```
                4  ***     BASIC - *WINTEK* BASIC INTERPRETER.
                5  *
                6  *       J. G. LETWIN, 09/76, FOR *WINTEK* CORPORATION.
                7  *
                8  *       H. W. SCHULTZ, 12/77, FOR HEATH CO.
                9  *
               10  *       J. G. LETWIN, 1/78, FOR HEATH COMPANY
               11  *
               12  *       G.    Chandler, 78/10, for Heath Co.
               13  *                     79/12
               14  *                     80/02
               15  *
               16  *
               17  *
               18  *       Issues:
               19  *
               20  *               110.01.00
               21  *               110.02.00                                  /79.05.sc/
               22  *               110.05.00                                  /79.12.GC/
               23  *                                                          /80.02.GC/


               25  ***     COPYRIGHT 09/1976, *WINTEK* CORPORATION, LAFAYETTE, IND.
               26  *
               27  *       COPYRIGHT 12/1977, 05/1979 HEATH CO.
               28  *
               29  *               HEATH CO.
               30  *               BENTON HARBOR, MI
               31  *               49022
               32  *
```

```
                        36  ****    ASSEMBLY CONSTANTS.
                        37
000.005                 38  CHANMAX EQU     5               MAXIMUM CHANNEL # = 5


                        40  **      RUN MODE FLAGS.
                        41  *
                        42  *       THESE ARE SET IN *RUNMODE*.
                        43
000.000                 44  RM.IMM  EQU     0               IMMEDIATE MODE
000.001                 45  RM.STE  EQU     01Q             STEP MODE
000.004                 46  RM.CON  EQU     04Q             CONTINUOUS MODE
000.200                 47  RM.HLT  EQU     200Q            HALT EXECUTION


                        49  **      MACHINE INSTRUCTIONS.
                        50  *
                        51
000.200                 52  MI.ADDB EQU     200Q            ADD B
000.303                 53  MI.JMP  EQU     303Q            JMP
000.077                 54  MI.CMC  EQU     077Q            CMC
000.072                 55  MI.LDA  EQU     072Q            LDA
000.076                 56  MI.MVIA EQU     076Q            MVI A
000.323                 57  MI.OUT  EQU     323Q            OUT
000.220                 58  MI.SUBB EQU     220Q            SUB B
000.000                 59  MI.NOP  EQU     0               NOP
000.311                 60  MI.RET  EQU     311Q            RET
000.333                 61  MI.IN   EQU     333Q            IN
000.041                 62  MI.LXIH EQU     041Q            LXI  H
000.021                 63  MI.LXID EQU     021Q            LXI  D
000.001                 64  MI.LXIB EQU     001Q            LXI  B


                        66  **      THE CT. SYMBOLS DEFINE INDEXED OF TOKENS.
                        67  *
                        68
                        69
                        70  **      CHARACTER TYPES.
                        71
000.000                 72          ORG     0
                        73
000.000                 74  CT.FIN  DS      1               00 OR ;
000.001                 75  CT.ALP  DS      1               ALPHABETIC
000.002                 76  CT.NUM  DS      1               NUMERIC
000.003                 77  CT.SEP  DS      1               UNSPECIFIED SEPERATOR
                        78
                        79  *       THE FOLLOWING ARE NOT COMPRESSED IN THE TEXT INTO THESE TOKENS,
                        80  *       BUT THE VARIOUS SCANNER ROUTINES RETURN THESE VALUES.
                        81
000.004                 82          ERRMI   10Q-*
000.004                 83          DS      10Q-*
000.010                 84          DS      1               PLACE HOLDER TO POSITION CT.EQ
000.000                 85          ERRNZ   *-011Q          REQUIRED FOR COMPARE PROCESSING
```

```
000.011                86  CT.EQ   DS      1               = 1
000.012                87  CT.GT   DS      1               > 2
000.013                88  CT.GE   DS      1               >= 3
000.014                89  CT.LT   DS      1               < 4
000.015                90  CT.LE   DS      1               <= 5
000.016                91  CT.NE   DS      1               <> 6
                       92
000.017                93  CT.PAL  DS      1               (
000.020                94  CT.PAR  DS      1               )
000.021                95  CT.PL   DS      1               +
000.022                96  CT.MI   DS      1               -
000.023                97  CT.MU   DS      1               *
000.024                98  CT.DI   DS      1               /
000.025                99  CT.EX   DS      1               ^
000.026               100  CT.CMA  DS      1               ,
000.027               101  CT.SEM  DS      1               ;
000.030               102  CT.QUO  DS      1               "
000.031               103  CT.PS   DS      1               #
                      104
                      105
                      106  **      BASIC VERBS AND KEYWORDS.
                      107
                      108
000.200               109          ORG     2000
000.200               110  CT.BLD  DS      1               BUILD   (MUST BE FIRST)
000.201               111  CT.BYE  DS      1               BYE
000.202               112  CT.CNT  DS      1               CONTINUE
000.203               113  CT.DEL  DS      1               DELETE
000.204               114  CT.LIS  DS      1               LIST
000.205               115  CT.REP  DS      1               REPLACE
000.206               116  CT.RUN  DS      1               RUN
000.207               117  CT.SAV  DS      1               SAVE
000.210               118  CT.SCR  DS      1               SCRATCH
000.211               119  CT.STE  DS      1               STEP
                      120
000.212               121  CT.RUA  EQU     *               FOLLOWING COMMANDS 'RUN USAGE ALLOWED'
                      122
000.212               123  CT.SYE  DS      1               SYNTAX ERROR
000.213               124  CT.CHA  DS      1               CHAIN
000.214               125  CT.CLR  DS      1               CLEAR
000.215               126  CT.CLO  DS      1               CLOSE
000.216               127  CT.CTL  DS      1               CNTRL
000.217               128  CT.DIM  DS      1               DIM
000.220               129  CT.FN   DS      1               FN
000.221               130  CT.FOR  DS      1               FOR
000.222               131  CT.FRE  DS      1               FREE
000.223               132  CT.FRZ  DS      1               FREEZE
000.224               133  CT.GOS  DS      1               GOSUB
000.225               134  CT.GOT  DS      1               GOTO
000.226               135  CT.IF   DS      1               IF
000.227               136  CT.LET  DS      1               LET
000.230               137  CT.LCK  DS      1               LOCK
000.231               138  CT.NXT  DS      1               NEXT
000.232               139  CT.OLD  DS      1               OLD
000.233               140  CT.ON   DS      1               ON
000.234               141  CT.OPE  DS      1               OPEN
```

```
000.235                 142  CT.OUT  DS      1               OUT
000.236                 143  CT.PAU  DS      1               PAUSE
000.237                 144  CT.POK  DS      1               POKE
000.240                 145  CT.PRT  DS      1               PRINT
000.241                 146  CT.REA  DS      1               READ
000.242                 147  CT.REM  DS      1               REMARK
000.243                 148  CT.RES  DS      1               RESTORE
000.244                 149  CT.RET  DS      1               RETURN
000.245                 150  CT.UNF  DS      1               UNFREEZE
000.246                 151  CT.UNL  DS      1               UNLOCK
000.247                 152  CT.UNS  DS      1               UNSAVE
                        153
000.250                 154  CT.IUA  EQU     *               PREVIOUS COMMANDS 'IMMEDIATE USAGE ALLOWED'
                        155
000.250                 156  CT.LIN  DS      1               LINE
000.251                 157  CT.DAT  DS      1               DATA
000.252                 158  CT.DEF  DS      1               DEF
000.253                 159  CT.END  DS      1               END
000.254                 160  CT.INP  DS      1               INPUT
000.255                 161  CT.STP  DS      1               STOP
                        162
000.256                 163  CT.CMD  EQU     *               PREVIOUS ARE VALID COMMANDS
                        164
                        165
                        166  **      BASIC PRE-DEFINED FUNCTIONS.
                        167
                        168
000.022                 169          ERRMI   300Q-*          CHECK FOR OVERLAP
000.256                 170          DS      300Q-*
                        171
                        172  *       THE FOLLOWING BITS ARE DESCRIBED IN THE SYMTAB DOCUMENTATION.
                        173  *       THEY ARE USED TO DECLARE VARIABLE TYPE.
                        174
000.001                 175  CF.STR  EQU     00000001B       IS STRING (NOT NUMERIC)
000.002                 176  CF.VEC  EQU     00000010B       IS VECTOR (NOT SCALAR)
000.004                 177  CF.FCN  EQU     00000100B       IS FUNCTION (NOT VALUE)
                        178
                        179
                        180
                        181  **      SYMBOL TYPE DECLARATIONS.
                        182  *
                        183  *       USED IN SYMBOL TABLE AND BY LEXICAL.
                        184
                        185
000.300                 186  CT.SNV  ORG     300Q+0
000.304                 187  CT.SNF  ORG     300Q+CF.FCN      SCALAR NUMERIC FUNCTION
000.301                 188  CT.SSV  ORG     300Q+CF.STR      SCALAR STRING VARIABLE
000.305                 189  CT.SSF  ORG     300Q+CF.STR+CF.FCN  SCALAR STRING FUNCTION
000.302                 190  CT.VNV  ORG     300Q+CF.VEC          VECTOR NUMERIC VARIABLE
000.303                 191  CT.VSV  ORG     300Q+CF.VEC+CF.STR  VECTOR STRING VALUE
000.300                 192  CT.VARL EQU     CT.SNV          LEAST VARIABLE INDEX
000.307                 193  CT.VARH EQU     300Q+CF.VEC+CF.STR+CF.FCN       HIGH VARIABLE INDEX
000.310                 194          ORG     300Q+CF.VEC+CF.STR+CF.FCN+1
                        195
                        196
                        197  *       VARIOUS NON-FUNCTION KEYWORDS.
```

```
                    198
000.310             199  CT.AND  DS      1          AND
000.311             200  CT.AS   DS      1          AS
000.312             201  CT.FIL  DS      1          FILE
000.313             202  CT.WRI  DS      1          WRITE
000.314             203  CT.NOT  DS      1          NOT
000.315             204  CT.OR   DS      1          OR
000.316             205  CT.THN  DS      1          THEN
000.317             206  CT.TO   DS      1          TO
                    207
                    208  *       FUNCTION DEFINITIONS
                    209
000.320             210  CT.FCN  EQU     *          ALL FUNCTIONS FOLLOW
                    211
000.320             212  CT.ABS  DS      1          ABS(
000.321             213  CT.ATN  DS      1          ATN(
000.322             214  CT.CHR  DS      1          CHR$(
000.323             215  CT.CIN  DS      1          CIN(
000.324             216  CT.COS  DS      1          COS(
000.325             217  CT.EXP  DS      1          EXP(
000.326             218  CT.INT  DS      1          INT(
000.327             219  CT.LNO  DS      1          LNO(
000.330             220  CT.LOG  DS      1          LOG(
000.331             221  CT.MAX  DS      1          MAX(
000.332             222  CT.MIN  DS      1          MIN(
000.333             223  CT.PAD  DS      1          PAD(
000.334             224  CT.PEK  DS      1          PEEK(
000.335             225  CT.PIN  DS      1          PIN(
000.336             226  CT.POS  DS      1          POS(
000.337             227  CT.RND  DS      1          RND(
000.340             228  CT.SEG  DS      1          SEG(
000.341             229  CT.SGN  DS      1          SGN(
000.342             230  CT.SIN  DS      1          SIN(
000.343             231  CT.SPC  DS      1          SPC(
000.344             232  CT.SQR  DS      1          SQR(
000.345             233  CT.STR  DS      1          STR$(
000.346             234  CT.TAB  DS      1          TAB(
000.347             235  CT.TAN  DS      1          TAN(
                    236
                    237  *       THE FOLLOWING FUNCTIONS REQUIRE STRING ARGUMENTS.
                    238
000.350             239  CT.SRA  EQU     *          REQUIRE STRING ARGUMENTS
000.350             240  CT.ASC  DS      1          ASC(
000.351             241  CT.LEF  DS      1          LEFT$(
000.352             242  CT.LEN  DS      1          LEN(
000.353             243  CT.MAT  DS      1          MATCH$(
000.354             244  CT.MID  DS      1          MID$(
000.355             245  CT.RIG  DS      1          RIGHT$(
000.356             246  CT.VAL  DS      1          VAL(
000.357             247  CT.FNM  DS      0          MAX FUNCTION VALUE
```

000.357 249 XTEXT MTR

```
                    252X **     MTR - PAM/8 EQUIVALENCES.
                    253X *
                    254X *      THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                    255X *      MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                    257X **     IO PORTS
                    258X
000.360             259X IP.PAD EQU     360Q            PAD INPUT PORT
000.360             260X OP.CTL EQU     360Q            CONTROL OUTPUT PORT
000.360             261X OP.DIG EQU     360Q            DIGIT SELECT OUTPUT PORT
000.361             262X OP.SEG EQU     361Q            SEGMENT SELECT OUTPUT PORT


                    264X **     FRONT PANEL CONTROL BITS.
                    265X
000.020             266X CB.SSI EQU     00010000B       SINGLE STEP INTERRUPT
000.040             267X CB.MTL EQU     00100000B       MONITOR LIGHT
000.100             268X CB.CLI EQU     01000000B       CLOCK INTERRUPT ENABLE
000.200             269X CB.SPK EQU     10000000B       SPEAKER ENABLE


                    271X **     MONITOR MODE FLAGS.
                    272X
000.000             273X DM.MR  EQU     0               MEMORY READ
000.001             274X DM.MW  EQU     1               MEMORY WRITE
000.002             275X DM.RR  EQU     2               REGISTER READ
000.003             276X DM.RW  EQU     3               REGISTER WRITE


                    278X **     USER OPTION BITS.
                    279X *
                    280X *      THESE BITS ARE SET IN CELL .MFLAG.
                    281X
000.200             282X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
000.100             283X UO.NFR EQU     CB.CLI          NO REFRESH OF FRONT PANEL
000.002             284X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
000.001             285X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING


                    287X **     MONITOR IDENTIFICATION FLAGS
                    288X *
                    289X *      THESE BYTES IDENTIFY THE ROM MONITOR.
                    290X *      THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                    291X
000.021             292X M.PAM8 EQU     021Q            'LXI' INSTRUCTION AT 000.000 IN PAM-8
000.303             293X M.FOX  EQU     303Q            'JMP' INSTRUCTION AT 000.000 IN FOX ROM
```

```
                        295X **     ROUTINE ENTRY POINTS.
