FPB3 DIAGNOSTIC LOADER FOR DISK

```
ERR LOC OBJECT CODE      ADDR STMT SOURCE STATEMENT

                               2         DECK  4
   0000                        3 FPB     START 0
   0015                        4         ORG   X'015'
                               5 ********************************************************************************
                               6 *                          CONTROL PROGRAM LOADER                              *
                               7 ********************************************************************************
                               8 *                                                                              *
                               9 *  THIS PROGRAM MODULE LOADS THE CONTROL PROGRAM, INCLUDING ITS SEC-           *
                              10 *  TION REFERENCE TABLE.  A ONE SECTOR BOOTSTRAP LOADER READS THIS             *
                              11 *  LOADER INTO CORE & BRANCHES TO IT.  LOAD CARDS RECOGNIZED BY THE            *
                              12 *  DCP LOADER INCLUDE                                                          *
                              13 *                                                                              *
                              14 *       TEXT                                                                   *
                              15 *       END                                                                    *
                              16 *       CPU                                                                    *
                              17 *       UDT                                                                    *
                              18 *                                                                              *
                              19 *  OTHER CARDS ARE IGNORED.                                                    *
                              20 *                                                                              *
                              21 ********************************************************************************
                              22
                         0018 23         USING BASE1,2
   0015 40404040         0018 24 ID      DC    CL4' '
   0019 C2 02 0018             25 ENT2    IA    BASE1,XR2               LOAD XR2 AS BASE
   001D B4 08 63               26         ST    MEXIT+3(,XR2),ARR       SAVE RETURN ADDRESS FOR 'M' CARDS
                         0020  27 SETWOK  EQU   *
   0020 C0 87 021E             28         B     UNPACK                  TO UNPACK ROUTINE
   0024 02               0024  29 NTWO    DC    XL1'02'                 LENGTH
   0025 01FF             0026  30 SOURCE  DC    AL2(DTABLE+1)           SOURCE ADDRESS
   0027 0018             0028  31         DC    AL2(ID)                 DESTINATION ADDRESS
                               32
   0029 BC 05 9A               33 SETSCN  MVI   RDDFC+1(,XR2),08        SET TO BEGIN SCAN AT CYLINDER 0
   002C BC 00 9B               34         MVI   RDDFC+2(,XR2),0         SET TO BEGIN AT HEAD 0 SECTOR 0.
   002F E0 87 BD               35 RDVTOC  B     STRTIO(,XR2)            TO I/O SUBROUTINE TO SEEK
   0032 00               0032  36 SEEKP1  DC    XL1'00'                 FUNCTION CODE
   0033 38 02 01FD             37         TBN   FLAG,BIT6
   0037 F2 90 03               38         JP    **6
   003A B5 10 63               39         L     MEXIT+3(,XR2),IAR       EXIT AFTER SEEKING TO VTOC
   003D E0 87 7E               40         B     READ(,XR2)              READ A VTOC ENTRY
   0040 6D 02 02 A7            41         CLC   2(,XR1),ACTCON(3,XR2)
   0044 F2 01 07               42         JNE   CKND
   0047 6D 02 07 00            43         CLC   7(,XR1),ID(3,XR2)
   004B F2 81 1B               44         JE    IDFND                   ID HAS BEEN FOUND
   004E BD 0C 9A               45 CKND    CLI   RDDFC+1(,XR2),X'0C'     TEST END OF VTOC
   0051 F2 81 03               46         JE    HD                      BRANCH IF ALL CYLINDERS ARE DONE
   0054 E0 87 17               47         B     RDVTOC(,XR2)
   0057 38 01 01FD             48 HD      TBN   FLAG,BIT7
   005B F2 10 07               49         JT    LDR
   005E 35 08 01FF             50         L     DTABLE+1,ARR
   0062 F0 3B 73               51 HPLHD   HPL   X'73',X'3B'
   0065 C0 87 0000             52 LDR     B     *-*
                         0068  53 XREF1   EQU   *-1
                               54
                               54
                               54
                               54
   0069 9C 01 9B 04            55 IDFND   MVC   RDDFC+2(2,XR2),4(,XR1)   PLACE PGM. ADDR. IN CONTROL FLD.
                               56
   006D E0 87 BD               57 PGMLOD  B     STRTIO(,XR2)            TO I/O SUBROUTINE TO SEEK
   0070 00               0070  58 SEEKP2  DC    XL1'00'                 FUNCTION CODE
   0071 E0 87 7E               59         B     READ(,XR2)              TO READ ROUTINE
                               60
   0074 38 20 01FD             61         TBN   FLAG,BIT2
   0078 C0 10 007C             62 MEXIT   BT    **4                     EXIT IF BIT 2 IS ON
                               63
   007C 7D E3 00               64         CLI   0(,XR1),C'T'            IS THIS A TEXT CARD ?
   007F C0 01 0A0B             65 CKCOM4  BNE   CKCOM1                  BR   TO NEXT TEST IF NOT.
                               66 **  NOTE - BRANCH ADDRESS OF PREVIOUS BRANCH IS ALTERED AFTER DCP
```

