979000
    IF LNKSOK:=(OPERAND(LINK,VLINK) AND                                      00980000
                LINKOK(VLINK)       AND %                                    00981000
                VLINK.FF=PREVLINK   AND %                                    00982000
               (IF LINK=MAXLNK                                               00983000
                THEN VLINK.CF=0 %                                            00984000
                ELSE VLINK.CF>LINK))THEN                                     00985000
    BEGIN %                                                                  00986000
        PREVLINK:=LINK; %                                                    00987000
        LINK:=VLINK.CF; %                                                    00988000
```

```
            IF AVALNKOK THEN %                                                  00989000
            IF BOOLEAN(VLINK.[1:1]) THEN %                                      00990000
            IF AVALNKOK:=MAXLNK>PREVLINK THEN %                                 00991000
            IF AVALNKOK:=(OPERAND(PREVLINK+1,VLINK) AND %                       00992000
                    VLINK.[1:17]=0) THEN %                                      00993000
            BEGIN %                                                             00994000
                AVAILN:=AVAILN+1; %                                             00995000
                AVAILT:=AVAILT+VLINK.FF; %                                      00996000
                AVAILM:=MAX(AVAILM,VLINK.FF); %                                 00997000
            END; %                                                              00998000
        END; %                                                                  00999000
    IF NOT ZEROK THEN WRITE(P,BADPRIMARY,0);                                    00999100
    IF LNKSOK THEN WRITE(P,RANGE,IF ZEROK THEN 0 ELSE OCTAL(MINLNK),            01000000
    OCTAL(MAXLNK))              ELSE                                            01001000
        BEGIN %                                                                 01002000
            IF SOMOKF:=PREVLINK>MINLNK THEN                                     01003000
            BEGIN %                                                             01004000
            WRITE(P,RANGE,IF ZEROK THEN 0 ELSE OCTAL(MINLNK),                   01005000
                OCTAL(PREVLINK));                                               01006000
            WRITE(P,BADPRIMARY,OCTAL(LINK));%                                   01007000
            END; %                                                              01008000
            MINBAD:=LINK;                                                       01009000
            PREVLINK:=0; %                                                      01010000
            LINK:=MAXLNK; %                                                     01011000
            WHILE OPERAND(LINK,VLINK) AND %                                     01012000
                LINKOK(VLINK)          AND                                      01013000
                VLINK.CF=PREVLINK      AND %                                    01014000
                LINK>VLINK.FF          DO %                                     01015000
            BEGIN %                                                             01016000
                PREVLINK:=LINK; %                                               01017000
                LINK:=VLINK.FF; %                                               01018000
            END; %                                                              01019000
            IF PREVLINK=0 THEN MAXBAD:=MAXCOR+1 ELSE %                          01020000
            WRITE(P,RANGE,OCTAL(MAXLNK),OCTAL(MAXBAD:=PREVLINK)); %             01021000
            WRITE(P,BADPRIMARY,OCTAL(LINK));%                                   01022000
            SOMOKB:=PREVLINK>0;                                                 01023000
        END; %                                                                  01024000
        NOWRAPAROUND:=TRUE;                                                     01025000
        IF AVALNKOK:=(OPERAND(AVAIL,PREVLINK) AND                               01026000
            PREVLINK=MAXLNK+1          AND %                                    01027000
            OPERAND(PREVLINK,VLINK)    AND %                                    01028000
            VLINK.[1:17]=0             AND %                                    01029000
            VLINK.FF=32767)            THEN %                                   01030000
        BEGIN %                                                                 01031000
            WHILE AVALNKOK AND %                                                01032000
                NOWRAPAROUND AND                                                01032100
                (IF LNKSOK THEN AVAILN GTR N ELSE TRUE) AND                     01033000
                (IF LNKSOK THEN AVAILT GTR T ELSE TRUE) AND                     01034000
            (IF LNKSOK THEN AVAILM GEQ M  ELSE TRUE) DO                         01035000
            IF NOWRAPAROUND:=(MAXCOR GTR LINK:=VLINK.CF                         01036000
                        AND LINK GTR 0) THEN                                    01036100
            IF AVALNKOK:=(OPERAND(LINK-1,VLINK) AND %                           01037000
                LINKOK(VLINK)          AND %                                    01038000
                BOOLEAN(VLINK.[1:1])   AND %                                    01039000
                OPERAND(LINK+1,VLINK)  AND %                                    01040000
                VLINK=PREVLINK         AND %                                    01041000
                OPERAND(LINK,VLINK)    AND %                                    01042000
                VLINK.[1:17]=0         AND %                                    01043000
                VLINK.CF#LINK)         THEN %                                   01044000
            BEGIN %                                                             01045000
```

```
                    N:=N+1;                                                    01046000
                    T:=T+VLINK.FF; %                                           01047000
                    M:=MAX(M,VLINK.FF); %                                      01048000
                    PREVLINK:=LINK; %                                          01049000
                END; %                                                         01050000
                IF NOT AVALNKOK THEN WRITE(P,BAD,"AVALBL",OCTAL(LINK)) ELSE    01051000
                IF LNKSOK AND                                                  01051100
                  NOT(AVALNKOK:=                                               01052000
                    LNKSOK AND                                                 01052100
                    AVAILN=N AND %                                             01053000
                    AVAILT=T AND %                                             01054000
                    AVAILM=M AND %                                             01055000
                    VLINK.CF=MAXLNK+1) THEN WRITE(P,AVLBAD) ELSE %             01056000
                WRITE(P,AVL,N,OCTAL(T),T,OCTAL(M),M);                          01057000
             TABLESLOC:=IF LNKSOK AND OPERAND(MAXLNK,T)                        01058000
                        THEN T.FF ELSE -1;                                     01059000
            END ELSE WRITE(P,BAD,"AVALBL",OCTAL(MAXLNK+1));                    01060000
            WRITE(P[DBL]); %                                                   01061000
        END; %                                                                 01062000
        ARRAY MIXSTK[0:43];                                                    01063000
        PROCEDURE GETSTACKSFROMTHEBED;                                         01064000
        BEGIN                                                                  01065000
            INTEGER I;                                                         01066000
            INTEGER MIX;                                                       01067000
        INTEGER M; BOOLEAN B;                                                  01068000
        FORMAT DUPBED(X8,"DUPLICATE BED ENTRY FOR MIX=",I2,                    01069000
                    " ,CHECK BED AT ADDRESS = ",A5,                            01070000
                    " (THIS ENTRY IGNORED)");                                  01071000
            REAL V,TYP;                                                        01072000
            IF OPERAND(JOBNUM,VJOBNUM) THEN                                    01073000
            IF VJOBNUM.[1:8]=0 AND                                             01074000
               VJOBNUM.[47:1]=0 AND                                            01075000
               PDATADESC(BED,VBED) AND                                         01076000
               VJOBNUM≤VBED.[8:10] THEN                                        01077000
            BEGIN                                                              01078000
                VBED:=VBED.CF;                                                 01079000
                FOR I:=0 STEP 2 UNTIL VJOBNUM DO                               01080000
                IF DESCRIPTOR(VBED+I,V,TYP) THEN                               01081000
            IF B:=((MIX:=V.[6:2]&TYP[43:45:3]) GTR 0                           01082000
                AND (M:=MIXSTK[MIX]) NEQ 0)    THEN                            01083000
        WRITE(P,DUPBED,MIX,OCTAL(VBED+I))                                      01084000
            ELSE                                                               01085000
            BEGIN                                                              01086000
                BEDSTK:=MAXSTK:=MAXSTK+1;                                      01087000
                    MIXSTK[MIX]:=BEDSTK;                                       01088000
                    STAX[BEDSTK]:=                                             01089000
                        V.FF&                                                  01090000
                        I[8:38:10]&                                            01091000
                        (REAL(MIX>0))[7:47:1];                                 01092000
                END;                                                           01093000
            END;                                                               01094000
        IF B THEN WRITE(P[DBL]);                                               01095000
        END GETSTACKSFROMTHEBED;                                               01096000
        PROCEDURE DUMPMEMORYANDNOTESTACKS(FROM,TOO);                           01097000
            VALUE FROM,TOO;                                                    01098000
            INTEGER FROM,TOO;                                                  01099000
        BEGIN                                                                  01100000
            BOOLEAN BEDDED;                                                    01101000
            DEFINE T=LINKTYPE#;                                                01101100
            REAL L;                                                            01102000
```

```
  FORMAT ITEM(X2,A3," =",2(X4,A5));                                    01103000
  REAL R,ADR,Q,LL; BOOLEAN INTR,QT;                                    01104000
REAL X, AVSIZE;                                                        01105000
REAL ISTACKV;BOOLEAN ISTACKOK,IRSTACK;                                 01106000
BOOLEAN MCPSAVE,MXNOTO;                                                01107100
DEFINE MXO=NOT MXNOTO#;                                                01107200
REAL DCQPTSTACK,DCQPTSTACKV;                                           01108000
BOOLEAN DCSTACK,DCHIT;                                                 01109000
DEFINE DC=DCQPTSTACK#,DCV=DCQPTSTACKV#;                                01110000
INTEGER N,STK;                                                         01112000
REAL MX;                                                               01113000
FORMAT AVAILABLE(X27,"**** AVAILABLE SIZE=",A5,"(",I5,")"),            01114000
       USERSTACKPRT(X27,"MIX=",I2," STACK-PRT"),                       01115000
       MIXUSE(X27,"MIX=",I2,X1,A6),                                    01116000
       SEGDICT(X27,"MIX=",I2," SEGMENT-DICTIONARY"),                   01117000
       MCPUSE(X27,"MCP=",A6),                                          01118000
       MCPSTACK(X27,"MCP INDEPENDENT-RUNNER STACK"),                   01119000
       INTARRAY(X27,"MCPS INTRNSC ARRAY"),                             01120000
       TABLEAREA(X27,"MCP TABLES");                                    01121000
ARRAY ATP[0:TYPMAX+1];                                                 01122000
NOTPRINTALL:=TRUE;                                                     01122100
FILLAREATYPE;                                                          01122200
IF DATACOM THEN                                                        01123000
BEGIN                                                                  01124000
     DCSTACK:=DC GTR 129 AND OPERAND(DC,DCV)                           01125000
             AND DCV NEQ 0 AND DCV.[1:32]=0                            01126000
             AND (DCV:=DCV.CF) GTR 1023 AND                            01127000
             OPERAND(DCV,R) AND R=0 AND                                01128000
             OPERAND(DCV-1,Q) AND Q.[9:6]=0 AND                        01129000
        (Q.[3:6]=0 OR Q.[3:6]=12) AND Q.[2:1]=1;                       01130000
END OF LOCATING DCQPTSTACK AFTER MANY CHECKS;                          01131000
ISTACKOK:=PDATADESC(ISTACK,ISTACKV) AND                                01132000
        ISTACKV.[8:10] GTR 0    AND                                    01133000
        (ISTACKV:=ISTACKV.CF) GTR 0;                                   01134000
INTR:=PDATADESC(INTRNSC,ADR) AND ADR.FF=0;                             01135000
  ADR:=ADR.CF;  FILL ATP[*] WITH                                       01136000
       "UNKNWN","CODE  ","DATA  ","IO-BUF",                            01137000
       "ALGFIB","INQBUF","COBFIB","INTSEG",                            01138000
      "HEADER","MAINT ","    10","SPACE ","STACK ",                    01139000
      "INT 13","SCRDIR","OPSET ","DIRTOP","SPOUT ",                    01140000
      "LABELS","JAR   ","CIDROW","INQPT ","INTRNC",                    01141000
      "RJEINP","DCQPT ","DALOC ","SHEET ","STAWRD",                    01141100
      "KEYIN ","    29","DC19Q ","    31","STAWD ",                    01141200
      "    33","    34","    35","    36","    37",                    01141300
      "    38","    39","    40",                                       01141400
      "@@@@@@";%"@" DENOTES UNRECOGNIZED LINK TYPE                     01142000
  WHILE FROM<TOO DO                                                    01143000
  BEGIN                                                                01144000
  IRSTACK:=FALSE;                                                      01145000
 DCHIT:=FALSE;                                                         01146000
     N:=(L:=M[FROM.ROW,FROM.COL]).CF;                                  01149000
     IF T:=L.[3:6] GTR TYPMAX THEN L.[3:6]:=T:=TYPMAX+1;% BAD TYP      01149100
     MXNOTO:=(MX:=L.[9:6]) NEQ 0;                                      01149200
     IF BOOLEAN(L.[1:1]) THEN                                          01150000
     IF NODUMP THEN ELSE                                               01150100
BEGIN                                                                  01151000
  AVSIZE:=                                                             01152000
  IF OPERAND(FROM+1,X) AND X.[1:17]=0 THEN                             01153000
  X.FF ELSE -10000;                                                    01154000
  WRITE(P[DBL],AVAILABLE,IF AVSIZE LSS 0 THEN "*****" ELSE             01155000
```