                        296X *
                        297X
000.000                 298X .IDENT EQU     0000A       IDENTIFICATION LOCATION
000.053                 299X .DLY   EQU     0053A       DELAY
001.267                 300X .LOAD  EQU     1267A       TAPE LOAD
001.374                 301X .DUMP  EQU     1374A       TAPE DUMP
002.136                 302X .ALARM EQU     2136A       ALARM ROUTINE
002.140                 303X .HORN  EQU     2140A       HORN
002.172                 304X .CTC   EQU     2172A       CHECK TAPE CHECKSUM
002.205                 305X .TPERR EQU     2205A       TAPE ERROR ROUTINE
002.264                 306X .PCHL  EQU     2264A       PCHL INSTRUCTION
002.265                 307X .SRS   EQU     2265A       SCAN RECORD START
002.325                 308X .RNP   EQU     2325A       READ NEXT PAIR
002.331                 309X .RNB   EQU     2331A       READ NEXT BYTE
002.347                 310X .CRC   EQU     2347A       CRC-16 CALCULATOR
003.017                 311X .WNP   EQU     3017A       WRITE NEXT PAIR
003.024                 312X .WNB   EQU     3024A       WRITE NEXT BYTE
003.122                 313X .DOD   EQU     3122A       DECODE FOR OCTAL DISPLAY
003.260                 314X .RCK   EQU     3260A       READ CONSOLE KEYSET
003.356                 315X .DODA  EQU     3356A       SEGMENT CODE TABLE

                        317X **     RAM CELLS USED BY H8MTR.
                        318X *
                        319X
040.000                 320X .START EQU     40000A      START DUMP ADDRESS
040.002                 321X .IOWRK EQU     40002A      IN OR OUT INSTRUCTION
040.005                 322X .REGI  EQU     40005A      DISPLAYED REGISTER INDEX
040.006                 323X .DSPROT EQU    40006A      PERIOD FLAG BYTE
040.007                 324X .DSPMOD EQU    40007A      DISPLAY MODE
040.010                 325X .MFLAG EQU     40010A      USER OPTION BYTE
040.011                 326X .CTLFLG EQU    40011A      PANEL CONTROL BYTE
040.013                 327X .ALEDS EQU     40013A      ABUSS LEDS
040.021                 328X .DLEDS EQU     40021A      DBUSS LEDS
040.024                 329X .ABUSS EQU     40024A      ABUSS REGISTER
040.027                 330X .CRCSUM EQU    40027A      CRCSUM WORD
040.031                 331X .TPERRX EQU    40031A      TAPE ERROR EXIT VECTOR
040.033                 332X .TICCNT EQU    40033A      CLOCK TICK COUNTER
040.035                 333X .REGPTR EQU    40035A      REGISTER POINTER
040.037                 334X .UIVEC EQU     40037A      USER INTERRUPT VECTORS
000.357                 335         XTEXT   ASCII

                        337X **     ASCII CHARACTER EQUIVALENCES.
                        338X
000.015                 339X CR     EQU     13          CARRIAGE RETURN
000.012                 340X LF     EQU     10          LINE FEED
000.200                 341X NULL   EQU     200Q        PAD CHARACTER
000.000                 342X NUL2   EQU     0
000.007                 343X BELL   EQU     7           BELL CHARACTER
000.177                 344X RUBOUT EQU     177Q
000.010                 345X BKSP   EQU     10Q         CTL-H
000.026                 346X C.SYN  EQU     26Q         SYNC
000.002                 347X C.STX  EQU     2           STX
```

```
000.047                 348X QUOTE   EQU     47Q
000.011                 349X TAB     EQU     11Q
000.033                 350X ESC     EQU     33Q
000.012                 351X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                 352X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                 353X FF      EQU     14Q             FORM FEED
000.001                 354X CTLA    EQU     01Q             CTL-A
000.002                 355X CTLB    EQU     02Q             CTL-B
000.003                 356X CTLC    EQU     03Q             CTL-C
000.004                 357X CTLD    EQU     04Q             CTL-D
000.017                 358X CTLO    EQU     17Q             CTL-O
000.020                 359X CTLP    EQU     20Q             CTL-P
000.021                 360X CTLQ    EQU     21Q             CTL-Q
000.023                 361X CTLS    EQU     23Q             CTL-S
000.032                 362X CTLZ    EQU     32Q             CTL-Z
000.357                 363          XTEXT   BECDEF


                        365X **      BASIC ERROR CODE DEFINITIONS.
                        366X
                        367X
000.200                 368X         ORG     128             USE 128 AND ABOVE
000.200                 369X BEC.CC  DS      1               CONTROL-C HIT
000.201                 370X BEC.CB  DS      1               CONTROL-B HIT
000.202                 371X BEC.DE  DS      1               DATA EXHAUSTED
000.203                 372X BEC.DO  DS      1               /0
000.204                 373X BEC.IN  DS      1               ILLEGAL NUMBER
000.205                 374X BEC.IU  DS      1               ILLEGAL USAGE
000.206                 375X BEC.LK  DS      1               DATA LOCK ENGAGED
000.207                 376X BEC.NV  DS      1               NEXT VARIABLE MISSING
000.210                 377X BEC.OV  DS      1               NUMERIC OVERFLOW
000.211                 378X BEC.RE  DS      1               RETURN ERROR
000.212                 379X BEC.SL  DS      1               STRING LENGTH
000.213                 380X BEC.SN  DS      1               STATEMENT NUMBER
000.214                 381X BEC.SY  DS      1               SYNTAX ERROR
000.215                 382X BEC.TC  DS      1               TYPE CONFLICT
000.216                 383X BEC.TO  DS      1               TABLE OVERFLOW
000.217                 384X BEC.SR  DS      1               SUBSCRIPT RANGE
000.220                 385X BEC.SC  DS      1               SUBSCRIPT COUNT
000.221                 386X BEC.ND  DS      1               NOT DIMENSIONED
000.222                 387X BEC.IC  DS      1               ILLEGAL CHARACTER
000.223                 388X BEC.UD  DS      1               UNDEFINED FUNCTION
000.224                 389X BEC.EN  DS      1               END
000.225                 390X BEC.ST  DS      1               STOP
000.226                 391X BEC.FAE DS      1               FILE ALREADY EXISTS
000.227                 392X BEC.ILF DS      1               ILLEGAL FILE NAME
000.230                 393X BEC.AC  DS      1               ILLEGAL ARGUMENT COUNT
000.231                 394X BEC.FND DS      1               FILE NOT OPEN
000.232                 395X BEC.LTL DS      1               LINE TOO LONG
000.233                 396X BEC.CIU DS      1               CHANNEL IN USE
000.234                 397          XTEXT   ECDEF
```

```
                    399X **     ERROR CODE DEFINITIONS.
                    400X
000.000             401X        ORG     0
000.000             402X        DS      1               NO ERROR #0
000.001             403X EC.EOF DS      1               END OF FILE
000.002             404X EC.EOM DS      1               END OF MEDIA
000.003             405X EC.ILC DS      1               ILLEGAL SYSCALL CODE
000.004             406X EC.CNA DS      1               CHANNEL NOT AVAILABLE
000.005             407X EC.DNS DS      1               DEVICE NOT SUITABLE
000.006             408X EC.IDN DS      1               ILLEGAL DEVICE NAME
000.007             409X EC.IFN DS      1               ILLEGAL FILE NAME
000.010             410X EC.NRD DS      1               NO ROOM FOR DEVICE DRIVER
000.011             411X EC.FNO DS      1               CHANNEL NOT OPEN
000.012             412X EC.ILR DS      1               ILLEGAL REQUEST
000.013             413X EC.FUC DS      1               FILE USAGE CONFLICT
000.014             414X EC.FNF DS      1               FILE NAME NOT FOUND
000.015             415X EC.UND DS      1               UNKNOWN DEVICE
000.016             416X EC.ICN DS      1               ILLEGAL CHANNEL NUMBER
000.017             417X EC.DIF DS      1               DIRECTORY FULL
000.020             418X EC.IFC DS      1               ILLEGAL FILE CONTENTS
000.021             419X EC.NEM DS      1               NOT ENOUGH MEMORY
000.022             420X EC.RF  DS      1               READ FAILURE
000.023             421X EC.WF  DS      1               WRITE FAILURE
000.024             422X EC.WPV DS      1               WRITE PROTECTION VIOLATION
000.025             423X EC.WP  DS      1               DISK WRITE PROTECTED
000.026             424X EC.FAP DS      1               FILE ALREADY PRESENT
000.027             425X EC.DDA DS      1               DEVICE DRIVER ABORT
000.030             426X EC.FL  DS      1               FILE LOCKED
000.031             427X EC.FAO DS      1               FILE ALREADY OPEN
000.032             428X EC.IS  DS      1               ILLEGAL SWITCH
000.033             429X EC.UUN DS      1               UNKNOWN UNIT NUMBER
000.034             430X EC.FNR DS      1               FILE NAME REQUIRED
000.035             431X EC.DIW DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036             432X EC.UNA DS      1               UNIT NOT AVAILABLE
000.037             433X EC.ILV DS      1               ILLEGAL VALUE
000.040             434X EC.ILO DS      1               ILLEGAL OPTION
000.041             435X EC.VPM DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042             436X EC.NVM DS      1               NO VOLUME PRESENTLY MOUNTED
000.043             437X EC.FOD DS      1               FILE OPEN ON DEVICE
000.044             438X EC.NPM DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045             439X EC.DNI DS      1               DISK NOT INITIALIZED
000.046             440X EC.DNR DS      1               DISK IS NOT READABLE
000.047             441X EC.DSC DS      1               DISK STRUCTURE IS CORRUPT
000.050             442X EC.NCV DS      1               NOT CORRECT VERSION OF HDOS
000.051             443X EC.NOS DS      1               NO OPERATING SYSTEM MOUNTED
000.052             444X EC.IOI DS      1               ILLEGAL OVERLAY INDEX
000.053             445X EC.OTL DS      1               OVERLAY TO LARGE
000.054             446         XTEXT   FBDEF
```

```
                    448X **     FILE BLOCK DEFINITIONS.
                    449X
000.000             450X        ORG     0
000.000             451X FB.CHA DS      1           CHANNEL NUMBER
000.001             452X FB.FLG DS      1           FLAGS
000.002             453X FB.FWA DS      2           BUFFER FWA
000.004             454X FB.PTR DS      2           BUFFER POINTER
000.006             455X FB.LIM DS      2           LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010             456X FB.LWA DS      2           LWA OF BUFFER
000.012             457X FB.NAM DS      4+8+4+1     NAME OF FILE
000.021             458X FB.NAML EQU     *-FB.NAM
000.033             459X FBENL  EQU     *           ENTRY LENGTH
000.033             460         XTEXT   DIRDEF


                    462X **     DIRECTORY ENTRY FORMAT.
                    463X
000.000             464X        ORG     0
                    465X
                    466X
000.377             467X DF.EMP EQU     377Q        FLAGS ENTRY EMPTY
000.376             468X DF.CLR EQU     376Q        FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                    469X
000.000             470X DIR.NAM DS     8           NAME
000.010             471X DIR.EXT DS     3           EXTENSION
000.013             472X DIR.PRO DS     1           PROJECT
000.014             473X DIR.VER DS     1           VERSION
000.015             474X DIRIDL EQU     *           FILE IDENTIFICATION LENGTH
                    475X
000.015             476X DIR.CLU DS     1           CLUSTER FACTOR
000.016             477X DIR.FLG DS     1           FLAGS
000.017             478X        DS      1           RESERVED
000.020             479X DIR.FGN DS     1           FIRST GROUP NUMBER
000.021             480X DIR.LGN DS     1           LAST GROUP NUMBER
000.022             481X DIR.LSI DS     1           LAST SECTOR INDEX (IN LAST GROUP)
000.023             482X DIR.CRD DS     2           CREATION DATE
000.025             483X DIR.ALD DS     2           LAST ALTERATION DATE
                    484X
000.027             485X DIRELEN EQU    *           DIRECTORY ENTRY LENGTH
000.027             486         XTEXT   IOCDEF


                    488X **     I/O CHANNEL DEFINITIONS.
                    489X
000.000             490X        ORG     0
                    491X
000.000             492X IOC.LNK DS     2           ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002             493X IOC.DDA DS     2           THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                    494X
000.004             495X IOC.FLG DS     1           FILE TYPE FLAGS
000.001             496X FT.DD  EQU     00000001B   =1 IF DIRECTORY DEVICE
000.002             497X FT.OR  EQU     00000010B   =1 IF OPEN FOR READ
000.004             498X FT.OW  EQU     00000100B   =1 IF OPEN FOR WRITE
000.010             499X FT.OU  EQU     00001000B   =1 IF OPEN FOR UPDATE
```

```
000.003                 500X IOC.SQL EQU     *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        501X
000.005                 502X IOC.GRT DS      2               ADDRESS OF GROUP RESERVATION TABLE
000.007                 503X IOC.SPG DS      1               SECTORS PER GROUP, THIS DEVICE
000.010                 504X IOC.CGN DS      1               CURRENT GROUP NUMBER
000.011                 505X IOC.CSI DS      1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 506X IOC.LGN DS      1               LAST GROUP NUMBER
000.013                 507X IOC.LSI DS      1               LAST SECTOR INDEX (IN LAST GROUP)
000.010                 508X IOC.DRL EQU     *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                        509X *                               THE CHANNEL TABLE
000.014                 510X IOC.DTA DS      2               DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 511X IOC.DES DS      2               SECTOR NUMBER OF DIRECTORY ENTRY
000.020                 512X IOC.DEV DS      2               DEVICE CODE
000.022                 513X IOC.UNI DS      1               UNIT NUMBER (0-9)
000.021                 514X IOC.DIL EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                        515X
000.023                 516X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                        517X
000.052                 518X IOCELEN EQU     *               IOC ENTRY LENGTH
                        519X
000.001                 520X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052                 521          XTEXT   HOSDEF


                        523X **      HOSDEF - DEFINE HOS PARAMETER.