FFB3 DIAGNOSTIC LOADER FOR DISK

```
ERR LOC OBJECT CODE      ADDR STMT SOURCE STATEMENT

                               67 **  LOADING COMPLETE TO PREPARE FOR SECTION LOADING.                *
   0083 9C 02 79 03            68         MVC   MOVE+3(3,XR2),3(,XR1) SET DESTINATION ADDRESS IN TEXT MOVE
   0087 9C 00 7A 01            69         MVC   MOVE+4(1,XR2),1(,XR1) PLACE INSTRUCTION LENGTH IN MOVE
   008B D2 01 04               70         LA    4(,XR1),XR1           INCREASE POINTER BY 4.
   008E 1C 00 0000 00          71 MOVE    MVC   *-*(*-*),*-*(,XR1)    MOVE TEXT DATA TO CORE
   0093 E0 87 55               72         B     PGMLOD(,XR2)          GO READ NEXT CARD
                               73
   0096 B? 08 98               74 READ    ST    REDEXT+3(,XR2),ARE    SAVE EXIT ADDRESS
   0099 B? 00 9C               75         MVI   RDDFC+3(,XR2),00      SET TO READ ONE SECTOR
   009C E0 87 BD               76         B     STRTIO(,XR2)          TO I/O SUBROUTINE TO READ
   009F 01                009F 77 READF  DC    XL1'01'               FUNCTION CODE
   00A0 AE 01 9B A3            78 ALC     ALC   RDDFC+2(2,XR2),ONESEC(,XR2)     STEP SECTOR NUMBER +1
   00A4 B8 60 9E               79         TBN   RDDFC+2(,XR2),X'60'             TEST FOR SECTOR NO'S
                               80 *                                            24 THRU 31.
   00A7 E0 10 88               81         BT    ALC(,XR2)                     IF EQUAL, STEP AGAIN
   00AA B5 01 9E               82         L     DFDR(,XR2),XR1        LOAD XR1 WITH ADDRESS OF READIN AREA
   00AD C0 87 0000             83 REDEXT  B     *-*                   EXIT
                               84
   00B1 00                 00B1 85 RDDFC  DC    XL1'00'               FLAG
   00B2 07                 00B2 86        DC    XL1'07'               CYLINDER
   00B3 04                 00B3 87        DC    XL1'04'               TRACK & SECTOR NO.
   00B4 00                 00B4 88        DC    XL1'00'               NUMBER OF SECTORS OR TRACKS
   00B5 0880               00B6 89 DFDR   DC    AL2(WORK)
   00B7 0000               00B8 90 SENSE  DC    XL2'0'
   00B9 0000               00BA 91 TWOZR  DC    XL2'0'
   00BB 04                 00BB 92 ONESEC DC    XL1'4'
   00BC 00                 00BC 93 SECTNO DC    XL1'0'
   00BD C1C3E3             00BF 94 ACTCON DC    CL3'ACT'
   00C0 0000               00C1 95 EXTSAV DC    XL2'0'
   00C2 00B1               00C3 96 NRMDFC DC    AL2(RDDFC)
   00C4 00C6               00C5 97 RECAB@ DC    AL2(*+2)   ********************************
   00C6 000000FF           00C9 98        DC    XL4'FF'    * MUST BE TOGETHER            *
                               99 *                         ********************************
   00CA 00CC               00CB 100 ALTDFC DC    AL2(ARDDFC)
   00CC 00                 00CC 101 ARDDFC DC    XL1'0'
   00CD 00                 00CD 102        DC    XL1'0'
   00CE 00                 00CE 103        DC    XL1'0'
   00CF 00                 00CF 104        DC    XL1'0'
   00D0 00D2               00D1 105 LAST@  DC    AL2(*+2)   ********************************
   00D2 00                 00D2 106        DC    XL1'0'     * MUST BE TOGETHER            *
   00D3 00                 00D3 107 LAST AD DC   XL1'0'     *                             *
   00D4 00                 00D4 108        DC    XL1'0'     *******************************
```

```
ERR LOC OBJECT CODE      ADDR STMT SOURCE STATEMENT

                          110 *            START I/O SUBROUTINE
                          111
                    00D5  112 STRTIO EQU   *
   00D5 B5 02 AB          113        L     NRMDFC(,XR2),XR2       LOAD BASE VALUE FOR START I/O
                    00B1  114        USING RDDFC,2
   00D9 B1 A6 12          115        LIO   NRMDFC(,XR2),CTLREG    LOAD CONTROL REG.
   00DF B4 08 9A          116 STPARM ST    SETXR1(,XR2),ARR       SAVE ADDRESS RECALL REGISTER
   00E2 B5 01 9A          117 LDPARM L     SETXR1(,XR2),XR1       LOAD XR1 AS PARAMETER POINTER
   00E5 9C 00 8C 00       118        MVC   SIO+1(1,XR2),0(,XR1)   SET FUNCTION CODE IN SIO
   00E9 B0 A6 07          119        SNS   SENSE(,XR2),CTLREG     SENSE THE CURRENT CTL. FLD. ADDR.
   00EC 79 07 00          120        TBF   0(,XR1),X'07'          TEST FOR SEEK
   00EF B5 01 07          121        L     SENSE(,XR2),XR1        LOAD XR1 AS POINTER
                          122 *                                   TO DISK CONTROL FIELD.
   00F2 F2 90 48          123        JF    ADREXT                 NO, SKIP SET ADDRESS.
                          124
   00F5 B1 A6 20          125        LIO   LAST@(,XR2),CTLREG
   00F8 F3 A1 01          126        SIO   1,X'A1'                READ ID
   00FB E1 A2 46          127        TIO   *(,XR2),BUSY
   00FE B9 03 21          128        TBF   LASTAD-1(,XR2),03         TEST FOR ALT. OR DEFECT.
   0101 F2 10 0C          129        JT    OK                        THE CYL IS OK
   0104 B1 A6 14          130        LIO   RECAB@(,XR2),CTLREG
   0107 F3 A0 00          131        SIO   0,X'A0'                   RECABRIATE
   010A E1 A2 55          132        TIO   *(,XR2),BUSY
   010D BC 00 22          133        MVI   LASTAD(,XR2),0
   0110 B1 A6 07          134 OK     LIO   SENSE(,XR2),CTLREG     RESTORE CTLREG
   0113 7C 00 03          135        MVI   3(,XR1),0              SET SEEK # TO 0
   0116 6D 00 01 22       136        CLC   1(,XR1),LASTAD(1,XR2)  COMPARE NEW ADDRESS WITH OLD
   011A F2 81 20          137        JE    ADREXT                 EQUAL, SEEK NOT NECESSARY
   011D F2 84 0B          138        JH    FWDSEK                 NEW ADDR. HIGHER, DO FORWARD SEEK
   0120 AC 00 82 22       139        MVC   SCRCH(1,XR2),LASTAD(,XR2)  PLACE LAST ADDR. IN WORKAREA
   0124 9F 00 82 01       140        SLC   SCRCH(1,XR2),1(,XR1)   SUBTRACT NEW ADDR. FROM LAST
   0128 F2 87 0B          141        J     SETADR                 PROCEED
                          142
   012B 7A 01 02          143 FWDSEK SBN   2(,XR1),01             SET BIT ON FOR FORWARD SEEK
   012E 9C 00 82 01       144        MVC   SCRCH(1,XR2),1(,XR1)   PLACE NEW ADDRESS IN WORK AREA
   0132 AF 00 82 22       145        SLC   SCRCH(1,XR2),LASTAD(,XR2)  SUBTRACT LAST ADDR. FROM NEW
                          146
                    0137  147 SCRCH  EQU   *+1
   0136 7C 00 03          148 SETADR MVI   3(,XR1),*-*            INSERT NO. OF TRACKS TO CROSS
   0139 9C 00 22 01       149        MVC   LASTAD(1,XR2),1(,XR1)  SAVE NEW ADDRESS
                    013D  150 ADREXT EQU   *
                          151
   013D B1 A4 05          152        LIO   DFDR(,XR2),X'A4'       LOAD DATA REGISTER
   0140 F3 00 00          153 SIO    SIO   0,0                    START I/O OPERATION
   0143 E1 A2 8E          154 WAIT   TIO   *(,XR2),BUSY           WAIT TILL COMPLETE
                          155
   0146 7B 01 02          156        SBF   2(,XR1),01             TURN OFF FOR. / REV. BIT
                          157
   0149 E1 A0 A1          158 TSTERR TIO   DSKERR(,XR2),NTRDY     BRANCH IF ERROR
                    014D  159 SETXR1 EQU   *+3
   014C C2 01 0000        160        LA    *-*,XR1                LOAD XR1 AS PARAMETER POINTER
   0150 B5 02 D9          161        L     RECARD+7(,XR2),XR2     LOAD XR2 AS BASE
                    0018  162        USING BASE1,2
   0153 D0 87 01          163        B      (,XR1)                EXIT
                          164
                    00B1  165        USING RDDFC,2
                    0156  166 DSKERR EQU   *
   0156 B1 A6 1A          167        LIO   ALTDFC(,XR2),CTLREG    LOAD CTL. REG. TO POINT TO ALT.FLD.
   0159 F3 A1 01          168 ALTSIO SIO   01,X'A1'               READ ID SIO
   015C E1 A2 A7          169 WAIT1  TIO   *(,XR2),BUSY           DELAY TILL FINISHED
   015F BD 02 1B          170        CLI   ARDDFC(,XR2),02        TEST FOR A DEFECTIVE TRACK
   0162 F2 81 09          171        JE    SETALT                 JUMP IF IT IS.
   0165 F0 3B 07          172 ERRHLT HPL   H7,HH                  ERROR HALT
   0168 B1 A6 12          173        LIO   NRMDFC(,XR2),CTLREG    LOAD CONTROL REG.
   016B F2 87 03          174        J     SAVPRM
   016E BC 01 1B          175 SETALT MVI   ARDDFC(,XR2),01              SET FLAG BIT TO ALTERNATE
   0171 AC 01 10 9A       176 SAVPRM MVC   EXTSAV(2,XR2),SETXR1(,XR2)    SAVE OLD PARAMETER PTR.
   0175 E0 87 2A          177        B     STPARM(,XR2)           TO I/O SUBROUTINE TO SEEK ALT. TK.
```

