LABEL 000000000PRINTER00175099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/PMERGE;END←

OBJECT /READ

SYMBOL/PMERGE

```
COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE                    *     00001000
         * FILE ID: SYMBOL/PMERGE TAPE ID: SYMBOL2/FILE000               *     00001500
         * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION         *     00002000
         * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED               *     00002500
         * EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON             *     00003000
         * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF               *     00003500
         * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232                *     00004000
         *                                                               *     00004500
         * COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION                *     00005000
         *  AA320206   AA386657                                          *;    00005500
%#############################################################################00010000
%                                                                             00020000
%            B-5700 PATCH/MERGE PROGRAM                                       00030000
%                  MARK XV.3.00                                               00040000
%                FEB  7, 1974                                                 00050000
%                                                                            00060000
% DIRECTORY BY SECTION:                                                      00070000
%       0     B-5700 PATCH/MERGE USER MANUAL IN XREF FORM.                   00080000
%       1     OUTER BLOCK DECLARATIONS AND DEFINES.                          00090000
%       2     UTILITY PROCEDURES.                                            00100000
%       3     PROCESS PROCEDURE.                                             00110000
%       4     INPUTPHASE PROCEDURE.                                          00120000
%       5     INPUTPHASE PROCEDURE DRIVER.                                   00130000
%       6     OUTPUTPHASE PROCEDURE.                                         00140000
%       7     OUTPUTPHASE PROCEDURE DRIVER.                                  00150000
%       8     PROCESS PROCEDURE DRIVER.                                      00160000
%       9     OUTER BLOCK DRIVER.                                            00170000
%                                                                            00180000
%############################################################################00190000
%                                                                            00200000
COMMENT DOCUMENT                                                            00210000
*DEFINE 1 = SKIP 2 INDENT 4 INDEX ONLY*                                     00220000
*DEFINE 2 = SKIP 4 UNDERLINE 1 INDENT 2 INDEX ONLY*                         00230000
*DEFINE 3 = SKIP 2 PAR INDENT 2 INDEX ONLY*                                 00240000
*DEFINE 4 = PAGE 1 SKIP 2 UNDERLINE 1 INDENT 2 INDEX ONLY*                  00250000
*DEFINE 5 = PAGE 1 SKIP 2 INDEX*                                            00260000
*PAGE NO SKIP 30 CENTER                                                     00270000
B-5700 PATCH/MERGE                                                          00280000
*SKIP 2 CENTER                                                             00290000
USER GUIDE                                                                  00300000
*SKIP 2 CENTER                                                             00310000
MARK XV.3.00                                                                00320000
*SKIP 30 INDENT 50                                                         00330000
FEB  1, 1974                                                                00340000
*TITLE CENTER                                                              00350000
B-5700 PATCH/MERGE USER GUIDE                                               00360000
*PAGE 1 SKIP 4 INDEX  UNDERLINE 1 CENTER                                    00370000
INTRODUCTION                                                                00380000
*PARAGRAPH 1,70,0                                                          00390000
THIS PROGRAM IS TO BE USED TO CREATE A MASTER PATCH DECK FROM               00400000
THE INDIVIDUAL PATCH DECKS RELEASED BY BURROUGHS ON LIBRARY                 00410000
DUMP TAPES. THIS MASTER PATCH DECK IS THEN USED TO COMPILE THE              00420000
PROGRAM FOR WHICH THE PATCHES WERE RELEASED.                                00430000
 OFFICIAL BURROUGHS PATCH RELEASES ASSUME THAT THIS PATCH/MERGE             00440000
 PROGRAM IS BEING USED TO COMPILE THE PROGRAM BEING PATCHED.                00450000
*PAGE 1 SKIP 2 INDEX  UNDERLINE 1 CENTER                                    00460000
INPUT PHASE                                                                 00470000
*PAR                                                                       00480000
DURING THE INPUT PHASE, PATCH/MERGE READS ALL CARDS                         00490000
INPUTTED TO IT FROM CARDS, IF SPECIFIED, FROM DISK. THE INPUT               00500000
```

```
IS ANALYZED TO SEE IF ALL THE PATCHES ARE FOR THE SAME                    00510000
PROGRAM, THERE ARE AS MANY CARDS IN EACH PATCH AS                         00520000
HAVE BEEN SPECIFIED, AND SO FORTH. IF ANY DISCREPANCIES ARE DETECTED,     00530000
THEN AN APPROPRIATE, SELF-EXPLANATORY ERROR MESSAGE IS EMITTED,           00540000
WHETHER LISTI OR LIST IS SET OR NOT.                                      00550000
*SKIP 1 PAR                                                               00560000
IF LISTI OR LIST IS SET, ALL INPUT WILL BE LISTED AS IT                   00570000
IS ENCOUNTERED. IF THERE ARE DUPLICATE PATCH NUMBERS IN DIFFERENT         00580000
FILES, THEN PATCH/MERGE WILL LIST THE DUPLICATES AND MARK                 00590000
THEM AS "DISCARDED" ON THE LISTING. IF THE DELETE OPTION HAS BEEN         00600000
SPECIFIED, THEN PATCH/MERGE WILL LIST ALL DELETED PATCHES, MARKING THEM   00610000
AS "DELETED" ON THE LISTING.                                              00620000
IF ANY PATCH IS DESIGNATED TO BE DELETED ON A "$@" CARD, BUT              00630000
PATCH/MERGE DOES NOT ENCOUNTER IT, THEN                                   00640000
THE DELETE        WILL BE                                                 00650000
IGNORED AND NO ACTION TAKEN.                                              00660000
ANY PATCHES PRESENT IN THE INPUT WHICH ARE DISCARDED OR DELETED           00670000
SHOULD NOT BE INCLUDED IN THE TOTAL NUMBER OF PATCHES                     00680000
                SPECIFIED ON THE "$." CARD.                               00690000
*SKIP 2 UNDERLINE 1 INDENT 2 INDEX ONLY                                   00700000
INPUT FILES                                                               00710000
*PAR                                                                      00720000
FROM ONE TO THREE FILES CAN BE USED AS INPUT TO PATCH/MERGE:              00730000
*UNDERLINE 1 SKIP 1 TAB 8,30,42,55                                        00740000
 FILE NAME/ FILE TYPE/ PRECEDENCE/ SYMBOL/                                00750000
*SKIP 0                                                                   00760000
    1. CARD                  CARD        PRIMARY        C                00770000
    2. PATCH/<PROGRAM ID>    DISK        SECONDARY      P                00780000
    3. PATCHES/<PROGRAM ID>  DISK        TERTIARY       D                00790000
*SKIP 1 PAR                                                               00800000
WHERE <PROGRAM ID> IS SPECIFIED ON A "$." CARD AND MAY BE UP ANY          00810000
COMBINATION OF ALPHANUMERIC CHARACTERS UP TO SEVEN CHARACTERS IN LENGTH.  00820000
*SKIP 1 PAR                                                               00830000
THE FILES, AS LISTED ABOVE, ARE IN DESCENDING                             00840000
ORDER OF PRECEDENCE. FOR EXAMPLE, IF THERE WERE A PATCH NUMBER            00850000
9 IN FILE CARD AND IN FILE PATCH/<PROGRAM ID>,                            00860000
THEN PATCH/MERGE WOULD USE THE PATCH FROM CARD AND                        00870000
DISCARD THE ONE IN PATCH/<PROGRAM ID>. SUPPOSING THERE WERE THREE         00880000
DIFFERENT PATCHES NUMBERED 27, PATCH/MERGE WOULD AUTOMATICALLY            00890000
USE THE ONE FROM THE FILE CARD AND DISCARD THE OTHER TWO.                 00900000
SEE FURTHER EXAMPLES LATER.                                               00910000
*SKIP 1 PAR                                                               00920000
THE NAME OF THE CARD READER INPUT FILE TO PATCH/MERGE MUST                00930000
BE CARD. (SEE THE EXAMPLES BELOW.) THIS IS THE ONLY                       00940000
PATCH/MERGE INPUT FILE THAT CAN BE LABEL-EQUATED. FOR EXAMPLE:            00950000
*SKIP 1                                                                   00960000
    CC FILE CARD=PMCARD READER                                            00970000
*SKIP 2 PAR                                                               00980000
THE THREE SYMBOLS INDICATED IN THE ABOVE TABLE ARE THOSE PRINTED ON       00990000
THE INPUT, CONFLICTS, OR  OUTPUT LISTINGS IF LISTI, CONFLICTS,            01000000
OR LISTG, RESPECTIVELY, IS SET. THEY INDICATE                             01010000
WHICH FILE A PARTICULAR PATCH IS BEING READ FROM.                         01020000
*PAGE 1 SKIP 2 UNDERLINE 1 INDEX                                          01030000
PATCH/MERGE CONTROL CARDS                                                 01040000
*PAR                                                                      01050000
THERE ARE SIX TYPES OF PATCH/MERGE CONTROL CARDS:                         01060000
*TAB 13,16,29,32                                                          01070000
    1./$@/    4./$-/                                                      01080000
    2./$./    5./$:/                                                      01090000
    3./$*/    6./$#/                                                      01100000
```

```
*2
"$@" CARD                                                                01110000
                                                                         01120000
*PAR                                                                     01130000
THE FIRST CARD (AFTER THE "CC DATA CARD") INPUTTED TO PATCH/MERGE        01140000
 MUST BE EITHER A "$@" CARD OR A "$." CARD. THE "$@" CARD IS OPTIONAL    01150000
AND, IF IT IS NOT SUPPLIED, THE OPTIONS CARD, LIST, AND ZIP              01160000
ARE AUTOMATICALLY SET BY DEFAULT.                                        01170000
*SKIP 1 PAR                                                             01180000
THE "$@" CARDS MUST HAVE THE CHARACTERS                                  01190000
"$@" IN COLUMNS ONE AND TWO. THE REST OF THE CARD MAY CONTAIN            01200000
ANY OPTIONS THE USER WISHES TO USE DURING THE CURRENT RUN OF             01210000
PATCH/MERGE. ANY OPTIONS NOT RECOGNIZED BY PATCH/MERGE                   01220000
WILL BE TREATED AS COMMENTS.                                             01230000
*SKIP 1 PAR                                                             01240000
IN GENERAL, THE "$@" CARD IS EQUIVALENT TO THE $-CARDS USED WITH         01250000
COMPILERS WITH ONE CRITICAL EXCEPTION:                                   01260000
"$@" OPTIONS CAN NEVER BE RESET IN ANY PARTICULAR RUN, THAT IS,          01270000
IF MORE THAN ONE "$@" CARD IS USED,  THEN THE SECOND, THIRD, ETC.        01280000
DOES NOT RESET ANY OPTIONS, BUT ONLY SETS                                01290000
THE OPTIONS SPECIFIED ON THAT PARTICULAR CARD.                           01300000
*SKIP 1 PAR ENDPAGE 7                                                   01310000
THE "$@" OPTIONS AVAILABLE FOR PATCH/MERGE ARE:                          01320000
*SKIP 1 INDENT 4 INDEX ONLY                                             01330000
    CARD                                                                 01340000
*PARAGRAPH 7,60,0 SINGLE                                                 01350000
                ALL INPUT TO PATCH/MERGE WILL BE FROM THE CARD           01360000
                READER ONLY. PATCH/MERGE WILL IGNORE ANY                 01370000
                PATCH FILES THAT MAY BE ON DISK.                         01380000
*1                                                                       01390000
    CONFLICTS                                                            01400000
*PAR SINGLE                                                             01410000
                PATCH/MERGE WILL LIST ON A LINE PRINTER                  01420000
                ANY CONFLICTS AMONG PATCHES AND INDICATE HOW             01430000
                THESE CONFLICTS WERE RESOLVED.                           01440000
 SEE THE SECTION "CONFLICTS (MERGE) PHASE."                              01450000
*SKIP 2 INDENT 4 INDEX ONLY ENDPAGE 18                                   01460000
    DELETE                                                               01470000
*PAR SINGLE                                                             01480000
                PATCH/MERGE WILL PASS OVER ANY PATCHES INPUTTED          01490000
                WHOSE NUMBERS FOLLOW THE DELETE    OPTION,               01500000
 BUT LISTING THE DELETED PATCHES ON THE LINE PRINTER IF LIST OR          01510000
 LISTI IS SET.                                                           01520000
                FOR EXAMPLE, SUPPOSE THE FOLLOWING CARD WERE             01530000
                INPUTTED TO PATCH/MERGE:                                 01540000
*SKIP 1 INDENT 15                                                       01550000
                    $@ DELETE 4,20,103                                   01560000
*SKIP 1 PAR SINGLE                                                      01570000
                PATCH/MERGE WOULD SKIP OVER PATCHES 4, 20, AND           01580000
                103 IF THEY EXISTED IN ANY OF THE PATCH/MERGE INPUT      01590000
                FILES FOR THAT PARTICULAR RUN. IF THE DELETE             01600000
                OPTION IS USED, IT MUST EITHER BE THE LAST OPTION        01610000
                SPECIFIED ON A PARTICULAR "$@" CARD, OR IT MUST          01620000
                BE THE ONLY OPTION SPECIFIED ON A PARTICULAR             01630000
                "$@" CARD. ANY OPTIONS FOLLOWING THE DELETE    OPTION    01640000
                ON ANY PARTICULAR "$@" CARD WILL BE IGNORED.             01650000
*1                                                                       01660000
    FINAL                                                                01670000
*PAR SINGLE                                                             01680000
                THE USE OF THIS OPTION WILL CAUSE THE OPTIONS:           01690000
        MERGE, ZIP, NEW DISK, NONO, AND LIST TO ALL                      01700000
```

```
                    BE SET.                                                  01710000
*1                                                                           01720000
    LIST                                                                     01730000
*PAR SINGLE                                                                  01740000
                    USE OF THIS OPTION WILL CAUSE THE OPTIONS: LISTI,        01750000
                    LISTG, AND CONFLICTS TO BE SET.                          01760000
*1                                                                           01770000
    LISTG                                                                    01780000
*PAR SINGLE                                                                  01790000
                    PATCH/MERGE WILL LIST THE SORTED PATCH DECK GENERATED.   01800000
*1                                                                           01810000
    LISTI                                                                    01820000
*PAR SINGLE                                                                  01830000
                    PATCH/MERGE WILL LIST ALL INPUT.                         01840000