                        524X *
                        525X
                        526X
000.026                 527X VERS    EQU     1*16+6          VERSION 1.6
                        528X
000.377                 529X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                        530X
                        531X
000.000                 532X         ORG     0
                        533X
                        534X *       RESIDENT FUNCTIONS
                        535X
000.000                 536X .EXIT   DS      1               EXIT (MUST BE FIRST)
000.001                 537X .SCIN   DS      1               SCIN
000.002                 538X .SCOUT  DS      1               SCOUT
000.003                 539X .PRINT  DS      1               PRINT
000.004                 540X .READ   DS      1               READ
000.005                 541X .WRITE  DS      1               WRITE
000.006                 542X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 543X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010                 544X .LOADO  DS      1               LOAD AN OVERLAY
000.011                 545X .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012                 546X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        547X
                        548X
                        549X *       *HDOSOVL0.SYS*  FUNCTIONS
                        550X
000.040                 551X         ORG     40A
                        552X
000.040                 553X .LINK   DS      1               LINK  (MUST BE FIRST)
```

```
000.041                 554X .CTLC   DS      1               CTL-C
000.042                 555X .OPENR  DS      1               OPENR
000.043                 556X .OPENW  DS      1               OPENW
000.044                 557X .OPENU  DS      1               OPENU
000.045                 558X .OPENC  DS      1               OPENC
000.046                 559X .CLOSE  DS      1               CLOSE
000.047                 560X .POSIT  DS      1               POSITION
000.050                 561X .DELET  DS      1               DELETE
000.051                 562X .RENAM  DS      1               RENAME
000.052                 563X .SETTP  DS      1               SETTOP
000.053                 564X .DECODE DS      1               NAME DECODE
000.054                 565X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                 566X .CLEAR  DS      1               CLEAR CHAN
000.056                 567X .CLEARA DS      1               CLEAR ALL CHANS
000.057                 568X .ERROR  DS      1               LOOKUP ERROR
000.060                 569X .CHFLG  DS      1               CHANGE FLAGS
000.061                 570X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                 571X .LOADD  DS      1               LOAD DEVICE DRIVER
                        572X
                        573X
                        574X *       *HDOSOVL1.SYS*  FUNCTIONS
                        575X
000.200                 576X         ORG     2000
                        577X
000.200                 578X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201                 579X .DMOUN  DS      1               DISMOUNT
000.202                 580X .MONMS  DS      1               MOUNT/NO MESSAGE
000.203                 581X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000.204                 582X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                 583          XTEXT   OVLDEF


                        585X **      OVERLAY TABLE ENTRYS.
                        586X
000.000                 587X         ORG     0
                        588X
000.000                 589X OVL.COD DS      2               FIRST SECTOR OF OVERLAY CODE
000.002                 590X OVL.SIZ DS      2               OVERLAY SIZE
000.004                 591X OVL.ENT DS      2               OVERLAY ENTRY POINT
000.006                 592X OVL.FLB DS      1               OVERLAY FLAG BYTE
000.007                 593X         DS      1               DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010                 594X OVL.ENS EQU     *               OVERLAY ENTRY SIZE
                        595X
                        596X *       OVERLAY INDICES
                        597X
000.000                 598X         ORG     0
                        599X
000.000                 600X OVL0    DS      1
000.001                 601X OVL1    DS      1
000.002                 602          XTEXT   HOSEQU
```

```
                    604X **      HDOS SYSTEM EQUIVALENCES.
                    605X *
                    606X
024.000             607X S.GRT0  EQU   24000A        SYSTEM AREA FOR  GRT0
025.000             608X S.GRT1  EQU   25000A        SYSTEM AREA FOR  GRT1
026.000             609X S.GRT2  EQU   26000A        SYSTEM AREA FOR  GRT2
                    610X
030.000             611X ROMBOOT EQU   30000A        ROM BOOT ENTRY
                    612X
040.100             613X         ORG   40100A        FREE SPACE FROM PAM-8
                    614X
040.100             615X         DS    8             JUMP TO SYSTEM EXIT
040.110             616X D.CON   DS    16            DISK CONSTANTS
040.130             617X SYDD    EQU   *             SYSTEM DISK ENTRY POINT
040.130             618X D.VEC   DS    24*3          SYSTEM ROM ENTRY VECTORS
040.240             619X D.RAM   DS    31            SYSTEM ROM WORK AREA
040.277             620X S.VAL   DS    36            SYSTEM VALUES
040.343             621X S.INT   DS    115           SYSTEM INTERNAL WORK AREAS
041.126             622X         DS    16
041.146             623X S.SOVR  DS    2             STACK OVERFLOW WARNING
041.150             624X         DS    42200A-*      SYSTEM STACK
001.032             625X STACKL  EQU   *-S.SOVR      STACK SIZE
                    626X
042.200             627X STACK   EQU   *             LWA+1 SYSTEM STACK
042.200             628X USERFWA EQU   *             USER FWA
042.200             629          XTEXT ESVAL


                    631X **      S.VAL - SYSTEM VALUE DEFINTIONS.
                    632X *
                    633X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                    634X *
                    635X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                    636X
                    637X
040.277             638X         ORG   S.VAL
                    639X
040.277             640X S.DATE  DS    9             SYSTEM DATE (IN ASCII)
040.310             641X S.DATC  DS    2             CODED DATE
040.312             642X S.TIME  DS    4             TIME FROM MIDNIGHT (IN TICS)
040.316             643X S.HIMEM DS    2             HARDWARE HIGH MEMORY ADRESS+1
                    644X
040.320             645X S.SYSM  DS    2             FWA RESIDENT SYSTEM
                    646X
040.322             647X S.USRM  DS    2             LWA USER MEMORY
                    648X
040.324             649X S.OMAX  DS    2             MAX OVERLAY SIZE FOR SYSTEM
                    650X
                    651X
                    652X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                    653X
000.200             654X CSL.ECH EQU   10000000B     SUPPRESS ECHO
000.002             655X CSL.WRP EQU   00000010B     WRAP LINES AT WIDTH
000.001             656X CSL.CHR EQU   00000001B     OPERATE IN CHARACTER MODE
```

```
                        657X
000.000                 658X I.CSLMD EQU     0              S.CSLMD IS FIRST BYTE
040.326                 659X S.CSLMD DS      1              CONSOLE MODE
                        660X
000.200                 661X CTP.BKS EQU     10000000B      TERMINAL PROCESSES BACKSPACES
000.040                 662X CTP.MLI EQU     00100000B      MAP LOWER CASE TO UPPER ON INPUT
000.020                 663X CTP.MLO EQU     00010000B      MAP LOWER CASE TO UPPER ON OUTPUT
000.010                 664X CTP.2SB EQU     00001000B      TERMINAL NEEDS TWO STOP BITS
000.002                 665X CTP.BKM EQU     00000010B      MAP BKSP (UPON INPUT) TO RUBOUT
000.001                 666X CTP.TAB EQU     00000001B      TERMINAL SUPPORTS TAB CHARACTERS
                        667X
000.001                 668X I.CONTY EQU     1              S.CONTY IS 2ND BYTE
000.000                 669X         ERRNZ   *-S.CSLMD-I.CONTY
040.327                 670X S.CONTY DS      1              CONSOLE TYPE FLAGS
000.002                 671X I.CUSOR EQU     2              S.CUSOR IS 3RD BYTE
000.000                 672X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330                 673X S.CUSOR DS      1              CURRENT CURSOR POSITION
000.003                 674X I.CONWI EQU     3              S.CONWI IS 4TH BYTE
000.000                 675X         ERRNZ   *-S.CSLMD-I.CONWI
040.331                 676X S.CONWI DS      1              CONSOLE WIDTH
                        677X
000.001                 678X CO.FLG  EQU     00000001B      CTL-O FLAG
000.200                 679X CS.FLG  EQU     10000000B      CTL-S FLAG
                        680X
000.004                 681X I.CONFL EQU     4              S.CONFL IS 5TH BYTE
000.000                 682X         ERRNZ   *-S.CSLMD-I.CONFL
040.332                 683X S.CONFL DS      1              CONSOLE FLAGS
                        684X
040.333                 685X S.CAADR DS      2              ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                 686X S.CCTAB DS      6              ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                 687          XTEXT   ESINT
```

```
                        689X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                        690X *
                        691X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                        692X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                        693X
                        694X
040.343                 695X         ORG     S.INT
                        696X
                        697X **      CONSOLE STATUS FLAGS
                        698X
040.343                 699X S.CDB   DS      1              CONSOLE DESCRIPTOR BYTE
000.000                 700X CDB.H85 EQU     00000000B
000.001                 701X CDB.H84 EQU     00000001B      =0 IF H8-5, =1 IF H8-4
040.344                 702X S.BAUD  DS      2              [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                        703X *                              [15]    =1 IF BAUD RATE => 2 STOP BITS
                        704X
                        705X **      TABLE ADDRESS WORDS
                        706X
040.346                 707X S.DLINK DS      2              ADDRESS OF DATA IN HDOS CODE
040.350                 708X S.OFWA  DS      2              FWA  OVERLAY  TABLE
040.352                 709X S.CFWA  DS      2              FWA  CHANNEL  TABLE
```

```
040.354              710X S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356              711X S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                     712X
                     713X **      DEVICE DRIVER DELAYED LOAD FLAGS
                     714X
040.360              715X S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362              716X S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364              717X S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365              718X         DS      1               HOLD PLACE
                     719X *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
040.366              720X S.DDDTA DS      2               DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370              721X S.DDOPC DS      1               OPEN OPCODE PENDEDING
                     722X
                     723X **      OVERLAY MANAGEMENT FLAGS
                     724X
000.001              725X OVL.IN  EQU     00000001B       IN MEMORY
000.002              726X OVL.RES EQU     00000010B       PERMINANTLY RESIDENT
000.014              727X OVL.NUM EQU     00001100B       OVERLAY NUMBER MASK
000.200              728X OVL.UCS EQU     10000000B       USER CODE SWAPPED FOR OVERLAY
                     729X
040.371              730X S.OVLFL DS      1               OVERLAY FLAG
040.372              731X S.UCSF  DS      2               FWA SWAPPED USER CODE
040.374              732X S.UCSL  DS      2               LENGTH SWAPPED USER CODE
040.376              733X S.OVLS  DS      2               SIZE OF OVERLAY CODE
041.000              734X S.OVLE  DS      2               ENTRY POINT OF OVERLAY CODE
                     735X
041.002              736X S.SSN   DS      2               SWAP AREA SECTOR NUMBER
041.004              737X S.OSN   DS      2               OVERLAY SECTOR NUMBER
                     738X
                     739X *       SYSCALL PROCESSING WORK AREAS
                     740X
041.006              741X S.CACC  DS      1               (ACC) UPON SYSCALL
041.007              742X S.CODE  DS      1               SYSCALL INDEX IN PROGRESS
                     743X
                     744X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                     745X
041.010              746X S.JUMPS DS      0               START OF DUMP VECTORS
041.010              747X S.SDD   DS      3               JUMP TO STAND-IN DEVICE DRIVER
041.013              748X S.FASER DS      3               JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016              749X S.DIREA DS      3               JUMP TO DIREAD (DISK FILE READ)
041.021              750X S.FCI   DS      3               JUMP TO FCI (FETCH CHANNEL INFO)
041.024              751X S.SCI   DS      3               JUMP TO SCI (STORE CHANNEL INFO)
041.027              752X S.GUP   DS      3               JUMP TO GUP (GET UNIT POINTER)
                     753X
041.032              754X S.MOUNT DS      1               <>0 IF THE SYSTEM DISK IS MOUNTED
041.033              755X S.DCS   DS      1               DEFAULT CLUSTER SIZE-1
                     756X
041.034              757X S.BOOTF DS      1               BOOT FLAGS
000.001              758X BOOT.P  EQU     00000001B       EXECUTE PROLOGUE UPON BOOTUP
                     759X
                     760X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                     761X
041.035              762X S.OVSTK DS      2               VALUE OF SP UPON SYSCALLS USING OVERLAY
                     763X
041.037              764X         DS      1               RESERVED
```

```
                        766X **     ACTIVE I/O AREA.
                        767X *
                        768X *      THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        769X *      CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        770X *      THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        771X *
                        772X *      NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
                        773X *      FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        774X *      8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        775X *      COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        776X *      BACKDATED AFTER PROCESSING.
                        777X
    041.040             778X AIO.VEC DS     3           JUMP INSTRUCTION
    041.041             779X AIO.DDA EQU    *-2         DEVICE DRIVER ADDRESS
    041.043             780X AIO.FLG DS     1           FLAG BYTE
    041.044             781X AIO.GRT DS     2           ADDRESS OF GROUP RESERV TABLE
    041.046             782X AIO.SPG DS     1           SECTORS PER GROUP
    041.047             783X AIO.CGN DS     1           CURRENT GROUP NUMBER
    041.050             784X AIO.CSI DS     1           CURRENT SECTOR INDEX
    041.051             785X AIO.LGN DS     1           LAST GROUP NUMBER
    041.052             786X AIO.LSI DS     1           LAST SECTOR INDEX
    041.053             787X AIO.DTA DS     2           DEVICE TABLE ADDRESS
    041.055             788X AIO.DES DS     2           DIRECTORY SECTOR
    041.057             789X AIO.DEV DS     2           DEVICE CODE
    041.061             790X AIO.UNI DS     1           UNIT NUMBER (0-9)
                        791X
    041.062             792X AIO.DIR DS     DIRELEN     DIRECTORY ENTRY
                        793X
    041.111             794X AIO.CNT DS     1           SECTOR COUNT
    041.112             795X AIO.EOM DS     1           END OF MEDIA FLAG
    041.113             796X AIO.EOF DS     1           END OF FILE FLAG
    041.114             797X AIO.TFP DS     2           TEMP FILE POINTERS
    041.116             798X AIO.CHA DS     2           ADDRESS OF CHANNEL BLOCK (IOC.DDA)


    041.120             800X S.SCR   DS     2           SYSTEM SCRATCH AREA ADDRESS
    041.122             801          XTEXT  MTRDEF


                        803X **     HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.