FFB3 DIAGNOSTIC LOADER FOR DISK

```
ERR LOC OBJECT CODE      ADDR STMT SOURCE STATEMENT

   0174 00               0174  178 SEEKF3 DC     XL1'00'                FUNCTION CODE
                         0018  179        USING  BASE1,2
   0175 B5 02 AB               180        L      NRMDFC(,XR2),XR2       L  D BASE VALUE FOR START I/O
                         00B1  181        USING  RDDFC,2
   0178 AC 01 9A 10            182        MVC    SETXR1(2,XR2),EXTSAV(,XR2)     RESTORE PAREMETER PTR.
   017C AC 02 1E 03            183        MVC    ARDDFC+3(3,XR2),RDDFC+3(,XR2)  TRANSFER C,S,N NUMBERS
   0180 E0 87 2D               184        B      LDPARM(,XR2)           TO RETRY ORIGINAL OPERATION
                               185
                               186 *              COME HERE TO READ A DATA RECORD
                               187
                         0018  188        USING  BASE1,2
   0183 34 08 0195             189 RECARD ST      RECEXT+3,ARR       SAVE ADDRESS RECALL REGISTER
   0187 C2 02 0018             190        LA     BASE1,XR2              LOAD XR2 AS BASE
   018B E0 87 BD               191        B      STRTIO(,XR2)           TO I/O SUBROUTINE TO SEEK
   018E 00               018E  192 SEEKF4 DC     XL1'00'                FUNCTION CODE
   018F E0 87 7E               193        B      READ(,XR2)         TO READ DATA RECORD
   0192 C0 87 0000             194 RECEXT 3      *-*                EXIT
                               195
                               196 **  THE FOLLOWING CODING COMPLETES THE SECTION LOADER.  IT IS          *
                               197 **  BYPASSED DURING DCP LOADING.  ONCE THE CONTROL PROGRAM IS LOADED,*
                               198 **  LINKAGES ARE SET UP SO THAT A BRANCH TO THE END CARD ROUTINE WILL*
                               199 **  BE SUBSTITUTED FOR THE DCP LOADER RESIDING AT HEX -A00-           *
                               200
   0196 7D 5C 00               201 CKCOM  CLI    0(,XR1),C'*'           JUMP   IF THIS IS NOT A COMMENT   *
   0199 F2 01 0F               202        JNE    CHKSSW                  CARD                             *
   019C 38 01 0208             203        TBN    SBYTE0,SSW07
   01A0 E0 10 55               204        BT     PGMLOD(,XR2)
   01A3 C0 87 021A             205        B      PRINT                  PRINT CONTENTS OF THIS CARD
   01A7 21               01A7  206        DC     XL1'21'                FLAGS
   01A8 E0 87 55               207        B      PGMLOD(,XR2)           GO READ NEXT CARD
                         020D  208 SBYTE5 EQU    X'20D'
   01AB 7D E2 00               209 CHKSSW CLI    0(,XR1),C'S'           SSW?
   01AE F2 01 27               210        JNE    CHKSE                  NO
   01B1 0F 03 020D 020D        211 SSWRD  SLC    SBYTE5(4),SBYTE5       YES, CLEAR SSW10 THRU SSW2F
   01B7 D2 01 05               212        LA     5(,XR1),XR1            PICK UP @ OF FIRST SSW
   01BA 34 01 01C4             213 CHKSS0 ST     SADDR,XR1              STORE @ TO PACK SSW
   01BE C0 87 0226             214        B      PACK
   01C2 02               01C2  215        DC     XL1'2'
   01C3 0000             01C4  216 SADDR  DC     AL2(*-*)
   01C5 0000             01C6  217 XREF5  DC     AL2(*-*)
   01C7 C0 87 0000             218        B      *-*
                         01CA  219 XREF4  EQU    *-1
   01CB 7D 6B 01               220        CLI    1(,XR1),C','           CHECK FOR ANOTHER SSW
   01CE D2 01 03               221        LA     3(,XR1),XR1            INCREMENT @ TO PICK UP NEXT SSW
   01D1 C0 81 01BA             222        BE     CHKSS0                 DO IT AGAIN IF ,
   01D5 E0 87 55               223        B      PGMLOD(,XR2)
   01D8 7D C5 00               224 CHKSE  CLI    0(,XR1),C'E'            IS THIS AN END CARD ?
   01DB E0 01 55               225        BNE    PGMLOD(,XR2)            GO READ NEXT CARD IF NO
   01DE 38 04 01FD             226        TBN    FLAG,BIT5
   01E2 D0 90 81               227        BF     129(,XR1)               BRANCH TO X'901'
   01E5 B5 01 63               228        L      NEXIT+3(,XR2),XR1
   01E8 D0 87 00               229        B      0(,XR1)                 RETURN TO DCP
                               230
                         0018  231 BASE1  EQU    ID
                         0880  232 WORK   EQU    X'880'
                         01FD  233 FLAG   EQU    X'1FD'
                         01FE  234 DTABLE EQU    X'1FE'
                         0068  235 LOADTB EQU    X'68'
```

```
ERR LOC OBJECT CODE       ADDR STMT SOURCE STATEMENT

   0A00                          237         ORG   X'A00'
                                 238
   0A00 0001              0A01   239 ONE     DC    IL2'1'
   0A02 09                0A02   240 NINE    DC    XL1'09'
   0A03 00                0A03   241 DEV     DC    XL1'0'
   0A04 0000              0A05   242 MASK    DC    XL2'0'
   0A06 C3C8C1C9D5        0A0A   243 CHAIN   DC    CL5'CHAIN'
                                 244
   0A0B 7D 5C 00                 245 CKCOM1  CLI   0(,XR1),C'*'      BRANCH IF NOT COMMENT CARD      *
   0A0E F2 01 13                 246         JNE   CKCPU                                           *
   0A11 C0 87 0B6F               247         B     DEFINE                                          *
   0A15 38 01 0208               248         TBN   SBYTE0,SSW07                                    *
   0A19 E0 10 55                 249         BT    PGMLOD(,XR2)      GO READ NEXT CARD             *
   0A1C C0 87 021A               250         B     PRINT                                           *
   0A20 21                0A20   251         DC    XL1'21'                                         *
   0A21 E0 87 55                 252         B     PGMLOD(,XR2)                                    *
   0A24 7D C3 00                 253 CKCPU   CLI   0(,XR1),C'C'      BRANCH IF NOT CPU DEFINITION CARD
   0A27 F2 01 2E                 254         JNE   CKUDT
   0A2A 1C 00 0200 04            255         MVC   SMOD(1),4(,XR1)   PUT SYSTEM MODEL INTO SRT
   0A2F C0 87 0B6F               256         B     DEFINE
   0A33 C0 87 0226               257         B     PACK              PACK CORE SIZE INTO SRT
   0A37 04                0A37   258         DC    IL1'4'
   0A38 0B89              0A39   259         DC    AL2(INPUT+9)
   0A3A 0203              0A3B   260         DC    AL2(SIZE)
   0A3C 7D F1 0B                 261         CLI   11(,XR1),C'1'     COLUMN 12 CONTAINS '1'
   0A3F F2 01 04                 262         JNE   *+7
   0A42 3A 80 0204               263         SBN   CPU,X'80'
   0A46 7D C2 04                 264         CLI   4(,XR1),C'B'      MAKE SURE SYSTEM MODEL IS B,C, OR D
   0A49 F2 82 06                 265         JL    CDERR0
   0A4C 7D C4 04                 266         CLI   4(,XR1),C'D'
   0A4F F2 04 3C                 267         JNH   NEXTR             YES, GO READ NEXT CARD
   0A52 F0 3B 6F                 268 CDERR0  HPL   H0,HH            *NO, CARD SET UP IMPROPERLY
   0A55 12 87 36                 269         J     NEXTR             GO READ NEXT CARD AFTER HALT RESET
                                 270
   0A58 7D E4 00                 271 CKUDT   CLI   0(,XR1),C'U'      BRANCH IF NOT UDT CARD
   0A5B F2 01 AD                 272         JNE   CKCHN
   0A5E C0 87 0B6F               273         B     DEFINE
   0A62 7D 40 03                 274         CLI   3(,XR1),C' '      CHECK THIRD COLUMN FOR BLANK
   0A65 F2 01 06                 275         JNE   PTRDC             JUMP IF NOT BLANK
   0A68 0F 2D 025F 025F          276         SLC   UTAB+45(46),UTAB+45  OUT PREVIOUS UDT ENTRIES
   0A6E D2 01 05                 277 PTRDC   LA    5(,XR1),XR1       POINT AT FIRST DEVICE CODE
   0A71 34 01 0A7B               278 ULP1    ST    UPTR,XR1          SET UP DEVICE CODE POINTER
   0A75 C0 87 0226               279         B     PACK              PACK DEVICE CODE
   0A79 02                0A79   280         DC    IL1'2'
   0A7A 0000              0A7B   281 UPTR    DC    AL2(*-*)
   0A7C 0A03              0A7D   282         DC    AL2(DEV)
   0A7E C2 02 022F               283         LA    UTAB-3,XR2        POINT AT DCP UNIT TABLE
   0A82 E2 02 03                 284 ULP2    LA    3(,XR2),XR2       INCREMENT UNIT TABLE POINTER
   0A85 B8 10 01                 285         TBN   1(,XR2),BIT3      BRANCH IF NOT LAST DCP ENTRY
   0A88 F2 90 0A                 286         JP    UDTA
   0A8B F0 3B 6F                 287         HPL   H0,HH            *RAN OUT ROOM IN UDT TABLE
   0A8E C2 02 0018               2     NEXTR LA    BASE1,XR2         LOAD XR2 AS BASE
   0A92 E0 87 55                 2           B     PGMLOD(,XR2)      GO READ NEXT CARD
   0A95 8D 00 00 0A03            2     UDTA  CLC   0(1,XR2),DEV      BRANCH TO OVERLAY IF THIS IS SAME AS
   0A9A F2 81 07                 291         JE    LDUDT               PREVIOUS ENTRY
   0A9D BD 00 00                 292         CLI   0(,XR2),X'0'      IS THIS AN UNUSED ENTRY
   0AA0 C0 01 0A82               293         BNE   ULP2              IF NOT UNUSED, GO CHECK NEXT
   0AA4 8C 00 00 0A03            294 LDUDT   MVC   0(1,XR2),DEV      SET UP THIS UDT ENTRY DEVICE CODE
   0AA9 BB 0F 01                 295         SBP   1(,XR2),X'0F'     CLEAR OPTION BITS
   0AAC BC 00 02                 296         MVI   2(,XR2),X'0'
   0AAF D2 01 01                 297         LA    1(,XR1),XR1
   0AB2 7D 60 00                 298         CLI   0(,XR1),C'-'      FOR DASH
   0AB5 F2 01 03                 299         JNE   *+6
   0AB8 D2 01 01                 300 ULP8    LA    1(,XR1),XR1
   0ABB 7D 40 00                 301         CLI   0(,XR1),C' '      IF BLANK ENCOUNTERED, CARD IS DONE
   0ABE C0 81 0A8E               302         BE    NEXTR
   0AC2 7D 6B 00                 303         CLI   0(,XR1),C','      IF COMMA ENCOUNTERED, GO TO NEXT
   0AC5 F2 01 07                 304         JNE   UDTB                DEVICE CODE
```