*1                                                                           01850000
    MERGE                                                                    01860000
*PAR SINGLE                                                                  01870000
IF AT LEAST ONE OF THE DISK FILES EXIST, INPUT TO PATCH/MERGE WILL BE         01880000
 BOTH FROM CARDS AND FROM DISK. PATCH/MERGE                                  01890000
                    WILL AUTOMATICALLY CHECK IF EITHER OR BOTH OF            01900000
                    THE FILES "PATCH/<PROGRAM ID>" OR                        01910000
                    "PATCHES/<PROGRAM ID>" ARE ON DISK. IF ONE OR            01920000
                    BOTH IS ON DISK, THEN PATCH/MERGE WILL AUTOMATICALLY     01930000
                    MERGE THE THREE INPUT FILES. IF NEITHER IS ON DISK,      01940000
                    THEN PATCH/MERGE WILL USE THE FILE CARD FOR              01950000
                    INPUT, ONLY.                                             01960000
*1                                                                           01970000
    NEW DISK                                                                 01980000
*PAR SINGLE                                                                  01990000
      PATCH/MERGE WILL CREATE A NEW, UNSORTED MASTER                         02000000
                    FILE ON DISK CONSISTING OF ALL PATCHES INPUTTED          02010000
                    WHICH WERE USED IN GENERATING THE SORTED PATCH DECK.     02020000
 PATCH/MERGE WILL NAME THIS FILE "PATCHES/<PROGRAM ID>".                     02030000
                         IF A PREVIOUS MASTER FILE EXISTED,                  02040000
                     IT WILL AUTOMATICALLY BE                                02050000
                    PURGED BEFORE THE NEW FILE IS CLOSED,                    02060000
AS WILL THE FILE PATCH/<PROGRAM ID>.                                         02070000
HOWEVER, IF ANY ERRORS ARE DETECTED BY PATCH/MERGE DURING                    02080000
 THE INPUT PHASE, THEN NO NEW DISK FILE WILL BE CREATED AND NO               02090000
DISK FILES WILL     BE PURGED, EVEN THOUGH NEW DISK IS SET.                  02100000
*SKIP 2 INDENT 4 INDEX ONLY ENDPAGE 8                                        02110000
    NONO                                                                     02120000
*PAR SINGLE                                                                  02130000
                    PATCH/MERGE WILL NOT ATTEMPT TO PUT THE PATCH NUMBER     02140000
                    ON EACH GENERATED PATCH CARD. IF THIS OPTION IS          02150000
                    *NOT* SET, THEN ON EACH PATCH CARD GENERATED THAT HAS    02160000
                    COLUMNS 68-72 BLANK, PATCH/MERGE WILL AUTOMATICALLY      02170000
                    INSERT THE PATCH NUMBER OF THE PATCH TO WHICH THE        02180000
                    CARD BELONGS.                                            02190000
*SKIP 2 INDENT 4 INDEX ONLY ENDPAGE 6                                        02200000
    PUNCHG                                                                   02210000
*PAR SINGLE                                                                  02220000
                    PATCH/MERGE WILL PUNCH OUT ON AN ON-LINE CARD            02230000
                    PUNCH THE GENERATED PATCH DECK, EXCLUSIVE OF             02240000
                    MCP AND PATCH/MERGE CONTROL CARDS.                       02250000
*1                                                                           02260000
    PUNCHI                                                                   02270000
*PAR SINGLE                                                                  02280000
                    PATCH/MERGE WILL PUNCH OUT ON AN ON-LINE CARD            02290000
                    PUNCH A DECK CONSISTING OF ALL INPUT PATCHES             02300000
```

```
                USED IN GENERATING THE SORTED PATCH DECK.                    02310000
* SKIP 2 INDENT 4 INDEX ONLY                                                 02320000
   ZIP                                                                       02330000
*PAR SINGLE                                                                  02340000
                PATCH/MERGE WILL AUTOMATICALLY ZIP THE GENERATED             02350000
                PATCH DECK.                                                  02360000
*1                                                                           02361000
NEWFILE                                                                      02361100
* PAR SINGLE                                                                02361150
PATCH/MERGE WILL CREATE A NEW UNSORTED FILE                                  02361200
  "PATCH/<PROGRAM ID>" ON DISK CONSISTING OF                                 02361300
  ALL THE PATCHES INPUTTED VIA THE CARD READER.                              02361400
IF A FILE EXISTS, IT WILL BE PURGED AFTER IT HAS                             02361500
BEEN PROCESSED AND A NEW FILE WILL BE CREATED                                02361600
PROVIDED THE FOLLOWING CONDITIONS ARE MET.  FIRST,                           02361700
 "NEW" OPTION MUST BE SET, SECOND, THERE MUST BE                             02361800
CARD INPUT, AND THIRD, ALL INPUT DISK AND CARD MUST BE                       02361900
ERROR FREE.                                                                  02362000
*2                                                                           02370000
"$." CARD                                                                    02380000
*PARAGRAPH 1,70,0                                                           02390000
THE "$." CARD MUST HAVE THE CHARACTERS                                       02400000
 "$." IN COLUMNS ONE AND TWO.  THE REST OF THE CARD MUST CONTAIN THE         02410000
 NUMBER OF PATCHES AND THE IDENTIFICATION OF THE PROGRAM WHICH THE           02420000
 PATCHES ARE FOR. THE FORMAT OF THIS CARD IS:                                02430000
*SKIP 1 INDENT 5                                                             02440000
     $, <NUMBER OF PATCHES> FOR <PROGRAM ID> <COMMENTS>                      02450000
*SKIP 1 PAR                                                                  02460000
WHERE <NUMBER OF PATCHES> MUST BE AN INTEGER BETWEEN ZERO AND 999,           02470000
INCLUSIVE, AND <PROGRAM ID> MAY BE ANY IDENTIFIER CONTAINING                 02480000
UP TO SEVEN ALPHANUMERIC CHARACTERS                                          02490000
IN LENGTH. IF <NUMBER OF PATCHES> IS ZERO, THEN  PATCH/MERGE WILL            02500000
AUTOMATICALLY CHANGE IT TO ONE.                                              02510000
*2                                                                           02520000
"$*" CARD                                                                    02530000
*PAR                                                                        02540000
PATCH/MERGE CREATES AND ZIPS A CONTROL DECK CONTAINING THE MERGED            02550000
PATCHES. THE MCP CONTROL CARDS USED IN THIS CONTROL DECK MUST BE             02560000
SPECIFIED BY CARDS WHICH HAVE THE CHARACTERS "$*" IN COLUMNS ONE AND         02570000
TWO. (PATCH/MERGE WILL AUTOMATICALLY SUBSTITUTE A QUESTION MARK AND          02580000
 A BLANK IN COLUMNS ONE AND TWO FOR THE "$*" SUPPLIED).                      02590000
      THE REMAINING SEVENTY COLUMNS CAN CONTAIN ANY CONTROL CARD             02600000
INFORMATION DESIRED.  THIS TYPE OF CARD MUST PRECEDE ALL PATCH/MERGE         02610000
CONTROL CARDS EXCEPT THE "$." OR "@" CARD.                                   02620000
 ANY INPUT TO PATCH/MERGE MUST CONTAIN AT LLAST                              02630000
 ONE "$*" CARD.                                                             02640000
*SKIP 1 PAR                                                                  02650000
          NOTE:THERE MUST BE AT LEAST ONE                                    02660000
PSEUDOREADER IN USE OR THE CONTROL DECK WILL NOT BE SCHEDULED BY THE         02670000
MCP.                                                                         02680000
*2                                                                           02690000
"$=" CARD                                                                    02700000
*PAR                                                                        02710000
PATCHES WHICH ARE NOT TO BE MERGED (E.G. DOLLAR CARDS) MAY BE                02720000
INCLUDED AS THE FIRST PATCH DECK.  THE HEADER CARD FOR THIS DECK             02730000
MUST HAVE THE CHARACTERS "$=" IN COLUMNS ONE AND TWO.  THE REMAINDER         02740000
OF THE CARD MAY CONTAIN COMMENTS.  THE CARDS IN THIS DECK WILL BE            02750000
THE FIRST CARDS OF THE MASTER PATCH DECK.                                    02760000
*2                                                                           02770000
"$!" CARD                                                                    02780000
```

```
*PAR                                                                    02790000
ACTUALLY, THE "$:" CARD IS NOT A CONTROL CARD, BUT A COMMENT            02800000
CARD. IT IS USED TO INTERSPERSE COMMENTS WITH PATCH/MERGE INPUT.        02810000
IN GENERAL, ALL "$:" CARDS WILL BE LISTED IF LISTI IS SET,              02820000
BUT WILL OTHERWISE BE IGNORED BY PATCH/MERGE. IF THE NEW DISK OPTION    02830000
HAS BEEN SET, THEN PATCH/MERGE WILL INCLUDE THESE COMMENT CARDS         02840000
IN THE NEW DISK FILE.                                                   02850000
* SKIP 4 UNDERLINE 1 INDENT 2 INDEX ONLY ENDPAGE 24                     02860000
"$#" CARD                                                               02870000
*PAR                                                                    02880000
 INDIVIDUAL PATCH DECKS MUST BEGIN WITH A CARD WHICH HAS "$#" IN        02890000
 COLUMNS ONE AND TWO AND THE PATCH IDENTIFICATION IN COLUMNS THREE       02900000
 THROUGH SEVENTY-TWO.  THE STANDARD FORMAT FOR THIS CARD IS:            02910000
*SKIP 1                                                                 02920000
     $# PATCH NUMBER <INTEGER> FOR <PROGRAM ID> CONTAINS <INTEGER> CARDS02930000
*SKIP 1 PAR                                                             02940000
WHERE THE <INTEGER> SPECIFYING PATCH NUMBER MUST BE AN INTEGER BETWEEN  02950000
ZERO AND 999, INCLUSIVE,  AND THE <INTEGER> SPECIFYING THE NUMBER       02960000
 OF CARDS                                                               02970000
MUST, MINIMALLY, BE ONE.                                                02980000
*SKIP 1 PAR                                                            02990000
 AN ERROR WILL BE INDICATED IF THE PATCH DECKS ARE NOT IN ASCENDING     03000000
 ORDER, IF THE PROGRAM IDENTIFICATION DOES NOT AGREE WITH THAT OF       03010000
 THE "$." CARD, IF THE NUMBER OF CARDS PRESENT DOES NOT                 03020000
 AGREE WITH THE NUMBER OF CARDS GIVEN ON THIS CARD, ETC.  ALL PATCHES   03030000
 RELEASED BY BURROUGHS HAVE THE PROPER HEADER CARD AS THE FIRST         03040000
CARD OF THE DECK.                                                       03050000
*PAGE 1 SKIP 2 INDEX UNDERLINE 1 CENTER                                 03060000
CONFLICTS (MERGE) PHASE                                                 03070000
*PAR                                                                    03080000
   BESIDES MERGING PATCH DECKS, PATCH/MERGE RESOLVES CONFLICTS AMONG    03090000
PATCH DECKS.  IF TWO OR MORE PATCH DECKS HAVE IDENTICAL SEQUENCE        03100000
NUMBERS, THEN PATCH/MERGE WILL DISCARD ALL BUT THE ONE FROM THE DECK    03110000
WITH THE HIGHEST PATCH NUMBER.                                          03120000
*SKIP 2        UNDERLINE 1      INDENT 2 INDEX ONLY                     03130000
"$VOID" CARDS                                                           03140000
*PAR                                                                    03150000
                                IF A VOID CARD IS TO BE DISCARDED,      03160000
SPECIAL ACTION IS TAKEN SO THAT THE VOIDING IS ACCOMPLISHED.  WHEN      03170000
A VOID CARD IS ENCOUNTERED, CARDS IN THE VOIDED RANGE ARE DISCARDED     03180000
FROM THE PATCH DECK IN WHICH THE VOID WAS FOUND, AND IN ALL DECKS OF    03190000
LOWER PATCH NUMBER.  VOID CARDS IN THE DECKS OF LOWER PATCH NUMBER      03200000
ARE ALSO SO HANDLED, WHEN THEY ARE ENCOUNTERED WHILE DISCARDING.  IF    03210000
THERE ARE CARDS IN DECKS WITH HIGHER PATCH NUMBERS THAN THE VOID        03220000
CARD WHICH FALL INTO THE VOIDING RANGE, THE NECESSARY VOID CARDS ARE    03230000
GENERATED SO THAT THE TAPE AREA IS VOIDED BUT THE PATCH CARDS FROM      03240000
THOSE HIGHER NUMBERED DECKS ARE NOT.                                    03250000
*SKIP 2 PAR                                                            03260000
                                NOTE: FOR THE PURPOSES OF THIS          03270000
PROGRAM, "VOID" CARDS MUST HAVE THEIR "$" IN COLUMN ONE, AND THE        03280000
VOIDING SEQUENCE AND RANGE MUST BE EIGHT DIGITS IN LENGTH.              03290000
THIS IS GOOD PRACTICE, ANYWAY.                                          03300000
*SKIP 2 UNDERLINE 1 INDENT 2 INDEX ONLY                                 03310000
"$VOIDT" CARDS                                                          03320000
*PAR                                                                    03330000
IN THE MERGE PHASE, "$VOIDT" CARDS ARE TREATED THE SAME AS "$VOID"      03340000
CARDS WITH RESPECT TO PATCHES LOWER IN NUMBER THAN THE PATCH            03350000
CONTAINING THE "$VOIDT" CARD. CARDS IN THE "$VOIDT" RANGE WHICH ARE     03360000
CONTAINED IN PATCHES WITH A NUMBER THE SAME AS OR HIGHER THAN THE PATCH 03370000
CONTAINING THE "$VOIDT" CARD ARE SIMPLY INSERTED IN THE GENERATED       03380000
```

```
PATCH DECK ALONG WITH THE "$VOIDT" CARD.                                    03390000
*SKIP 2 UNDERLINE 1 INDENT 2 INDEX ONLY                                     03400000
WORK FILES                                                                  03410000
*PAR                                                                        03420000
   THE WORK FILES USED BY PATCH/MERGE HAVE TWENTY ROWS OF 300 RECORDS       03430000
EACH.  THE NUMBER OF ROWS MAY BE CHANGED BY SPECIFYING THE DESIRED          03440000
VALUE ON A "COMMON" CONTROL CARD WHEN EXECUTING PATCH/MERGE.                03450000
*PAGE 1 SKIP 2 INDEX UNDERLINE 1 CENTER                                     03460000
DECK SET UP AND EXAMPLES.                                                   03470000
*2                                                                          03480000
INPUT FROM CARDS ONLY                                                       03490000
*PAR                                                                        03500000
LET US ASSUME THAT WE WANT TO COMPILE                                       03510000
A PROGRAM WHICH HAS A SOURCE FILE, SOURCE/PRGXII, ON TAPE.                  03520000
WE WANT TO ADD THREE PATCHES TO THIS FILE, AND WE HAVE THE                  03530000
THREE PATCHES PUNCHED ON CARDS SUITABLE FOR INPUT TO PATCH/MERGE.           03540000
WE COULD USE THE FOLLOWING DECK SET UP:                                     03550000
*SKIP 2  INDENT 5 SINGLE                                                    03560000
*INVALID ALL 3                                                              03570000
    EXECUTE PATCH/MERGE                                                     03580000
    DATA CARD                                                               03590000
*INDENT 5 SINGLE                                                            03600000
     $, 3 PATCHES FOR PRGXII                                                03610000
     $*COMPILE OBJECT/TST WITH ALGOL FOR LIBRARY                            03620000
     $*ALGOL FILE TAPE=SOURCE/PRGXII TAPE                                   03630000
     $*DATA CARD                                                            03640000
     $-                                                                     03650000
     $TAPE LIST                                                             03660000
     $#PATCH NUMBER 1 FOR PRGXII CONTAINS 20 CARDS                          03670000