                        804X
    000.000             805X         ORG    0
    000.000             806X M.SYSN  DS     1           SYSCALL ITTERATION COUNT
    000.001             807X M.SALO  DS     1           STAND-ALONE FLAG
    000.002             808X M.CSLC  DS     1           LINES IN CONSOLE BUFFER
    000.003             809X M.CPRE  DS     1           CONSOLE PREVIOUS CHARACTER
    000.004             810X M.CRUB  DS     1           CONSOLE RUBOUT FLAG
    000.005             811X M.CINT  DS     1           CONSOLE INTERRUPT FLAG
    000.006             812X M.CIN   DS     2           CONSOLE CB IN POINTER
    000.010             813X M.COUT  DS     2           CONSOLE CB OUT POINTER
    000.012             814X M.CFWA  DS     2           CONSOLE CB FWA POINTER
    000.014             815X M.CLWA  DS     2           CONSOLE CB LWA POINTER
```

```
000.016                 816X M.CDLY  DS      1               CONSOLE PAD CHARACTER COUNT
000.017                 817X M.CDCA  DS      2               ADDRESS OF CHARACTER BEING PADDED
000.021                 818          XTEXT   FILDEF

                        820X **      FILDEF - FILE TYPE DEFINITIONS.
                        821X *
                        822X *       DB      377Q,FT.XXX
                        823X
                        824X
000.000                 825X FT.ABS  EQU     0               ABSOLUTE BINARY
000.001                 826X FT.PIC  EQU     1               POSITION INDEPENDANT CODE
000.002                 827X FT.REL  EQU     2               RELOCATABLE CODE
000.003                 828X FT.BAC  EQU     3               COMPILED BASIC CODE
000.021                 829          XTEXT   ABSDEF

                        831X **      ABS FORMAT EQUIVALENCES.
                        832X
000.000                 833X         ORG     0
                        834X
000.000                 835X ABS.ID  DS      1               377Q = BINARY FILE FLAG
000.001                 836X         DS      1               FILE TYPE (FT.ABS)
000.002                 837X ABS.LDA DS      2               LOAD ADDRESS
000.004                 838X ABS.LEN DS      2               LENGTH OF ENTIRE RECORD
000.006                 839X ABS.ENT DS      2               ENTRY POINT
                        840X
000.010                 841X ABS.COD DS      0               CODE STARTS HERE
```

```
                                844
042.170                         845          ORG     USERFWA-ABS.COD
042.170  377 000                846          DB      377Q,FT.ABS             ABS FILE
042.172  200 042                847          DW      USERFWA         LOAD ADDRESS
042.174  225 050                848          DW      LOADL-USERFWA   SIZE
042.176  210 112                849          DW      PRS             ENTRY
                                850

                                852  **      LOW-MEMORY CELLS USED BY BASIC
                                853
042.200                         854          DS      2               ACCX TYPE
042.202                         855  ACCX    DS      4
042.206                         856          DS      2               ACCY TYPE
042.210                         857  ACCY    DS      4
                                858
                                859  **      SPECIAL LEXICAL VARIABLE AREA.
                                860  *
                                861  *       VARIABLES ARE STORED HERE SO THAT BASIC CAN QUICKLY TELL THAT
                                862  *       THEY DONT RESIDE IN THE SYMBOL TABLE BY SIMPLY CHECKING THE
                                863  *       BANK ADDRESS.
                                864
042.214  000                    865          DB      0               TYPE OF LEXC
042.215  000 000 000            866  LEXC    DB      0,0,0,0         SPECIAL LEXICAL UNDEFINED VARIABLE VALUE
                                867
042.221  000                    868          DB      0               TYPE OF LEXB
042.222  000 000 000            869  LEXB    DB      0,0,0,0         SPECIAL LEXICAL CELL FOR NUMERIC LITTERALS
                                870
042.226                         871  LEXLIM  EQU     *               ALL SYMTAB VARIABLES OCCUR IN HIGHER MEM

                                873  **      FBLIST - FILE BLOCK LIST.                               /80.02.GC/
                                874  *
                                875  *       FBLIST CONTAINS TTHE FILE BLOCK FOR ALL POSSIBLE USER
                                876  *       CHANNELS, IN ORDER #1 TO #N.
                                877  *
                                878  *       THE FIRST ENTRY IN FBLIST IS NOT A USER ACCESSABLE FILE, BUT IS
                                879  *       THE SYSTEM'S INTERNAL WORK FILE.
                                880  *       THE 2ND ENTRY IS CHANNEL #2, THE 3RD CHANNEL #3, ETC.
                                881  *
                                882  *       NOTE:   These tables were moved to the front to avoid problems
                                883  *               when overlays are loaded, etc.                  /80.02.SC/
                                884
042.226  351 114                885  FBUFAD  DW      MTAREA+3        CURRENT CONTENTS OF FILTAB+MT.FWA
                                886
042.230                         887  FBLIST  DS      0
                                888
042.230  000 000                889  FBSCR   DB      0,0             CHANNEL AND STATUS
042.232  000 000 000            890          DW      0,0,0,0+512     USE  *HDOS*  SCRATCH RAM (PRS INITIALIZES IT)
042.242                         891          DS      FB.NAML
                                892
                                893  *       CHANNEL #1
                                894
042.263  001 000                895          DB      1,0             CHANNEL AND STATUS
```

```
042.265  351 114 351   896          DW      MTAREA+3,MTAREA+3,MTAREA+3,MTAREA+3+256
042.275                897          DS      FB.NAML
                       898
                       899  *       CHANNEL #2
                       900
042.316  002 000       901          DB      2,0             CHANNEL AND STATUS
042.320  351 115 351   902          DW      MTAREA+3+256,MTAREA+3+256,MTAREA+3+256,MTAREA+3+256+256
042.330                903          DS      FB.NAML
                       904
                       905  *       CHANNEL #3
                       906
042.351  003 000       907          DB      3,0             CHANNEL AND STATUS
042.353  351 116 351   908          DW      MTAREA+3+512,MTAREA+3+512,MTAREA+3+512,MTAREA+3+512+256
042.363                909          DS      FB.NAML
                       910
                       911  *       CHANNEL #4
                       912
043.004  004 000       913          DB      4,0             CHANNEL AND STATUS
043.006  351 117 351   914          DW      MTAREA+3+768,MTAREA+3+768,MTAREA+3+768,MTAREA+3+768+256
043.016                915          DS      FB.NAML
                       916
                       917  *       CHANNEL #5
                       918
043.037  005 000       919          DB      5,0             CHANNEL AND STATUS
043.041  351 120 351   920          DW      MTAREA+3+1024,MTAREA+3+1024,MTAREA+3+1024,MTAREA+3+1024+256
043.051                921          DS      FB.NAML



043.072  123 131 060   923  DEFALTF DB      'SYOBAS'        PROGRAM FILE DEFAULTS           /80.02,GC/
043.100  123 131 060   924  DEFALTD DB      'SYODAT'        DATA    FILE DEFAULTS           /80.02,GC/
```

```
                                926  ***     BASIC - MAIN EXEC LOOP.
                                927  *
                                928
043.106                         929  START   EQU     *
043.106  041 370 100            930          LXI     H,CBINT
043.111  076 002                931          MVI     A,CTLB
043.113  377 041                932          DB      SYSCALL,.CTLC   SETUP CTL-B HANDLER
043.115  041 363 100            933          LXI     H,CCINT
043.120  076 003                934          MVI     A,CTLC
043.122  377 041                935          DB      SYSCALL,.CTLC   SETUP CTL-C HANDLER
                                936
                                937  **      ACCEPT COMMAND OR TEXT.
                                938
043.124                         939  RESTART EQU     *               RESTART ADDRESS
                                940
                                941  *       AM IN COMMAND MODE, RESTORE SYSTEM TO COMMAND MODE.
                                942
043.124  061 200 042            943          LXI     SP,STACK        RESTORE STACK POINTER
043.127  041 124 043            944          LXI     H,RESTART
043.132  345                    945          PUSH    H               SET *RETURN ADDRESS*
043.133  257                    946          XRA     A
000.000                         947          ERRNZ   MI.NOP
043.134  062 111 076            948          STA     PNTC            CLEAR TOKEN PIPELINE
043.137  062 142 112            949          STA     CTLFLAG         CLEAR CTL-C AND CTL-B FLAGS
043.142  062 326 040            950          STA     S.CSLMD
000.000                         951          ERRNZ   RM.IMM
043.145  062 301 114            952          STA     RUNMOD          SET IMMEDIATE MODE
043.150  315 115 074            953          CALL    FOC             FILE OPEN CLEANUP
043.153  076 001                954          MVI     A,1
043.155  062 253 112            955          STA.    COLCNTS+0       SET COLUMN NUMBER FOR CONSOLE (*PRINT* CMD)
043.160  315 354 111            956          CALL    $CCO            CLEAR CTL-O
043.163  315 031 112            957          CALL    $GNL            GUARANTEE NEW LINE
043.166  315 136 031            958          CALL    $TYPTX
043.171  252                    959          DB      '*'+200Q        PROMPT
043.172  315 364 065            960          CALL    ICL             INPUT COMMAND LINE
043.175  322 203 043            961          JNC     BAS1            NO CTL-C HIT
                                962
                                963  *       CTL-C HIT, CLEAR CONSOLE AND RESTART
                                964
043.200  377 007                965          DB      SYSCALL,.CLRCO
043.202  311                    966          RET     RESTART         START AGAIN
                                967
043.203  302 152 070            968  BAS1    JNZ     ERR.SY          SYNTAX ERROR IN STATEMENT       /80.01.GC/
043.206  001 265 112            969          LXI     B,LINE
043.211  315 230 072            970          CALL    CNC             CLASSIFY NEXT CHARACTER         /80.01.GC/
043.214  247                    971          ANA     A               SEE IF KEYWORD
043.215  372 233 043            972          JM      BAS3            IS KEYWORD
043.220  376 002                973          CPI     CT.NUM                                          /80.01.GC/
043.222  302 240 043            974          JNZ     BAS2            IS NOT A NUMBER                 /80.01.GC/
                                975
                                976  *       HAVE STATEMENT WITH NUMBER.
                                977
043.225  315 021 045            978          CALL    CLR1            CLEAR REFERENCES TO TEXT
043.230  303 270 070            979          JMP     MTL             INSERT TEXT LINE
                                980
                                981  *       IS KEYWORD, SEE IF ALLOWED IMMEDIATE USAGE
```

```
                                982
043.233  376 250                983  BAS3    CPI     CT.IUA          IMMEDIATE USAGE ALLOWED?
043.235  322 125 070              984          JNC     ERR.IU          ILLEGAL USAGE
                                985
043.240  257                    986  BAS2    XRA     A
000.000                         987          ERRNZ   RM.IMM          SET IMMEDIATE MODE
                                988  *       JMP     EXEC            EXECUTE IN IMMEDIATE MODE
```

```
                                    991  **      EXEC - EXECUTE BASIC STATEMENTS.
                                    992  *
                                    993  *       EXEC CAUSES ONE OR MORE BASIC STATEMENTS TO BE EXECUTED.
                                    994  *
                                    995  *       ENTRY   (CURNUM) = CURNET LINE NUMBER
                                    996  *               (CURADR) = CURRENT LINE ADDRESS
                                    997  *               (A) = RUN MODE CONTROL
                                    998  *               (BC) = TEXT START ADDRESS
                                    999  *                                  (STEP, IMMEDIATE, CONTINUOUS)
                                   1000  *       EXIT    WHEN MODE CONTROL IS CLEARED, OR AT END OF LINE
                                   1001  *               FOR STEP AND IMMEDIATE MODES.
                                   1002  *       USES    ALL
                                   1003
                                   1004
     043.241                       1005  EXEC    EQU     *
     043.241  062 301 114          1006          STA     RUNMOD          SET RUN MODE
                                   1007
                                   1008  *       PERFORM THE NEXT COMMAND.
                                   1009
     043.244  041 124 043          1010  EXEC1   LXI     H,RESTART       SET ABNORMAL EXIT ADDRESS
     043.247  042 077 075          1011          SHLD    ILMA
     043.252  257                 1012          XRA     A
     043.253  062 140 112          1013          STA     IOCHAN          SET OUTPUT TO CONSOLE
     043.256  315 072 076          1014          CALL    PNT             PREVIEW NEXT TOKEN
     000.000                       1015          ERRNZ   CT.FIN
     043.261  247                 1016          ANA     A
     043.262  302 372 043          1017          JNE     EXEC3
     043.265  315 056 071          1018          CALL    ANT             CLEAR 'PNT' PIPELINE
                                   1019
                                   1020  *       END OF STATEMENT.
                                   1021
     043.270                       1022  EXEC2   EQU     *
     043.270  315 201 044          1023          CALL    EXEC7           SAVE CURRENT TEXT ADDRESS
                                   1024
                                   1025  *       CHECK FOR CONTROL CHARACTERS.
                                   1026
     043.273  041 142 112          1027          LXI     H,CTLFLAG
     043.276  176                 1028          MOV     A,M
     043.277  037                 1029          RAR
     043.300  332 106 070          1030          JC      ERR.CC          CONTROL-C HIT
     043.303  037                 1031          RAR
     043.304  334 215 044          1032          CC      EXEC8           USER INTERRUPT
                                   1033
                                   1034  *       CHECK FOR HALT
                                   1035
     043.307  072 301 114          1036          LDA     RUNMOD
     043.312  147                 1037          MOV     H,A
     043.313  247                 1038          ANA     A
     000.000                       1039          ERRNZ   RM.HLT-200Q
     043.314  370                 1040          RM                      AM TO HALT
                                   1041
                                   1042  *       SETUP CORRECT DISPLAY MODE FOR FPLEDS.
                                   1043
     043.315  076 000              1044          MVI     A,0             (A) = MODE INDEX
     043.316                       1045  FPMODE  EQU     *-1
     043.317  021 244 044          1046          LXI     D,EXECA
```

```
043,322  203          1047          ADD     E
043,323  137          1048          MOV     E,A
043,324  032          1049          LDAX    D               (A) = FLAG VALUE
043,325  062 010 040  1050          STA     .MFLAG          SET TYPE OF DISPLAY
                      1051
                      1052  *       CHECK TO SEE IF ANOTHER STATEMENT ON THIS LINE
                      1053
043,330  012          1054          LDAX    B
043,331  003          1055          INX     B
043,332  247          1056          ANA     A
043,333  302 244 043  1057          JNZ     EXEC1           DO NEXT STATEMENT
043,336  174          1058          MOV     A,H             (A) = RUNMODE
043,337  247          1059          ANA     A
000,000               1060          ERRNZ   RM.HLT-2000     REMOVE HALT FLAG
043,340  310          1061          RZ                      IMMEDIATE MODE
                      1062
                      1063  *       ADVANCE TO NEXT PROGRAM LINE.