FFB3 DIAGNOSTIC LOADER FOR DISK

```
ERR LOC OBJECT CODE       ADDR STMT SOURCE STATEMENT

   0AC8 D2 01 02                  305         LA    2(,XR1),XR1
   0ACB C0 87 0A71                306         B     ULP4
   0ACF 7D F0 00                  307 UDT8    CLI   0(,XR1),X'F0'          CHANGE EBCDIC 0-8 TO BINARY
   0AD2 F2 02 05                  308         JNL   UDTC
   0AD5 4E 00 00  A02             309         ALC   0(1,XR1),NINE
   0ADA 7B F0 00                  310 UDTC    SBP   0(,XR1),X'F0'
   0ADD 0C 01 0A05 0A01           311         MVC   MASK(2),ONE            INITIALIZE MASK WITH LAST BIT ON
   0AE3 4F 00 00 0A01             312 ULP3    SLC   0(1,XR1),ONE           SHIFT BIT TO PROPER POSITION TO
   0AE8 F2 82 0A                  313         JL    UDTD
   0AEB 0E 01 0A05 0A05           314         ALC   MASK(2),MASK
   0AF1 C0 87 0AE3                315         B     ULP3
   0AF5 0C 00 0B02 0A04           316 UDTD    MVC   USET1+1(1),MASK-1      LOAD MASK INTO SET BITS ON
   0AFB 0C 00 0B05 0A05           317         MVC   USET2+1(1),MASK         INSTRUCTIONS
   0B01 BA 00 01                  318 USET1   SBN   1(,XR2),*-*            TURN ON PROPER OPTION BIT
   0B04 BA 00 02                  319 USET2   SBN   2(,XR2),*-*
   0B07 C0 87 0AB8                320         B     ULP4                   GO LOOK AT NEXT OPTION NUMBER
                                  321
   0B0B 4D 04 07 0A0A             322 CKCHN   CLC   7(5,XR1),CHAIN         BRANCH IF NOT CHAIN IMAGE CONTROL
   0B10 F2 01 17                  323         JNE   CKSSW                    CARD
   0B13 C0 87 0226                324         B     PACK                   PACK THIS PRINT IMAGE DATA CARD
   0B17 F0              0B17      325         DC    IL1'240'                 INTO PROPER LOCATION
   0B18 097B            0B19      326         DC    AL2(INPUT+251)         SOURCE
   0B1A 0877            0B1B      327         DC    XL2'877'               DESTINATION
   0B1C 7D F0 0B                  328         CLI   11(,XR1),C'0'          TEST FOR 120 CHARACTER CHAIN
   0B1F F2 01 04                  329         JNE   *+7                    JUMP IF NOT 120 CHARACTER
   0B22 3C FF 0878                330         MVI   X'878',X'FF'           SET FLAG TO SAY 120 CHARACTER
   0B26 C0 87 0A8E                331         B     NEXTR
                                  332
   0B2A 7D E2 00                  333 CKSSW   CLI   0(,XR1),C'S'
   0B2D F2 01 1A                  334         JNE   CKEND
   0B30 C2 02 0018                335         LA    BASE1,XR2
   0B34 0F 05 020D 020D           336         SLC   SBYTE5(6),SBYTE5       CLEAR SSW00 THRU SSW3F
   0B3A 0C 01 01C6 1FF7           337         MVC   XREF5(2),X'1FF7'       SET UP LINKAGE TO SENSE SWITCH RTN
   0B40 0C 01 01CA 1FF9           338         MVC   XREF4(2),X'1FF9'
   0B46 C0 87 01B1                339         B     SSWRD                  PROCESS THE SSW CARD
                                  340
   0B4A 7D C5 00                  341 CKEND   CLI   0(,XR1),C'E'            BRANCH IF NOT END CARD
   0B4D C0 01 0A8E                342         BNE   NEXTR
   0B51 0C 01 0068 1FFF           343         MVC   XREF1(2),X'1FFF'
   0B57 0C 01 01CA 1FF9           344         MVC   XREF4(2),X'1FF9'
   0B5D 0C 01 01C6 1FF7           345         MVC   XREF5(2),X'1FF7'
   0B63 C2 01 0196                346         LA    CKCOM,XR1
   0B67 34 01 0082                347         ST    CKCOMA+3,XR1
   0B6B 35 10 1FFB                348         L     XREF3,IAR                GO BEGIN DCP
   0B6F 35 10 1FFD                349 DEFINE  L     XREF2,IAR
                        1FFD      350 XREF2   EQU   X'1FFD'
                        1FFB      351 XREF3   EQU   X'1FFB'
                                  352
                                  353 ***********
                                  354 * EQUATES ***************************************************************
                                  355 ***********
                        0880      356 INPUT   EQU   X'880'
                        0200      357 SMOD    EQU   X'200'                 SYSTEM MODEL IDENT LOCATION
                        0203      358 SIZE    EQU   X'203'                 CORE SIZE OF SYSTEM
                        0204      359 CPU     EQU   X'204'                 CPU OPTIONS
                        0208      360 SBYTE0  EQU   X'208'                 COMMON SENSE SWITCHES
                        0001      361 SSW07   EQU   X'01'                  SSW TO LOAD SECTION AND GO
                        0232      362 UTAB    EQU   X'232'                 FIRST BYTE OF UDT TABLE
                        0212      363 TEST    EQU   X'212'                 ENTRY TO CHECK CONSOLE SWITCHES
                        021A      364 PRINT   EQU   X'21A'                 ENTRY TO DCP PRINT
                        021E      365 UNPACK  EQU   X'21E'                 ENTRY TO UNPACK ROUTINE
                        0226      366 PACK    EQU   X'226'                 ENTRY TO PACK SUBROUTINE
                        0010      367 IAR     EQU   X'10'                  INSTRUCTION ADDRESS REGISTER
                        0008      368 ARR     EQU   X'08'                  ADDRESS RECALL REGISTER
                        0001      369 XR1     EQU   1                      INDEX REGISTER 1
                        0002      370 XR2     EQU   2                      INDEX REGISTER 2
                        006F      371 H0      EQU   X'6F'
                        0007      372 H7      EQU   X'07'
```