```
            OCTAL(AVSIZE),AVSIZE);                                          01156000
          AVALNK:=TRUE;                                                     01156400
          PRINT(FROM,IF DUMPAVAIL OR                                        01158000
            NEEDCHECKAVAILNKS THEN N ELSE FROM+3);                          01159000
          AVALNK :=FALSE;                                                   01159100
        END                                                                 01160000
          ELSE                                                              01161000
          BEGIN                                                             01162000
      QT:=OPERAND(FROM+1,Q);                                                01163000
       MCPSAVE:=QT AND Q NEQ 0 AND L.[2:13]=4096 AND Q.FF NEQ 0             01164000
          AND Q.CF GEQ 129 AND Q.CF LEQ PRTMAX;                             01164100
       DCHIT:=(DCV=1)=FROM;                                                 01164200
       IRSTACK:= (CONTROLDESC(FROM+2,R) AND                                 01164300
          R.FF=ISTACKV+1) OR                                                01164400
          (CONTROLDESC(FROM+3,R) AND                                        01164500
          R.FF=ISTACKV+1);                                                  01164600
       IF NODUMP THEN ELSE                                                  01164900
       IF MXNOTO AND T=2 AND                                                01165000
      OPERAND(FROM+2,X) AND X=("BBBB"&"BBBB"[1:25:23]) AND                  01166000
       ((CONTROLDESC(FROM+3,X) AND X.FF=0)                                  01167000
                        OR                                                  01168000
       (OPERAND(FROM+3,X) AND X=0 AND CONTROLDESC(FROM+4,X) AND X.FF=0))    01169000
       THEN WRITE(P[DBL],USERSTACKPRT,MX) ELSE                              01170000
       IF (MXNOTO AND T=1 AND L.[2:1]=1 AND                                 01171000
                        OPERAND(FROM+2,X) AND X.[1:37]=0 AND                01172000
                        OPERAND(FROM+3,X) AND X.[1:2] LEQ 2 AND             01173000
                        X.[3:5]=0 AND X.CF NEQ 0)                           01174000
       THEN WRITE(P[DBL],SEGDICT,MX)        ELSE                           01175000
       IF MXNOTO THEN WRITE(P[DBL],MIXUSE,MX,ATP[T]) ELSE                   01176000
       IF MXO AND ((T=1 AND QT) OR MCPSAVE) THEN                            01177000
       DISPLAY(Q.CF,FALSE) ELSE                                             01177100
       IF QT AND T=7 AND INTR AND OPERAND(ADR+(LL:=Q.[8:10]),R)             01178000
      AND R:=R.CF>1023 AND LL LEQ INTMAX THEN BEGIN                         01179000
       WRITE(LINE[*],ITEM,OCTAL(LL),OCTAL(R),OCTAL(R+Q.FF-1));              01180000
       MOVE(INAMS[INAME[LL].CF],LINE[4],INAME[LL].FF);                      01181000
       WRITE(P[DBL],7,LINE[*]); END ELSE                                    01182000
       IF (DCSTACK AND DCHIT) OR                                            01183000
            L.[2:1]=1 AND(T=0 OR T=12) AND MXO AND                          01184000
          ISTACKOK        AND                                               01185000
            IRSTACK                                                         01186000
       THEN WRITE(P[DBL],MCPSTACK) ELSE                                     01190000
       IF FROM=TABLESLOC THEN WRITE(P[DBL],TABLEAREA) ELSE                  01191000
       IF FROM=ADR-2 THEN WRITE(P[DBL],INTARRAY) ELSE                       01192000
       WRITE(P[DBL],MCPUSE,ATP[T]);                                         01193000
         STK:=-1;    BEDDED:=FALSE;                                         01194000
          WHILE BEDSTK GEQ STK:=STK+1 AND NOT BEDDED DO                     01195000
           IF BEDDED:=                                                      01196000
      (LL:=STAX[STK].CF GTR FROM AND N GTR LL) THEN                         01197000
       IF STAX[STK].FF=0 THEN STAX[STK].FF:=FROM+2;                         01198000
    IF NOT BEDDED AND  (IRSTACK  OR                                         01199000
          T=12 OR                                                           01200000
          (DCSTACK AND DCHIT))  THEN                                        01202000
    STAX[MAXSTK:=MAXSTK+1]:=(N-1)&(FROM+2)[18:33:15]                        01203000
                        &1[8:47:1];                                         01204000
       NOTPRINTCALL:=NOTPRINTCALL AND NOT MCPSAVE;                          01208100
       PRINT(FROM,IF( ONEMIX NEQ 0 AND ONEMIX NEQ MX AND MX NEQ 0)          01209000
            OR (NONORMALCODE AND ((T=7 OR T=13) OR                          01210000
                (T=1 AND MX NEQ 0)))                                        01211000
            OR (NOMCPCODE AND (L.[3:12]=64 OR MCPSAVE))                     01212000
                          THEN FROM+2 ELSE N);                              01215000
```

```
        MCPSAVE:=FALSE;                                                    01215100
            END;                                                           01216000
            FROM:=N;                                                       01218000
        END;                                                               01219000
        NOTPRINTCALL:=FALSE;                                               01219100
END DUMP MEMORY AND NOTE STACKS;                                           01220000
ARRAY MCPROG[0:255];                                                       01221000
INTEGER MAXMCPROG,ESP;                                                     01222000
ARRAY OUTERBLOCK[0:0];%                                                    01223000
PROCEDURE SEQUENCE(ARAY,LIM);                                              01224000
    VALUE LIM;                                                             01225000
    ARRAY ARAY[0];                                                         01226000
    INTEGER LIM;                                                           01227000
BEGIN                                                                      01228000
    INTEGER T,L;                                                           01229000
    REAL V;                                                                01230000
    STREAM PROCEDURE MOVE(S,D,D32,M32);                                    01231000
        VALUE D32,M32;                                                     01232000
    BEGIN                                                                  01233000
        SI:=S; DI:=D;                                                      01234000
        D32(DS:=32 WDS);                                                   01235000
        DS:=M32 WDS;                                                       01236000
    END MOVE;                                                              01237000
    T:=LIM;                                                                01238000
    WHILE (T:=T-1)≥0 DO                                                    01239000
    BEGIN                                                                  01240000
        I:=LIM;                                                            01241000
        L:=(V:=ARAY[T]).CF;                                                01242000
        WHILE ARAY[I].CF>L DO                                              01243000
            I:=I-1;                                                        01244000
        IF (L:=I-T)>0 THEN                                                 01245000
        BEGIN                                                              01246000
            MOVE(ARAY[T+1],ARAY[T],L.[37:6],L.[43:5]);                     01247000
            ARAY[I]:=V;                                                    01248000
        END;                                                               01249000
    END;                                                                   01250000
END SEQUENCE;                                                              01251000
INTEGER BADSAVEPRTLOC;                                                     01251500
PROCEDURE GETSORTANDLISTMCPROG(B); VALUE B; BOOLEAN B;                     01252000
BEGIN                                                                      01253000
    REAL R;                                                                01254000
    INTEGER TYP;                                                           01255000
FORMAT TOTALPROCS(///"### A TOTAL OF ",I3," MCP PROCEDURES ",              01256000
        "WERE PRESENT IN MEMORY"),                                         01257000
        BADLOC(//"*** AT LEAST ",I4," MCP PRT LOCATIONS DO NOT ",          01257100
            "CONTAIN PROGRAM OR DATA DESCRIPTORS FOR SAVE CODE ",          01257110
            "AS THEY SHOULD..."),                                          01257120
        F(X8,"PRESENT MCP PROCEDURES"//                                    01258000
            "PRT",X5,"DESCRIPTOR",X8,"THRU"//);                            01259000
    IF NOT B THEN % SORT/*DONT* LIST                                       01259500
    BEGIN                                                                  01259600
    IF DESCRIPTOR(ESPBIT,R,TYP) AND TYP="75" THEN                          01260000
    MCPROG[MAXMCPROG:=0]:=                                                 01261000
        (ESP:=R.CF)&                                                       01262000
        (ESP+R.[8:10]-1)[18:33:15]&                                        01263000
        ESPBIT[8:38:10]                                                    01264000
    ELSE MAXMCPROG:=-1;                                                    01265000
    IF MAXMCPROG NEQ -1 THEN                                               01265900
    OUTERBLOCK[0]:=(PRTMAX+1)&(ESP-1) CTF;                                 01266000
    FOR I:=129 STEP 1 UNTIL PRTMAX DO                                      01267000
```

```
            IF DESCRIPTOR(I,R,TYP) AND                                      01268000
             (TYP="75" OR TYP="77" OR TYP="74") AND                         01269000
               R.CF≠ESP           THEN                                      01270000
            MCPROG[MAXMCPROG:=MAXMCPROG+1]:=                                01271000
                R.CF&                                                       01272000
                (R.CF+R.[8:10]=1)[18:33:15]&                                01273000
                I[8:38:10];                                                 01274000
            SEQUENCE(MCPROG,MAXMCPROG);                                     01275000
            END ELSE % LIST ONLY                                            01275500
            BEGIN                                                           01275600
            NEXTPAGE;                                                       01276000
            WRITE(P,F);                                                     01277000
            SGLTOG:=TRUE;                                                   01277100
            FOR I:=0 STEP 1 UNTIL MAXMCPROG DO                              01278000
            DISPLAY(MCPROG[I].[8:10],TRUE);                                 01279000
            SGLTOG:=FALSE;                                                  01279100
            WRITE(P,TOTALPROCS,MAXMCPROG+1);                                01280000
             IF BADSAVEPRTLOC NEQ 0 THEN WRITE(P,BADLOC,BADSAVEPRTLOC);     01280300
            END;                                                            01280500
        END GET SORT AND LIST PRESENT MCP PROGRAM SEGMENTS;                 01281000
    DEFINE GETANDSORTMCPROG=                                                01281400
           GETSORTANDLISTMCPROG(FALSE)#;                                    01281410
    DEFINE LISTMCPROG=                                                      01281420
           GETSORTANDLISTMCPROG(TRUE)#;                                     01281430
    PROCEDURE DUMPMCPSAVECODE;                                              01281500
    % SEPARATES & IDENTIFIES SAVE PROCEDURES & ARRAYS                       01281510
    BEGIN                                                                   01281520
        BOOLEAN OK;                                                         01281530
        INTEGER T,SIZE;                                                     01281535
        REAL THISPROG;                                                      01281538
        INTEGER I,J,LASTADR,NEXTPROG;                                       01281540
        OK:=SGLTOG:=TRUE;                                                   01281550
        LASTADR:=ESP;                                                       01281560
        FOR I:=0 STEP 1  WHILE OK ANB I LEQ MAXMCPROG DO                    01281570
        IF OK:=(THISPROG:=MCPROG[I]).FF LSS MINLNK THEN                     01281580
        BEGIN                                                               01281590
             DISPLAY(THISPROG,[8:10],TRUE);                                 01281600
             PRINT(THISPROG.CF,LASTADR:=THISPROG.FF+1) ;                    01281610
             IF OK:=(NEXTPROG:=MCPROG[I+1].CF) LSS  MINLNK THEN             01281620
             WHILE NEXTPROG NEQ LASTADR DO                                  01281630
             BEGIN                                                          01281640
                  FOR J:=129 STEP 1 UNTIL PRTMAX DO                         01281650
                  IF PDATADESC(J,T) THEN                                    01281660
                  IF (SIZE:=T.[8:10]) NEQ 0 THEN                            01281670
                  IF (T:=T.CF) = LASTADR THEN                               01281680
                  BEGIN                                                     01281690
                       DISPLAY(J,TRUE);                                     01281700
                       J:=1023;                                             01281710
                       PRINT(T,MIN(LASTADR:=T+SIZE,NEXTPROG));              01281715
                       IF LASTADR GTR NEXTPROG THEN NEXTPROG:=LASTADR;      01281720
                  END;                                                      01281725
                  IF J GTR 1023 THEN ELSE BEGIN % A PRT CELL HAS BEEN ZAPPED01281727
                  PRINT(T:=LASTADR,LASTADR:=NEXTPROG);                      01281728
                  BADSAVEPRTLOC:=BADSAVEPRTLOC+1; END;                      01281729
             END;                                                           01281730
        END;                                                                01281735
        PRINT(LASTADR,MINLNK);                                              01281740
        SGLTOG:=FALSE;                                                      01281745
    END DUMPMCPSAVECODE;                                                    01281750
        ARRAY INTSP[0:INAMESIZE-1];                                         01282000
```