                      1064
043,341  012          1065          LDAX    B
043,342  003          1066          INX     B
043,343  157          1067          MOV     L,A             SET LINE NUMBER
043,344  012          1068          LDAX    B
043,345  003          1069          INX     B
043,346  147          1070          MOV     H,A
043,347  042 133 112  1071          SHLD    CURNUM
043,352  245          1072          ANA     L               (A) = PRODUCT OF LINE NUMBER BYTES
043,353  074          1073          INR     A
043,354  076 253       1074          MVI     A,CT.END
043,356  312 040 044  1075          JZ      EXEC6           END OF TEXT - GENERATE 'END'
043,361  315 201 044  1076          CALL    EXEC7
043,364  376 001       1077          CPI     RM.STE
043,366  310          1078          RE                      DONE STEPPING
043,367  303 244 043  1079          JMP     EXEC1           PROCESS NEXT STATEMENT
                      1080
                      1081  *       PROCESS LINE.
                      1082
043,372  315 007 044  1083  EXEC3   CALL    EXEC4
                      1084
                      1085  *       RETURN FROM STATEMENT PROCESSOR. MUST HAVE END OF STATEMENT.
                      1086
043,375  315 305 077  1087  EXEC3.5 CALL    RNT
044,000  000          1088          DB      CT.FIN          REQUIRE CT.FIN
044,001  315 357 073  1089          CALL    DTS             DELETE TEMP STRINGS
044,004  303 270 043  1090          JMP     EXEC2
                      1091
                      1092
044,007  376 200       1093  EXEC4   CPI     CT.BLD
044,011  332 125 070  1094          JC      ERR.IU          ILLEGAL USAGE
                      1095
044,014  376 256       1096          CPI     CT.CMD
044,016  322 374 050  1097          JNC     LET             MUST BE 'LET', IS NOT COMMAND
                      1098
044,021  376 212       1099          CPI     CT.RUA
044,023  322 035 044  1100          JNC     EXEC5           RUN USAGE ALLOWED
                      1101
044,026  072 301 114  1102          LDA     RUNMOD
```

```
000.000                 1103            ERRNZ   RM.IMM
044.031  247           1104            ANA     A
044.032  302 125 070    1105            JNE     ERR.IU          ILLEGAL USAGE FOR IMMEDIATE MODE
                        1106
044.035  315 056 071    1107  EXEC5     CALL    ANT             ACCEPT NEXT TOKEN
044.040  326 200        1108  EXEC6     SUI     200Q            REMOVE BIAS
044.042  315 061 031    1109            CALL    $TJMP           ENTER PROCESSOR
                        1110
044.045  247 044        1111            DW      BUILD
044.047  337 044        1112            DW      BYE
044.051  163 045        1113            DW      CONT            CONTINUE
044.053  162 046        1114            DW      DELETE
044.055  020 051        1115            DW      LIST
044.057  233 053        1116            DW      REPLACE
044.061  155 045        1117            DW      RUN
044.063  302 053        1118            DW      SAVE
044.065  351 044        1119            DW      SCRATCH
044.067  356 053        1120            DW      STEP
                        1121
044.071  152 070        1122            DW      ERR.SY          LEXICAL SYNTAX ERROR FOUND
044.073  205 045        1123            DW      CHAIN
044.075  363 044        1124            DW      CLEAR
044.077  260 045        1125            DW      CLOSE
044.101  320 045        1126            DW      CNTRL           CNTRL
044.103  236 046        1127            DW      DIM             DIMENSION
044.105  152 070        1128            DW      ERR.SY          FN
044.107  060 047        1129            DW      FOR
044.111  213 047        1130            DW      FREE
044.113  336 047        1131            DW      FREEZE
044.115  026 050        1132            DW      GOSUB
044.117  031 050        1133            DW      GOTO
044.121  051 050        1134            DW      IF
044.123  374 050        1135            DW      LET
044.125  175 051        1136            DW      LOCK
044.127  203 051        1137            DW      NEXT
044.131  332 051        1138            DW      OLD
044.133  355 051        1139            DW      ON
044.135  036 052        1140            DW      OPEN
044.137  220 052        1141            DW      OUT
044.141  251 052        1142            DW      PAUSE
044.143  336 052        1143            DW      POKE
044.145  343 052        1144            DW      PRINT
044.147  171 053        1145            DW      READ
044.151  121 050        1146            DW      IF2             REM
044.153  053 045        1147            DW      RESTORE
044.155  242 053        1148            DW      RETURN
044.157  041 054        1149            DW      UNFREZ          UNFREEZE
044.161  176 051        1150            DW      UNLOCK
044.163  065 054        1151            DW      UNSAVE
                        1152
044.165  137 050        1153            DW      LINPUT
044.167  342 077        1154            DW      SES             DATA
044.171  133 046        1155            DW      DEF
044.173  044 047        1156            DW      END
044.175  150 050        1157            DW      INPUT
044.177  030 054        1158            DW      STOP
```

```
                                1160  *       END OF EXEC SEQUENCE. SAVE TEXT POINTER.
                                1161
044.201  072 301 114            1162  EXEC7   LDA     RUNMOD
044.204  346 177                1163          ANI     377Q-RM.HLT
000.000                         1164          ERRNZ   RM.IMM
044.206  310                    1165          RZ                      AM IN IMMEDIATE MODE
044.207  140                    1166          MOV     H,B             (HL) = TEXT ADDRESS
044.210  151                    1167          MOV     L,C
044.211  042 135 112            1168          SHLD    CURADR
044.214  311                    1169          RET


                                1171  **      CTL-B (USER INTERRUPT) HIT
                                1172
044.215  021 000 000            1173  EXEC8   LXI     D,0             (DE) = INTERRUPT EXIT ADDRESS
044.216                         1174  ACTLB   EQU     *-2
044.220  172                    1175          MOV     A,D
044.221  263                    1176          ORA     E
044.222  312 111 070            1177          JZ      ERR.CB          NO USER PROCESSING
                                1178
                                1179  *       USER PROGRAM PROCESSING SPECIFIED.
                                1180
044.225  176                    1181  EXEC9   MOV     A,M
044.226  346 375                1182          ANI     377Q-CFCTLB
044.230  167                    1183          MOV     M,A             CLEAR FLAG
044.231  341                    1184  EXEC10  POP     H               DISCARD 'RETURN ADDRESS'
044.232  353                    1185          XCHG                    (HL) = TEXT ADDRESS
044.233  315 143 100            1186          CALL    SRA             SAVE TEXT RETURN ADDRESS
044.236  315 042 050            1187          CALL    GOTO2           PROCESS AS GOTO
044.241  303 375 043            1188          JMP     EXEC3.5         EXIT FROM GOSUB
                                1189
                                1190  **      TABLE OF .MFLAG VALUES FOR DISPLAY CONTROL.
                                1191
044.244  301                    1192  EXECA   DB      UO.NFR+UO.HLT+UO.CLK    NO DISPLAY
044.245  203                    1193          DB      UO.DDU+UO.HLT+UO.CLK    DISABLE UPDATE
044.246  201                    1194          DB      UO.HLT+UO.CLK           LEAVE ON AND UPDATING
                                1195
000.044                         1196  .       SET     */256
000.000                         1197          ERRNZ   EXECA/256-.     ASSUME IN SAME BANK
                                1198


                                1201  **      BUILD - PROCESS BUILD COMMAND.
                                1202  *
                                1203  *       BUILD N,M
                                1204  *
                                1205  *       STARTING AT LINE N, INCREMENT BY M
                                1206
                                1207
044.247                         1208  BUILD   EQU     *
044.247  315 313 075            1209          CALL    LFC             CHECK FOR DATA LOCK
044.252  315 235 052            1210          CALL    OUT1            (DE) = INC, (HL) = VAL
```

```
044.255  325          1211  BLD1    PUSH    D               SAVE INC
044.256  345          1212          PUSH    H               SAVE NUMBER
044.257  353          1213          XCHG
044.260  315 206 072  1214          CALL    CLN             CHECK FOR LEGAL NUMBER
044.263  315 206 100  1215          CALL    TDI             TYPE LINE NUMBER
044.266  315 364 065  1216          CALL    ICL             ACCEPT NEW LINE
044.271  332 124 043  1217          JC      RESTART         CTL-C HIT
044.274  302 320 044  1218          JNZ     BLD2            ERROR IN LINE
044.277  041 265 112  1219          LXI     H,LINE
044.302  321          1220          POP     D
044.303  325          1221          PUSH    D               (DE) = NUMBER
044.304  315 304 070  1222          CALL    MTLO            INSERT TEXT LINE
044.307  341          1223          POP     H               (HL) = NUMBER
044.310  321          1224          POP     D               (DE) = INC
044.311  031          1225          DAD     D
044.312  332 122 070  1226          JC      ERR.IN          OVERFLOW
044.315  303 255 044  1227          JMP     BLD1
                      1228
                      1229  *       ERROR IN LINE
                      1230
044.320  315 136 031  1231  BLD2    CALL    $TYPTX
044.323  207          1232          DB      BELL+200Q
044.324  076 214      1233          MVI     A,BEC.SY
044.326  046 012      1234          MVI     H,NL
044.330  377 057      1235          DB      SYSCALL,.ERROR  SHOW ERROR
044.332  341          1236          POP     H
044.333  321          1237          POP     D
044.334  303 255 044  1238          JMP     BLD1            RE-TRY LINE ENTRY


                      1240  ***     BYE - RETURN TO HDOS.
                      1241  *
                      1242  *       BYE
                      1243
                      1244
044.337               1245  BYE     EQU     *
044.337  315 313 075  1246          CALL    LFC             CHECK FOR DATA LOCK
044.342  315 146 071  1247          CALL    AYS             ARE YOU SURE?
044.345  300          1248          RNE                     NOT SURE
044.346  257          1249          XRA     A
044.347  377 000      1250          DB      SYSCALL,.EXIT   EXIT


                      1252  **      SCRAT - SCRATCH SYSTEM.
                      1253  *
                      1254  *       DESTROY TEXT, CLEAR VARIABLES.
                      1255
                      1256
044.351               1257  SCRATCH EQU     *
044.351  315 313 075  1258          CALL    LFC             CHECK FOR DATA LOCK
044.354  315 146 071  1259          CALL    AYS             ARE YOU SURE?
044.357  300          1260          RNE                     NOT SURE
```

```
044.360  315 320 077  1261  SCR.    CALL    SCRA            INSERT DUMMY LAST LINE INTO TEXT TABLE
                       1262  *       JMP     CLEAR

                       1264  **      CLEAR - MASTER CLEAR.
                       1265  *
                       1266  *       CLEAR RESETS ALL CONTROL STRUCTURES:
                       1267  *
                       1268  *       1) GOSUB STACK
                       1269  *       2) 'FOR' STACK
                       1270  *       3) NEXT STATEMENT INDEX
                       1271  *       4) CLEAR VARIABLE LIST
                       1272  *       5) DATA POINTER
                       1273
                       1274
044.363                1275  CLEAR   EQU     *
044.363  315 313 075  1276          CALL    LFC             CHECK FOR DATA LOCK
044.366  315 056 071  1277          CALL    ANT
000.000                1278          ERRNZ   CT.FIN
044.371  247         1279          ANA     A
044.372  302 062 045  1280          JNZ     CLR2            HAVE VARIABLE
044.375  315 357 073  1281  CLEAR.  CALL    DTS
045.000  041 000 000  1282          LXI     H,0
045.003  042 112 112  1283          SHLD    STRTAB+MT,LEN
045.006  042 066 112  1284          SHLD    SYMTAB+MT.LEN
045.011  056 200      1285          MVI     L,200Q
045.013  042 143 112  1286          SHLD    STRVI           CLEAR STRING INDEX
045.016  315 171 072  1287          CALL    CLF             CLEAR FILE STRUCTURES
                       1288
                       1289
                       1290  *       ENTRY POINT FOR ROUTINES TO CLEAR REFERENCES TO TXTTAB.
                       1291
045.021  041 000 000  1292  CLRI    LXI     H,0             ENTRY TO JUST CLEAR TXTTAB REFERENCES
045.024  042 073 112  1293          SHLD    FORTAB+MT.LEN
045.027  042 100 112  1294          SHLD    GOSTAB+MT.LEN
045.032  042 105 112  1295          SHLD    WRKTAB+MT.LEN
045.035  042 216 044  1296          SHLD    ACTLB
045.040  056 300      1297          MVI     L,300Q
045.042  042 366 073  1298          SHLD    DTSA            CLEAR TEMP ONDEX
045.045  041 345 114  1299          LXI     H,MTAREA-1
045.050  042 135 112  1300          SHLD    CURADR          CLEAR ADDREAA

                       1302  **      RESTORE - RESTORE DATA POINTER
                       1303  *
                       1304  *       RESTORE
                       1305
                       1306
045.053  041 345 114  1307  RESTORE LXI     H,MTAREA-1
045.056  042 303 114  1308          SHLD    DATPTR
045.061  311          1309          RET
                       1310
```

```
                                1311  *       CLEAR VARIABLE
                                1312
045.062  376 300                1313  CLR2    CPI     CT.VARL
045.064  332 152 070            1314          JC      ERR.SY          NOT VARIABLE
045.067  376 306                1315          CPI     CT.SSF+1
045.071  322 152 070            1316          JNC     ERR.SY          NOT VARIABLE
045.074  147                    1317          MOV     H,A             SAVE (A) IN H
045.075  076 042                1318          MVI     A,LEXLIM/256
045.077  272                    1319          CMP     D
045.100  320                    1320          RNC                     IS NOT IN SYMBOL TABLE
045.101  174                    1321          MOV     A,H             (A) = VARIABLE TYPE
045.102  041 006 000            1322          LXI     H,6             (HL) = SIZE TO CLEAR
045.105  346 002                1323          ANI     CF.VEC
045.107  312 132 045            1324          JZ      CLR3            NOT VECTOR
045.112  032                    1325          LDAX    D
045.113  247                    1326          ANA     A
045.114  372 132 045            1327          JM      CLR3            IS FUNCTION
045.117  325                    1328          PUSH    D               SAVE ADDR OF AREA+2
045.120  345                    1329          PUSH    H               SAVE #6
045.121  023                    1330          INX     D
045.122  023                    1331          INX     D
045.123  353                    1332          XCHG
045.124  136                    1333          MOV     E,M
045.125  043                    1334          INX     H
045.126  126                    1335          MOV     D,M             (DE) = SIZE OF ARRAY
045.127  341                    1336          POP     H               (HL) = 6
045.130  031                    1337          DAD     D               (HL) = TOTAL SIZE
045.131  321                    1338          POP     D               (DE) = VARIABLE AREA+2
045.132  033                    1339  CLR3    DCX     D
045.133  033                    1340          DCX     D               (DE) = VARIABLE FWA
045.134  345                    1341          PUSH    H               SAVE COUNT TO REMOVE
045.135  052 064 112            1342          LHLD    SYMTAB+MT.FWA
045.140  173                    1343          MOV     A,E             COMPUTE INDEX INTO SYMTAB
045.141  225                    1344          SUB     L
045.142  157                    1345          MOV     L,A
045.143  172                    1346          MOV     A,D
045.144  234                    1347          SBB     H
045.145  147                    1348          MOV     H,A             (HL) = INDEX
045.146  321                    1349          POP     D               (DE) = DELETE COUNT
045.147  315 203 104            1350          CALL    $DBT            DELETE FROM SYMTAB
045.152  064 112                1351          DW      SYMTAB+1
045.154  311                    1352          RET                     DONE


                                1354  **      RUN - BEGIN EXECUTION.