      .     .     .                                                        03680000
      .     .     .                                                        03690000
     <PATCH DECK>                                                           03700000
      .     .     .                                                        03710000
      .     .     .                                                        03720000
     $#PATCH NUMBER 2 FOR PRGXII CONTAINS 4 CARDS                           03730000
      .     .     .                                                        03740000
      .     .     .                                                        03750000
     <PATCH DECK>                                                           03760000
      .     .     .                                                        03770000
      .     .     .                                                        03780000
     $#PATCH NUMBER 3 FOR PRGXII CONTAINS 14 CARDS                          03790000
      .     .     .                                                        03800000
      .     .     .                                                        03810000
     <PATCH DECK>                                                           03820000
      .     .     .                                                        03830000
      .     .     .                                                        03840000
*SINGLE  INDENT 5                                                           03850000
*INVALID 3                                                                  03860000
    END.                                                                    03870000
*SKIP 2 PAR                                                                 03880000
SINCE THE "$@" CARD IS MISSING FROM THE ABOVE INPUT DECK, THE               03890000
OPTIONS CARD,LIST, AND ZIP WILL AUTOMATICALLY BE SET BY DEFAULT.            03900000
THEREFORE, EVEN IF ONE OR BOTH OF THE FILES "PATCH/PRGXII" OR               03910000
"PATCHES/PRGXII" ARE ON DISK, PATCH/MERGE WILL NOT ATTEMPT TO               03920000
USE THEM AS INPUT, BUT USE ONLY THOSE PATCHES INCLUDED IN THE CARD DECK.    03930000
*SKIP 2 PAR                                                                 03940000
NOTE THAT, ALTHOUGH PATCH/MERGE DOES NOT REQUIRE QUOTES AROUND A            03950000
<PROGRAM ID>              WHICH HAS A SPECIAL CHARACTER(S),                 03960000
ANY MCP CONTROL CARDS USED DO.                                              03970000
*PAGE 1 SKIP 2 PAR                                                         03980000
```

```
   THE FOLLOWING IS AN EXAMPLE OF A DECK WHICH WOULD COMPILE THE MCP     03990000
FROM THE TWO PATCH DECKS GIVEN:                                         04000000
*SKIP 2  INDENT 5 SINGLE                                                04010000
*INVALID ALL 1                                                          04020000
 EXECUTE PATCH/MERGE                                                    04030000
 FILE CARD=PMCARD                                                       04040000
 DATA PMCARD                                                            04050000
*INDENT 5 SINGLE                                                        04060000
      $@CARD LISTI LISTG NONO                                           04070000
      $,2 PATCHES FOR MCP                                               04080000
      $*EXECUTE ESPOL/DISK TO COMPILE THE MCP.                          04090000
      $* FILE DISK=MCP/DISK.                                           04100000
      $*FILE TAPE=SYMBOL/DCESPSY.                                       04110000
      $*FILE LINE=LINE BACK UP DISK                                     04120000
      $*FILE CARD=MCPDECK                                               04130000
      $*DATA MCPDECK                                                    04140000
      $=DOLLAR CARDS NECESSARY FOR MCP COMPILATION                      04150000
      $SET DATACOM=FALSE,DEBUGGING=FALSE,DUMP=FALSE                     04160000
      $SET CHECKLINK=TRUE,BREAKOUT=TRUE,DISKLOG=FALSE                   04170000
      $SET DFX=FALSE,INQUIRY=TRUE,AUXMEM=FALSE                          04180000
      $SET SHAREDISK=FALSE,B6500LOAD=TRUE,DCLOG=FALSE                   04190000
      $SET DCSPO=FALSE,SAVERESULTS=FALSE,STATISTICS=FALSE               04200000
      $TAPE LIST PRT                                                    04210000
      $#PATCH NUMBER 1 FOR MCP CONTAINS 40 CARDS                        04220000
          .       .       .                                             04230000
          .       .       .                                             04240000
          .       .       .                                             04250000
              <PATCH DECK>                                              04260000
          .       .       .                                             04270000
          .       .       .                                             04280000
          .       .       .                                             04290000
      $#PATCH NUMBER 2 FOR MCP CONTAINS 10 CARDS                        04300000
          .       .       .                                             04310000
          .       .       .                                             04320000
          .       .       .                                             04330000
              <PATCH DECK>                                              04340000
          .       .       .                                             04350000
          .       .       .                                             04360000
          .       .       .                                             04370000
      *SINGLE  INDENT 5                                                 04380000
      *INVALID 1                                                        04390000
       END                                                              04400000
      *SKIP 2 PAR                                                       04410000
      THE ABOVE DECK WOULD CAUSE PATCH/MERGE TO READ ONLY THE CARDS,    04420000
            IGNORING ANY DISK FILES WHICH MAY BE ON DISK, SINCE THE OPTION 04430000
      CARD IS SET. THE INPUT WOULD BE LISTED BECAUSE LISTI IS SET, AND THE 04440000
      GENERATED PATCH DECK WOULD ALSO BE LISTED BECAUSE LISTG IS SET.   04450000
      IF ANY CONFLICTS EXIST BETWEEN THE TWO PATCHES INPUTTED, THEY WOULD 04460000
      NOT BE LISTED BECAUSE THE OPTION CONFLICTS IS NOT SET.            04470000
       PATCH NUMBERS WOULD NOT BE PLACED ONTO THE GENERATED OUTPUT      04480000
       BECAUSE NONO IS SET.                                             04490000
                                                        LIKEWISE,       04500000
      THE GENERATED DECK WOULD NOT BE ZIPPED BECAUSE THE OPTION ZIP     04510000
       IS NOT SET.                                                      04520000
      *4                                                                04530000
      INPUT FROM CARDS AND DISK                                         04540000
      *PAR                                                              04550000
      PATCH/MERGE WILL ACCEPT INPUT FROM ONE, TWO, OR THREE FILES:      04560000
      ONE CARD FILE AND TWO DISK FILES. ALL PATCH/MERGE CONTROL CARDS   04570000
      INCLUDING:                                                        04580000
```

```
*SKIP 1 INDENT 5                                                    04590000
 1. $@                                                              04600000
 2. $.                                                              04610000
 3. $*                                                              04620000
 4. $=                                                              04630000
*3                                                                  04640000
  MUST BE ENTERED FROM A CARD READER FILE WHOSE NAME IS CARD.       04650000
IN OTHER WORDS, ALL PATCH/MERGE AND $ CONTROL CARDS                 04660000
UP TO BUT NOT INCLUDING THE FIRST "$#" CARD MUST BE ENTERED FROM    04670000
A CARD READER FILE LABELED CARD.                                    04680000
*3                                                                  04690000
FOR EXAMPLE, ASSUMING THAT AT LEAST ONE DISK FILE WITH PATCHES      04700000
IN IT WERE PRESENT ON DISK, ONE MIGHT ENTER THE FOLLOWING FROM      04710000
A CARD READER:                                                      04720000
*SKIP 1 SINGLE  INDENT 5                                            04730000
*INVALID ALL 4                                                      04740000
    EXECUTE PATCH/MERGE                                             04750000
    DATA CARD                                                       04760000
*INDENT 5 SINGLE                                                   04770000
     $@ FINAL                                                       04780000
     $. 5 PATCHES FOR ALGOLX                                        04790000
     $*COMPILE ALGBL/DISK WITH ALGOL TO LIBRARY                     04800000
     $*ALGOL FILE TAPE=SYMBOL/ALGOLSY                               04810000
     $*FILE LINE=LINE PRINT OR BACK UP                              04820000
     $*DATA CARD                                                    04830000
     $=                                                             04840000
     $TAPE SINGLE PRT SEQXEQ                                        04850000
     $TAPE SEQXEQ CHECK                                             04860000
*SINGLE  INDENT 5                                                   04870000
*INVALID 4                                                          04880000
    END.                                                            04890000
*3                                                                  04900000
IF THE ABOVE DECK WERE INPUT FROM  A CARD READER, THEN              04910000
A TOTAL OF FIVE                                                     04920000
PATCHES (INDICATED BY THE "$." CARD) MUST BE INCLUDED IN            04930000
ONE OR BOTH FILES                                                   04940000
ON DISK NAMED "PATCH/ALGOLX" OR "PATCHES/ALGOLX".                   04950000
*SKIP 2 PAR                                                        04960000
SINCE THE OPTION FINAL IS SET:                                      04970000
*SKIP 1 PAR                                                        04980000
                    1. THE OPTIONS MERGE, LISTI,                    04990000
 CONFLICTS, LISTG, NONO,                                            05000000
NEW DISK, AND ZIP WILL AUTOMATICALLY BE SET BY PATCH/MERGE.         05010000
THEREFORE, PATCH/MERGE WILL LOOK FOR THE FILES PATCH/ALGOLX AND     05020000
PATCHES/ALGOLX ON DISK.                                             05030000
*SKIP 1 PAR                                                        05040000
                  2. THE INPUT WILL BE LISTED SINCE LISTI IS SET.   05050000
*SKIP 1 PAR                                                        05060000
3. THE CONFLICTS (IF ANY) WILL BE LISTED, SINCE CONFLICTS IS SET.   05070000
*SKIP 1 PAR                                                        05080000
4. THE                                                              05090000
GENERATED PATCH DECK WILL BE LISTED, SINCE LISTG IS SET.            05100000
*SKIP 1 PAR                                                        05110000
5. THE GENERATED                                                    05120000
PATCH DECK WILL BE ZIPPED, SINCE ZIP IS SET.                        05130000
*SKIP 1 PAR                                                        05140000
                    6. PATCH NUMBERS WILL NOT                       05150000
 BE PUT IN COLUMNS 68-72 OF THE GENERATED PATCH CARDS, SINCE        05160000
 NONO IS SET.                                                       05170000
*SKIP 1 PAR                                                        05180000
```

```
                                        7.  FINALLY, A NEW FILE NAMED  05190000
PATCHES/ALGOLX WILL BE CREATED ON DISK CONTAINING ALL THE PATCHES       05200000
USED IN GENERATING THE SORTED PATCH DECK,                               05210000
AND THE OLD FILES  PATCH/ALGOLX AND PATCHES/ALGOLX WILL BE PURGED.      05220000
*QUIT                                                                   05230000
;                                                                       05240000
%                                                                       10000000
COMMENT##################################################################10010000
                    OUTER BLOCK DECLARATIONS AND DEFINES                10020000
##########################################################################;10030000
%                                                                       10040000
BEGIN                                                                   10050000
        REAL PATCHMAX,COLUMN,COL,NUMERR,TIME1;                          10060000
COMMENT  ALL DEFINES USED IN PATCH/MERGE ARE HERE;                      10070000
DEFINE                                                                  10080000
        BADCARD             = 4#,                                       10090000
        BADDOLATCD          = 18#,                                      10100000
        BADVOID             = 19#,                                      10110000
        BADVOIDRANGE        = 21#,                                      10120000
        BADVOIDSEQ          = 20#,                                      10130000
        DUPATCH             =-13#,                                      10140000
        EVOLVE              = POINTER(TOP[DECK,8])+3#,                  10150000
        EXTRACDS            = 11#,                                      10160000
        EXTRAPATS           = 17#,                                      10170000
        HOWMANYCDS          =-7#,                                       10180000
        HOWMANYPAT          = 1#,                                       10190000
        MEDIUM[PATCH]       = ARR[PATCH].[15:15]#,                      10200000
        MISSCDS             = 10#,                                      10210000
        MISSPATS            = 16#,                                      10220000
        NOCDS               = 15#,                                      10230000
        NODOLASTCD          = 14#,                                      10240000
        NODOLCARD           = 3#,                                       10250000
        NODOLPERCD          = 0#,                                       10260000
        NONAME              = 2#,                                       10270000
        NOTINSEQ            =-9#,                                       10280000
        NOPATNO             = 5#,                                       10290000
        PATCHNUMBER[PATCH]  = ARR[PATCH].[33:15]#,                      10300000
        PATNOTINSEQ         = 12#,                                      10310000
        PTN[KARD]           = NAME[2+KARD]#,                            10320000
        QUEST               = 3"1460000000000000" FOR 2#,               10330000
        REVOLVER            = POINTER(TOP[DECK,10])#,                   10340000
        STOPPER             = 98765#,                                   10350000
        WHATPROG            =-6#,                                       10360000
        WRONGPROG           =-8#,                                       10370000
        WHICHONE[WH]        = POINTER(NAME[1])+WH#,                     10380000
        DUMMY= #                                                        10390000
        ;                                                               10400000
ALPHA PATCHID;                                                          10410000
FILE CARD(2,10,150),                                                    10420000
     MASTERFILE DISK SERIAL[20:1000](2,10,150),                         10430000
     PATCHDECK DISK SERIAL [20:1000](2,10,150),                         10440000
     LINE 18(5,15),                                                     10450000
     SORTLINE DISK SERIAL [20:1000] (2,15,150),                         10460000