PPB3 DIAGNOSTIC LOADER FOR DISK

```
ERR LOC OBJECT CODE       ADDR STMT SOURCE STATEMENT

                          006C  373 HC       EQU   X'6C'
                          003B  374 HH       EQU   X'3B'                HALT DISPLAY CODE -H-
                          0020  375 BIT2     EQU   X'20'
                          0010  376 BIT3     EQU   X'10'
                          0008  377 BIT4     EQU   X'08'
                          0004  378 BIT5     EQU   X'04'
                          0002  379 BIT6     EQU   X'02'
                          0001  380 BIT7     EQU   X'01'
                          00A2  381 BUSY     EQU   X'A2'                EQUATE FOR B LEVEL
                          00A6  382 CTLREG   EQU   X'A6'                EQUATE FOR B LEVEL
                          00A0  383 NTRLY    EQU   X'A0'                EQUATE FOR B LEVEL
                                384
                                385 *
                                386 **  EXECUTION ENTRY POINT.
                                387 *
  0373 3C 40 097F               388 BEGIN    MVI   X'97F',C' '          CLEAR PRINT IMAGE AND DATA FIELDS
  0377 0C FE 097E 097F          389          MVC   X'97E'(255),X'97F'
  037D 0C 03 087F 0890          390          MVC   X'87F'(4),X'890'
  0383 C2 02 0018               391          LA    BASE1,XR2            LOAD XR2 AS BASE
  0387 30 A3 0C0B               392          SNS   SNSARE,X'A3'                                      *
  038B 38 01 0C0A               393          TBN   SNSARE-1,X'01'       IS SYSTEM MOD B              *
  038F F2 90 20                 394          JP    MODLD                                             *
  0392 3C 04 0A8D               395          MVI   NEXTR-1,X'04'        SET MOD B HALTS              *
  0396 3C 78 0A8C               396          MVI   NEXTR-2,X'78'                                     *
  039A 3C 78 0A53               397          MVI   CDERR0+1,X'78'                                    *
  039E 3C 04 0A54               398          MVI   CDERR0+2,X'04'                                    *
  03A2 3C 78 0162               399          MVI   ERRHLT+1,X'78'
  03A6 3C 3C 0163               400          MVI   ERRHLT+2,X'3C'
  03AA 3C 78 0063               401          MVI   HPLHD+1,X'78'
  03AE 3C 6C 0064               402          MVI   HPLHD+2,X'6C'
  03B2 0C 00 0C0C 0011          403 MODLD    MVC   DISKTP(1),X'11'      MOVE SWITCH FROM BOOT TO LOADER  *
  03B8 88 00 1A 0C0C            404          MZZ   SEEKP1(,XR2),DISKTP
  03BD 88 00 58 0C0C            405          MZZ   SEEKP2(,XR2),DISKTP
  03C2 88 00 0174 0C0C          406          MZZ   SEEKP3,DISKTP
  03C8 88 00 018E 0C0C          407          MZZ   SEEKP4,DISKTP
  03CE 88 00 87 0C0C            408          MZZ   READP(,XR2),DISKTP
  03D3 88 00 0156 0C0C          409          MZZ   ALTSIO+1,DISKTP
  03D9 39 08 0C0C               410          TBP   DISKTP,X'08'         TEST FOR RUN ON REMOVABLE DISK
  03DD F2 10 21                 411          JT    SETENT                JUMP IF YES
  03E0 BA 08 1A                 412          SBN   SEEKP1(,XR2),X'08'
  03E3 BA 08 58                 413          SBN   SEEKP2(,XR2),X'08'
  03E6 3A 08 0174               414          SBN   SEEKP3,X'08'
  03EA 3A 08 018E               415          SBN   SEEKP4,X'08'
  03EE BA 08 87                 416          SBN   READP(,XR2),X'08'
  03F1 3A 08 0156               417          SBN   ALTSIO+1,X'08'
  03F5 3A 08 0140               418          SBN   WAIT+1,X'08'
  03F9 3A 08 0159               419          SBN   WAIT1+1,X'08'
  03FD 3A 08 0146               420          SBN   TSTERR+1,X'08'
  0C01 0C 03 01FF 0C10          421 SETENT   MVC   X'1FF'(4),ENTRYS      MOVE ENTRIES POINTS
  0C07 E0 87 55                 422          B     PGMLOD(,XR2)         GO BEGIN LOADING
  0C0A 0000                     0C0B 423 SNSARE DC    XL2'0'                                          *
  0C0C 00                       0C0C 424 DISKTP DC    XL1'00'
  0C0D 0183                     0C0E 425         DC    AL2(RECARD)
  0C0F 0019                     0C10 426 ENTRYS  DC    AL2(ENT2)
                                0B73 427         END   BEGIN
```