                                1355  *
                                1356  *       RUN IS THE SAME AS
                                1357  *
                                1358  *       CLEAR: CONTINUE
                                1359
                                1360
045.155  315 313 075            1361  RUN     CALL    LFC             CHECK FOR DATA LOCK
045.160  315 375 044            1362          CALL    CLEAR.
```

```
                                1364  **      CONT - RESUME EXECUTION.
                                1365  *
                                1366
                                1367
045.163  076 004               1368  CONT    MVI     A,RM.CON        (A) = NEW RUN MODE
045.165  052 135 112           1369  CONT1   LHLD    CURADR
045.170  104                   1370          MOV     B,H
045.171  115                   1371          MOV     C,L             (BC) = CURRENT TEXT ADDRESS
045.172  315 241 043           1372          CALL    EXEC            EXECUTE WITH REQUESTED MODE
045.175  001 345 114           1373          LXI     B,ZERO          POINT TO ZERO BYTE
000.000                        1374          ERRNZ   RM.IMM
045.200  257                   1375          XRA     A
045.201  062 301 114           1376          STA     RUNMOD          RESTORE IMMEDIATE MODE
045.204  311                   1377          RET


                                1379  ***     CHAIN - CHAIN TO NEW PROGRAM.
                                1380  *
                                1381  *       CHAIN <STRING> [<LINE NUMBER>]
                                1382  *
                                1383  *       LEAVE DATA, VARIABLES, AND CHANNELS INTACT
                                1384
                                1385
045.205                         1386  CHAIN   EQU     *
045.205  341                   1387          POP     H               ** KLUDGE ** TO CLEAN STACK FOR RECURSIVE CALL TO *CONT*
045.206  341                   1388          POP     H
045.207  315 053 072           1389          CALL    CFN             COPY FILE NAME
045.212  315 072 076           1390          CALL    PNT             SEE IF LINE # FOLLOWS
045.215  247                   1391          ANA     A
000.000                        1392          ERRNZ   CT.FIN
045.216  312 231 045           1393          JZ      CHAIN1          NO LINE NUMBER
045.221  315 223 072           1394          CALL    CMA             GOBBLE COMMA
045.224  315 033 074           1395          CALL    ELN             EVAL LINE NUMBER
045.227  366 001                1396          ORI     1               CLEAR 'Z'
045.231  325                   1397  CHAIN1  PUSH    D               SAVE LINE NUMBER (GARBAGE IF NO NUMBER)
045.232  365                   1398          PUSH    PSW             'Z' SET IF NO LINE NUMBER
045.233  315 206 077           1399          CALL    RNP             READ NEW PROGRAM
045.236  361                   1400          POP     PSW             'Z' SET IF NO NUMBER
045.237  321                   1401          POP     D               (DE) = NUMBER
045.240  312 163 045           1402          JZ      CONT            JUST CONTINUE
                                1403
                                1404  *       HAD LINE NUMBER, NOW FIND IT
                                1405
045.243  315 242 074           1406          CALL    FLN             FIND LINE BY NUMBER
045.246  332 147 070           1407          JC      ERR.SN
045.251  053                   1408          DCX     H               POINT TO TERMINATOR OF PREVIOUS LINE
045.252  042 135 112           1409          SHLD    CURADR
045.255  303 163 045           1410          JMP     CONT            PROCESS AS CONTINUE
```

```
                                1412  ***     CLOSE - CLOSE FILE.
                                1413  *
                                1414  *       CLOSE   #I [,#J,...,#N]
                                1415  *
                                1416  *       CLOSE FILES #I THROUGH #N
                                1417  *
                                1418  *       NO ERROR MESSAGE IF FILE ALREADY CLOSED.
                                1419
                                1420
045.260                         1421  CLOSE   EQU     *
045.260  315 273 073            1422          CALL    DCN.            DECODE CHANNEL NUMBER
045.263  305                    1423          PUSH    B               SAVE TEXT POINTER
045.264  072 140 112            1424          LDA     IOCHAN
045.267  075                    1425          DCR     A
045.270  315 005 072            1426          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
045.273  332 304 045            1427          JC      CLOSE1          CHANNEL DOSENT EXIST
045.276  315 335 102            1428          CALL    $FCLO           CLOSE IT
045.301  315 326 073            1429          CALL    DNF             DELETE NON-OPEN FILE BLOCKS
045.304  301                    1430  CLOSE1  POP     B
045.305  315 072 076            1431          CALL    PNT             CHECK NEXT TOKEN
000.000                         1432          ERRNZ   CT.FIN
045.310  247                    1433          ANA     A
045.311  310                    1434          RZ                      DONE WITH STATEMENT
045.312  315 223 072            1435          CALL    CMA             REQUIRE COMMA
045.315  303 260 045            1436          JMP     CLOSE           CRACK ANOTHER


                                1438  **      CNTRL - CONTROL COMMAND.
                                1439  *
                                1440  *       CNTRL I,J
                                1441  *
                                1442  *       I=0     SET CTL-B PROCESSOR LINE
                                1443  *               J=N             LINE NUMBER
                                1444  *
                                1445  *       I=1     SET PRINTING MODE
                                1446  *               J=N             SET SCIENTIFIC THRESHOLD
                                1447  *
                                1448  *       I=2     SET DISPLAY MODE
                                1449  *               J=0  DISPLAYS OFF
                                1450  *               J=1  DISPLAYS REFRESHED, NOT UPDATED
                                1451  *               J=2  DISPLAYS REFRESHED AND UPDATED
                                1452  *
                                1453  *       I=3     SET TAB SIZE
                                1454  *               J=NN            WIDTH OF TAB FIELD
                                1455  *
                                1456  *       I=4     SET OVERLAY FLAG
                                1457  *               J=0  USE MAXIMUM AMOUNT OF MEMORY
                                1458  *               J=1  ALLOW OVERLAY TO REMAIN RESIDENT
                                1459
                                1460
045.320                         1461  CNTRL   EQU     *
045.320  315 235 052            1462          CALL    OUT1            (L) = I, (E) = J
045.323  175                    1463          MOV     A,L
045.324  376 005                1464          CPI     CNTLMX
```

```
045.326  322 122 070  1465          JNC     ERR.IN          TOO BIG A NUMBER
045.331  315 076 031  1466          CALL    $TBRA
045.334  005          1467  CNTLA   DB      CNTL1-*
045.335  016          1468          DB      CNTL2-*
045.336  033          1469          DB      CNTL3-*
045.337  044          1470          DB      CNTL4-*
045.340  101          1471          DB      CNTL5-*
000.005               1472  CNTLMX  EQU     *-CNTLA         MAX NUMBER OF FUNCTIONS - 1
                      1473
                      1474
                      1475  *       SET CTL-B PROCESSOR.
                      1476
045.341  315 242 074  1477  CNTL1   CALL    FLN             FIND LINE BY NUMBER
045.344  332 147 070  1478          JC      ERR.SN          NOT FOUND
045.347  042 216 044  1479          SHLD    ACTLB           SET ADDRESS
045.352  311          1480          RET
                      1481
                      1482  *       SET SCIENTIFIC THRESHOLD
                      1483
045.353  173          1484  CNTL2   MOV     A,E             SET THRESHOLD
045.354  074          1485          INR     A
045.355  376 010       1486          CPI     7+1                                     /78.10.GC/
045.357  322 122 070  1487          JNC     ERR.IN          LIM SIZE DUE TO ACC. OF FLT. PT./78.10.GC/
045.362  062 360 110  1488          STA     FTAC
045.365  062 370 110  1489          STA     FTAD                                    /78.10.GC/
045.370  311          1490          RET
                      1491
                      1492  *       SET DISPLAY MODE.
                      1493
045.371  173          1494  CNTL3   MOV     A,E
045.372  376 003       1495          CPI     3
045.374  322 122 070  1496          JNC     ERR.IN          IF ILLEGAL VALUE
045.377  062 316 043  1497          STA     FPMODE          SET DISPLAY MODE
046.002  311          1498          RET
                      1499
                      1500  *       SET TAB SIZE
                      1501
046.003  173          1502  CNTL4   MOV     A,E
046.004  247          1503          ANA     A
046.005  312 122 070  1504          JZ      ERR.IN          BAD VALUE
046.010  062 062 053  1505          STA     PRIC
                      1506
046.013  257          1507          XRA     A                                       /80.01.GC/
046.014  041 331 040  1508          LXI     H,S.CONWI                               /80.01.GC/
046.017  203          1509  CNTL43  ADD     E                                       /80.01.GC/
046.020  332 033 046  1510          JC      CNTL46                                  /80.01.GC/
046.023  276          1511          CMP     M                                       /80.01.GC/
046.024  332 017 046  1512          JC      CNTL43          NOT >= CONSOLE WIDTH    /80.01.GC/
046.027  312 033 046  1513          JZ      CNTL46          IS AN INTEGRAL MULTIPLE /80.01.GC/
046.032  223          1514          SUB     E                                       /80.01.GC/
046.033  223          1515  CNTL46  SUB     E                                       /80.01.GC/
046.034  074          1516          INR     A               ADJUST AT THE LIMIT POINTS /80.01.GC/
                      1517
046.035  062 050 053  1518          STA     PRIB            SET TAB-FIELD WRAP WIDTH
046.040  311          1519          RET
                      1520
```

```
                        1521  *       SET OVERLAY LOAD OPTIONS
                        1522
046.041  172            1523  CNTL5   MOV     A,D
046.042  247            1524          ANA     A                                       /78.10.GC/
046.043  302 122 070    1525          JNZ     ERR.IN          BAD VALUE               /78.10.GC/
046.046  263            1526          ORA     E
046.047  376 002        1527          CPI     1+1                                     /78.10.GC/
046.051  322 122 070    1528          JNC     ERR.IN                                  /78.10.GC/
046.054  062 141 112    1529          STA     OVLMAN          SET OVERLAY MANAGE FLAGS
046.057  247            1530          ANA     A                                       /78.10.GC/
046.060  312 115 074    1531          JZ      FOC             OPEN TABLES             /78.10.GC/
                        1532
                        1533  *       GET THE NEW OVERLAY MEMORY                      /80.01.GC/
                        1534
046.063  315 054 031    1535          CALL    $SAVALL                                 /80.01.GC/
046.066  315 230 074    1536          CALL    FOP.            SQUEEZE TABLES          /80.01.GC/
046.071  345            1537          PUSH    H               SAVE LWA                /80.01.GC/
                        1538
046.072  052 350 040    1539          LHLD    S.OFWA                                  /80.01.GC/
000.000                 1540          ERRNZ   OVLO                                    /80.01.GC/
046.075  021 006 000    1541          LXI     D,OVL.FLB                               /80.01.GC/
046.100  031            1542          DAD     D               HL = ADDR. OF FLAG BYTE /80.01.GC/
046.101  176            1543          MOV     A,M             A  = FLAG BYTE          /80.01.GC/
046.102  346 001        1544          ANI     OVL.IN                                  /80.01.GC/
046.104  052 320 040    1545          LHLD    S.SYSM                                  /80.01.GC/
046.107  021 360 377    1546          LXI     D,-16                                   /80.01.GC/
046.112  031            1547          DAD     D               LEAVE SOME SLOP         /80.01.GC/
046.113  302 126 046    1548          JNZ     CNTL52          ALREADY IN MEMORY       /80.01.GC/
                        1549
                        1550  *       LEAVE ROOM FOR THE OVERLAY                      /80.01.GC/
                        1551
046.116  353            1552          XCHG                                            /80.01.GC/
046.117  052 324 040    1553          LHLD    S.OMAX                                  /80.01.GC/
046.122  315 224 030    1554          CALL    $CHL            HL = -HL                /80.01.GC/
046.125  031            1555          DAD     D                                       /80.01.GC/
                        1556
046.126  321            1557  CNTL52  POP     D                                       /80.01.GC/
046.127  353            1558          XCHG                    DE = PROSPECTUS, HL = LIMIT /80.01.GC/
046.130  303 152 074    1559          JMP     FOC1.3                                  /80.01.GC/


                        1561  **      DEF - DEFINE FUNCTION.
                        1562  *
                        1563  *       1 LINE FUNCTIONS:
                        1564  *
                        1565  *       DEF FN X(P1,...,PN) = EXPR
                        1566
                        1567
046.133                 1568  DEF     EQU     *
046.133  315 305 077    1569          CALL    RNT
046.136  220            1570          DB      CT.FN           REQUIRE 'FN'
046.137  315 263 075    1571          CALL    IVT             INSERT VECTOR IN TABLE
046.142  032            1572          LDAX    D
046.143  075            1573          DCR     A
```

```
046.144  362 152 070  1574           JP      ERR.SY          IS DIMENSIONED
                      1575
                      1576  *        IS SINGLE LINE DEFINITION.