     PUNCH 0(5,10);                                                     10470000
POINTER P,P1,Q,Q1;                                                      10480000
BOOLEAN CONFLICTS,ERRORS,LISTI,LISTG,                                   10490000
        SAVEIT,GOTOIT,NOHEADING,WCON,ANOND,NOMOREOPCARDS,               10500000
        CARDSONLY,PUNCHI,PUNCHG,NOMAST,DELETE, NEWFILES;                10510000
ARRAY IMAGE[0:2,0:16],NAME[0:4];                                        10520000
INTEGER ARRAY DEL[0:19];                                                10530000
```

```
INTEGER KARD;                                                          10540000
LABEL  AXIT;                                                           10550000
%                                                                      20000000
COMMENT##########################################################################20010000
                    UTILITY PROCEDURES                                 20020000
##########################################################################;20030000
%                                                                      20040000
PROCEDURE DATIME;                                                      20050000
    BEGIN                                                              20060000
    INTEGER H,MIN,Q;                                                   20070000
%#######################################################################20080000
    ALPHA PROCEDURE DATER(DATE); VALUE DATE; REAL DATE;                20090000
        BEGIN                                                          20100000
        ARRAY A[0:1];                                                  20110000
        POINTER SI,DI;                                                 20120000
        A[0]:=DATE;                                                    20130000
        SI:=POINTER(A)+2; DI:=POINTER(A[1]);                           20140000
        REPLACE DI:DI BY SI:SI FOR 2,"/";                              20150000
        REPLACE DI:DI BY SI:SI FOR 2,"/";                              20160000
        REPLACE DI BY SI FOR 2;                                        20170000
        DATER:=A[1];                                                   20180000
        END OF DATER;                                                  20190000
%                                                                      20200000
    H:=TIME1 DIV 216000; MIN:=(TIME1 DIV 3600) MOD 60;                 20210000
    WRITE(LINE[DBL],                                                   20220000
        <X16,"BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK ",             20230000
        "XV.3.00"                                                      20240000
        ,"  ",A6,"DAY, ",0,",  ",I2,":",A2,X1,A1,"M.",//>,              20250000
        TIME(6),DATER(TIME(5)),12×REAL(Q:=H MOD 12=0)+Q,               20260000
        Q:=MIN MOD 10+(MIN DIV 10)×64,                                 20270000
        IF H≥12 THEN "P" ELSE "A");                                    20280000
    NOHEADING:=FALSE;                                                  20290000
    END OF DATIME;                                                     20300000
%#######################################################################20310000
PROCEDURE ERROR(NUMBER,PARAMETER);                                     20320000
    VALUE NUMBER,PARAMETER;                                            20330000
    REAL NUMBER,PARAMETER;                                             20340000
    BEGIN                                                              20350000
    SWITCH FORMAT ERR:=                                                20360000
    (X21,"I CANNOT FIND YOUR $. CARD ××××××××××××××××××",X*), %0      20370000
    (X21,"TELL ME HOW MANY PATCHES YOU HAVE ××××××××××××",X*), %1      20380000
    (X21,"TELL ME THE NAME OF YOUR PROGRAM ×××××××××××××",X*), %2      20390000
    (X21,"I NEED A DOLLAR CARD AT THIS POINT ××××××××××",X*), %3       20400000
    (X21,"THIS CARD SHOULD NOT BE HERE ××××××××××××××××",X*), %4       20410000
    (X21,"I HAVE TO KNOW THE PATCH NUMBER ×××××××××××××",X*), %5       20420000
    (X21,"YOU DONT SAY WHAT PROGRAM PATCH #"I3" IS FOR ××××××"),%6      20430000
    (X21,"TELL ME HOW MANY CARDS ARE IN THIS PATCH ××××××",X*), %7      20440000
    (X21,"PATCH #"I3" IS FOR THE WRONG PROGRAM ××××××××××××"),%8       20450000
    (X21,"THIS CARD IS NOT IN SEQUENCE IN PATCH #"I3" ×××××××"),%9     20460000
    (X21,"THERE ARE "I3" MISSING CARDS IN THIS PATCH ××××××"),  %10    20470000
    (X21,"THERE ARE "I3" EXTRA CARDS IN THIS PATCH ×××××××××"),  %11   20480000
    (X21,"PATCH #"I3" IS OUT OF SEQUENCE ××××××××××××××××××××"),%12   20490000
    (X21,"YOU ARE ALLOWED ONLY ONE PATCH #"I3" ××××××××××××××"),%13    20500000
    (X21,"YOU FORGOT THE $* CARDS ××××××××××××××××××××××××",X*), %14  20510000
    (X21,"PLEASE PUT AT LEAST ONE CARD IN PATCH #"I3" ×××××××"),%15    20520000
    (X21,I3," MISSING PATCHES ××××××××××××××××××××××××××××"),  %16    20530000
    (X21,I3," MORE PATCHES THAN YOU SAID THERE WOULD BE ×"),  %17       20540000
    (X21,"THIS $@ CARD IS OUT OF PLACE ××××××××××××××××××",X*), %18    20550000
    (X21,"THIS KIND OF VOID CARD IS NOT ALLOWED ××××××××××",X*), %19    20560000
    (X21,"THE FOLLOWING VOID RANGE MUST HAVE 8 DIGITS ×××",X*), %20     20570000
```

```
        (X21,"CHECK THE VOID RANGE, IT IS INCORRECT xxxxxxxxx",X*); %21      20580000
        FORMAT FLAG("xxxxxxxxxxxxxxxxxxxx ",X47,"xxxxxxxxxxxxxx",           20590000
                    "xxxxxxxx ERROR",I3);                                   20600000
        LABEL E;                                                            20610000
    %                                                                       20620000
        NUMERR:=NUMERR+1;                                                   20630000
        IF NOT LISTI AND NOHEADING THEN DATIME;                             20640000
        IF NUMBER LSS 0 THEN WRITE(LINE[DBL]) ELSE                          20650000
            IF PARAMETER LSS 0 THEN                                         20660000
                BEGIN                                                       20670000
                WRITE(LINE,10,IMAGE[KARD,*]);                               20680000
                PARAMETER:=ABS(PARAMETER);                                  20690000
                END;                                                        20700000
            NUMBER:=ABS(NUMBER);                                            20710000
        WRITE(LINE[NO],FLAG,NUMERR);                                        20720000
        WRITE(LINE[DBL],ERR[NUMBER],PARAMETER);                             20730000
        WRITE(LINE);                                                        20740000
        ERRORS:=TRUE;                                                       20750000
        IF NUMBER LSS 5 THEN GO AXIT                                        20760000
        END ERROR PROCEDURE;                                                20770000
    %###################################################################################20780000
    PROCEDURE MASTHEAD(N); VALUE N; INTEGER N;                              20790000
        BEGIN                                                               20800000
        SWITCH FORMAT F:=(X10,"      INPUT       "),                        20810000
                         (X10," GENERATED OUTPUT "),                        20820000
                         (X10,"    CONFLICTS     "),                        20830000
                         ("**********",X18,"**********",                    20840000
                          "****************************************",        20850000
                          "************************"/);                     20860000
    %                                                                       20870000
        WRITE(LINE[NO],F[N]);                                               20880000
        WRITE(LINE[DBL],F[3]);                                              20890000
        NOMAST:=FALSE                                                       20900000
        END OF MASTHEAD;                                                    20910000
    %###################################################################################20920000
    PROCEDURE GETDELETE;                                                    20930000
    BEGIN                                                                   20940000
        INTEGER I;  LABEL EGRESS;                                           20950000
    %                                                                       20960000
        FOR I:=0 STEP 1 UNTIL 19 DO                                         20970000
        BEGIN                                                               20980000
            SCAN Q:Q1 FOR COLUMN:COLUMN UNTIL GEQ 0;                        20990000
            IF COLUMN LEQ 0 THEN GO EGRESS;                                 21000000
            SCAN Q1:Q FOR COLUMN:COLUMN WHILE GEQ 0;                        21010000
            DEL[I]:=INTEGER(Q,DELTA(Q,Q1));                                 21020000
            DELETE:=TRUE;                                                   21030000
        END;                                                                21040000
    EGRESS:                                                                 21050000
    END GETDELETE;                                                          21060000
    %###################################################################################21070000
    PROCEDURE SETOPTIONS;                                                   21080000
        BEGIN  INTEGER COL; LABEL INIT;                                     21090000
    %                                                                       21100000
    INIT:                                                                   21110000
        COLUMN:=80;                                                         21120000
    DO  BEGIN                                                               21130000
        SCAN Q:Q1 FOR COLUMN:COLUMN UNTIL IN ALPHA;                         21140000
        SCAN Q1:Q FOR COLUMN:COLUMN WHILE IN ALPHA;                         21150000
        CASE (IF COL:=DELTA(Q,Q1).GTR 9 THEN 0 ELSE COL) OF                 21160000
            BEGIN                                                           21170000
```

```
            ;                                                    %0          21180000
            ;                                                    %1          21190000
            ;                                                    %2          21200000
            IF REAL(Q,3)="ZIP" THEN GOTOIT:=TRUE ELSE            %3          21210000
             IF REAL(Q,3)="NEW" THEN SAVEIT:=TRUE;                           21220000
            IF REAL(Q,4)="CARD" THEN CARDSONLY:=TRUE ELSE        %4          21230000
             IF REAL(Q,4)="LIST" THEN LISTI:=LISTG:=CONFLICTS:=TRUE ELSE     21240000
              IF REAL(Q,4)="NONO" THEN ANONO:=TRUE;                          21250000
            IF REAL(Q,5)="MERGE" THEN CARDSONLY:=FALSE ELSE      %5          21260000
             IF REAL(Q,5)="LISTI" THEN LISTI:=TRUE ELSE                      21270000
              IF REAL(Q,5)="LISTG" THEN LISTG:=TRUE ELSE                     21280000
               IF REAL(Q,5)="FINAL" THEN GOTOIT:=SAVEIT:=LISTI:=             21290000
                                         LISTG:=CONFLICTS:=ANONO:=           21300000
                                         NOT(CARDSONLY:=FALSE);              21310000
            IF REAL(Q,6)="PUNCHI" THEN PUNCHI:=TRUE ELSE         %6          21320000
             IF REAL(Q,6)="PUNCHG" THEN PUNCHG:=TRUE ELSE                    21330000
              IF REAL(Q,6)="DELETE" THEN GETDELETE;                          21340000
            IF REAL(Q,7) = "NEWFILE" THEN NEWFILES := TRUE;      %7          21350000
            ;                                                    %8          21360000
            IF REAL(Q,7)="CONFLIC" THEN CONFLICTS:=TRUE;         %9          21370000
            END;                                                             21380000
        END UNTIL COLUMN LEQ 0;                                              21390000
        IF LISTI THEN                                                        21400000
            BEGIN                                                            21410000
            IF NOHEADING THEN DATIME;                                        21420000
            IF NOMAST THEN MASTHEAD(0);                                      21430000
            WRITE(LINE,1C,IMAGE[KARD,*])                                     21440000
            END;                                                             21450000
        READ(CARD,10,IMAGE[KARD,*]);                                         21460000
        IF REAL(Q1:=POINTER(IMAGE[KARD,*]),2)="$@" THEN GO INIT;             21470000
        NOMOREOPCARDS:=TRUE;                                                 21480000
        END OF SETOPTIONS;                                                   21490000
    %################################################################################21500000
        PROCEDURE GETDOLPERCD;                                               21510000
        BEGIN                                                                21520000
        INTEGER COL;                                                         21530000
        %                                                                    21540000
        IF Q1 NEQ "$." THEN ERROR(NODOLPERCD,-1);                            21550000
        COL:=78;                                                             21560000
        DO  BEGIN                                                            21570000
            SCAN Q:Q1 FOR COL:COL UNTIL IN ALPHA;                            21580000
            SCAN Q1:Q FOR COL:COL WHILE IN ALPHA;                            21590000
            IF COL LEQ 0 THEN ERROR(NONAME,-1);                              21600000
            END UNTIL REAL(Q,3)="FOR" AND DELTA(Q,Q1)=3;                     21610000
        P1:=POINTER(IMAGE[KARD,*]);                                          21620000
        SCAN P:P1 FOR COLUMN:78-COL WHILE LSS 0;                             21630000
        IF COLUMN LEQ 0 THEN ERROR(HOWMANYPAT,-1);                           21640000
        SCAN P1:P FOR COLUMN:COLUMN UNTIL LSS 0;                             21650000
        IF (PATCHMAX:=INTEGER(P,DELTA(P,P1)))LSS 0 THEN ERROR(HOWMANYPAT,-1); 21660000
        SCAN P1:Q1 FOR COLUMN:COL UNTIL IN ALPHA;                            21680000
        SCAN P:P1 FOR COLUMN:COLUMN WHILE IN ALPHA;                          21690000
        IF COLUMN LEQ 0 THEN ERROR(NONAME,-1);                               21700000
        REPLACE POINTER(NAME) BY " " FOR 1 WORDS;                            21710000
        REPLACE POINTER(NAME) BY "0",P1 FOR MIN(DELTA(P1,P),7);              21720000
        PATCHID:=REAL(P1,MIN(PATCHID:=DELTA(P1,P),6)) & PATCHID[41:5:6];     21730000
        IF (LISTI OR LISTG OR CONFLICTS) AND NOHEADING THEN DATIME;          21740000
        IF LISTI THEN                                                        21750000
            BEGIN                                                            21760000
            IF NOMAST THEN MASTHEAD(0);                                      21770000
            WRITE(LINE,10,IMAGE[KARD,*])                                     21780000
```

```
    END;                                                                21790000
END GETDOLPERCD;                                                        21800000
%                                                                       30000000
COMMENT##########################################################################30010000
                    PROCESS PROCEDURE                                   30020000
##################################################################################;30030000
%                                                                       30040000
 PROCEDURE PROCESS;                                                     30050000
    BEGIN                                                               30060000