FPB3 DIAGNOSTIC LOADER FOR DISK

CROSS-REFERENCE

| SYMBOL | T | LEN | VALUE | DEFN | REFERENCES |
|---|---|---|---|---|---|
| ACTCON | A | 003 | 00BF | 0094 | 0041 |
| ADREXT | A | 001 | 0139 | 0150 | 0123 0137 |
| ALC | A | 004 | 00A0 | 0078 | 0081 |
| ALTDFC | A | 002 | 00CB | 0100 | 0167 |
| ALTSIO | A | 003 | 0155 | 0168 | 0409* 0417* |
| ARDDFC | A | 001 | 00CC | 0101 | 0100 0170 0175* 0183* |
| ARR | C | 001 | 0008 | 0368 | 0026 0050* 0074 0116 0189 |
| BASE1 | A | 004 | 0018 | 0231 | 0023 0025 0162 0179 0188 0190 0288 0335 0391 |
| BEGIN | A | 004 | 0B73 | 0388 | 0427 |
| BIT2 | C | 001 | 0020 | 0375 | 0061 |
| BIT3 | C | 001 | 0010 | 0376 | 0285 |
| BIT4 | C | 001 | 0008 | 0377 | |
| BIT5 | C | 001 | 0004 | 0378 | 0226 |
| BIT6 | C | 001 | 0002 | 0379 | 0037 |
| BIT7 | C | 001 | 0001 | 0380 | 0048 |
| BUSY | C | 001 | 00A2 | 0381 | 0127 0132 0154 0169 |
| CDERR0 | A | 003 | 0A52 | 0268 | 0265 0397* 0398* |
| CHAIN | A | 005 | 0A0A | 0243 | 0322 |
| CHKSE | A | 003 | 01D8 | 0224 | 0210 |
| CHKSSW | A | 003 | 01AB | 0209 | 0202 |
| CHKSSO | A | 004 | 01BA | 0213 | 0222 |
| CKCHN | A | 005 | 0B0B | 0322 | 0272 |
| CKCOM | A | 003 | 0196 | 0201 | 0346 |
| CKCOMA | A | 004 | 007F | 0065 | 0347* |
| CKCOM1 | A | 003 | 0A0B | 0245 | 0065 |
| CKCPU | A | 003 | 0A24 | 0253 | 0246 |
| CKEND | A | 003 | 0B4A | 0341 | 0334 |
| CKND | A | 003 | 004E | 0045 | 0042 |
| CKSSW | A | 003 | 0B2A | 0333 | 0323 |
| CKUDT | A | 003 | 0A58 | 0271 | 0254 |
| CPU | C | 001 | 0204 | 0359 | 0263* |
| CTLREG | C | 001 | 00A6 | 0382 | 0115* 0119 0125* 0130* 0134* 0367* 0373* |
| DEFINE | A | 004 | 0B6F | 0349 | 0247 0256 0273 |
| DEV | A | 001 | 0A03 | 0241 | 0282 0290 0294 |
| DFDR | A | 002 | 00B6 | 0089 | 0082 0152 |
| DISKTP | A | 001 | 0C0C | 0424 | 0403* 0404 0405 0406 0407 0408 0409 0410 |
| DSKERR | A | 001 | 0152 | 0166 | 0158 |
| DTABLE | C | 001 | 01FE | 0234 | 0030 0050 |
| ENTRYS | A | 002 | 0C10 | 0426 | 0421 |
| ENT2 | A | 004 | 0019 | 0025 | 0426 |
| ERRHLT | A | 003 | 0161 | 0172 | 0399* 0400* |
| EXTSAV | A | 002 | 00C1 | 0095 | 0176* 0182 |
| FPB | A | 001 | 0000 | 0003 | |
| FLAG | C | 001 | 01FD | 0233 | 0037 0048 0061 0226 |
| FWDSEK | A | 003 | 0127 | 0143 | 0138 |
| HC | C | 001 | 006C | 0373 | |
| HD | A | 004 | 0057 | 0048 | 0046 |
| HH | C | 001 | 0038 | 0374 | 0172 0268 0287 |
| HPLHD | A | 003 | 0062 | 0051 | 0401* 0402* |
| HO | C | 001 | 006F | 0371 | 0268 0287 |
| H7 | C | 001 | 0007 | 0372 | 0172 |
| IAR | C | 001 | 0010 | 0367 | 0039* 0348* 0349* |
| ID | A | 004 | 0018 | 0024 | 0031 0043 0231 |
| IDFND | A | 004 | 0069 | 0055 | 0044 |
| INPUT | C | 001 | 0880 | 0356 | 0253 0326 |
| LAST@ | A | 002 | 00D1 | 0105 | 0125 |
| LASTAD | A | 001 | 00D3 | 0107 | 0128 0133* 0136 0139 0145 0147* |
| LDPARM | A | 003 | 00DF | 0117 | 0184 |
| LDR | A | 004 | 0065 | 0052 | 0049 |
| LDUDT | A | 005 | 0AA4 | 0294 | 0291 |
| LOADTB | C | 001 | 0068 | 0235 | |
| MASK | A | 002 | 0A05 | 0242 | 0311* 0314 0314* 0316 0317 |
| MEXIT | A | 004 | 0078 | 0062 | 0026* 0039 0228 |
| MODLD | A | 006 | 0BB2 | 0403 | 0394 |
| MOVE | A | 005 | 008F | 0071 | 0068* 0069* |
| NEXTR | A | 004 | 0A8E | 0288 | 0267 0269 0302 0331 0342 0395* 0396* |