                      1577
046.147  076 201      1578           MVI     A,201Q
046.151  022          1579           STAX    D
046.152  023          1580           INX     D
046.153  353          1581           XCHG
046.154  161          1582           MOV     M,C
046.155  043          1583           INX     H
046.156  160          1584           MOV     M,B             SET FUNCTION ADDRESS
046.157  303 342 077  1585           JMP     SES             SKIP TO STATEMENT END AND EXIT


                      1587  **       DELETE - DELETE LINES.
                      1588  *
                      1589  *        DELETE NNN,MMM
                      1590
                      1591
046.162               1592  DELETE   EQU     *
046.162  315 313 075  1593           CALL    LFC             CHECK FOR DATA LOCK
046.165  315 036 057  1594           CALL    EVALI           (DE) = 1ST LINE NUMBER
046.170  315 223 072  1595           CALL    CMA             REQUIRE ','
046.173  315 242 074  1596           CALL    FLN             FIND LINE BY NUMBER
046.176  345          1597           PUSH    H               SAVE ADDRESS
046.177  315 036 057  1598           CALL    EVALI
046.202  023          1599           INX     D
046.203  315 242 074  1600           CALL    FLN             FIND LAST
046.206  353          1601           XCHG
046.207  341          1602           POP     H               (HL) = FWA, (DE) = LWA
046.210  175          1603           MOV     A,L
046.211  223          1604           SUB     E
046.212  137          1605           MOV     E,A
046.213  174          1606           MOV     A,H
046.214  232          1607           SBB     D
046.215  127          1608           MOV     D,A             (DE) = BYTE COUNT TO DELETE
046.216  322 152 070  1609           JNC     ERR.SY          FIRST > LAST
046.221  325          1610           PUSH    D               SAVE COUNT
046.222  021 032 263  1611           LXI     D,-MTAREA
046.225  031          1612           DAD     D               (HL) = TABLE INDEX OF 1ST LINE TO DELETE
046.226  321          1613           POP     D               (DE) = COUNT
046.227  067          1614           STC                     NUMBER IS NEG.  SET 17TH BIT OF NUMBER
046.230  315 213 104  1615           CALL    $IBT            REMOVE BYTES
046.233  057 112      1616           DW      TXTTAB+1
046.235  311          1617           RET
```

```
                                1619  **      DIM - PROCESS DIMENSION DECLARATION.
                                1620  *
                                1621  *       DIM ITEM1(X1,...,XN),...,ITEMN(X1,...,XP)
                                1622
                                1623
046.236                         1624  DIM     EQU     *
046.236  052 086 112            1625          LHLD    SYMTAB+MT.LEN
046.241  042 034 047            1626          SHLD    DIMA            SET BEFORE SYMTAB LEN
046.244  041 033 047            1627          LXI     H,DIM5
046.247  042 077 075            1628          SHLD    ILMA            SET ABORT PROCESSOR
046.252  315 263 075            1629          CALL    IVT             INSERT VECTOR IN SYMBOL TABLE
                                1630
046.255  315 000 073            1631          CALL    CSI             (DE) = INDEX INTO SYMTAB
046.260  325                    1632          PUSH    D               SAVE INDEX INTO SYMTAB
                                1633
                                1634  *       DECODE AND STORE DIMENSION BOUNDS IN VECTAB.
                                1635
046.261  041 001 000            1636          LXI     H,1             (HL) = ARRAY SIZE ACCUMULATOR
046.264  134                    1637          MOV     E,H             (E) = 0 = DIMENSION COUNT
046.265  034                    1638  DIM2    INR     E               INCREMENT DIMENSION COUNT
046.266  325                    1639          PUSH    D
046.267  315 036 057            1640          CALL    EVALI           EVALUATE NUMERIC EXPRESSION
046.272  023                    1641          INX     D               (DE) = BOUND+1
046.273  325                    1642          PUSH    D               SAVE BOUND
046.274  305                    1643          PUSH    B               SAVE (BC)
046.275  104                    1644          MOV     B,H             (BC) = CURRENT ARRAY SIZE
046.276  115                    1645          MOV     C,L
046.277  315 337 030            1646          CALL    $MU66           (HL) = NEW ARRAY SIZE
046.302  302 160 070            1647          JNZ     ERR.TO          OVERFLOW
046.305  301                    1648          POP     B
046.306  343                    1649          XTHL                    PUSH SIZE UNDER DIMENSION BOUND
046.307  345                    1650          PUSH    H
046.310  041 002 000            1651          LXI     H,2
046.313  021 064 112            1652          LXI     D,SYMTAB+1
046.316  315 026 071            1653          CALL    AMB             ALLOCATE 2 BYTES TO STORE BOUND .
046.321  321                    1654          POP     D               (DE) = DIMENSION BOUND
046.322  163                    1655          MOV     M,E
046.323  043                    1656          INX     H
046.324  162                    1657          MOV     M,D             STORE IN TABLE
046.325  315 056 071            1658          CALL    ANT             ACCEPT NEXT TOKEN
046.330  341                    1659          POP     H               (HL) = ARRAY SIZE
046.331  321                    1660          POP     D               (E) = DIMENSION COUNT
046.332  376 026                1661          CPI     CT.CMA
046.334  312 265 046            1662          JE      DIM2            GET ANOTHER
046.337  376 020                1663          CPI     CT.PAR
046.341  302 152 070            1664          JNE     ERR.SY          REQUIRE )
                                1665
                                1666  *       READ ALL BOUNDS. SET SUBSCRIPT COUNT IN SYMTAB.
                                1667
046.344  173                    1668          MOV     A,E             (A) = SUBSCRIPT COUNT
046.345  321                    1669          POP     D               (DE) = INDEX INTO SYMBOL
                                1670
046.346  315 366 072            1671          CALL    CSA             (DE) = ABSOLUTE ADDRESS IN SYMTAB
                                1672
046.351  325                    1673          PUSH    D
046.352  022                    1674          STAX    D               SET DIMENSION COUNT
```

```
046.353  051            1675          DAD     H               (HL) = 2*(HL)
046.354  332 122 070    1676          JC      ERR.IN          TOO LARGE
046.357  051            1677          DAD     H               (HL) = 4*HL
046.360  332 122 070    1678          JC      ERR.IN          TOO LARGE
                        1679
                        1680  *       INSERT LENGTH OF AREA IN HEADER, (HL) = STORAGE NEEDED
                        1681
046.363  207            1682          ADD     A               (A) = NUMBER OF DIMENSIONS *2
046.364  353            1683          XCHG
046.365  046 000        1684          MVI     H,0
046.367  157            1685          MOV     L,A             (HL) = LENGTH OF BOUNDS
046.370  031            1686          DAD     D               (HL) = TOTAL LENGTH
046.371  353            1687          XCHG
046.372  343            1688          XTHL                    (HL) = ADDRESS OF HEADER;((SP)) = STORAGE NEEDED
046.373  043            1689          INX     H
046.374  043            1690          INX     H
                        1691
046.375  163            1692          MOV     M,E
046.376  043            1693          INX     H
046.377  162            1694          MOV     M,D             SET TOTAL LENGTH
047.000  341            1695          POP     H               (HL) = LENGTH OF VALUE STORE AREA
047.001  021 064 112    1696          LXI     D,SYMTAB+1
047.004  345            1697          PUSH    H               SAVE COUNT
047.005  315 026 071    1698          CALL    AMB             ALLOCATE MEMORY
047.010  321            1699          POP     D               (DE) = COUNT
                        1700
                        1701  *       ZERO NEWLY CREATED VALUES.
                        1702
047.011  066 000        1703  DIM3    MVI     M,0
047.013  033            1704          DCX     D               ZERO ENTRYS
047.014  043            1705          INX     H
047.015  172            1706          MOV     A,D
047.016  263            1707          ORA     E
047.017  302 011 047    1708          JNZ     DIM3
                        1709
                        1710  *       DONE WITH DECLARATION, SEE IF ANOTHER FOLLOWS.
                        1711
047.022  315 056 071    1712          CALL    ANT             GET NEXT TOKEN
047.025  376 026        1713          CPI     CT.CMA
047.027  300            1714          RNE                     NOT COMMA
047.030  303 236 046    1715          JMP     DIM             PROCESS ANOTHER
                        1716
                        1717  *       ERROR OCCURED, PUT SYMBOL TABLE BACK.
                        1718
047.033  041 000 000    1719  DIM5    LXI     H,0
047.034                 1720  DIMA    EQU     *-2             PREVIOUS LENGTH
047.036  042 066 112    1721          SHLD    SYMTAB+MT.LEN
047.041  303 124 043    1722          JMP     RESTART         EXIT: RESTART RESTORES ABORT ADDRESS
```

```
                        1724  **      END - END PROGRAM.
                        1725  *
                        1726
                        1727
047.044  041 345 114   1728  END     LXI     H,MTAREA-1
047.047  042 135 112   1729          SHLD    CURADR          SET EXECUTION ADDRESS TO TOP
047.052  076 224       1730          MVI     A,BEC.EN
047.054  365           1731          PUSH    PSW             SAVE CODE
047.055  303 063 075   1732          JMP     ILM             ISSUE LINE MESSAGE


                        1734  **      FOR - PERFORM 'FOR' LOOP.
                        1735  *
                        1736  *       FOR VAR = VAL1 TO VAL2 [STEP VAL3]
                        1737  *
                        1738  *       KEPT ON 'FOR' STACK:
                        1739  *
                        1740  *       1) INDEX VARIABLE ADDRESS (2BYTES)
                        1741  *       2) STEP VALUE (4 BYTES)
                        1742  *       3) FINAL VALUE (4 BYTES)
                        1743  *       4) LOOP ADDRESS (2 BYTES)
                        1744  *
                        1745  *       IF THE 'FOR' VARIABLE IS ALREADY PRESENT IN THE 'FOR' STACK,
                        1746  *       REMOVE IT AND THEN ADD IT TO THE END.
                        1747
                        1748
047.060                 1749  FOR     EQU     *
047.060  315 362 077   1750          CALL    SFS             SEARCH 'FOR' STACK
047.063  315 000 073   1751          CALL    CSI             CONVERT TO INDEX                /80.01.GC/
047.066  325           1752          PUSH    D
047.067  302 104 047   1753          JNZ     FOR1            NONE PRE-EXISTING
047.072  053           1754          DCX     H
047.073  053           1755          DCX     H
047.074  021 014 000   1756          LXI     D,12
047.077  315 203 104   1757          CALL    $DBT            REMOVE FROM TABLE
047.102  071 112       1758          DW      FORTAB+1
                        1759
                        1760  *       ALLOCATE SPACE FOR ENTRY.
                        1761
047.104                 1762  FOR1    EQU     *
047.104  041 014 000   1763          LXI     H,12
047.107  021 071 112   1764          LXI     D,FORTAB+1
047.112  315 026 071   1765          CALL    AMB             ALLOCATE 12 BYTES
047.115  321           1766          POP     D               (DE) = FOR INDEX
047.116  315 366 072   1767          CALL    CSA             CONVERT BACK TO ABS. AFTER DEL  /80.01.GC/
                        1768
                        1769  *       STORE THE KEY ENTRY
                        1770
047.121  033           1771          DCX     D                                               /80.01.GC/
047.122  033           1772          DCX     D                                               /80.01.GC/
047.123  032           1773          LDAX    D                                               /80.01.GC/
047.124  167           1774          MOV     M,A                                             /80.01.GC/
047.125  023           1775          INX     D                                               /80.01.GC/
047.126  043           1776          INX     H                                               /80.01.GC/
```

```
047.127  032           1777          LDAX    D                                /80.01.GC/
047.130  167           1778          MOV     M,A                              /80.01.GC/
047.131  023           1779          INX     D                                /80.01.GC/
047.132  043           1780          INX     H                                /80.01.GC/
047.133  315 000 073   1781          CALL    CSI             CONVERT IT TO AN INDEX  /80.01.GC/
                       1782
047.136  076 300       1783          MVI     A,CT.SNV
047.140  315 377 050   1784          CALL    LET.            ASSIGN VALUE
047.143  315 305 077   1785          CALL    RNT
047.146  317           1786          DB      CT.TO           REQUIRE *TO*
047.147  315 022 057   1787          CALL    EVALN
047.152  043           1788          INX     H               GO PAST 'STEP' VALUE
047.153  043           1789          INX     H
047.154  043           1790          INX     H
047.155  043           1791          INX     H
047.156  315 051 076   1792          CALL    MOV4            STORE LIMIT
047.161  021 370 377   1793          LXI     D,-8
047.164  031           1794          DAD     D               (HL) = ADDRESS FOR STEP
047.165  315 056 071   1795          CALL    ANT             ACCEPT NEXT TOKEN
047.170  021 147 112   1796          LXI     D,FP1,0
047.173  376 211       1797          CPI     CT,STE
047.175  314 022 057   1798          CE      EVALN           EVALUATE STEP VALUE
047.200  315 051 076   1799          CALL    MOV4            STORE STEP
047.203  043           1800          INX     H               SKIP 'LIMIT'
047.204  043           1801          INX     H
047.205  043           1802          INX     H
047.206  043           1803          INX     H
047.207  161           1804          MOV     M,C
047.210  043           1805          INX     H
047.211  160           1806          MOV     M,B             STORE STATEMENT RETURN ADDRESS
047.212  311           1807          RET


                       1809  **      FREE - TYPE FREE SPACE.