```
INTEGER INTSPMAX;                                                    01283000
PROCEDURE GETSORTANDLISTINTRINSICS;                                  01284000
BEGIN                                                                01285000
    INTEGER IMAX,TYP;                                                01286000
    REAL ADR,R,L;                                                    01287000
    REAL MIX;                                                        01287100
    FORMAT ITEM(X2,A3," =",2(X4,A5)),                                01288000
      BADINTO("*** BAD INTRNSC[0]"),                                 01288100
        F(X8,"PRESENT INTRINSICS"//                                  01289000
              "INDEX",X7,"FROM",X5,"THRU"/);                         01290000
    INTSPMAX:=-1;                                                    01291000
    IF DESCRIPTOR(INTRNSC,ADR,TYP) AND TYP="50" THEN                 01292000
    BEGIN                                                            01293000
        IMAX:=ADR.[8:10]-1;                                          01294000
        ADR:=ADR.CF;                                                 01295000
        IF OPERAND(ADR,R) AND R.[1:38]=0                             01295100
        AND R NEQ 0 AND R LEQ IMAX THEN IMAX:=R ELSE MIX:=-1;        01295200
        IF IMAX>0 THEN                                               01296000
        FOR I:=1 STEP 1 UNTIL IMAX DO                                01297000
        IF OPERAND(ADR+I,R) AND R.CF>1023 THEN                       01298000
        INTSP[INTSPMAX:=INTSPMAX+1]:=                                01299000
          IF I=17 THEN R.CF&(R.CF+3) CTF & I[8:38:10] ELSE           01299100
            R.CF&                                                    01300000
            (IF OPERAND(R.CF+1,L) AND                                01301000
                0<(L:=L.FF) AND                                      01302000
                L≤1023 THEN                                          01303000
            R.CF+L-1 ELSE R.CF)[18:33:15]&                           01304000
            I[8:38:10];                                              01305000
    END;                                                             01306000
    IF INTSPMAX≥0 THEN                                               01307000
    BEGIN                                                            01308000
        NEXTPAGE;                                                    01309000
        DISPLAY(INTRNSC,TRUE);                                       01310000
        IF MIX LSS 0 THEN WRITE(P[DBL],BADINTO);                     01310100
        WRITE(P[DBL],F);                                             01311000
        SEQUENCE(INTSP,INTSPMAX);                                    01312000
        FOR I:=0 STEP 1 UNTIL INTSPMAX DO                            01313000
        BEGIN                                                        01314000
            WRITE(LINE[*],ITEM,OCTAL(L:=INTSP[I].[8:10]),            01315000
                  OCTAL(INTSP[I].CF),                                01316000
                  OCTAL(INTSP[I].FF));                               01317000
            IF L≤INTMAX THEN                                         01318000
            MOVE(INAMS[INAME[L].CF],LINE[4],                         01319000
                        INAME[L].FF);                                01320000
            WRITE(P[DBL],Z,LINE[*]);                                 01321000
        END;                                                         01322000
    END;                                                             01323000
END GET SORT AND LIST PRESENT INTRINSICS;                            01324000
STREAM PROCEDURE MOVC(S,SP,D,DP,W,C);                                01325000
    VALUE SP,DP,W,C;                                                 01326000
BEGIN                                                                01327000
    SI:=S; SI:=SI+SP;                                                01328000
    DI:=D; DI:=DI+DP;                                                01329000
    W(DS:=S CHR); DS:=C CHR;                                         01330000
END MOVC;                                                            01331000
BOOLEAN PROCEDURE WITHIN(ARAY,AMAX,ITEM,RESULT,PLUS);                01332000
    VALUE AMAX,ITEM;                                                 01333000
    ARRAY ARAY[0];                                                   01334000
    INTEGER AMAX,ITEM,RESULT,PLUS;                                   01335000
BEGIN                                                                01336000
```

```
        BOOLEAN FOUND;                                                  01337000
        LABEL EXIT;                                                     01338000
        IF AMAX≥0 THEN                                                  01339000
        FOR I:=0 STEP 1 UNTIL AMAX DO                                   01340000
        IF FOUND:=                                                      01341000
            (ARAY[I].CF≤ITEM AND ITEM ≤ARAY[I].FF) THEN                 01342000
        BEGIN                                                           01343000
            RESULT:=ARAY[I].[8:10];                                     01344000
            PLUS:=ITEM-ARAY[I].CF;                                      01345000
            GO TO EXIT;                                                 01346000
        END;                                                            01347000
        EXIT: WITHIN:=FOUND;                                            01348000
    END WITHIN;                                                         01349000
    ARRAY PROGS[0:3,0:255]; INTEGER PROWS;                              01350000
    INTEGER PROCEDURE SPREAD(AT,F,C);VALUE AT;                          01351000
        INTEGER AT,F,C;             FORWARD;                            01352000
    REAL LOCIRCW;                                                       01352500
    PROCEDURE DUMPSTACK(INX);                                           01353000
        VALUE INX;                                                      01354000
        INTEGER INX;                                                    01355000
    BEGIN                                                               01356000
        INTEGER TOS,BOS,SEG,ADR,H,L,PRO,I;                              01357000
        INTEGER IY;                                                     01357100
        REAL V,R;                                                       01358000
        REAL TOSORIG,X,Y; BOOLEAN WOOPS;                                01359000
        LABEL ERROWT;                                                   01360000
        LABEL DUMPIT;                                                   01360100
        LABEL SKIPEEE;                                                  01361000
        LABEL SQUEEZE,CUTBACK,XIT;                                      01362000
        DEFINE CD=WOOPS #;                                              01363000
        DEFINE DELTA=50#;                                               01364000
        DEFINE MAXSTACKSIZE=512#;% MAX STACK BELOW TOS WE"LL DUMP       01365000
        DEFINE SPEC=BOOLEAN(STAX[INX],[4:1])#;                          01365100
        BOOLEAN NORMAL,FOUND;                                           01366000
        FORMAT HD(X8,"STACK FROM ",A5," DOWN TO ",A5),                  01367000
               TRITEM(A5," = ",A6,2(X1,A5)),                            01368000
               S(X8,"BED[",A3,"] =",A6,X1,A5,X1,A5,", MASK= ",          01369000
                 A6,2(X1,A5)),                                          01370000
               C(X8,3A6,X1,A4,"(",I4,")"," + ",                         01371000
                    A4,"(",I4,":" I1,")");                              01372000
        NEXTITEM;                                                       01373000
        WRITE(P[DBL],HD,                                                01374000
              IF (TOS:=STAX[INX].CF) LSS 127 THEN "*****"               01375000
                ELSE OCTAL(TOS),                                        01376000
              IF (BOS:=STAX[INX].FF) LSS 64 THEN "*****"                01377000
                ELSE OCTAL(BOS));                                       01378000
      IF BOS LSS 64 THEN BOS:=0;                                        01379000
      IF TOS LSS 127 THEN TOS:=0;                                       01380000
    IF TOS=0 THEN GO ERROWT;%E.G. IN EARLY SELECTRUN                    01381000
    IF SPEC THEN GO DUMPIT;                                             01381100
        NORMAL:=BOOLEAN(STAX[INX],[7:1]);                               01382000
        IF INX=0 OR INX>BEDSTK THEN                                     01383000
        BEGIN                                                           01384000
     IF NORMAL THEN                                                     01385000
     BEGIN                                                              01386000
         WRITE(P[DBL],TRITEM,OCTAL(TOS),OCTAL(SPREAD(TOS,H,L)),         01387000
               OCTAL(H),OCTAL(L));                                      01388000
        TOS:=TOS-1;                                                     01389000
    END;                                                                01390000
    IF NOT NORMAL AND BOS=64 THEN                                       01391000
```

```
FOR TOS:=127 STEP -1 UNTIL ACTUALPRIBASE DO                              01392000
BEGIN                                                                    01393000
    WRITE(LINE[*],TRITEM,OCTAL(TOS),OCTAL(SPREAD(TOS,H,L)),              01394000
          OCTAL(H),OCTAL(L));                                            01395000
    MOVE(XNAMS[XNAME[TOS].CF],LINE[5],XNAME[TOS].FF);                    01396000
    WRITE(P[DBL],8,LINE[*]);                                             01397000
END;                                                                     01398000
    IF BOOLEAN(STAX[INX].[6:1]) THEN BEGIN                               01399000
       TOSORIG:=TOS;                                                     01400000
    SQUEEZE:DO TOS:=TOS-1 UNTIL (WOOPS:=TOS LEQ BOS) OR                  01401000
                          CONTROLDESC(TOS,X);                            01402000
    IF WOOPS THEN BEGIN TOS:=TOS+10; GO XIT END;                         01403000
    CUTBACK:IF (Y:=X.FF) GEQ BOS AND                                     01404000
             Y LEQ BOS +1          AND                                   01405000
            CD:=CONTROLDESC(Y,X) THEN                                    01406000
             BEGIN                                                       01407000
                 TOS:=MIN(TOSORIG,TOS+DELTA);                            01408000
                 GO XIT;                                                 01409000
             END                                                         01410000
             ELSE                                                        01411000
             IF Y GTR BOS +1 AND  Y LSS TOS                              01412000
                 AND CD      THEN                                        01413000
                GO CUTBACK;                                              01414000
        GO SQUEEZE;                                                      01415000
    XIT:                                                                 01416000
    END;                                                                 01417000
          IF BOS=64 THEN                                                 01418500
          IF OPERAND(107,V) AND V=(Y:="EEEE"&"EEEE"[1:25:23]) THEN       01418510
          BEGIN                                                          01418520
              FOR I:=108 STEP 1 UNTIL 111 DO                             01418530
              IF OPERAND(I,V) AND V NEQ Y THEN I:=112;                   01418540
              IF I=113 THEN FOR    TOS:=111 STEP -1 UNTIL 108 DO         01418560
              WRITE(P[DBL],TRITEM,OCTAL(TOS),                            01418570
                                  OCTAL(R:=SPREAD(TOS,H,L)),             01418575
                        OCTAL(H),OCTAL(L));                              01418580
          END;                                                           01418590
        SKIPEEE:                                                         01418600
            IF OPERAND(TOS,V) THEN                                       01419000
IF ((V="EEEE"&"EEEE"[1:25:23]) AND NOT NORMAL) OR                        01420000
            ((V="[[[["&"[[[["[1:25:23]) AND                              01421000
                   NORMAL)              THEN                             01422000
            WHILE OPERAND(TOS-1,R) AND R=V DO                            01423000
            TOS:=TOS-1;                                                  01424000
        END                                                              01435000
        ELSE                                                             01436000
        BEGIN                                                            01437000
    WRITE(P[DBL],S,OCTAL(V+STAX[INX].[8:10]),                            01438000
        OCTAL(SPREAD(V+V+V8ED.CF,H,L)),                                  01439000
        OCTAL(H),OCTAL(L),                                               01440000
        OCTAL(R+SPREAD(V+1,H,L)),                                        01441000
        OCTAL(H),OCTAL(L));                                              01442000
        IF OCTAL(R.[30:6])="74" THEN                                     01443000
        IF WITHIN(MCPROG,MAXMCPROG,V:=V.CF,SEG,ADR) THEN                 01444000
        BEGIN                                                            01445000
            WRITE(LINE[*],C,"COMPLE","X SLEE","P AT  ",                  01446000
               OCTAL(SEG),SEG,OCTAL(ADR),ADR,R.[40:2]);                  01447000
            MOVE(NAMS[NAME[SEG].CF],LINE[7],                             01448000
                       NAME[SEG].FF);                                    01449000
            WRITE(P[DBL],15,LINE[*]);                                    01450000
        END                                                              01451000
```