    ARRAY SEQ,RECORD,ENDOFFILE[0:PATCHMAX+3],       -                   30070000
         LASTCONTROLCARDIMAGE[0:11], TOP[0:PATCHMAX+3,0:11],            30080000
         ARR[0:PATCHMAX+3];                                             30090000
    REAL LASTCONTROLCARD,PATCHRECORD,TOPDOG,TOPDOGSEQ,VOIDSEQ,          30100000
         VOIDINGSEQ,ENDSOFFILE;                                         30110000
%                                                                       40000000
COMMENT##########################################################################40010000
                    INPUTPHASE PROCEDURE                                40020000
##################################################################################;40030000
%                                                                       40040000
    PROCEDURE INPUTPHASE;                                               40050000
         BEGIN                                                          40060000
         LABEL THATSALL,EGRESS,READCC,NEXTDECK,NEXTCARD,PASS,EOF,READCD;40070000
         POINTER INPOINTER,STARPOINTER,P,P1;                            40080000
         BOOLEAN FIRSTCARD;                                             40090000
         REAL PATCH,RECORDNO,REC,SEQUENCE,SIZE,T,CHAR,LASTPATCHNO;      40100000
FILE NEWDISK DISK SERIAL [20:1000]"PATCHES"/"MCP"(1,10,150,SAVE 99),    40110000
     OLDDISK DISK SERIAL "PATCHES"/"MCP"(2,10,150),                     40120000
     PATCHFILE DISK SERIAL [20:1000]"PATCH"/"MCP"(2,10,150,SAVE 60),    40121000
     NEWPATCHES DISK SERIAL "PATCH  "/"MCP"(2,10,150);                  40130000
SWITCH FILE KARDSW:=OLDDISK,NEWPATCHES,CARD;                            40140000
%##############################################################################40150000
PROCEDURE FLUSH; FORWARD;                                               40160000
%##############################################################################40170000
PROCEDURE GETPATCHNO(KARD); VALUE KARD; INTEGER KARD; FORWARD;          40180000
%##############################################################################40190000
BOOLEAN PROCEDURE DELETEIT;                                             40200000
    BEGIN    INTEGER I; POINTER KEY;                                    40210000
    LABEL XIT;                                                          40220000
%                                                                       40230000
    KEY:=POINTER(IMAGE[KARD,*]);                                        40240000
    FOR I:=0 STEP 1 WHILE I LSS 20 AND DEL[I] NEQ 0 DO                  40250000
        BEGIN                                                           40260000
        IF T=DEL[I] THEN                                                40270000
            BEGIN                                                       40280000
            IF LISTI THEN                                               40290000
                BEGIN                                                   40300000
                REPLACE KEY+80 BY " ",                                  40310000
                                  WHICHONE[KARD] FOR 1,                 40320000
                                  " " FOR 2,PTN[KARD] FOR 3 DIGITS,     40330000
                                  " "," DELETED   ### ";                40340000
                WRITE(LINE[DBL]);                                       40350000
                WRITE(LINE[DBL],13,IMAGE[KARD,*]);                      40360000
                END;                                                    40370000
            FLUSH;                                                      40380000
            GETPATCHNO(KARD);                                           40390000
            DELETEIT:=TRUE;                                             40400000
            GO TO XIT                                                   40410000
            END                                                         40420000
        END;                                                            40430000
XIT:                                                                    40440000
```

```
END DELETEIT;                                                                    40450000
%#################################################################################40460000
PROCEDURE CHECKVOID;                                                             40470000
    BEGIN    LABEL XIT;                                                          40480000
    REAL T;                                                                      40490000
%                                                                                40500000
    SCAN P:INPOINTER FOR COLUMN:72 UNTIL IN ALPHA;                               40510000
    IF COLUMN LEQ 0 THEN GO TO XIT;                                              40520000
    SCAN P1:P FOR COLUMN:COLUMN WHILE IN ALPHA;                                  40530000
    IF COLUMN LEQ 0 THEN GO TO XIT;                                              40540000
    IF (T:=DELTA(P,P1))=4 OR T=5 THEN                                            40550000
    BEGIN                                                                        40560000
     IF REAL(P,4)="VOID" THEN ELSE                                               40570000
      IF REAL(P,5)="VOIDT" THEN ELSE GO XIT;                                     40580000
    END                                                                          40590000
    ELSE GO XIT;                                                                 40600000
    SCAN P:P1 FOR COLUMN:COLUMN UNTIL GEQ 0;                                     40610000
    IF COLUMN LEQ 0 THEN GO TO XIT;                                              40620000
    SCAN P1:P FOR COLUMN:COLUMN UNTIL LSS 0;                                     40630000
    IF COLUMN LSS 0 THEN ERROR(BADVOID,1);                                       40640000
    IF DELTA(P,P1) NEQ 8 THEN                                                    40650000
        BEGIN                                                                    40660000
        ERROR(BADVOIDSEQ,1);                                                     40670000
        GO TO XIT                                                                40680000
        END;                                                                     40690000
    IF INTEGER(P,8) LEQ INTEGER(POINTER(IMAGE[KARD,9]),8) THEN                   40700000
        ERROR(BADVOIDRANGE,1);                                                   40710000
    IF REAL(INPOINTER,2)=" $" THEN ERROR(BADVOID,1);                             40720000
XIT:                                                                             40730000
    END CHECKVOID;                                                               40740000
%#################################################################################40750000
PROCEDURE GETPATCHNO(KARD); VALUE KARD; INTEGER KARD;                            40760000
    BEGIN                                                                        40770000
    LABEL XIT,NUTS,OWT;                                                          40780000
    POINTER R,R1;                                                                40790000
%                                                                                40800000
    IF PTN[KARD] = STOPPER THEN GO XIT,                                          40810000
    R1:=POINTER(IMAGE[KARD,*]);                                                  40820000
    IF REAL(R1,2) NEQ "$#" THEN                                                  40830000
        BEGIN                                                                    40840000
        ERROR(NOPATNO,1);                                                        40850000
OWT:                                                                             40860000
        PTN[KARD]:=STOPPER;                                                      40870000
        GO XIT                                                                   40880000
        END;                                                                     40890000
    SCAN R:R1 FOR COL:80 WHILE LSS "0";                                          40900000
    IF COL LEQ 0 THEN                                                            40910000
        BEGIN                                                                    40920000
NUTS:                                                                            40930000
        ERROR(NOPATNO,1);                                                        40940000
        GO OWT                                                                   40950000
        END;                                                                     40960000
    SCAN R1:R FOR COL:COL UNTIL LSS "0";                                         40970000
    IF COL LEQ 0 THEN GO NUTS;                                                   40980000
    PTN[KARD]:=INTEGER(R,DELTA(R,R1));                                           40990000
XIT:                                                                             41000000
    END GETPATCHNO;                                                              41010000
%#################################################################################41020000
PROCEDURE HANDLEENDOFFILE;                                                       41030000
    BEGIN                                                                        41040000
```

```
      PTN[KARD]:=STOPPER;                                                    41050000
      IF (PTN[0]+PTN[1]+PTN[2])=(3×(STOPPER)) THEN                           41060000
           BEGIN                                                             41070000
           CHAR:=0; % SO PROPER EXIT WILL BE MADE AT THATSALL                41080000
           GO TO (IF PATCHMAX = 0 THEN EGRESS ELSE THATSALL);                41090000
           END;                                                              41100000
      IF PTN[0]+PTN[1]=2×STOPPER THEN CARDSONLY:=TRUE;                       41110000
      END OF HANDLEENDOFFILE;                                                41120000
%#############################################################################41130000
PROCEDURE ALIGN;                                                             41140000
      BEGIN                                                                  41150000
%                                                                            41160000
      FOR KARD:=0 STEP 1 UNTIL 2 DO                                          41170000
       WHILE PTN[KARD] LEQ LASTPATCHNO DO                                    41180000
           BEGIN                                                             41190000
           IF LISTI THEN                                                     41200000
                BEGIN                                                        41210000
                REPLACE POINTER(IMAGE[KARD,10]) BY " ",                      41220000
                                        WHICHONE[KARD] FOR 1,                41230000
                                        " " FOR 2,                           41240000
                                        PTN[KARD] FOR 3 DIGITS,              41250000
                                        " "," DISCARDED ### ";               41260000
                WRITE(LINE[DBL]);                                            41270000
                WRITE(LINE[DBL],13,IMAGE[KARD,*])                            41280000
                END;                                                         41290000
           FLUSH;                                                            41300000
           GETPATCHNO(KARD);                                                 41310000
           END;                                                              41320000
      KARD:=IF PTN[1] LEQ PTN[0] THEN 1 ELSE 0;                              41330000
      KARD:=IF PTN[2] LEQ PTN[KARD] THEN 2 ELSE KARD;                        41340000
      END ALIGN;                                                             41350000
%#############################################################################41360000
PROCEDURE FLUSH;                                                             41370000
      BEGIN                                                                  41380000
      POINTER PTR;                                                           41390000
      LABEL OWT,EOF;                                                         41400000
%                                                                            41410000
      PTR:=POINTER(IMAGE[KARD,*]);                                           41420000
      WHILE TRUE DO                                                          41430000
           BEGIN                                                             41440000
           READ(KARDSW[KARD],10,IMAGE[KARD,*]) [EOF];                        41450000
           IF REAL(PTR,2)="$#" THEN GO OWT;                                  41460000
           IF LISTI THEN WRITE(LINE,13,IMAGE[KARD,*]);                       41470000
           END;                                                              41480000
      WHILE FALSE DO                                                         41490000
EOF:                                                                         41500000
           HANDLEENDOFFILE;                                                  41510000
OWT:                                                                         41520000
      END FLUSH;                                                             41530000
%#############################################################################41540000
PROCEDURE INITIALIZE;                                                        41550000
      BEGIN                                                                  41560000
      LABEL EOC,XIT;                                                         41570000
      ARRAY TEMP[0:9];                                                       41580000
      INTEGER INDEX;                                                         41590000
%                                                                            41600000
      IF CARDSONLY THEN                                                      41610000
           BEGIN                                                             41620000
           PTN[0]:=PTN[1]:=STOPPER;                                          41630000
           GO XIT                                                            41640000
```

```
            END;                                                          41650000
        FOR INDEX:=0,1 DO                                                 41660000
            BEGIN                                                         41670000
            SEARCH(KARDSW[INDEX],TEMP[*]);                                41680000
            IF TEMP[0] LEQ 0 THEN                                         41690000
EOC:                                                                      41700000
             PTN[INDEX]:=STOPPER ELSE                                     41710000
                BEGIN                                                     41720000
                READ(KARDSW[INDEX],10,IMAGE[INDEX,*]) [EOC];              41730000
                GETPATCHNO(INDEX)                                         41740000
                END;                                                      41750000
            END;                                                          41760000
XII:                                                                      41770000
    END INITIALIZE;                                                       41780000
%                                                                         50000000
COMMENT###################################################################50010000
                    INPUTPHASE PROCEDURE DRIVER                           50020000
#########################################################################;50030000
%                                                                         50040000
            FILL NEWDISK WITH *,NAME[0];                                  50050000