FPB3 DIAGNOSTIC LOADER FOR DISK

CROSS-REFERENCE

| SYMBOL | T | LEN | VALUE | DEFN | REFERENCES |
|---|---|---|---|---|---|
| NINE | A | 001 | 0A02 | 0240 | 0309 |
| NRMDFC | A | 002 | 00C3 | 0096 | 0113 0115 0173 0180 |
| NTRDY | C | 001 | 00A0 | 0383 | 0158 |
| NTWO | A | 001 | 0024 | 0029 | |
| OK | A | 00 | 010C | 0134 | 0129 |
| ONE | A | 002 | 0A01 | 0239 | 0311 0312 |
| ONESEC | A | 001 | 00BB | 0092 | 0078 |
| PACK | C | 001 | 0226 | 0366 | 0215 0257 0279 0324 |
| PGMLOD | A | 003 | 006D | 0057 | 0072 0204 0207 0223 0225 0249 0252 0289 0422 |
| PRINT | C | 001 | 021A | 0364 | 0205 0250 |
| PTPDC | A | 003 | 0A6E | 0277 | 0275 |
| RDDFC | A | 001 | 00B1 | 0085 | 0033* 0034* 0045 0055* 0075* 0078* 0079 0096 0114 0165 0181 0183 |
| RDVTOC | A | 003 | 002F | 0035 | 0047 |
| READ | A | 003 | 0096 | 0074 | 0030 0059 0193 |
| READF | A | 001 | 009F | 0077 | 0403* 0416* |
| RECABA | A | 002 | 00C5 | 0097 | 0130 |
| RECARD | A | 004 | 0183 | 0189 | 0161 0425 |
| RECEXT | A | 004 | 0192 | 0194 | 0169* |
| REDEXT | A | 004 | 00AD | 0083 | 0174* |
| SADDR | A | 002 | 01C4 | 0216 | * |
| SAVPRM | A | 004 | 016D | 0176 | 0124 |
| SBYTE0 | C | 001 | 0208 | 0360 | 03 0248 |
| SBYTE5 | C | 001 | 020D | 0208 | 211 0211* 0336 0336* |
| SCRCH | A | 001 | 0133 | 0147 | 0139* 0140* 0144* 0145* |
| SECTNO | A | 001 | 00BC | 0093 | |
| SEEKP1 | A | 001 | 0032 | 0036 | 0404* 0412* |
| SEEKP2 | A | 001 | 0070 | 0058 | 0405* 0413* |
| SEEKP3 | A | 001 | 0174 | 0178 | 0406* 0414* |
| SEEKP4 | A | 001 | 018E | 0192 | 0407* 0415* |
| SENSE | A | 002 | 00B8 | 0090 | 0119* 0121 0134 |
| SETADB | A | 003 | 0132 | 0148 | 0141 |
| SETALT | A | 003 | 016A | 0175 | 0171 |
| SETENT | A | 006 | 0C01 | 0421 | 0411 |
| SETSCN | A | 003 | 0029 | 0033 | |
| SETWOK | A | 001 | 0020 | 0027 | |
| SETXR1 | A | 001 | 014B | 0159 | 0116* 0117 0176 0182* |
| SIO | A | 003 | 013C | 0153 | 0118* |
| SIZE | C | 001 | 0203 | 0358 | 0260 |
| SMOD | C | 001 | 0200 | 0357 | 0255* |
| SNSARE | A | 002 | 0C0B | 0423 | 0392* 0393 |
| SOURCE | A | 002 | 0026 | 0030 | |
| SSWRD | A | 006 | 01B1 | 0211 | 0339 |
| SSW07 | C | 001 | 0001 | 0361 | 0203 0248 |
| STPARM | A | 003 | 00DB | 0116 | 0177 |
| STRTIO | A | 001 | 00D5 | 0112 | 0035 0057 0076 0191 |
| TEST | C | 001 | 0212 | 0363 | |
| TSTERR | A | 003 | 0145 | 0158 | 0420* |
| TWOZR | A | 002 | 00BA | 0091 | |
| UDTA | A | 005 | 0A95 | 0290 | 0286 |
| UDTB | A | 003 | 0ACF | 0307 | 0304 |
| UDTC | A | 003 | 0ADA | 0310 | 0308 |
| UDTD | A | 006 | 0AE5 | 0316 | 0313 |
| ULP1 | A | 004 | 0A71 | 0278 | 0306 |
| ULP2 | A | 003 | 0A82 | 0284 | 0293 |
| ULP3 | A | 005 | 0A8E | 0312 | 0315 |
| ULP4 | A | 003 | 0AB2 | 0300 | 0320 |
| UNPACK | C | 001 | 021E | 0365 | 0028 |
| UPTR | A | 002 | 0A7B | 0281 | 0278* |
| USRT1 | A | 003 | 0B01 | 0318 | 0316* |
| USRT2 | A | 003 | 0B04 | 0319 | 0317* |
| UTAB | C | 001 | 0232 | 0362 | 0276 0276* 0283 |
| WAIT | A | 003 | 013F | 0154 | 0418* |
| WAIT1 | A | 003 | 0158 | 0169 | 0419* |
| WORK | C | 001 | 0880 | 0232 | 0089 |
| XREF1 | A | 001 | 0068 | 0053 | 0343* |
| XREF2 | C | 001 | 1FFD | 0350 | 0349 |