```
            ELSE WRITE(P);                                                  01452000
        END;                                                                01453000
        IF BOS=0 THEN BOS:=MAX(0,TOS-(IF NORMAL THEN MAXSTACKSIZE-1         01454000
                ELSE 128)) ELSE                                             01454100
        IF BOS>TOS THEN BOS:=MAX(0,TOS-1);                                  01455000
        IF TOS-BOS GEQ MAXSTACKSIZE                                         01456000
          THEN BOS:=TOS-MAXSTACKSIZE+1;                                     01457000
DUMPIT:%                                                                    01457100
        IF MYSTACKADR GTR 0 AND MYSTACKSIZE=0 THEN                          01457110
        IF NOT MYSTACKDUMPED THEN                                           01457120
        IF MYSTACKADR LEQ TOS THEN                                          01457130
        IF MYSTACKADR GTR BOS THEN                                          01457140
        MYSTACKDUMPED:=TRUE;                                                01457150
        FOR I:=TOS STEP -1 UNTIL BOS DO                                     01458000
        BEGIN                                                               01459000
            WRITE(P[DBL],TRITEM,OCTAL(I),OCTAL(R+SPREAD(I ,H,L)),           01460000
                  OCTAL(H),                                                 01461000
                  OCTAL(L));                                                01462000
            PRO:=-1;                                                        01463000
            FOUND:=FALSE;                                                   01464000
IF CONTROLDESC(I,X) AND (X.CF GTR 0) AND                                    01465000
            X.FF GTR BOS AND X.FF LSS TOS AND I NEQ IY THEN                 01465100
            BEGIN                                                           01466000
                IF FOUND:=                                                  01467000
                  WITHIN(MCPROG,MAXMCPROG,L.CF,SEG,ADR) AND                 01468000
                TRUE  THEN                                                  01469000
                BEGIN                                                       01470000
                  WRITE(LINE[*],C,"MCP SE","GMENT ","AT    ",               01471000
                 OCTAL(SEG),SEG,OCTAL(ADR),ADR,R.[40:2]);                   01472000
                    MOVE(NAMS[NAME[SEG].CF],LINE[7],                        01473000
                         NAME[SEG].FF);                                     01474000
                END                                                         01475000
                ELSE                                                        01476000
          IF FOUND:=WITHIN(OUTERBLOCK,0,L.CF,SEG,ADR) THEN                  01477000
        WRITE(LINE[*],C,"MCP OU","TER BL","OCK   ",                         01478000
            0,0,OCTAL(L),L,R.[40:2])                                        01479000
          ELSE IF NORMAL THEN                                               01480000
         BEGIN                                                              01480100
            IF FOUND:=WITHIN(INTSP,INTSPMAX,L.CF,SEG,ADR) THEN              01481000
            BEGIN                                                           01482000
                WRITE(LINE[*],C,"INTRIN","SIC NU","MBER  ",                 01483000
                 OCTAL(SEG),SEG,OCTAL(ADR),ADR,R.[40:2]);                   01484000
                 MOVE(INAMS[INAME[SEG].CF],LINE[7],                         01485000
                      INAME[SEG].FF);                                       01486000
            END                                                             01487000
            ELSE                                                            01488000
            WHILE (PRO:=PRO+1)≤PROWS AND NOT FOUND DO                       01490000
            IF FOUND:=WITHIN(PROGS[PRO,*],PROGS[PRO,255],                   01491000
                L,CF,SEG,ADR)                THEN                           01492000
            WRITE(LINE[*],C,"SEGMEN","T NUMB","ER    ",                     01493000
                OCTAL(SEG),SEG,OCTAL(ADR),ADR,R.[40:2]);                    01494000
        END;                                                                01495000
    END ELSE                                                                01495100
            IF NOT FOUND AND NORMAL THEN                                    01496000
            IF OPERAND(I,X) AND X.[1:1]=1 AND X.[3:1]=0                     01497000
            THEN%POSSIBLE CODE (OVERLAID)                                   01498000
            IF FOUND:=( (Y:=X.FF) GTR BOS   AND                             01499000
                        Y LSS I          AND                                01500000
                        (SEG:=X.CF) NEQ 0      AND                          01501000
                        SEG LSS 1023          AND                           01502000
```

```
                    CONTROLDESC(Y,R))                                    01503000
                  THEN%                                                  01504000
                BEGIN                                                    01504100
                IY:=Y;                                                   01504200
                  WRITE(LINE[*],C,"SEGMEN","T NUMB","ER(*) ",            01505000
                        OCTAL(SE$),SEG,OCTAL(ADR*R.CF),ADR,X,[10:2]);    01506000
                END;                                                     01506100
                  IF FOUND THEN                                          01507000
              WRITE(PIDBL,15,LINE[*]);                                   01508000
        END;                                                             01509000
ERROWT:                                                                  01510000
END DUMPING A STACK;                                                     01511000
INTEGER PROCEDURE SPREAD(AT,F,C),                                        01512000
    VALUE AT;                                                            01513000
    INTEGER AT,F,C;                                                      01514000
BEGIN                                                                    01515000
    SPREAD:=CHRS(M[AT.ROW,AT.COL],0,3);                                  01516000
    C:=(F:=CHRS(M[AT.ROW,AT.COL],3,5)).CF;                               01517000
    F:=F.FF;                                                             01518000
END SPREAD;                                                              01519000
PROCEDURE DUMPROGRAMS;                                                   01520000
BEGIN                                                                    01521000
    INTEGER MIX,TYP,PRTH,PRTF,H,F,C,FPB,SD,AIT,S;                        01522000
INTEGER L; ALPHA STARBLANK;                                              01523000
    INTEGER A;                                                           01524000
    REAL JAR00;                                                          01525000
    REAL STARS,JAR0,PRT0;                                                01526000
    BOOLEAN PRTOK;                                                       01526100
    FORMAT HD(A5," = ",A6,2(X1,A5),X2,A1,X1,*A6),                        01527000
     H1(X8,"PROGRAM:",X8,"/",X7,", MIX =",I2);                           01528000
    INTEGER PCOL,SIZ,RO,CL,RP;                                           01529000
    REAL SEGS,SGM;                                                       01530000
    REAL SEG,ADR;                                                        01531000
    ARRAY LSTD[0:3];                                                     01532000
    FORMAT SEGH(X8,"PRESENT PROGRAM SEGMENTS"//                          01533000
            "SEGMENT SIZE  AT    THRU"//),                               01534000
    BADSIZE(8("*"),," INVALID SEGMENT DICTIONARY SIZE ",8("*")),         01535000
    BADSDESC(8("*"),," BAD SEGMENT DICTIONARY DESCRIPTOR ",8("*")),      01536000
        SEGMENT(X3,2(A4,X1),2(X1,A5),X2,A3,A1),                          01537000
            PD(A5," = ",A6,X1,2(A5,X1),", R+",A4,", +",A4);              01538000
    STARS:="****"; STARS.[6:18]:=STARS.[24:18];                          01539000
    IF DESCRIPTOR(JAR,JAR0,TYP) AND TYP="50" THEN                        01540000
    JAR0:=JAR0.CF;                                                       01541000
    IF (PRTOK:=PDATADESC(PRT,PRT0)) THEN                                 01542000
    PRTH:=SPREAD(PRT,PRTF,PRT0);                                         01543000
    IF PRT0>0 THEN                                                       01544000
    FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                    01545000
    IF ONEMIX=0 OR ONEMIX=MIX THEN                                       01545100
IF DESCRIPTOR(PRT0+MIX,R,TYP) AND TYP="50" THEN                          01546000
    BEGIN                                                                01550000
    NEXTPAGE;                                                            01551000
        WRITE(LINE[*],H1,MIX));                                          01552000
        IF                                                               01553000
            IF JAR0=0 THEN FALSE                                         01554000
            ELSE                                                         01555000
            DESCRIPTOR(JAR0+MIX,JAR00,TYP) AND TYP="50"                  01556000
            AND (JAR00<-JAR00.CF) GTR 0          THEN                    01557000
        BEGIN                                                            01558000
            MOVC(M[(A:=JAR00).ROW,A.COL],1,                              01559000
                LINE[2],1,0,7);                                          01560000
```

```
        MOVC(M[(A:=A+1),ROW,A,COL],1,                                   01561000
            LINE[3],1,0,7);                                             01562000
    END                                                                 01563000
    ELSE                                                                01564000
    BEGIN                                                               01565000
        MOVC(STARS,1,LINE[2],1,0,7);                                    01566000
        MOVC(STARS,1,LINE[3],1,0,7);                                    01567000
    END;                                                                01568000
    WRITE(PIOBL],15,LINE[*]);                                           01569000
    NEXTITEM;                                                           01570000
    WRITE(P,HD,                                                         01571000
        OCTAL(PRT),                                                     01572000
        OCTAL(PRTH),OCTAL(PRTF),OCTAL(PRTO)," ",                        01573000
        2,"PRT[*,","*]    ",                                            01574000
        OCTAL(PRTO+MIX),                                                01575000
        OCTAL(SPREAD(PRTO+MIX,F,R)),OCTAL(F),OCTAL(R),                  01576000
        IF F=0 THEN " " ELSE "*",                                       01577000
        3,"PRT[MI","X,*]"," ","CF=R  ",                                 01578000
        OCTAL(R+3),                                                     01579000
        OCTAL(H:=SPREAD(R+3,F,FPB)),OCTAL(F),OCTAL(FPB),                01580000
        IF OCTAL(H.[30:6])="50" THEN " " ELSE "*",                      01581000
        2,"R+3, F","PB    ",                                            01582000
        OCTAL(R+4),                                                     01583000
        OCTAL(H:=SPREAD(R+4,F,SD)),OCTAL(F),OCTAL(SD),                  01584000
        IF OCTAL(H.[30:6])="50" THEN " " ELSE "*",                      01585000
        4,"R+4, S","EGMENT"," DICTI","ONARY ",                          01586000
        OCTAL(R+6),                                                     01587000
        OCTAL(H:=SPREAD(R+6,F,AIT)),OCTAL(F),OCTAL(AIT),                01588000
        IF BOOLEAN(H.[30:1]) THEN " " ELSE "*",                         01589000
        2,"R+6, A","IT    ",                                            01590000
        OCTAL(R+7),                                                     01591000
        OCTAL(H:=SPREAD(R+7,F,C)),OCTAL(F),OCTAL(C),                    01592000
        IF H.[30:2]=3 AND F<R AND C=0 THEN " " ELSE "*",                01593000
        7,"R+7, L","AST MS","CW FOR","WHICH"," MSFF ",                  01594000
          "WAS FA","LSE   ",                                            01595000
        OCTAL(R+8),                                                     01596000
        OCTAL(H:=SPREAD(R+8,F,C)),OCTAL(F),OCTAL(C),                    01597000
        " ",                                                            01598000
        2,"R+10, ","INCW  ",                                            01599000
        OCTAL(R+9),                                                     01600000
        OCTAL(H:=SPREAD(R+9,F,C)),OCTAL(F),OCTAL(C),                    01601000
        IF H=0 AND F=0 THEN " " ELSE "*",                               01602000
        9,"R+11, ","LITERA","L FOR ","LAST C","OMMUNI",                 01603000
          "CATE OR","PROGR","AM REL","EASE  ",                          01604000
        OCTAL(R+10),                                                    01605000
        OCTAL(H:=SPREAD(R+10,S,C)),OCTAL(S),OCTAL(C),                   01606000
        IF OCTAL(H.[30:6])="50" AND 0<S AND S<R AND R=C                 01607000
        THEN " " ELSE "*",                                              01608000
        5,"R+12, ","FF = B","OTTOM ","OF THE"," STACK");                01609000
    IF CONTROLDESC(R+8,H) AND (LOCIRCW:=H.CF) GTR 1024                  01609100
    AND H.FF GTR 1024 THEN ELSE LOCIRCW:=-1;                            01609200
    NEXTITEM;                                                           01610000
    PROGS[PROWS:=0,255]:=PCOL:=-1;                                      01611000
    IF DESCRIPTOR(R+4,SD,TYP) AND TYP="50" THEN                         01612000
    IF OPERAND(SD:=SD,CF,SEGS) AND SEGS.[1:37]=0 THEN                   01613000
    BEGIN                                                               01614000
        WRITE(P,SEGH);                                                  01615000
        FOR SEG:=1 STEP 1 UNTIL SEGS DO                                 01616000
        IF OPERAND(SD+SEG,SGM) AND                                      01617000
            (ADR:=SGM.FF)>1023 THEN                                     01618000
```

```
                BEGIN                                                    01619000
                    SIZ:=                                                01620000
          IF (OPERAND(ADR-1,SIZ)  AND                                    01621000
              SIZ.CF=SEG)                          OR                    01623000
              (OPERAND(ADR-1,SIZ)      AND                               01624000
              OPERAND(ADR-2,H)      AND                                  01625000
              H.[3:6]=7        AND                                       01626000
              SIZ.[8:10]=SGM.CF)        THEN                            01627000
                      SIZ.FF ELSE 0;                                     01628000
                    IF BOOLEAN(SGM.[2:1]) THEN                           01629000
              BEGIN                                                      01630000
                  L:=SGM.CF;                                             01631000
                  STARBLANK:="*";                                        01632000
                  FOR I:=0 STEP 1 UNTIL INTSPMAX DO                      01633000
                  IF INTSP[I].[8:10]=L THEN                              01634000
                  BEGIN                                                  01635000
                          IF L=17 THEN SIZ:=4;                           01635100
                      STARBLANK:=" ";                                    01636000
                      I:=INTSPMAX+2;                                     01637000
                  END;                                                   01638000
                  WRITE(LINE[*],SEGMENT,OCTAL(SEG),OCTAL(SIZ),           01639000
                      OCTAL(ADR),IF SIZ=0 THEN STARS ELSE                01640000
                      OCTAL(ADR+SIZ-1),OCTAL(L),STARBLANK);              01641000
                  IF L LEQ INTMAX THEN                                   01642000
                  MOVE(INAMS[INAME[L].CF],LINE[4],INAME[L].FF);          01643000
                  WRITE(P,7,LINE[*]);                                    01644000
              END                                                        01645000
                    ELSE                                                 01646000
                    BEGIN                                                01647000
                      IF (PCOL:=PCOL+1)=255 THEN                         01648000
                      BEGIN                                              01649000
                          PROGS[PROWS,255]:=254;                         01650000
                          PROWS:=PROWS+1;                                01651000
                          PCOL:=0;                                       01652000
                      END;                                               01653000
                      PROGS[PROWS,PCOL]:=                                01654000
                          ADR&                                           01655000
                          (IF SIZ=0 THEN 0 ELSE ADR+SIZ-1)[18:33:15]&    01656000
                          SEG[8:38:10];                                  01657000
                    END;                                                 01658000
                END;                                                     01659000
                IF (PROGS[PROWS,255]:=PCOL)≥0 THEN                       01660000
                BEGIN                                                    01661000
                    FOR RO:=0 STEP 1 UNTIL PROWS DO                      01662000
                    SEQUENCE(PROGS[RO,*],PROGS[RO,255]);                 01663000
                    SEGS:=PROWS+1;                                       01664000
                    FOR I:=0 STEP 1 UNTIL PROWS DO                       01665000
                    BEGIN                                                01666000
                        SEGS:=SEGS+PROGS[I,255];                         01667000
                        LSTO[I]:=0;                                      01668000
                    END;                                                 01669000
                    FOR SEG:=1 STEP 1 UNTIL SEGS DO                      01670000
                    BEGIN                                                01671000
                        RO:=0;                                           01672000
                        IF PROWS>0 THEN                                  01673000
                        FOR I:=1 STEP 1 UNTIL PROWS DO                   01674000
                        IF LSTD[RO]=255 OR                               01675000
                            (LSTD[I]<255 AND                             01676000
                            PROGS[I,LSTD[I]].CF<PROGS[RO,LSTD[RO]].CF)   01677000
                                                          THEN           01678000
```