            FILL OLDDISK WITH *,NAME[0];                                  50060000
            FILL NEWPATCHES WITH *,NAME[0];                               50070000
             FILL PATCHFILE WITH *, NAME[0];                              50071000
            INPOINTER:=POINTER(IMAGE[KARD,*]);                            50080000
            REC:=1;                                                       50090000
            WRITE(MASTERFILE,*,RECORDNO:=1);                              50100000
            PATCH:=LASTCONTROLCARD:=LASTPATCHNO:=-1;                      50110000
            INITIALIZE;                                                   50120000
READCC:                                                                   50130000
            READ(CARD,10,IMAGE[KARD,*])[EOF];                             50140000
            IF LISTI AND CHAR:=REAL(INPOINTER+1,1)NEQ "#" THEN            50150000
                WRITE(LINE,10,IMAGE[KARD,*]); % # CARDS LISTED LATER      50160000
            REC:=REC+1;                                                   50170000
            IF REAL(INPOINTER,1) NEQ "$" THEN                             50180000
                ERROR(NODOLCARD,1);                                       50190000
            IF CHAR="@" THEN                                              50200000
                BEGIN                                                     50210000
                ERROR(BADDOLATCD,1);                                      50220000
                GO TO READCC                                              50230000
                END;                                                      50240000
            IF CHAR="*" THEN                                              50250000
                BEGIN           COMMENT CONTROL CARD IMAGE;               50260000
                REPLACE INPOINTER BY GUEST;                               50270000
                IF LASTCONTROLCARD≥0 THEN                                 50280000
                    BEGIN                                                 50290000
                    LASTCONTROLCARDIMAGE[9]:=LASTCONTROLCARD+1;           50300000
                    WRITE(PATCHDECK, 10, LASTCONTROLCARDIMAGE[*]);        50310000
                    IF LISTG THEN                                         50320000
                       WRITE(SORTLINE,12,LASTCONTROLCARDIMAGE[*]);        50330000
                    END;                                                  50340000
                LASTCONTROLCARD:=LASTCONTROLCARD+1;                       50350000
                REPLACE POINTER(LASTCONTROLCARDIMAGE[*]) BY               50360000
                    INPOINTER FOR 10 WORDS;                               50370000
                GO TO READCC                                              50380000
                END;                                                      50390000
    READCD:                                                               50400000
            WRITE(PATCHDECK,10,LASTCONTROLCARDIMAGE[*]);                  50410000
            IF LISTG THEN                                                 50420000
               WRITE(SORTLINE,12,LASTCONTROLCARDIMAGE[*]);                50430000
            PATCHRECORD:=LASTCONTROLCARD+1;                               50440000
```

```
            IF CHAR="-" THEN     COMMENT DOLLAR CARDS, ET, AL;                    50450000
                BEGIN                                                             50460000
PASS:                                                                             50470000
                READ(CARD,10,IMAGE[KARD,*]) [EOF];                                50480000
                REC:=REC+1;                                                       50490000
                IF CHAR:=REAL(INPOINTER,2)="$#" THEN GO TO NEXTDECK;              50500000
                IF LISTI THEN WRITE(LINE,10,IMAGE[KARD,*]);                       50510000
                IF CHAR="$*" OR CHAR="$-" THEN ERROR(BADCARD,1);                  50520000
                WRITE(PATCHDECK,10,IMAGE[KARD,*]);                                50530000
                    IF LISTG THEN                                                 50540000
                      WRITE(SORTLINE,12,IMAGE[KARD,*]);                           50550000
                PATCHRECORD:=PATCHRECORD+1;                                       50560000
                GO TO PASS                                                        50570000
                END;                                                              50580000
            IF LASTCONTROLCARD LSS 0 THEN ERROR(NODDLASTCD,1);                    50590000
            IF CHAR="#" THEN                                                      50600000
                BEGIN                                                             50610000
NEXTDECK:                                                                         50620000
                GETPATCHNO(KARD);                                                 50630000
                ALIGN;                                                            50640000
                INPOINTER:=POINTER(IMAGE[KARD,*]);                                50650000
                STARPOINTER:=POINTER(IMAGE[KARD,9]);                              50660000
                FIRSTCARD:=TRUE;                                                  50670000
                IF LASTPATCHNO GTR T:=PTN[KARD] THEN                              50680000
                    ERROR(PATNOTINSEQ,T);      %PATCH OUT OF SEQ                  50690000
                REPLACE POINTER(IMAGE[KARD,10]) BY " " FOR 2 WORDS;               50700000
                REPLACE POINTER(IMAGE[KARD,10])+1 BY WHICHONE[KARD] FOR 1,        50710000
                    " " FOR 2,PTN[KARD] FOR 3 DIGITS;                             50720000
                IF DELETE THEN IF DELETEIT THEN GO TO NEXTDECK;                   50730000
                IF LISTI THEN                                                     50740000
                    BEGIN                                                         50750000
                    WRITE(LINE[DBL]); WRITE(LINE[DBL]);                           50760000
                    WRITE(LINE[NO],12,IMAGE[KARD,*]);                             50770000
                    END;                                                          50780000
            IF LASTPATCHNO=LASTPATCHNO:=T THEN %DUP. PATCH #                      50790000
                ERROR(DUPATCH,T);                                                 50800000
            P1:=INPOINTER; COLUMN:=80;                                            50810000
            DO  BEGIN                                                             50820000
                SCAN P:P1 FOR COLUMN:COLUMN UNTIL IN ALPHA;                       50830000
                SCAN P1:P FOR COLUMN:COLUMN WHILE IN ALPHA;                       50840000
                IF COLUMN LEQ 0 THEN ERROR(WHATPROG,1);                           50850000
                END UNTIL (REAL(P,3)="FOR" AND DELTA(P,P1)=3)OR                   50860000
                    COLUMN LEQ 0;                                                 50870000
            IF COLUMN LEQ 0 THEN                                                  50880000
                BEGIN                                                             50890000
                ERROR(WHATPROG,T);                                                50900000
                GO TO NEXTCARD                                                    50910000
                END;                                                              50920000
            SCAN P:P1 FOR COLUMN:COLUMN UNTIL IN ALPHA;                           50930000
            SCAN P1:P FOR COLUMN:COLUMN WHILE IN ALPHA;                           50940000
            IF COLUMN≤0 THEN                                                      50950000
                ERROR(WHATPROG,T);                                                50960000
            IF PATCHID NEQ REAL(P,MIN(T:=DELTA(P,P1),6)) & T[41:5:6]              50970000
                THEN ERROR(WRONGPROG,LASTPATCHNO);                                50980000
            SCAN P:P1 FOR COLUMN:COLUMN UNTIL =" ";                               50985000
            SCAN P1:P FOR COLUMN:COLUMN WHILE <"0",                               50990000
            SCAN P:P1 FOR COLUMN:COLUMN UNTIL <"0";                               51000000
            IF COLUMN LEQ 0 THEN ERROR(HOWMANYCDS,1);                             51010000
            PATCH:=PATCH+1;                                                       51020000
            SIZE:=INTEGER(P1,DELTA(P1,P));                                        51030000
```

```
                PATCHNUMBER[PATCH]:=LASTPATCHNO;                                51040000
                MEDIUM[PATCH]:=KARD;                                            51050000
                IF NOT ERRORS AND SAVEIT THEN                                   51060000
                 BEGIN                                                          51061000
                   WRITE(NEWDISK,10,IMAGE[KARD,*]);                             51070000
                    IF NEWFILES AND KARD = 2 THEN                               51071000
                    WRITE(PATCHFILE, 10, IMAGE[KARD,*]);                        51072000
                 END;                                                           51073000
                IF LISTI THEN WRITE(LINEIDBL]);                                 51080000
NEXTCARD:                                                                       51090000
                READ(KARDSW[KARD],10,IMAGE[KARD,*])[EOF];                       51100000
                IF (REAL(STARPOINTER+/,1)="*") OR                               51110000
                 (REAL(INPOINTER,2)="%:") THEN % PATCH/MERGE COMMENT CARDS      51120000
                    BEGIN                                                       51130000
                    IF LISTI THEN WRITE(LINE,12,IMAGE[KARD,*]);                 51140000
                    IF NOT ERRORS AND SAVEIT THEN                               51142000
                 BEGIN                                                          51143000
                    WRITE(NEWDISK,10,IMAGE[KARD,*]);                            51150000
                    IF NEWFILES AND KARD = 2 THEN                               51151000
                    WRITE(PATCHFILE, 10, IMAGE[KARD,*]);                        51152000
                END;                                                            51153000
                %                                                               51160000
                    IF PUNCHI THEN WRITE(PUNCH,10,IMAGE[KARD,*]);               51170000
                    GO TO NEXTCARD;                                             51180000
                    END;                                                        51190000
                REC:=REC+1;                                                     51200000
                IF REAL(INPOINTER,2)=" $" THEN CHECKVOID;                       51210000
                IF CHAR:=REAL(INPOINTER,1)="$" THEN                             51220000
                     BEGIN                                                      51230000
                IF CHAR:=REAL(INPOINTER+1,1)="@" THEN                           51240000
                     BEGIN                                                      51250000
                     IF LISTI THEN WRITE(LINE,12,IMAGE[KARD,*]);                51260000
                     ERROR(BADDOLAICD,1);                                       51270000
                     GO TO NEXTCARD                                             51280000
                     END;                                                       51290000
                     IF CHAR="#" OR CHAR="*" OR                                 51300000
                        CHAR="-" THEN                                           51310000
                          BEGIN                                                 51320000
THATSALL:                 ENDOFFILE[PATCH]:=RECORDNO;                           51330000
                          IF NOT FIRSTCARD THEN                                 51340000
                           IF (T:=RECORDNO-RECORD[PATCH]) LSS SIZE THEN         51350000
                               ERROR(MISSCDS,SIZE-T) ELSE                       51360000
                               IF T GTR SIZE THEN                               51370000
                                    ERROR(EXTRACDS,T-SIZE);                     51380000
                          IF CHAR="*" OR CHAR="-" THEN ERROR(BADCARD,REC);      51390000
                          IF CHAR="#" THEN                                      51400000
                              BEGIN                                             51410000
                              IF FIRSTCARD THEN ERROR(NOCDS,LASTPATCHNO);       51420000
                              GO TO NEXTDECK                                    51430000
                              END;                                              51440000
                          GO TO EGRESS                                          51450000
                          END ELSE CHECKVOID;                                   51460000
                     END;                                                       51470000
                IF LISTI THEN WRITE(LINE,12,IMAGE[KARD,*]);                     51480000
                IF NOT ERRORS AND SAVEIT THEN                                   51490000
                BEGIN                                                           51491000
                   WRITE(NEWDISK,10,IMAGE[KARD,*]);                             51500000
                   IF NEWFILES AND KARD = 2 THEN                                51501000
                   WRITE(PATCHFILE, 10, IMAGE[KARD,*]);                         51502000
                END;                                                            51503000
```

```
                IF FIRSTCARD THEN                                                   51510000
                     BEGIN                                                          51520000
                     READ (IMAGE[KARD,*],10,TOP[PATCH,*]);                          51530000
                     FIRSTCARD:=FALSE;                                              51540000
                     RECORD[PATCH]:=RECORDNO-1;                                     51550000
                     SEQUENCE:=SEQ[PATCH]:=                                         51560000
                               INTEGER(POINTER(TOP[PATCH,9]),8);                    51570000
                     GO TO NEXTCARD                                                 51580000
                     END;                                                           51590000
                IF SEQUENCE≥SEQUENCE:=INTEGER(POINTER(IMAGE[KARD,9]),8)             51600000
                     THEN ERROR(NOTINSEQ,LASTPATCHNO);                              51610000
                WRITE(MASTERFILE,10,IMAGE[KARD,*]);                                 51620000
                RECORDNO:=RECORDNO+1;                                               51630000
                GO TO NEXTCARD                                                      51640000
                END;                                                                51650000
           ERROR(BADCARD,1);                                                        51660000
EOF:                                                                                51670000
           HANDLEENDOFFILE;                                                         51680000
           CHAR:="#";                                                               51690000
           GO TO(IF PATCH<0 THEN NEXTDECK ELSE THATSALL); % MORE PATCHES.           51691000
EGRESS:                                                                             51700000
           IF PATCH NEQ PATCHMAX:=PATCHMAX-1 THEN                                   51710000
           IF PATCH LSS PATCHMAX THEN                                               51720000
                BEGIN                                                               51730000
                WRITE(LINE[DBL]);                                                   51740000