FPB3 DIAGNOSTIC LOADER FOR DISK

CROSS-REFERENCE

| SYMBOL | T | LEN | VALUE | DEFN | REFERENCES |
|---|---|---|---|---|---|
| XREF3 | C | 001 | 1FPB | 0351 | 0348 |
| XREF4 | A | 001 | 01CA | 0219 | 0338* 0344* |
| XREF5 | A | 002 | 01C6 | 0217 | 0337* 0345* |
| XR1 | C | 001 | 0001 | 0369 | 0041 0043 0065 0064 0068 0069 0070 0070* 0071 0082* 0117* 0119 |
| | | | | | 0120 0121* 0135 0136 0140 0143 0144 0148 0149 0156 0160* 0163 |
| | | | | | 0201 0209 0212 0212* 0213 0220 0221 0221* 0224 0227 0228* 0229 |
| | | | | | 0245 0253 0255 0261 0264 0266 0271 0274 0277 0277* 0278 0297 |
| | | | | | 0297* 0298 0300 0300* 0301 0303 0305 0305* 0307 0309 0310 0312 |
| | | | | | 0322 0328 0333 0341 0346* 0347 |
| XR2 | C | 001 | 0002 | 0370 | 0025* 0026 0033 0034 0035 0039 0040 0041 0043 0045 0047 0055 |
| | | | | | 0057 0059 0068 0069 0072 0074 0075 0076 0078 0078 0079 0081 |
| | | | | | 0082 0113 0113* 0115 0116 0117 0118 0119 0121 0125 0127 0128 |
| | | | | | 0130 0132 0133 0134 0136 0139 0139 0140 0144 0145 0145 0149 |
| | | | | | 0152 0154 0158 0161 0161* 0167 0169 0170 0173 0175 0176 0176 |
| | | | | | 0177 0180 0180* 0182 0182 0183 0183 0184 0190* 0191 0193 0204 |
| | | | | | 0207 0223 0225 0228 0249 0252 0283* 0284 0284* 0285 0288* 0289 |
| | | | | | 0290 0292 0294 0295 0296 0318 0319 0335* 0391* 0404 0405 0408 |
| | | | | | 0412 0413 0416 0422 |

TOTAL STATEMENTS FLAGGED IN THIS ASSEMBLY = 0

FPB3 DIAGNOSTIC LOADER FOR DISK

OBJECT CARD LISTING

THE CHARACTER ' INDICATES A BLANK COLUMN AND THE CHARACTERS D̄ Ē H̄ INDICATE NUMERIC SHIFT.

| CL 1 THROUGH 16 | CL 17 THROUGH 32 | CL 33 THROUGH 48 | CL 49 THROUGH 64 | CL 65 THROUGH 80 | CL 81 THROUGH 96 |
|---|---|---|---|---|---|
| T+-A!&DA'&<HB'AS | 4BF¦'/OH;'-G"'AS | @BID̄@'I?-/#4'+'H | A"¬H̄&'*M&Q=BG¬W4 | B'D̄¬2'&)_'-*'@YD | $?&0'"/OFPB30001 |
| T+-BH̄W?H̄A'=BGE3- | A'¬72D'*5B'G"@C_ | 3OH*''IOAWOL-/#4 | '8H)=+B'A"*'&'G1 | '80C''&Y.X'I9'90 | ';-D'P'MFPB30002 |
| T+-CE4-DDG''''''C | -/5O4BIS@'I3-/#4 | A,-F$Y#/-W='&S.M | AX%8G''''A0&'BH | ''''''A'CA0=<'''B | 1'<Q')A-PPB30003 |
| T+-D''''''"0C<''' | '''CK''''_&H̄,%ĒQ | K_'SĒ_&PĒX'B<'.B | WA7HG'.MAA"H̄&K.F | WH¦+/';PSJ,0CH¬H | &C.D'#Q-PPB30004 |
| T+-D#Z/L3Y'C/Y¬O | @'BH̄1Z-)@''(_''D | S@YD-@Y&.,'BRHZ@ | '--G2/0_:'&H̄*'HH | A,0BRHX0''90'H-P | 1Z'M'5/'PPB30005 |
| T+-E6@0''8ĒH̄+;0D | B8ĒB/0-D''.MB6)B | G'$PWP?+/';PSZ#4 | BP"H̄AB¬'#A#PWD?H̄ | G'*0AP:0ADI.-/2Y | '_&H'"AMPPB30006 |
| T+-P1D̄:0AW/8%'/8 | C8H*_('-AV*EB'AT | -/#4'8H)=0H*''G5 | *';HAC3-A'-T-DEP | '/0HĒH;BGNP7S'¦H | AI0@'9-@?PB30007 |
| T+'GD̄'0H:('-7K'&M | 4'&GD0H*8I-H'''' | 'CH*''G5,')HA'@B | A'$,-/5N'1&C-'NM | 8A'G'4IBA_&ET4H* | '''''12@PPB30008 |
| T+-Y:''DI''''0@T | A2)N'P'C2'J¦'/0_ | ?+'DRB+'&N*BG'/Y | /8H)N¬*<'@-D>G'' | B''L'/0_?0H*BI-& | HS&H'@Y<PPB30009 |
| T+-Z5'771B"HAACD̄ | ''-J'0-L2--R'1'L | 2AC30+6"2/3R'9'C | 2'Ē7'/0_?¬M'C@-D | FC24BP0I¬4-DE('D | H̄;@''00&PPB30010 |
| T+-D̄O/0HW'-''B-¦ | B'-H?8-HC>A'A@Z' | H̄@C_?0-H'F+BGNQ4 | '''YC@YDG?&''0'D | H̄-Y0'''YC>0@A?'' | B4-D'¬/OPPB30011 |
| T+-,,'P5-'¦HA''H | A'P5''<BABY9'ĒOC | 2'&¬K'&.'/0Z1¬¬' | '@-H&L-''B-I#@'' | <'&YEB-E¦'''H̄'¬H̄ | BB-8''T8PPB30012 |
| T+-%W'&YEB-P'/0, | TC''.'-YDC''.A&Y | E>-'A>-'BOH*H̄>D4 | DAOYH̄@-DP0H*BI? | I;0/?¬¬'.@-DD¦¦@ | H;<''':1MPPB30013 |
| T+-_//0D̄+¬;9'@-D | ĒO-H'F'@L'-4BC&O | A'*Q¬'00A'*Y¬=*B | G'$E'1&C''&D̄+C'D | ĒA""C'DA2/"?C'D | A1/@'$Y4PPB30014 |
| T+->*'@HA'RQ4'&B | B(J'¬=3M&G"4@&'V | "C¦8I¬-V"C'<H¬0S | '0-H'PCBTC'%9'90 | H̄@Z'-¦'&H̄TL18B?0 | @;'Y',%PPB30015 |
| T+-?P:30D8V&@;'E | S¦COAQ318'P<@$'A | UC''<C''JS''ĒC'2 | H'E-<C'-''P&<C'- | ''Q8<CH-'/00<8'' | AN-0'N.DPPB30016 |
| T+'0&CCUHC'32DBP | :@AD̄:BE-:B'E4+-- | AT,YH/3YH'NQ:B'E | '+--AOLYH'MQ<'0G | "CAC-/5M''''A-0' | R''''0H*PPB30017 |
| EB7(*E7*=-JC"PH$ | ="7M&P¦'''¦''C'' | F%''ASC''R'A | S0''Q'''''''''' | 09310317710 | 40571&'YPPB30018 |

---------- LAST PAGE ----------