```
                    RO:=I;                                                  01679000
                    WRITE(P,SEGMENT,                                        01680000
                          OCTAL((SGM:=PROGS[RO,LSTD[RO]]).[8:10]),          01681000
                          IF SGM.FF=0 THEN STARS ELSE                       01682000
                             OCTAL(SGM.FF-SGM.CF+1),                        01683000
                          OCTAL(SGM.CF),IF SGM.FF=0 THEN                    01684000
                          STARS ELSE OCTAL(SGM.FF));                        01684100
                    LSTD[RO]:=LSTD[RO]+1;                                   01685000
                END;                                                        01686000
            END;                                                            01687000
END                                                                         01688000
    ELSE WRITE(P[DBL],BADSIZE) ELSE WRITE(P[DBL],BADSDESC);                 01689000
     IF (AIT:=MIXSTK[MIX]) = 0 THEN                                         01690000
        STAX[0]:=                                                           01691000
            (R-1)&                                                          01692000
        (S-1)[18:33:15]&                                                    01693000
          1[7:47:1] ELSE                                                    01694000
    BEGIN                                                                   01694100
     IF STAX[AIT].FF=0 THEN STAX[AIT].FF:=S-1;                              01695000
    MIXSTK[MIX]:=0;                                                         01695100
    END;                                                                    01695200
     DUMPSTACK(AIT);                                                        01696000
IF ONEMIX NEQ 0 AND ONEMIX=MIX THEN MIX:=MIXMAX+1; % XIT                    01697000
    END;                                                                    01698000
    IF PRTOK THEN                                                           01698100
    FOR MIX:=1 STEP 1 UNTIL MIXMAX DO                                       01698200
    IF(AIT:=MIXSTK[MIX]) NEQ 0 THEN STAX[AIT].[7:1]:=0;                     01698300
    COMMENT THE ABOVE WILL CAUSE ANY STACKS ASLEEP, ASSOCIATED WITH         01698400
    A MIX INDEX, BUT NOT DUMPED IN THIS PROCEDURE,TO BE DUMPED              01698500
    LATER AS CONTROL STATE STATES.  ;                                       01698600
END DUMPING NORMAL STATE PROGRAM INFO;                                      01699000
PROCEDURE DUMPCONTROLSTACKS;                                                01700000
BEGIN                                                                       01701000
    FORMAT H(X8,"CONTROL STATE STACKS");                                    01702000
    REAL V,R,A;                                                             01703000
     INTEGER INX,I;                                                         01704000
    NEXTPAGE;                                                               01705000
    WRITE(P[DBL],H);                                                        01706000
     STAX[0]:=127&64[18:33:15];                                             01707000
    DUMPSTACK(0);                                                           01708000
    NEXTPAGE;                                                               01709000
     I:=-1;                                                                 01710000
    IF DESCRIPTOR(ISTACK,A,R) AND R="50" THEN                               01711000
    IF (V:=A.[8:10])>0 AND (A:=A.CF)>0 THEN                                 01712000
     I:=A+V-1;                                                              01713000
 IF BEDSTK GTR 0 THEN                                                       01714000
 FOR INX:=1 STEP 1 UNTIL BEDSTK DO                                          01715000
    IF BOOLEAN(STAX[INX].[7:1]) THEN ELSE                                   01716000
BEGIN                                                                       01717000
     R:=STAX[INX].CF;                                                       01718000
     IF R GTR A AND R LEQ I THEN                                            01719000
     BEGIN % ISTACK ASLEEP .                                                01720000
          STAX[INX].FF:=A;                                                  01721000
          NEXTITEM;                                                         01722000
          DISPLAY(ISTACK,TRUE);                                             01723000
          I:=0;% SO DONT DUMP ISTACK TWICE                                  01724000
     END SPECIAL ISTACK STUFF;                                              01725000
     DUMPSTACK(INX);                                                        01726000
NEXTPAGE;                                                                   01727000
END OF DUMPING BEDDED CONTROL STATE STACKS ;                                01728000
```

```
      IF I GTR 0 THEN % ISTACK AWAKE                                    01729000
      BEGIN                                                             01730000
           STAX[0]:=I&A[10:33:15];                                      01731000
           NEXTITEM;                                                    01732000
           DISPLAY(ISTACK,TRUE);                                        01733000
           DUMPSTACK(0);                                                01734000
         NEXTPAGE;                                                      01735000
        END;                                                            01736000
    IF MAXSTK GTR BEDSTK THEN %MORE C.S. STACKS TO DP                   01737000
    FOR INX:=BEDSTK+1 STEP 1 UNTIL MAXSTK DO                            01738000
    BEGIN                                                               01739000
      DUMPSTACK(INX);                                                   01740000
      IF INX LSS MAXSTK THEN NEXTPAGE;                                  01741000
    END;                                                                01742000
    IF NOT MYSTACKDUMPED THEN                                           01742100
    IF MYSTACKADR GTR 0 THEN                                            01742110
    BEGIN % SPECIAL DUMP OF USER SELECTED "STACK" AREA                  01742120
      STAX[0]:=MYSTACKADR & MAX(0,MYSTACKADR+1 -                        01742130
                (IF MYSTACKSIZE NEQ 0 THEN MYSTACKSIZE ELSE 200)) CTF & 01742140
                1[4:47:1];                                              01742150
      DUMPSTACK(0);                                                     01742160
      MYSTACKDUMPED:=TRUE;                                              01742165
    END;                                                                01742170
      END DUMPING CONTROL STATE STACKS;                                 01743000
      DEFINE MAXMESSAGES=100#;                                          01747000
          STREAM PROCEDURE MOVD(S,D,W);                                 01750000
              VALUE W;                                                  01751000
          BEGIN                                                         01752000
              LABEL EXIT;                                               01753000
              SI:=S; DI:=D;                                             01754000
              W(8(IF SC="←" THEN JUMP OUT 2 TO EXIT; DS:=CHR));         01755000
              SI:=SI-1; DI:=DI-1;                                       01756000
              EXIT: DS:=CHR;                                            01757000
      END MOVD;                                                         01757100
      PROCEDURE PRINTQUEUE(PROC,CODE); VALUE PROC,CODE;                 01757300
              INTEGER PROC,CODE;                                        01757310
      BEGIN                                                             01757320
          BOOLEAN B;                                                    01757330
          INTEGER C;                                                    01757332
        REAL A,R,S,T,J,USERCODE;                                        01757335
          FORMAT Q1(A5," = ",2(O,X1),I2,"/",I2,X89),                    01757340
                 Q2(A5," = ",2(O,X1),I2,"/",I2,X1,"USER=",A1,A6,X76)    01757350
                ,Q3(A5,X3,2(2(O,X1),X1),X2,X72)                         01757352
                 ;                                                      01757355
          NEXTITEM;                                                     01757400
          C:=CODE.FF;                                                   01757410
          CODE:=0 & CODE CTC;                                           01757415
          IF CODE=1 THEN                                                01757420
          BEGIN                                                         01757425
              DISPLAY(MIXMASK,FALSE);                                   01757434
              DISPLAY(INFOMASK1,FALSE);                                 01757436
              DISPLAY(INFOMASK2,FALSE);                                 01757440
              DISPLAY(CCMASK1,FALSE);                                   01757450
              DISPLAY(CCMASK2,FALSE);                                   01757455
          END;                                                          01757460
        DISPLAY(PROC,FALSE);                                            01757470
          I:=0;                                                         01757500
          IF OPERAND(PROC,A) AND A.[9:9]=511 AND (R:=A.CF) NEQ PROC     01757505
      THEN                                                              01757510
          DO BEGIN                                                      01757520
```

```
                    IF NOT OPERAND(R,A) THEN I:=MAXMESSAGES+1 ELSE                  01757530
                    CASE CODE OF                                                    01757550
                    BEGIN                                                           01757560
1                      BEGIN % 0(E.G. STATIONMESSAGEHOLDER)                         01757570
2                      WRITE(LINE[*],Q1, OCTAL(R),OCTAL(HIHALF(R)),                 01757580
3                            OCTAL(LOHALF(R)),A.[9:4],A.[14:4]);                    01757590
4   MOVD(M[(R+1 +C),ROW,(R+1 +C),COL],LINE[4],                                      01757600
5                         MIN(11,MAXCOR-11));                                       01757610
6                      WRITE(P,15,LINE[*]);                                         01757620
7                      END; % 0                                                     01757630
8                      BEGIN % 1(LOOKQ)                                             01757700
9                      B:=OPERAND(R+1,USERCODE);                                    01757710
10                     WRITE(LINE[*],Q2,OCTAL(R),OCTAL(HIHALF(R)),                  01757720
11                           OCTAL(LOHALF(R)),A.[9:4],A.[14:4],                     01757730
12                           IF B THEN USERCODE.[6:6] ELSE " ",                     01757740
13                           IF B THEN USERCODE.[12:36] ELSE "      ");             01757750
14                     WRITE(P,15,LINE[*]);                                         01757760
15                     PRINT(R+REAL(B),R+9);                                        01757770
16                     END; % 1                                                     01757780
17                     BEGIN % 2(E.G. DC19Q)                                        01757800
18                     WRITE(LINE[*],Q1,OCTAL(R),OCTAL(HIHALF(R)),                  01757810
19                           OCTAL(LOHALF(R)),A.[9:4],A.[14:4]);                    01757820
20                     WRITE(P,15,LINE[*]);                                         01757830
21                     PRINT(R+1,R+5);                                              01757840
22                     IF OPERAND(R+1,S) AND(S:=S.CF) NEQ R+1 THEN                  01757845
23                     DO BEGIN                                                     01757850
24                          WRITE(LINE[*],Q3,OCTAL(S),OCTAL(HIHALF(S)),             01757855
25                                OCTAL(LOHALF(S)),OCTAL(HIHALF(S+1)),              01757860
26                                OCTAL(LOHALF(S+1)));                              01757865
27                          IF OPERAND(S+1,T) AND T.CF NEQ S+1 THEN                 01757870
28                          MOVD(M[(S+2).ROW,(S+2).COL],LINE[6],                    01757875
29                              MIN(9,MAXCOR-9));                                   01757880
30                          WRITE(P,15,LINE[*]);                                    01757885
31                          END UNTIL NOT OPERAND(S,T) OR                           01757890
32                                    (S:=T.CF)=R+1 OR                              01757895
33                                    (J:=J+1) GTR MAXMESSAGES; J:=0;               01757897
34                        END; % 2                                                  01757900
35                     END CASE;                                                    01757920
36                END UNTIL I:=I+1 GTR MAXMESSAGES OR                               01757950
37                          (R:=A.CF)=PROC OR (A.[9:4]=0);                          01757960
38          END PRINTQUEUE;                                                       01757980
39      PROCEDURE DUMPMCPINFO;                                                    01758000
40      BEGIN                                                                     01758100
41          REAL R,A,N,L,TA,TS,MA,MS,LA,LS,RA,RS,PA,PS;                           01758200
42          INTEGER TYP,S,C;                                                      01758300
43          BOOLEAN TINUOK;                                                       01758400
44          BOOLEAN COMSAVE;                                                      01758500
45          FORMAT SE(A3," = ",2(O,X1),X24,"(",O,X1,O,")"),                       01759000
46      NULLMESSAGES("### NO SPO MESSAGES QUEUED"),                               01760000
47      NULLSLATE("### NO INDEPENDENT-RUNNERS TO BE INITIATED"),                  01761000
48            NOMEMXI(  ,"TOGLE.[15:6]",X2," = ",A2,", NOMEM"),                   01762000
49            TFXI(X8,"BIT ",I2," = ",I1,", ",X28),                               01763000
50              TF(X8,"BIT ",I2," = ",I1,", ",*A6/);                              01764000
51          BOOLEAN STREAM PROCEDURE BITON(W,B);                                  01765000
52              VALUE B;                                                          01766000
53          BEGIN                                                                 01767000
54              SI:=W; SKIP B SB;                                                 01768000
55              IF SB THEN TALLY:=1;                                              01769000
56              BITON:=TALLY;                                                     01770000
57          END;                                                                  01771000
```