                ERROR(MISSPATS,PATCHMAX-PATCH)                                      51750000
                END ELSE                                                            51760000
                BEGIN                                                               51770000
                WRITE(LINE[DBL]);                                                   51780000
                ERROR(EXTRAPATS,PATCH-PATCHMAX)                                     51790000
                END;                                                                51800000
           IF NOT ERRORS AND SAVEIT THEN                                            51810000
                BEGIN                                                               51820000
                CLOSE(OLDDISK,PURGE);                                               51830000
                CLOSE(NEWPATCHES,PURGE);                                            51840000
                IF NEWFILES THEN LOCK(PATCHFILE);                                   51841000
                LOCK(NEWDISK)                                                       51850000
                END                                                                 51860000
           END INPUTPHASE;                                                          51870000
      %                                                                             60000000
      COMMENT##############################################################################60010000
                          OUTPUTPHASE PROCEDURE                                     60020000
      ##############################################################################;60030000
      %                                                                             60040000
          PROCEDURE OUTPUTPHASE;                                                    60050000
           BEGIN                                                                    60060000
           REAL I,J,NEXTDOG,NEXTDOGSEQ,LASTDOG,LASTVOIDTRANGE;                      60070000
           POINTER INP;                                                             60080000
           BOOLEAN PASSEDNONE,FIRSTTIME,VOITAP,TOPVOITAP;                           60090000
           LABEL  NEXT,ENDOFLOOP,EOS;                                               60100000
      %#################################################################################60110000
           PROCEDURE REED(DECK);                                                    60120000
                VALUE DECK;                                                         60130000
                REAL DECK;                                                          60140000
                BEGIN                                                               60150000
                LABEL EGRESS;                                                       60160000
                REAL I;                                                             60170000
      %                                                                             60180000
                IF RECORD[DECK]<0 THEN GO TO EGRESS;                                60190000
                IF (I:=RECORD[DECK]:=RECORD[DECK]+1)≥ENDOFFILE[DECK]THEN            60200000
```

```
                    BEGIN                                                    60210000
                    RECORD[DECK]:=-ENDOFFILE[DECK];                          60220000
                    ENDSOFFILE:=ENDSOFFILE+1;                                60230000
                    REPLACE POINTER(TOP[DECK,*]) BY " " FOR 72,              60240000
                        "9" FOR 8;                                           60250000
                    SEQ[DECK]:=99999999;                                     60260000
                    GO TO EGRESS                                             60270000
                    END;                                                     60280000
                READ(MASTERFILE[1],10,TOP[DECK,*]);                          60290000
                SEQ[DECK]:=INTEGER(POINTER(TOP[DECK,9]),8);                  60300000
EGRESS:                                                                      60310000
            END READ A CARD FROM A PATCH DECK;                               60320000
%#############################################################################60330000
            PROCEDURE RITE(DECK);                                            60340000
                VALUE DECK;                                                  60350000
                REAL DECK;                                                   60360000
%                                                                            60370000
                BEGIN                                                        60380000
                IF NOT ANOND THEN IF EVOLVE="      " THEN REPLACE EVOLVE     60390000
                    BY "%",PATCHNUMBER[DECK] FOR 3 DIGITS,"-";               60400000
                IF LISTG THEN                                                60410000
                    BEGIN                                                    60420000
                    KARD:=MEDIUM[DECK];                                      60430000
                    REPLACE REVOLVER BY " ",WHICHONE[KARD]FOR 1," "FOR 2,    60440000
                            PATCHNUMBER[DECK] FOR 3 DIGITS," " FOR 9;        60450000
                    WRITE(SORTLINE,12,TOP[DECK,*]);                          60460000
                    END;                                                     60470000
                IF PUNCHG THEN WRITE(PUNCH,10,TOP[DECK,*]);                  60480000
                WRITE(PATCHDECK,10,TOP[DECK,*]);                             60490000
                PATCHRECORD:=PATCHRECORD+1;                                  60500000
                REED(DECK);                                                  60510000
                END OUTPUT OF A PATCH CARD;                                  60520000
%#############################################################################60530000
            PROCEDURE INSERTVOID(DECK,VOIDSEQUENCE,SEQUENCE);                60540000
                VALUE DECK,VOIDSEQUENCE,SEQUENCE;                            60550000
                REAL DECK,VOIDSEQUENCE,SEQUENCE;                             60560000
%                                                                            60570000
                BEGIN                                                        60580000
                ENDSOFFILE:=ENDSOFFILE-REAL(RECORD[DECK]<0);                 60590000
                RECORD[DECK]:=ABS(RECORD[DECK])-1;                           60600000
                REPLACE POINTER(TOP[DECK,*]) BY "$VOID ",                    60610000
                    VOIDSEQUENCE FOR 8 DIGITS," " FOR 57,                    60620000
                    SEQ[DECK]:=SEQUENCE FOR 8 DIGITS;                        60630000
            END CREATE A VOID CARD;                                          60640000
%#############################################################################60650000
            BOOLEAN PROCEDURE VOIDCARD(DECK);                                60660000
                VALUE DECK;                                                  60670000
                REAL DECK;                                                   60680000
                BEGIN                                                        60690000
                LABEL EGRESS;                                                60700000
                POINTER P;                                                   60710000
%                                                                            60720000
                IF NOT(VOIDCARD:=REAL(P:=POINTER(TOP[DECK,*]),1)="$")THEN    60730000
                    GO TO EGRESS;                                            60740000
                SCAN P:P+1 FOR COLUMN:71 WHILE = " ";                        60750000
                IF NOT(VOIDCARD:=REAL(P,4)="VOID") THEN                      60760000
                    GO TO EGRESS;                                            60770000
                IF NOT(VOITAP:=REAL(P,5)="VOIDT") THEN                       60780000
                SCAN P:P+4 FOR COLUMN:COLUMN-4 WHILE=" " ELSE                60790000
                SCAN P:P+5 FOR COLUMN:COLUMN-5 WHILE=" ";                    60800000
```

```
                IF VOIDCARD:=COLUMN GEQ 8 THEN                                  60810000
                    VOIDSEQ:=INTEGER(P,8);                                      60820000
EGRESS:                                                                         60830000
            END SCAN TO SEE IF IT IS A VOID CARD;                               60840000
%###########################################################################60850000
            PROCEDURE WRITECONFLICT(TYPE,DECK);                                 60860000
                VALUE TYPE,DECK;                                                60870000
                REAL TYPE,DECK;                                                 60880000
                BEGIN                                                           60890000
                POINTER P,INP;                                                  60900000
                INTEGER KARD;                                                   60910000
                LABEL EGRESS;                                                   60920000
%                                                                               60930000
                IF SEQ[DECK]=99999999 THEN GO TO EGRESS;                        60940000
                KARD:=MEDIUM[TOPDOG];                                           60950000
                INP:=POINTER(IMAGE[KARD,*]);                                    60960000
                IF FIRSTTIME THEN                                               60970000
                    BEGIN                                                       60980000
                    IF CONFLICTS AND NOT WCON THEN                              60990000
                        BEGIN                                                   61000000
                        WCON:=TRUE;                                             61010000
                        IF LISTI THEN WRITE(LINE[PAGE]);                        61020000
                        MASTHEAD(2)                                             61030000
                        END;                                                    61040000
                    WRITE(LINE[DBL]);                                           61050000
                    FIRSTTIME:=FALSE;                                           61060000
                IF TYPE NEQ 6 THEN                                              61061000
                BEGIN                                                           61062000
                    IF VOIDINGSEQ=0 THEN                                        61070000
                        REPLACE INP BY POINTER(TOP[TOPDOG,*]) FOR               61080000
                            10 WORDS;                                           61090000
                    REPLACE INP+80 BY " ",WHICHONE[KARD] FOR 1," "FOR 2,        61100000
                    PATCHNUMBER[TOPDOG] FOR 3 DIGITS,                           61110000
                        " CONFLICTED WITH:"," " FOR 10;                         61120000
                    WRITE(LINE,14,IMAGE[KARD,*]);                               61130000
                END;                                                            61131000
                    END;                                                        61140000
                REPLACE  P:INP BY POINTER(TOP[DECK,*]) FOR 10 WORDS,            61150000
                    " ",WHICHONE[KARD] FOR 1," " FOR 2,                         61160000
                PATCHNUMBER[DECK] FOR 3 DIGITS," " FOR 4;                       61170000
                CASE TYPE OF                                                    61180000
                    BEGIN                                                       61190000
                    REPLACE P BY "DISCARDED", " " FOR 11;                       61200000
                    REPLACE P BY "SEQUENCE CHANGED", " " FOR 4;                 61210000
                    REPLACE P BY "CREATED", " " FOR 13;                         61220000
                    REPLACE P BY "VOIDED", " " FOR 14;                          61230000
                    REPLACE P BY "VOID RANGE EXTENDED ";                        61240000
                    REPLACE P BY "VOID RANGE DECREASED";                        61250000
                    REPLACE P BY "RANGE WAS: ",VOIDINGSEQ FOR 8 DIGITS,         61251000
                        "  ";                                                   61252000
                    END;                                                        61260000
                WRITE(LINE,14,IMAGE[KARD,*]);                                   61270000
EGRESS:                                                                         61280000
                END OUTPUT OF CONFLICTS;                                        61290000
%###########################################################################61300000
            PROCEDURE PASS(DECK,TILSEQ);                                        61310000
                VALUE DECK, TILSEQ;                                             61320000
                REAL DECK,TILSEQ;                                               61330000
                BEGIN                                                           61340000
                REAL FIRSTSEQ,LASTSEQ;                                          61350000
```

```
                BOOLEAN DONE;                                                   61360000
                LABEL EGRESS;                                                   61370000
%                                                                               61380000
                FIRSTSEQ:=SEQ[DECK];                                            61390000
                DO   BEGIN                                                      61400000
                     IF SEQ[DECK]≥TILSEQ THEN GO TO EGRESS;                     61410000
                     LASTSEQ:=SEQ[DECK];                                        61420000
                     IF VOIDCARD(DECK) THEN                                     61430000
                          IF VOIDSEQ GTR TILSEQ THEN                            61440000
                               BEGIN                                            61450000
                               VOIDINGSEQ:=TILSEQ:=VOIDSEQ;                     61460000
                               LASTDOG:=DECK;                                   61470000
                               IF DONE:=CONFLICTS THEN                          61480000
                                    WRITECONFLICT(4,DECK);                      61490000
                               END;                                             61500000
                     IF CONFLICTS THEN                                          61510000
                          IF NOT DONE THEN                                      61520000
                               WRITECONFLICT(3,DECK);                           61530000
                     READ(DECK);                                                61540000
                     END UNTIL DONE:=FALSE;                                     61550000
EGRESS:                                                                         61560000
                END PASS CARDS FROM A PATCH DECK;                               61570000
%                                                                               70000000
COMMENT##########################################################################70010000
                     OUTPUTPHASE PROCEDURE DRIVER                               70020000
##########################################################################;     70030000
          IF PATCHMAX LSS 0 THEN                                                70040000
          BEGIN                                                                 70041000
               REPLACE POINTER(TOP[TOPDOG:=0,*]) BY                             70042000
                    "END.","  " FOR 66,"9" FOR 8;                              70043000