```
ARRAY TB[0:143];                                                    01772000
   REAL UA,US,IA,IS,FA,FS;                                          01773000
   FORMAT LUN(A3),                                                  01774000
      LQUEHDR("EL",X2,"IOQUE-INDEX", X2,"DISK-ADDRESS"),            01774100
      BADLQUENTRY(A2,"  # BAD ENTRY IS  ",2(D,X1)),                 01774110
      BADLQUE("## THE LQE IS INCORRECT"),                           01774120
      FLQUE(A2,X6,A3,X8,A1,A6),                                     01774130
      NULLQUE("### NO ENTRIES IN THE LQUE"),                        01774140
       FT(A1,X1,A2,X1,A1,X1,A3,X1,A2,X1,A1,X1,A2,X1,A3),            01775000
       RT(A3,X1,A2,X1,A4,X1,A6,X1,A3),                              01776000
       PT(A1,X1,A1,X1,A5,X1,A5,X1,A6),                              01777000
       IOATH(X8,"FIELDS OF WORDS IN THE I/O ASSIGNMENT TABLES:"//   01778000
       X8,"TINU"/                                                   01779000
       X12,"  [0:3]"/                                               01780000
       X12,"1 [3:5]   HARDWARE UNIT NUMBER"/                        01781000
       X12,"  [8:3]"/                                               01782000
       X12,"2 [11:7]  POWER OF 2"/                                  01783000
       X12,"  [18:6]"/                                              01784000
       X12,"3 [24:1]  IN=0, OUT=1"/                                 01785000
       X12,"  [25:5]"/                                              01786000
       X12,"4 [30:18] UNIT MNEMONIC"//                              01787000
       X8,"RDCTABLE"/                                               01788000
       X12,"  [0:8]"/                                               01789000
       X12,"1 [8:6]   MIX INDEX IF ASSIGNED"/                       01790000
       X12,"2 [14:10] REEL NUMBER"/                                 01791000
       X12,"3 [24:17] CREATION DATE"/                               01792000
       X12,"4 [41:7]  CYCLE"//                                      01793000
       X8,"PRNTABLE"/                                               01794000
       X12,"  [0:1]"/                                               01795000
       X12,"1 [1:1]   IF WRITE RING PRESENT"/                       01796000
       X12,"  [2:13]"/                                              01797000
       X12,"2 [15:15] ADDRESS OF TOP I/O DESCRIPTOR"/               01798000
       X12,"3 [30:18] PHYSICAL REEL NUMBER"//                       01799000
       "LUN",X6,                                                    01800000
       "TINU",X24,                                                  01801000
       "MULTITABLE",X4,                                             01802000
       "LABELTABLE",X4,                                             01803000
       "RDCTABLE",X24,                                              01804000
       "PRNTABLE"/                                                  01805000
       X9,                                                          01806000
       X2,"1",X4,"2",X6,"3",X4,"4",X8,                              01807000
       X14,X14,                                                     01808000
       X4,"1",X2,"2",X4,"3",X6,"4",X8,                              01809000
       X2,"1",X7,"2",X5,"3",X5);                                    01810000
       BOOLEAN PROCEDURE VARIFY(WHAT,A,S,B);                        01811000
          VALUE WHAT,B;                                             01812000
       INTEGER WHAT;                                                01813000
       REAL A,S;                                                    01814000
       BOOLEAN B;                                                   01814500
BEGIN                                                               01815000
          DISPLAY(WHAT,B);                                          01816000
       VARIFY:=                                                     01817000
          DESCRIPTOR(WHAT,A,S) AND                                  01818000
          S="50" AND                                                01819000
          (S:=A.[8:10])>0 AND                                       01820000
          (A:=A.CF)>0 AND                                           01821000
          (A+S-1)<MAXCOR;                                           01822000
END VARIFY;                                                         01823000
DEFINE VERIFY(VERIFY1,VERIFY2,VERIFY3)=                             01823100
          VARIFY(VERIFY1,VERIFY2,VERIFY3,FALSE)#;                   01823300
```

```
        FORMAT IOQSH("FIELDS OF WORDS IN THE I/O-QUEUE TABLES"//                 01824000
            X8,"UNIT"/                                                          01825000
            X12,"  [0:1]"/                                                      01826000
            X12,"1 [1:4]   UNIT TYPE"/                                          01827000
            X12,"2 [5:8]   ERROR FIELD"/                                        01828000
            X12,"3 [13:1]  UNIT NOT READY"/                                     01829000
            X12,"4 [14:1]  ERROR FLAG"/                                         01830000
            X12,"5 [15:1]  WAITING FOR AN I/O CHANNEL"/                         01831000
            X12,"6 [16:2]  I/O IN PROCESS"/                                     01832000
            X12,"7 [18:15] INDEX OF FIRST I/O REQUEST"/                         01833000
            X12,"8 [33:15] INDEX OF LAST I/O REQUEST"//                         01834000
            X8,"LOCATQUE"/                                                      01835000
            X12,"  [0:3]   = 5, DESCRIPTOR BITS"/                              01836000
            X12,"1 [3:5]   MIX INDEX"/                                          01837000
            X12,"  [8:4]"/                                                      01838000
            X12,"2 [12:6]  LOGICAL UNIT NUMBER"/                                01839000
            X12,"3 [18:15] INDEX OF NEXT I/O REQUEST"/                          01840000
            X12,"4 [33:15] ADDRESS OF I/O DESCRIPTOR"//                         01841000
        "LUN/  TINU  ",                                                         01842000
            "UNIT",X31,                                                         01843000
            "IOQUE",X19,                                                        01844000
            "LOCATQUE",X21,                                                     01845000
            "FINALQUE",X11/                                                     01846000
            "INDEX",X7,                                                         01847000
            X2,"1",X2,"2",X3,"3",X1,"4",X1,"5",                                 01848000
                X1,"6",X1,"7",X5,"8",X11,X24,                                   01849000
            X2,"1",X5,"2",X2,"3",X5,"4"//                                       01850000
                                          ),                                    01851000
        IFO(A5,X2,A3),                                                          01852000
              UFO(A1,X1,A2,X1,A3,4(X1,A1),2(X1,A5)),                            01853000
              WFO(0,X1,0),                                                      01854000
              LFO(A1,3(X1,A2),2(X1,A5));                                        01855000
          NEXTPAGE;                                                             01856000
          COMSAVE:=COMMON;                                                      01856100
          COMMON:=BOOLEAN(64); % OCTAL ONLY WHEN CALL PRINT                     01856200
          DISPLAY(TOGLE,FALSE);                                                 01857000
          IF OPERAND(TOGLE,R) THEN                                              01858000
          BEGIN                                                                 01859000
      WRITE(P[DBL],NOMEMXI,OCTAL(R,[15:6]));                                    01860000
      FOR I:=1 STEP 1 UNTIL 11 DO                                               01861000
          IF BITON(R,I) THEN WRITE(P,IF,I,1,-1);                                01862000
          FILL TB[*] WITH                                                       01863000
          "       ","       ","       ",%00%BIT 00(NEVER USED)                 01864000
          "       ","       ","       ",%03%BIT 01                             01865000
          "       ","       ","       ",%06%BIT 02                             01866000
          "       ","       ","       ",%09%BIT 03                             01867000
          "       ","       ","       ",%12%BIT 04                             01868000
          "       ","       ","       ",%15%BIT 05                             01869000
          "       ","       ","       ",%18%BIT 06                             01870000
          "       ","       ","       ",%21%BIT 07                             01871000
          "       ","       ","       ",%24%BIT 08                             01872000
          "       ","       ","       ",%27%BIT 09                             01873000
          "       ","       ","       ",%30%BIT 10                             01874000
          "       ","       ","       ",%33%BIT 11                             01875000
          "DIRECTOR","YTOG    ","       ",%36%BIT 12                           01876000
          "SEPTICTA","NKING   ","       ",%39%BIT 13                           01877000
          "BREAKTOG","        ","       ",%42%BIT 14                           01878000
          "       ","       ","       ",%45%BIT 15                             01879000
          "       ","       ","       ",%48%BIT 16                             01880000
          "       ","       ","       ",%51%BIT 17                             01881000
```

```
                "        ","        ","        ",%54%BIT 18                    01882000
                "        ","        ","        ",%57%BIT 19                    01883000
                "        ","        ","        ",%60%BIT 20                    01884000
                "CDFREE  ","        ","        ",%63%BIT 21                    01885000
                "FINDINGA","DDRESS  ","        ",%66%BIT 22                    01886000
                "SCRATCHD","IRECTORY","READY   ",%69%BIT 23                    01887000
                "MCPFREE ","        ","        ",%72%BIT 24                    01888000
                "INQPTSTO","PPED    ","        ",%75%BIT 25                    01889000
                "DCQPTSTO","PPED    ","        ",%78%BIT 26                    01890000
                "DCWAITIN","G       ","        ",%81%BIT 27                    01891000
                "SMWSTOPP","ED      ","        ",%84%BIT 28                    01892000
                "NINETEEN","NOTREADI","NG      ",%87%BIT 29                    01893000
                "STARTOG ","        ","        ",%90%BIT 30                    01894000
                "EGGSELEC","TSTUPPED","        ",%93%BIT 31                    01895000
                "REMOTELO","GFRLE   ","        ",%96%BIT 32                    01896000
                "SPOEDNUL","LOG     ","        ",%99%BIT 33                    01897000
                "INTFREE ","        ","        ",%102%BIT 34                   01898000
                "QTRDY   ","        ","        ",%105%BIT 35                   01899000
                "NOBACKTA","LK      ","        ",%108%BIT 36                   01900000
                "KEYBOARD","READY   ","        ",%111%BIT 37                   01901000
                "BUMPTUTI","TIME    ","        ",%114%BIT 38                   01902000
                "ABORTABL","E       ","        ",%117%BIT 39                   01903000
                "NSECONDR","EADY    ","        ",%120%BIT 40                   01904000
                "HOLDFREE","        ","        ",%123%BIT 41                   01905000
                "USERDISK","READY   ","        ",%126%BIT 42                   01906000
                "STOREDY ","        ","        ",%129%%BIT 43                  01907000
                "STACKUSE","        ","        ",%132%BIT 44                   01908000
                "SHEETFRE","E       ","        ",%135%BIT 45                   01909000
                "STATUSBI","T       ","        ",%138%BIT 46                   01910000
                "HP2TOG  ","        ","        ",%141%BIT 47                   01911000
        FOR I:=12,13,14,21 STEP 1 UNTIL 47 DO                                  01912000
        BEGIN                                                                  01913000
            WRITE(LINE[*],IFXI,I,REAL(BITON(R,I)));                            01914000
            MOVC(TB[3*I],0,LINE[2],4,3,0);                                     01915000
            WRITE(P[DBL],6,LINE[*]);                                           01916000
        END;                                                                   01917000
            END;                                                               01918000
            DISPLAY(NOPROCESSTOG,FALSE);                                       01918200
            NEXTITEM;                                                          01919000
            DUMPARRAY(TAR);                                                    01919100
            DISPLAY(TOGLE,FALSE); % FOR COMPARISION                            01919200
            NEXTITEM;                                                          01919300
            NEXTPAGE;                                                          01919400
            DISPLAY(OPTION,FALSE);                                             01920000
            IF OPERAND(OPTION,R) THEN                                          01921000
            BEGIN                                                              01922000
                IF BITON(R,1) THEN                                             01924000
                WRITE(P,IF,I,1,-1);                                            01925000
                FILL TB[*] WITH                                                01926000
                    "USEDRA","      ",                                         01927000
                    "USEDRB","      ",                                         01928000
                    "BOJMES","S     ",                                         01929000
                    "EOJMES","S     ",                                         01930000
                    "OPNMES","S     ",                                         01931000
                    "TERMGO","      ",                                         01932000
                    "GIVEDA","TE    ",                                         01933000
                    "GIVETI","ME    ",                                         01934000
                    "SAMEBR","EAKTAP","E     ",                               01935000
                    "AUTOPR","INT   ","      ",                               01936000
                    "CLEARW","RS, ST","OPUNT ",                               01937000
```

```
                "DISCON","DC  NO","TIFYOP",                                  01938000
                "COPNME","SS    ",                                           01939000
                "CLOSEM","ESS   ",                                           01940000
                "ERRORM","SG    ",                                           01941000
                "RETMSG","      ",                                           01942000
                "LIBMSG","      ",                                           01943000
                "SCHEDM","SG    ",                                           01944000
                "SECMSG","      ",                                           01945000
                "DSKTOG","      ",                                           01946000
                "RELTOG","      ",                                           01947000
                "PBDREL","      ",                                           01948000
                "CHECKL","INK   ",                                           01949000
                "DISKMS","G     ",                                           01950000
                "NOT US","ED    ",                                           01950100
                "NOT US","ED    ",                                           01950200
                "USEPBD","      ",                                           01951000
                "SVPBT ","      ",                                           01952000
                "RSTOG ","      ",                                           01953000
                "AUTOUN","LD    ",                                           01954000
                "RNALL ","      ",                                           01955000
                "CODECL","AY    ",                                           01955100
                "COREST","      ",                                           01955200
                "DATACL","AY    ",                                           01955300
                "NOT US","ED    ",                                           01955305
                "NOT US","ED    ",                                           01955310
                "NOT US","ED    ",                                           01955320
                "NOT US","ED    ",                                           01955330
                "NOT US","ED    ",                                           01955340
                "STOPTE","ST    ",                                           01955350
                "PUNCHL","CK    ",                                           01955360
                "CDONLY","      ",                                           01955370
                "PRTONL","Y     ",                                           01955380
                "SEPARA","TE    ",                                           01955390
                "NOT US","ED    ";                                           01955400
            WRITE(P,TF,2,R.[2:1],2,"MOD3IO","S     ");                       01956000
            FOR I:=3 STEP 1 UNTIL 35 DO                                      01960000
        WRITE(P,TF,I,REAL(BITON(R,I)),2,                                     01961000
              TB[A:=28+2×(35-I)],TB[A+1]);                                   01962000
            FOR I:=36 STEP 1 UNTIL 39 DO                                     01963000
            WRITE(P,TF,I,REAL(BITON(R,I)),3,                                 01964000
                TB[A:=16+3×(39-I)],TB[A+1],TB[A+2]);                         01965000
            FOR I:=40 STEP 1 UNTIL 47 DO                                     01966000
            WRITE(P,TF,I,REAL(BITON(R,I)),2,                                 01967000
                TB[A:=2×(47-I)],TB[A+1]);                                    01968000
        END;                                                                 01969000
        NEXTITEM;                                                            01970000
        DISPLAY(MESSAGEHOLDER,FALSE);                                        01971000
        I:=0;                                                                01972000
        IF OPERAND(MESSAGEHOLDER,R) AND (R:=R,CF)≠0 THEN                     01973000
        DO                                                                   01974000
        BEGIN                                                                01975000
            WRITE(LINE[*],ITEM,OCTAL(R),                                     01976000
        OCTAL(HIHALF(R)),OCTAL(LOHALF(R)),"     ");                          01977000
            MOVD(ME(R+1),ROW,(R+1).COL],LINE[4],MIN(9,MAXCOR-R));            01978000
            WRITE(P,15,LINE[*]);                                             01979000
        END                                                                  01980000
        UNTIL                                                                01981000
        (I:=I+1) GEQ MAXMESSAGES OR                                          01982000
            NOT OPERAND(R,R) OR                                              01983000
    (R:=R,FF)=0                                                              01984000
```

```
        ELSE WRITE(P,NULLMESSAGES);                                    01985000
            IF DATACOM THEN                                            01985100
            BEGIN                                                      01985105
                PRINTQUEUE(STATIONMESSAGEHOLDER,0);                    01985120
                PRINTQUEUE(LOOKQ,1);                                   01985150
                PRINTQUEUE(DC19Q,2);                                   01985200
                PRINTQUEUE(PINGQ,0);                                   01985250
                PRINTQUEUE(ILL,(O&3 CTF));                             01985300
                IF RJE THEN                                            01985350
                PRINTQUEUE(RJEWAITQ,0);                                01985375
                NEXTPAGE;                                              01985380
                PRINTARRAY(STATION);                                   01985385
                FOR I:=0 STEP 1 UNTIL 15 DO PAROW(STATION,I);          01985390
                NEXTPAGE;                                              01986000
            END ELSE NEXTITEM;                                         01986100
            DISPLAY(NSLATE,FALSE);                                     01987000
            DISPLAY(LSLATE,FALSE);                                     01988000
            IF VARIFY(SLATE,A,S,TRUE) AND                              01989000
                S.[47:1]=0 AND                                         01990000
                OPERAND(NSLATE,N) AND                                  01991000
                N.[1:8]=0 AND                                          01992000
                N<S AND                                                01993000
                N.[47:1]=0 AND                                         01994000
                OPERAND(LSLATE,L) AND                                  01995000
                L.[1:8]=0 AND                                          01996000
                L<S AND                                                01997000
                L.[47:1]=0 AND                                         01998000
                L≠N THEN                                               01999000
            DO                                                         02000000
            BEGIN                                                      02001000
            N:=REAL(BOOLEAN(N+2) AND BOOLEAN(S-1));                    02002000
                WRITE(LINE[*],SE,OCTAL(N),                             02003000
                   OCTAL(HIHALF(A+N)),OCTAL(LOHALF(A+N)),              02004000
                   OCTAL(HIHALF(A+N+1)),OCTAL(R:=LOHALF (A+N+1)));     02005000
                IF 129≤(R:=R.CF) AND R≤PRTMAX THEN                     02006000
                MOVE(NAMS[NAME[R].CF],LINE[3],                         02007000
                        NAME[R].FF);                                   02008000
                WRITE(P,15,LINE[*]);                                   02009000
             END UNTIL N=L                                             02010000
             ELSE WRITE(P,NULLSLATE);                                  02011000
            NEXTITEM; NEXTPAGE;                                        02012000
            SGLTOG:=TRUE;                                              02012100
    IF TINUOK:=VERIFY(TINU,TA,TS) AND                                  02013000
               VERIFY(MULTITABLE,MA,MS) AND                            02014000
               VERIFY(LABELTABLE,LA,LS) AND                            02015000
               VERIFY(RDCTABLE,RA,RS) AND                              02016000
               VERIFY(PRNTABLE,PA,PS) THEN                             02017000
            BEGIN                                                      02018000
                S:=MAX(TS,MS,LS,RS,PS)-1;                              02019000
                WRITE(P,IOATH);                                        02020000
                FOR I:=0 STEP 1 UNTIL S DO                             02021000
                BEGIN                                                  02022000
                WRITE(TB[*],LUN,OCTAL(I));                             02023000
                IF I<TS THEN                                           02024000
                BEGIN                                                  02025000
                    WRITE(LINE[*],FT,                                  02026000
                        (A:=HIHALF(TA+I)).[24:3],                      02027000
                        OCTAL(A.[27:5]),                               02028000
                        A.[32:3],                                      02029000
                        OCTAL(A.[35:7]),                               02030000
```

```
                OCTAL(A.[42:6]),                                        02031000
                (A:=LOHALF(TA+I)).[24:1],                               02032000
                OCTAL(A.[25:5]),                                        02033000
                A.[30:18]);                                             02034000
            MOVC(LINE[0],0,TB[1],1,2,6);                                02035000
        END;                                                            02036000
        IF I<MS THEN                                                    02037000
            MOVC(M[(MA+I).ROW,(MA+I).COL],0,TB[4],5,1,0);               02038000
        IF I<LS THEN                                                    02039000
            MOVC(M[(LA+I).ROW,(LA+I).COL],0,TB[6],3,1,0);               02040000
            IF I<RS THEN                                                02041000
            BEGIN                                                       02042000
                WRITE(LINE[*],RT,                                       02043000
                OCTAL((A:=HIHALF(RA+I)).[24:8]),                        02044000
                OCTAL(A.[32:6]),                                        02045000
                OCTAL(A.[38:10]),                                       02046000
                OCTAL((A:=LOHALF(RA+I)).[24:17]),                       02047000
                OCTAL(A.[41:7]));                                       02048000
                MOVC(LINE[0],0,TB[8],1,2,6);                            02049000
            END;                                                        02050000
            IF I<PS THEN                                                02051000
            BEGIN                                                       02052000
                WRITE(LINE[*],PT,                                       02053000
                (A:=HIHALF(PA+I)).[24:1],                               02054000
                A.[25:1],                                               02055000
                OCTAL(A.[26:13]),                                       02056000
                OCTAL((A:=LOHALF(PA+I)&A[15:39:9]).[15:15]),            02057000
                OCTAL(A.[30:18]));                                      02058000
                MOVC(LINE[0],0,TB[11],5,2,6);                           02059000
            END;                                                        02060000
            WRITE(P,15,TB[*]);                                          02061000
        END;                                                            02062000
    END;                                                                02063000
    NEXTPAGE;                                                           02064000
    DISPLAY(IOQUESLOTS,FALSE);                                          02064500
    DISPLAY(IOQUEAVAIL,FALSE);                                          02065000
    IF  VERIFY(UNIT,UA,US) AND                                          02066000
        VERIFY(IOQUE,IA,IS) AND                                         02067000
        VERIFY(LOCATQUE,LA,LS) AND                                      02068000
        VERIFY(FINALQUE,FA,FS) THEN                                     02069000
    BEGIN                                                               02070000
        S:=MAX(US,IS,LS,FS)-1;                                          02071000
        WRITE(P,IOQSH);                                                 02072000
        FOR I:=0 STEP 1 UNTIL S DO                                      02073000
        BEGIN                                                           02074000
     WRITE(TB[*],IFU,OCTAL(I),IF TINUOK THEN                            02075000
          LOHALF(TA+I).[30:18] ELSE "***");                             02076000
            IF I<US THEN                                                02077000
            BEGIN                                                       02078000
                WRITE(LINE[*],UFU,                                      02079000
                (A:=HIHALF(UA+I)).[24:1],                               02080000
                OCTAL(A.[25:4]),                                        02081000
                OCTAL(A.[29:8]),                                        02082000
                A.[37:1],                                               02083000
                A.[38:1],                                               02084000
                A.[39:1],                                               02085000
                A.[40:2],                                               02086000
                OCTAL((A:=LOHALF(UA+I)&A[18:42:6]).[18:15]),            02087000
                OCTAL(A.[33:15]));                                      02088000
                MOVC(LINE[0],0,TB[1],4,3,4);                            02089000
```

```
                END;                                                    02090000
                IF I<IS THEN                                            02091000
                BEGIN                                                   02092000
                    WRITE(LINE[*],WFU,                                  02093000
                    OCTAL(HIHALF(IA+I)),                                02094000
                    OCTAL(LOHALF(IA+I)));                               02095000
                    MOVC(LINE[0],0,TB[5],7,2,1);                        02096000
                END;                                                    02097000
                IF I<LS THEN                                            02098000
                BEGIN                                                   02099000
                    WRITE(LINE[*],LFU,                                  02100000
                    (A:=HIHALF(LA+I)).[24:3],                           02101000
                    OCTAL(A,[27:5]),                                    02102000
                    OCTAL(A,[32:4]),                                    02103000
                    OCTAL(A,[36:6]),                                    02104000
                    OCTAL((A:=LOHALF(LA+I)&A[18:42:6]),[18:15]),        02105000
                    OCTAL(A,[33:15]));                                  02106000
                    MOVC(LINE[0],0,TB[8],7,2,6);                        02107000
                END;                                                    02108000
                IF I<FS THEN                                            02109000
                BEGIN                                                   02110000
                    WRITE(LINE[*],WFU,                                  02111000
                    OCTAL(HIHALF(FA+I)),                                02112000
                    OCTAL(LOHALF(FA+I)));                               02113000
                    MOVC(LINE[0],0,TB[12],4,2,1);                       02114000
                END;                                                    02115000
                WRITE(P,15,TB[*]);                                      02116000
            END;                                                        02117000
        NEXTITEM;                                                       02117050
        PRINTARRAY(CHANNEL);                                            02117100
        NEXTITEM;                                                       02117125
        DISPLAY(DISKCUNT,FALSE);                                        02117150
        IF DFX THEN BEGIN                                               02117155
        DISPLAY(EUW,FALSE);                                             02117160
        PRINTARRAY(EUQ);                                                02117200
        END;                                                            02117205
        IF SHAREDISK THEN                                               02117400
        BEGIN                                                           02117410
            NEXTITEM;                                                   02117420
            DISPLAY(LQAVAIL,FALSE);                                     02117430
            IF VARIFY(LQUE,UA,US,TRUE) AND                              02117440
            OPERAND(LQAVAIL,FA) AND FA GEQ 0 AND FA LEQ US THEN         02117450
            IF FA=0 THEN WRITE(P,NULLQUE) ELSE                          02117460
            BEGIN                                                       02117470
                WRITE(P[DBL],LQUEHDR);                                  02117480
                S:=MIN(20,FA-1);                                        02117490
                FOR C:=0 STEP 1 UNTIL S DO                              02117500
                IF NOT OPERAND(UA+C,L) OR (LA:=L.[1:7]) GEQ 64 THEN     02117510
                WRITE(P,BADLQUENTRY,OCTAL(C),OCTAL(HIHALF(L)),          02117520
                      OCTAL(LOHALF(L))) ELSE                            02117530
                WRITE(P,FLQUE,OCTAL(C),OCTAL(LA),L.[8:4],L.[12:36])     02117540
            END ELSE WRITE(P,BADLQUE);                                  02117550
        END LQUE;                                                       02117560
        END;                                                            02118000
        COMMON:=COMSAVE;                                                02118100
        SGLTOG:=FALSE;                                                  02118200
    END DUMPING MCP INFO;                                               02119000
    PROCEDURE DUMPAUXMEM;                                               02119100
    BEGIN                                                               02119110
        FORMAT AUX("DR",A1," MEMORY DUMP"),                             02119120
```