               GO TO ENDOFLOOP;                                                 70044000
          END;                                                                  70045000
          INP:=POINTER(IMAGE[KARD,*]);                                          70050000
NEXT:     TOPDOGSEQ:=SEQ[TOPDOG:=PATCHMAX];                                     70060000
          FIRSTTIME:=TRUE;                                                      70070000
          DO   BEGIN                                                            70080000
               PASSEDONCE:=TRUE;                                                70090000
               FOR I:=TOPDOG-1 STEP -1 UNTIL 0 DO                               70100000
                    BEGIN                                                       70110000
                    IF (J:=SEQ[I])<TOPDOGSEQ THEN                               70120000
                         BEGIN                                                  70130000
                         FIRSTTIME:=TRUE;                                       70140000
                         TOPDOGSEQ:=J;                                          70150000
                         TOPDOG:=I;                                             70160000
                         END                                                    70170000
                    ELSE                                                        70180000
                         IF J=TOPDOGSEQ THEN                                    70190000
                              BEGIN                                             70200000
                              IF VOIDCARD(I) THEN                               70210000
                                   IF VOIDSEQ≤(J+1) THEN                        70220000
                                        BEGIN                                   70230000
                                        IF CONFLICTS THEN                       70240000
                                             WRITECONFLICT(0,I);                70250000
                                        READ(I);                                70260000
                                        END                                     70270000
                                   ELSE                                         70280000
                                        BEGIN                                   70290000
                                        REPLACE POINTER(TOP[I,9]) BY            70300000
                                             SEQ[I]:=J+1 FOR 8 DIGITS;          70310000
                                        IF CONFLICTS THEN                       70320000
```

```
                                        WRITECONFLICT(1,I);                 70330000
                                    END                                     70340000
                               ELSE                                         70350000
                                    BEGIN                                   70360000
                                    REPLACE INP BY POINTER(TOP[TOPDOG,*]) 70370000
                                       FOR 10 WORDS;                        70380000
                                    IF CONFLICTS THEN                       70390000
                                       WRITECONFLICT(0,I);                  70400000
                                    READ(I);                                70410000
                                    END;                                    70420000
                               PASSEDNONE:=FALSE;                           70430000
                               END;                                         70440000
          END;                                                              70450000
     END UNTIL PASSEDNONE OR SEQ[TOPDOG] = 99999999;                        70460000
IF VOIDCARD(TOPDOG) THEN                                                    70470000
     BEGIN                                                                  70480000
     TOPVOITAP:=VOITAP;                                                     70490000
     IF CONFLICTS THEN                                                      70500000
          REPLACE INP BY POINTER(TOP[TOPDOG,*]) FOR 10 WORDS;               70510000
     VOIDINGSEQ:=VOIDSEQ;                                                   70520000
     IF NOT VOITAP THEN                                                     70530000
     DO   BEGIN                                                             70540000
          READ(TOPDOG);                                                     70550000
          IF CONFLICTS THEN                                                 70560000
               IF SEQ[TOPDOG]<VOIDSEQ THEN                                  70570000
                    WRITECONFLICT(3,TOPDOG);                                70580000
          END UNTIL SEQ[TOPDOG]≥VOIDSEQ;                                    70590000
     NEXTDOG:=0;                                                            70600000
     LASTDOG:=TOPDOG;                                                       70610000
     FOR I:=TOPDOG-1 STEP -1 UNTIL 0 DO                                     70620000
          PASS(I,VOIDINGSEQ);                                               70630000
     NEXTDOGSEQ:=VOIDINGSEQ;                                                70640000
     FOR I:=LASTDOG STEP 1 UNTIL PATCHMAX DO                                70650000
          IF SEQ[I]<NEXTDOGSEQ THEN                                         70660000
               IF VOIDCARD(I) THEN                                          70670000
                    BEGIN                                                   70680000
                    IF NOT VOITAP THEN                                      70690000
                    BEGIN                                                   70700000
                    IF VOIDSEQ≤VOIDINGSEQ THEN                              70710000
                         BEGIN                                              70720000
                         IF CONFLICTS THEN                                  70730000
                              WRITECONFLICT(0,I);                           70740000
                         END                                                70750000
                    ELSE                                                    70760000
                         BEGIN                                              70770000
                         VOIDINGSEQ:=SEQ[I];                                70780000
                         IF CONFLICTS THEN                                  70790000
                              WRITECONFLICT(5,I);                           70800000
                         END                                                70810000
                    END                                                     70820000
                    END                                                     70830000
               ELSE                                                         70840000
                    NEXTDOGSEQ:=SEQ[NEXTDOG:=I];                            70850000
IF NOT TOPVOITAP THEN                                                       70860000
IF NEXTDOG NEQ 0 THEN                                                       70870000
     BEGIN                                                                  70880000
     INSERTVOID(LASTDOG,NEXTDOGSEQ,TOPDOGSEQ);                              70890000
     IF CONFLICTS THEN                                                      70900000
          BEGIN                                                             70910000
          FIRSTTIME:=TRUE;                                                  70920000
```

```
                    REPLACE INP BY POINTER(TOP[TOPDOG:=NEXTDOG,*])                70930000
                         FOR 10 WORDS;                                            70940000
                    WRITECONFLICT(2,LASTDOG);                                     70950000
                    END;                                                          70960000
               RITE(LASTDOG);                                                     70970000
               INSERTVOID(LASTDOG,VOIDINGSEQ,NEXTDOGSEQ+1);                       70980000
               IF CONFLICTS THEN                                                  70990000
                    WRITECONFLICT(2,LASTDOG);                                     71000000
               GO TO ENDOFLOOP;                                                   71010000
               END;                                                               71020000
          IF NOT TOPVOITAP THEN                                                   71030000
               INSERTVOID(TOPDOG,VOIDINGSEQ,TOPDOGSEQ)                            71040000
          ELSE BEGIN                                                              71041000
               REPLACE POINTER(TOP[TOPDOG,*]) BY "$VOIDT ",                       71042000
                 MAX(LASTVOIDTRANGE,VOIDINGSEQ) FOR 8 DIGITS,                     71043000
                 " " FOR 57,TOPDOGSEQ FOR 8 DIGITS;                               71044000
                    IF VOIDINGSEQ LSS LASTVOIDTRANGE THEN                         71045000
                         IF CONFLICTS THEN WRITECONFLICT(6,TOPDOG) ELSE           71046000
                    ELSE LASTVOIDTRANGE:=VOIDINGSEQ;                              71047000
               END;                                                               71048000
               RITE(TOPDOG);                                                      71050000
               VOIDINGSEQ:=0;                                                     71060000
               END                                                                71070000
          ELSE                                                                    71080000
               RITE(TOPDOG);                                                      71090000
ENDOFLOOP:                                                                        71100000
          IF ENDSOFFILESPATCHMAX THEN GO TO NEXT;                                 71110000
          RITE(TOPDOG);        COMMENT BE SURE THERE IS A 9"S CARD;               71120000
          REPLACE POINTER(TOP[TOPDOG,*]) BY QUEST, "END.", " " FOR 90;            71130000
          TOP[TOPDOG,9]:=PATCHRECORD;                                             71140000
          IF LISTG THEN WRITE(SORTLINE,12,TOP[TOPDOG,*]);                         71150000
          WRITE(PATCHDECK,10,TOP[TOPDOG,*]);                                      71160000
          LASTCONTROLCARDIMAGE[9]:=PATCHRECORD;                                   71170000
          WRITE(PATCHDECK[LASTCONTROLCARD],10,LASTCONTROLCARDIMAGE[*]);           71180000
          IF LISTG THEN                                                           71190000
             WRITE(SORTLINE[LASTCONTROLCARD],10,LASTCONTROLCARDIMAGE[*]);         71200000
          IF GOTOIT THEN ZIP WITH PATCHDECK; %FIRE UP BEFORE PRINTING PAT         71210000
          IF LISTG THEN                                                           71220000
               BEGIN                                                              71230000
               REWIND(SORTLINE);                                                  71240000
               IF LISTI OR CONFLICTS THEN WRITE(LINE[PAGE]);                      71250000
               MASTHEAD(1);                                                       71260000
               WHILE TRUE DO                                                      71270000
                    BEGIN                                                         71280000
                    READ(SORTLINE,12,IMAGE[KARD,*]) [EOS];                        71290000
                    WRITE(LINE,12,IMAGE[KARD,*])                                  71300000
                    END                                                           71310000
               END;                                                               71320000
          EOS:                                                                    71330000
          END OUTPUT PHASE;                                                       71340000
%                                                                                 80000000
COMMENT###################################################################80010000
                    PROCESS PROCEDURE DRIVER                                      80020000
###################################################################;             80030000
%                                                                                 80040000
     REPLACE POINTER(NAME[1]) BY "DPC";                                           80050000
     REPLACE POINTER(LASTCONTROLCARDIMAGE[10]) BY                                 80060000
        " " FOR 2 WORDS;                                                          80070000
     REPLACE POINTER(IMAGE[KARD,10]) BY " " FOR 2 WORDS;                          80080000
     INPUTPHASE;                                                                  80090000
```

```
      CLOSE(CARD);                                                          80100000
      IF ERRORS THEN GC AXIT;                                               80110000
      REWIND(MASTERFILE);                                                   80120000
      MASTERFILE.ACCESS:=RANDOM;                                            80130000
      OUTPUTPHASE;                                                          80140000
      END OF PROCESS;                                                       80150000
%                                                                           90000000
COMMENT##############################################################################90010000
                    OUTER BLOCK DRIVER                                      90020000
##############################################################################;90030000
%                                                                           90040000
TIME1:=TIME(1);                                                             90050000
NOHEADING:=NOMAST:=TRUE;                                                    90060000
KARD:=2;                                                                    90070000
CARDSONLY:=TRUE;                                                            90080000
READ(CARD,10,IMAGE[KARD,*]);                                                90090000
IF REAL(Q1:=POINTER(IMAGE[KARD,*]),2)="$@" THEN SETOPTIONS ELSE             90100000
     LISTI:=LISTG:=CONFLICTS:=GOTOIT:=TRUE; % DEFAULT OPTIONS.              90110000
GETDOLPERCD;                                                                90120000
PROCESS;                                                                    90130000
AXIT:                                                                       90140000
IF NOT NOHEADING THEN                                                       90150000
     BEGIN                                                                  90160000
     FORMAT PAN("PROCESSOR TIME =",I5," SECONDS.",                          90170000
               /"I/O TIME =",I5," SECONDS."),                               90180000
            E(////"NUMBER OF ERRORS DETECTED =",F4.0);                      90190000
     IF NOT ERRORS THEN WRITE(LINE[DBL]) ELSE                               90200000
          BEGIN                                                             90210000
          FILE SPO 11(1,10);                                                90220000
          WRITE(SPO,<13," ERRORS DETECTED.">,NUMERR);                       90230000
          END;                                                              90240000
     WRITE(LINE,E,NUMERR);                                                  90250000
     WRITE(LINE,PAN,TIME(2)/60,TIME(3)/60)                                  90260000
     END                                                                    90270000
END OF PATCH MERGE.                                                         90280000
END;END.        LAST CARD ON OCRDING TAPE                                   99999999
```

LABEL 000000000PRINTER001/5099CC EX OBJECT/READ;FILE SOURCEFILE=SYMBOL/PMERGE;END+ OBJECT /READ