```
            AUXMESS("# PRINTING CONTENTS OF DR",A1,"...");                02119130
        LABEL TRYANOTHER,EXIT;                                            02119140
        BOOLEAN COMSAVE;                                                  02119142
        RELOAD:=TRUE;                                                     02119145
        TRYANOTHER:%                                                      02119150
        IF NOMO THEN GO EXIT;                                             02119160
        NEXTPAGE;                                                         02119170
        LOAD;                                                             02119180
        IF AUXTYPE NEQ 0 THEN % AUXMEM PRESENT                            02119190
        BEGIN                                                             02119200
             NEXTITEM ;                                                   02119210
             WRITE(P,AUX,16+AUXTYPE DIV 4);                               02119220
             WRITE(SPO,AUXMESS,16+AUXTYPE DIV 4);                         02119230
             NEXTITEM;                                                    02119240
             COMSAVE:=COMMON;                                             02119244
             COMMON:=BOOLEAN(512); % ALPHA/OCTAL                          02119245
             PRINT(0,32768);                                              02119250
             COMMON:=COMSAVE;                                             02119255
             GO TRYANOTHER;                                               02119260
        END;                                                              02119270
        EXIT:RELOAD:=FALSE;                                               02119280
   END DUMPAUXMEM;                                                        02119290
   PROCEDURE FIXFID;                                                      02119300
   BEGIN                                                                  02119310
        REAL A;                                                           02119320
        FORMAT FNEWFILE("#NEXT FILE TO BE ANALYZED IS MEMORY/",A1,A6);    02119330
        STREAM PROCEDURE MINUS1(N,A); VALUE N;                            02119340
        BEGIN SI:=LOC N; SI:=SI+5; DI:=A; DI:=DI+5; DS:=3 SUB END;        02119350
        MINUS1(1,TDFID);                                                  02119360
        FILL MDUMP WITH **,TDFID;                                         02119370
        WRITE(SPO,FNEWFILE,TDFID.[6:6],TDFID);                            02119380
   END;                                                                   02119390
   FORMAT COLHDR("B5500 COLLATING SEQUENCE"//X8,"01234567"/),             02119500
          FPUNT("HANG CAUSED BY: ",X24),                                  02119505
          COLINE(X5,I1,X2,X8);                                            02119510
   ARRAY COLLATE[0:7];                                                    02119600
    INTEGER DUMPD;                                                        02120000
  PROCEDURE INITIALIZE;                                                   02120100
    COMMENT:ZERO VARIABLES & ARRAYS IN THE EVENT MORE THAN ONE FILE IS     02120110
    PROCESSED IN A GIVEN RUN;                                             02120115
    BEGIN                                                                 02120120
    INTEGER K,J;                                                          02120125
    LABEL EXIT;                                                           02120130
    IF NOT RELOAD THEN GO EXIT;                                           02120135
    COMMENT:ZERO REALS & INTEGERS;                                        02120140
    I:=                                                                   02120145
    R:=                                                                   02120150
    VJOBNUM:=                                                             02120155
    VBED:=                                                                02120160
    TABLESLOC:=                                                           02120165
    0;                                                                    02120170
    MINLNK:=                                                              02120175
    MINBAD:=                                                              02120180
    MAXBAD:=                                                              02120185
    MAXLNK:=                                                              02120190
    PRTCODE:=                                                             02120195
    0;                                                                    02120200
    MAXMCPROG:=                                                           02120205
    ESP:=                                                                 02120210
    BADSAVEPRTLOC:=                                                       02120215
```

```
        INTSPMAX:=                                                          02120220
        PROWS:=                                                             02120225
        0;                                                                  02120230
        MAXSTK:=                                                            02120235
        BEDSTK:=                                                            02120240
        DUMPD:=                                                             02120245
        0;                                                                  02120300
        COMMENT:MAKE BOOLEANS FALSE;                                        02120400
        LNKSOK:=                                                            02120405
        AVALNKOK:=                                                          02120410
        SOMOKF:=                                                            02120415
        SOMOKB:=                                                            02120420
        NEEDCHECKAVAILNKS:=                                                 02120425
        FALSE;                                                              02120430
        BADCOMMENT:=                                                        02120435
        FALSE;                                                              02120500
        COMMENT:ZERO ARRAYS;                                                02120600
        FOR K:=0 STEP 1 UNTIL MIXMAX DO                                     02120605
            MIXSTK[K]:=                                                     02120610
            0;                                                              02120615
        FOR K:=0 STEP 1 UNTIL INAMESIZE-1 DO                                02120650
            INTSP[K]:=                                                      02120655
            0;                                                              02120660
        EXIT:                                                               02120700
    END INITIALIZE;                                                         02120705
    RESTART:%%                                                              02120900
        INITIALIZE;                                                         02120910
        LOAD;                                                               02121000
        DUMPMCPSPRT;                                                        02122000
        IF NOT COMMON THEN                                                  02122050
        IF DUMPCESSPOOLONLY THEN                                            02122100
        BEGIN                                                               02122200
            DUMPCESSPOOL;                                                   02122300
            GO EOPROG;                                                      02122400
        END;                                                                02122500
        IF NOT COMMON THEN                                                  02123000
        BEGIN                                                               02124000
            CHECKMEMORYLINKS;                                               02125000
            GETSTACKSFROMTHEBED;                                            02126000
            GETANDSORTMCPROG;                                               02126500
        END;                                                                02127000
            IF COMNT THEN BEGIN                                             02128000
                WRITE(P[DBL],STARS);                                        02129000
            IF NOT COMMON THEN                                              02129050
            IF OPERAND(107,PROWS) AND PROWS NEQ "EEEE"&"EEEE"[1:25:23] THEN 02129100
            IF PDATADESC(PUNTER,R) AND R.CF LSS 4096 THEN                   02129110
            BEGIN                                                           02129120
                I:=R.[8:10];                                                02129130
                R:=R.CF;                                                    02129140
                IF PROWS GEQ R AND PROWS LEQ R+I THEN                       02129150
                BEGIN                                                       02129160
                    WRITE(LINE[*],FPUNT);                                   02129170
                    MOVD(M[PROWS.ROW*PROWS.COL],LINE[2],3);                 02129180
                    WRITE(P[DBL],15,LINE[*]);                               02129190
                END;                                                        02129200
            END;                                                            02129210
                IF BADCOMMENT THEN WRITE(P[DBL],COMNTPAR) ELSE              02130000
                BEGIN                                                       02131000
                WRITE(P,10,COMMT[*]);                                       02132000
                MOVE(COMMT[10],COMMT[0],40);                                02133000
```

```
            WRITE(P,10,COMMT[*]);                                          02134000
            END;                                                           02135000
            WRITE(P[PAGE],STARS);                                          02136000
        END;                                                               02137000
         IF NOT COMMON THEN                                                02137100
         BEGIN                                                             02137200
         IF NODUMP THEN ELSE                                               02138000
         IF OPERAND(16,R) AND R GEQ 12 AND R LEQ 15                        02139000
            THEN BEGIN                                                     02140000
         NEEDCHECKAVAILNKS:=NOT AVALNKOK;                                  02141000
PRINT(0,R);                                                                02142000
IF OPERAND(107,PROWS) AND PROWS="EEEE"&"EEEE"[1:25:23]                     02142100
THEN IF OPERAND(R+96,PROWS) AND PROWS NEQ "EEEE"&"EEEE"[1:25:23]           02142120
THEN WRITE(P[DBL],ITEM,OCTAL(R),OCTAL(HIHALF(PROWS:=R+96)),                02142200
            OCTAL(LCHALF(PROWS)));                                         02142250
DONTPRINTLINKS:=TRUE;                                                      02142260
PRINT(R+1,16);                                                             02142300
          END ELSE PRINT(0,36);                                            02143000
  DONTPRINTLINKS:=TRUE;                                                    02143100
  IF NODUMP THEN ELSE                                                      02144000
  IF MINLNK GTR DUMPD:=36 THEN                                             02145000
  BEGIN                                                                    02146000
       DUMPD:=MINLNK;                                                      02147000
       PRINT(36,64); % INTERRUPT CODE                                      02148000
       PRINT(64,112); % INTERRUPT STACK                                    02149000
       PRINT(112,PRTMAX+1); % PRT                                          02149100
       IF NOT NOMCPCODE THEN                                               02149200
       IF (R:=OUTERBLOCK[0])=0 THEN PRINT(36,MINLNK) ELSE                  02149300
       BEGIN                                                               02149350
            PRINT(R.CF,R.FF+ 1); % OUTERBLOCK                              02149400
            DUMPMCPSAVECODE; % SAVE PROCEDURES/ARRAYS                      02149450
       END;                                                                02150000
    END;                                                                   02150050
    DONTPRINTLINKS:=FALSE;                                                 02150075
    IF LNKSOK THEN                                                         02151000
       DUMPMEMORYANDNOTESTACKS(MINLNK,DUMPD:=MAXLNK)                       02152000
    ELSE                                                                   02153000
    BEGIN                                                                  02154000
       IF SOMOKF THEN                                                      02155000
          DUMPMEMORYANDNOTESTACKS(DUMPD,DUMPD:=MINBAD);                    02156000
       IF SOMOKB THEN                                                      02157000
       BEGIN                                                               02158000
          PRINT(DUMPD,MAXBAD);                                             02160000
          DUMPMEMORYANDNOTESTACKS(MAXBAD,DUMPD←MAxLNK);                    02161000
       END;                                                                02162000
    END;                                                                   02163000
        END; % IF NOT COMMON                                               02163100
    PRINT(DUMPD,MAXCOR+1);                                                 02165000
    IF NODUMP THEN                                                         02165100
    BEGIN                                                                  02165110
       NODUMPTOG:=TRUE;                                                    02165120
       COMMON:=COMMON AND NOT BOOLEAN(384);                                02165130
    END;                                                                   02165140
    IF NOT COMMON THEN                                                     02166000
    BEGIN                                                                  02167000
       LISTMCPROG;                                                         02168000
       DUMPMCPINFO;                                                        02169000
       GETSORTANDLISTINTRINSICS;                                           02170000
       DUMPROGRAMS;                                                        02171000
       DUMPCONTROLSTACKS;                                                  02172000
```

```
        PRINTSELECTEDARRAYS;                                                    02172100
        DUMPAUXMEM;                                                             02172200
    END;                                                                        02173000
EOPROG:%                                                                        02173400
    FILL COLLATE[*] WITH                                                        02173500
    OCT0001020304050607,                                                        02173510
    OCT1011121314151617,                                                        02173520
    OCT2021222324252627,                                                        02173530
    OCT3031323334353637,                                                        02173540
    OCT4041424344454647,                                                        02173550
    OCT5051525354555657,                                                        02173560
    OCT6061626364656667,                                                        02173570
    OCT7071727374757677;                                                        02173580
    WRITE(P[PAGE]);                                                             02173590
    WRITE(P,COLHDR);                                                            02173600
    FOR I:=0 STEP 1 UNTIL 7 DO                                                  02173610
    BEGIN                                                                       02173620
        WRITE(LINE[*],COLINE,I);                                                02173630
        MOVE(COLLATE[I],LINE[1],1);                                             02173640
        WRITE(P,2,LINE[*]);                                                     02173650
    END;                                                                        02173660
    IF MULTI AND TDFID.[30:18] GTR 1 THEN                                       02173700
    BEGIN                                                                       02173710
        IF NOT REPEATING THEN                                                   02173720
        BEGIN                                                                   02173730
            REPEATING:=TRUE;                                                    02173740
            SPACE(MDUMP,-1)[XX:XX];                                             02173750
            XX:CLOSE(MDUMP,*);                                                  02173760
        END;                                                                    02173770
        FIXFID;                                                                 02173780
        RELOAD:=TRUE;                                                           02173790
        PRINTCOMMONVALUES;                                                      02173791
        WRITE(P[DBL]); WRITE(P[DBL]);                                           02173792
        WRITE(P,FINI);                                                          02173793
        NEXTPAGE;                                                               02173795
        IF NODUMPTOG THEN BEGIN                                                 02173796
        NODUMPTOG:=FALSE;                                                       02173797
        COMMON:=COMMON OR BOOLEAN(384) END;                                     02173798
        MYSTACKADR:=-1;                                                         02173799
        GO RESTART;                                                             02173800
    END;                                                                        02173810
    END;% % % % % % % % % % % % END INNER BLOCK % % % % % % % % % % %           02174000
PRINTCOMMONVALUES;                                                              02174500
WRITE(P[DBL]);WRITE(P[DBL]);                                                    02175000
WRITE(P,FINI);                                                                  02176000
END.                                                                            02178000
END;END.        LAST CARD ON OCRDING TAPE                                       99999999
      0000000000000000)x≥A40XF
```

LABEL 000000000PRINTER00175100CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/DUMPANL+0000000 OBJECT /READ