                       1810  *
                       1811  *       FREE
                       1812
                       1813
047.213                1814  FREE    EQU     *
047.213  305           1815          PUSH    B               SAVE (BC)
047.214  041 061 112   1816          LXI     H,MTABIND+MT.LEN
047.217  345           1817          PUSH    H               SAVE TABLE INDEX ON STACK
047.220  006 021       1818          MVI     B,MTABL*2+1     (B) = NUMBER OF TABLES * 2 +1
047.222  041 272 047   1819          LXI     H,FREEA         (HL) = HEADER MESSAGES
                       1820
047.225  377 003       1821  FREE1   DB      SYSCALL,.PRINT  PRINT HEADER
047.227  315 136 031   1822          CALL    $TYPTX
047.232  040 075 240   1823          DB      ' =',' '+200Q
047.235  343           1824          XTHL                    (HL) = ADDRESS OF INDEX
047.236  136           1825          MOV     E,M
047.237  043           1826          INX     H
047.240  126           1827          MOV     D,M
047.241  043           1828          INX     H
047.242  043           1829          INX     H
```

```
047.243  043          1830          INX     H
047.244  043          1831          INX     H
047.245  343          1832          XTHL
047.246  005          1833          DCR     B
047.247  304 264 047  1834          CNZ     TDI.            TYPE VALUE IF NOT LAST ONE
047.252  005          1835          DCR     B
047.253  362 225 047  1836          JP      FREE1           MORE TO GO
047.256  341          1837          POP     H               DISCARD TABLE ADDRESS
047.257  315 127 072  1838          CALL    $CFS            COMPUTE FREE SPACE
047.262  353          1839          XCHG
047.263  301          1840          POP     B               RESTORE (BC)
                      1841
                      1842  **      TDI. - TYPE DECIMAL INTEGER FOLLOWED BY $CRLF
                      1843
047.264  315 206 100  1844  TDI.    CALL    TDI
047.267  303 312 111  1845          JMP     $CRLF
                      1846
047.272               1847  FREEA   EQU     *               TABLE OF TABLE NAMES
047.272  124 145 170  1848          DB      'Tex','t'+200Q
047.276  123 171 155  1849          DB      'Sym','b'+200Q
047.302  106 157 162  1850          DB      'For','1'+200Q
047.306  107 163 165  1851          DB      'Gsu','b'+200Q
047.312  127 157 162  1852          DB      'Wor','k'+200Q
047.316  123 164 162  1853          DB      'Str','n'+200Q
047.322  124 123 164  1854          DB      'TSt','r'+200Q
047.326  106 151 154  1855          DB      'Fil','e'+200Q
047.332  106 162 145  1856          DB      'Fre','e'+200Q


                      1858  ***     FREEZE - FREEZE PROGRAM AND BASIC.
                      1859  *
                      1860  *       FREEZE <STRING>
                      1861  *
                      1862  *       FREEZE THE BASIC PROGRAM, BASIC, AND ALL MEMORY ONTO
                      1863  *       FILE 'STRING'
                      1864
                      1865
047.336               1866  FREEZE  EQU     *
047.336  315 041 072  1867          CALL    CFN.            COPY FILE NAME, DO FILE OPEN PRESET
047.341  257          1868          XRA     A
047.342  315 005 072  1869          CALL    CFA             PRESET FOR I/O OPERATION
047.345  021 057 054  1870          LXI     D,UNFREZA       (DE) = DEFAULTS
047.350  315 030 101  1871          CALL    $FOPEW          OPEN FOR WRITE
047.353  345          1872          PUSH    H               SAVE FB ADDRESS
047.354  052 127 112  1873          LHLD    MEML
047.357  021 200 335  1874          LXI     D,-USERFWA
047.362  031          1875          DAD     D               (HL) = LENGTH
047.363  042 022 050  1876          SHLD    FREZEB
047.366  021 016 050  1877          LXI     D,FREZEA        (DE) = HEADER ADDRESS
047.371  343          1878          XTHL                    (HL) = FB ADDRESS, ((SP)) = LEN
047.372  001 010 000  1879          LXI     B,FREZEAL
047.375  315 047 102  1880          CALL    $FWRIB          WRITE HEADER
050.000  301          1881          POP     B               (BC) = LEN OF PROGRAM
050.001  021 200 042  1882          LXI     D,USERFWA
```

```
050.004  315 047 102  1883           CALL    $FWRIB          WRITE IT
050.007  315 335 102  1884           CALL    $FCLO           CLOSE FILE
050.012  001 345 114  1885           LXI     B,ZERO          NO MORE TEXT LINE
050.015  311          1886           RET                     LET HIM KEEP RUNNING
                      1887
050.016  377 000      1888  FREZEA   DB      377Q,FT.ABS     ABS HEADER FOR IMAGE
050.020  200 042      1889           DW      USERFWA
050.022  000 000      1890  FREZEB   DW      0               LENGTH
050.024  106 043      1891           DW      START           ENTRY ADDRESS
000.010               1892  FREZEAL  EQU     *-FREZEA        LENGTH OF HEADER


                      1894  **       GOSUB - CALL SUBROUTINE.
                      1895  *
                      1896  *        GOSUB EXIR
                      1897
                      1898
050.026               1899  GOSUB    EQU     *
050.026  315 143 100  1900           CALL    SRA             STACK RETURN ADDRESS
                      1901  *        JMP     GOTO            PROCESS AS GOTO


                      1903  **       GOTO - GO TO STATEMENT.
                      1904  *
                      1905  *        GOTO EXPR
                      1906
                      1907
050.031               1908  GOTO     EQU     *
050.031  315 033 074  1909           CALL    ELN             EVAL LINE NUMBER
050.034  315 242 074  1910  GOTO1    CALL    FLN             FIND LINE BY NUMBER
050.037  332 147 070  1911           JC      ERR.SN          CANT FIND IT
                      1912
050.042               1913  GOTO2    EQU     *
050.042  053          1914           DCX     H               (HL) = PREVIOUS LINE TERMINATOR
050.043  104          1915           MOV     B,H
050.044  115          1916           MOV     C,L
050.045  042 135 112  1917           SHLD    CURADR          SAVE CURRENT TEXT ADDRESS
050.050  311          1918           RET                     LET EXEC 'FIND' NEW LINE


                      1920  **       IF - PROCESS IF STATEMENT.
                      1921  *
                      1922  *        IF EXPR THEN <STATEMENT>
                      1923  *        IF EXPR THEN <STATEMENT NUMBER>
                      1924
                      1925
050.051               1926  IF       EQU     *
050.051  315 036 057  1927           CALL    EVALI           EVALUATE EXPRESSION
                      1928
                      1929  *        WILL EXECUTE, REQUIRE 'THEN'
```

```
                                1930
050.054  315 056 071           1931            CALL    ANT             GET NEXT TOKEN
050.057  376 225               1932            CPI     CT.GOT
050.061  312 127 050           1933            JE      IF3             IS IF <EXPR> GOTO <EXPR>
050.064  376 316               1934            CPI     CT.THN
050.066  302 152 070           1935            JNE     ERR.SY          NOT *THEN*
050.071  173                   1936            MOV     A,E             (A) = TEST CODE
050.072  037                   1937            RAR
050.073  322 121 050           1938            JNC     IF2             FALSE - WILL SKIP
050.076  315 126 100           1939            CALL    SOB             SKIP OVER BLANKS
050.101  012                   1940            LDAX    B
050.102  376 060               1941            CPI     '0'
050.104  332 114 050           1942            JC      IF0             NOT DIGIT - MUST BE STATEMENT
050.107  376 072               1943            CPI     '9'+1
050.111  332 031 050           1944            JC      GOTO            IS DIGIT - MUST BE GOTO
050.114  341                   1945  IF0       POP     H
050.115  303 244 043           1946            JMP     EXEC1           PROCESS AS STATEMENT
                                1947
                                1948  *         SKIP REST OF LINE.
                                1949
050.120  003                   1950  IF1       INX     B
050.121  012                   1951  IF2       LDAX    B
050.122  247                   1952            ANA     A
050.123  302 120 050           1953            JNZ     IF1             SKIP STATEMENT
050.126  311                   1954            RET                     DONE
                                1955
                                1956  *         IF <EXPR> GOTO <EXPR>
                                1957
050.127  173                   1958  IF3       MOV     A,E
050.130  037                   1959            RAR
050.131  322 121 050           1960            JNC     IF2             IF TO SKIP
050.134  303 031 050           1961            JMP     GOTO            PROCESS GOTO
```

```
                                1963  **        LINE INPUT - INPUT ONE LINE FROM CONSOLE.
                                1964  *
                                1965  *         SAME AS *INPUT*, EXCEPT THAT THE FIRST VARIABLE MUST
                                1966  *         BE A STRING VARIABLE, AND THE FIRST LINE IS
                                1967  *         TAKEN AS THE VALUE.
                                1968
                                1969
050.137                         1970  LINPUT    EQU     *
050.137  315 305 077           1971            CALL    RNT
050.142  254                   1972            DB      CT.INP          REQUIRE *INPUT*
050.143  076 001               1973            MVI     A,1
050.145  303 151 050           1974            JMP     INP1            PROCESS AS INPUT
```

```
                                1976  **      INPUT - INPUT FROM CONSOLE.
                                1977  *
                                1978  *       INPUT "PROMPT";V1,...,VN
                                1979
                                1980
050.150  257                   1981  INPUT   XRA     A
050.151  062 370 050           1982  INP1    STA     INPUTA          SAVE FLAG FOR LINE INPUT
050.154  315 253 073           1983          CALL    DCN             DECODE CHANNEL NUMBER
050.157  305                   1984          PUSH    B               SAVE (BC)(
050.160  315 354 111           1985          CALL    $CCO            CLEAR CTL-O
050.163  301                   1986          POP     B
050.164  315 072 076           1987          CALL    PNT             PEEK AT NEXT TOKE
050.167  041 371 050           1988          LXI     H,INPUTB        ASSUME '?' PROMPT
050.172  376 027               1989          CPI     CT.SEM
050.174  302 205 050           1990          JNE     INP2            MAY HAVE PROMPT
050.177  315 056 071           1991          CALL    ANT             NO PROMPT, GOBBLE ;
050.202  303 233 050           1992          JMP     INP4            PROVIDE DEFALT PROMPT
                                1993
050.205  376 301               1994  INP2    CPI     CT.SSV          SCALAR STRING VALUE
050.207  302 233 050           1995          JNE     INP4            NO PROMPT
                                1996
                                1997  *       HAVE PROMPT
                                1998
050.212  072 140 112           1999          LDA     IOCHAN
050.215  247                   2000          ANA     A
050.216  314 200 100           2001          CZ      TCS             TYPE CHARACTER STRING IFF CONSOLE INPUT
050.221  315 056 071           2002          CALL    ANT             ACCEPT ALREADY PROCESSED STRING
050.224  315 305 077           2003          CALL    RNT
050.227  027                   2004          DB      CT.SEM          REQUIRE ;
050.230  041 373 050           2005          LXI     H,INPUTC        SUPPRESS OUR PROMPT
                                2006
                                2007  *       READY TO INPUT VALUES
                                2008
050.233  072 140 112           2009  INP4    LDA     IOCHAN
050.236  247                   2010          ANA     A               SEE IF OUTPUT TO CONSOLE
050.237  302 244 050           2011          JNZ     INP4.5          DISK I/O, NO PROMPT
050.242  377 003               2012          DB      SYSCALL,.PRINT  PRINT PROMPT
050.244                        2013  INP4.5  EQU     *                                       /80.01.GC/
                                2014
                                2015  *       MAKE SURE WE HAVE VARIABLES
                                2016
050.244  315 072 076           2017          CALL    PNT                                     /80.01.GC/
050.247  376 300               2018          CPI     CT.VARL         LOWEST VARIABLE         /80.01.GC/
050.251  332 152 070           2019          JC      ERR.SY          < LOWEST VARIABLE       /80.01.GC/
050.254  376 310               2020          CPI     CT.VARH+1                               /80.01.GC/
050.256  322 152 070           2021          JNC     ERR.SY          > HIGHEST VARIABLE      /80.01.GC/
                                2022
050.261  041 266 112           2023          LXI     H,LINE+1                                /80.01.GC/
050.264  315 141 077           2024          CALL    RLF             READ LINE FROM FILE
050.267  332 106 070           2025          JC      ERR.CC          IF CTL-C HIT
050.272  072 370 050           2026          LDA     INPUTA
050.275  247                   2027          ANA     A
050.276  041 266 112           2028          LXI     H,LINE+1        ASSUME START LINE AT FIRST CHARACTER
050.301  312 356 050           2029          JZ      INP6            IS REGULAR INPUT
                                2030
                                2031  *       IS LINE INPUT, ENCLOSE LINE IN QUOTES
```

```
                                2032
050.304  315 072 076  2033  INP5    CALL    PNT             CHECK INPUT VARIABLE
050.307  346 375      2034          ANI     377Q-CF.VEC     ALLOW VECTORS, TOO
050.311  376 301      2035          CPI     CT.SSV
050.313  302 152 070  2036          JNE     ERR.SY          MUST BE SCALAR STRING VALUE
050.316  345          2037          PUSH    H               SAVE DATA POINTER
050.317  315 136 075  2038          CALL    IST             INSERT SYMBOL IN TABLE
050.322  341          2039          POP     H               (HL) = DATA POINTER
050.323  325          2040          PUSH    D               SAVE TARGET VARIABLE INDEX
050.324  305          2041          PUSH    B               SAVE TEXT POINTER
050.325  104          2042          MOV     B,H
050.326  115          2043          MOV     C,L             (BC) = INPUT TEXT ADDRESS
050.327  315 012 055  2044          CALL    LEX11.5         BUILD INTO STRING
050.332  001 005 000  2045          LXI     B,5
050.335  033          2046          DCX     D
050.336  041 201 042  2047          LXI     H,ACCX-1
050.341  315 252 030  2048          CALL    $MOVE           MOVE TEMP DESCRIPTOR INTO ACCX
050.344  301          2049          POP     B
050.345  321          2050          POP     D
050.346  315 366 072  2051          CALL    CSA             CONVERT INDEX TO ABSOLUTE
050.351  076 301      2052          MVI     A,CT.SSV        IS STRING ASSIGNMENT
050.353  303 202 071  2053          JMP     AVV             ASSIGN VALUE TO VARIABLE, EXIT 'INPUT' PROCESSING
                                2054
                                2055  *         ASSIGN VALUES
                                2056
050.356  315 135 076  2057  INP6    CALL    PVI             PERFORM VALUE INPUT
050.361  310          2058          RE                      DONE
050.362  041 371 050  2059          LXI     H,INPUTB        USE '?' PROMPT
050.365  303 233 050  2060          JMP     INP4            INPUT MORE
                                2061
050.370  000          2062  INPUTA  DB      0               <>0 IF LINE INPUT
050.371  077 240      2063  INPUTB  DB      '?',' '+200Q    DEFAULT PROMPT
050.373  200          2064  INPUTC  DB      200Q            NULL PROMPT


                                2066  **        LET - ASSIGN VALUE.
                                2067  *
                                2068  *         LET VAL = EXPR
                                2069
                                2070
050.374  315 136 075  2071  LET     CALL    IST             PREPARE VALUE FOR ASSIGNMENT
                                2072
050.377  365          2073  LET.    PUSH    PSW             SAVE TYPE
                                2074
051.000  325          2075          PUSH    D               SAVE INDEX
051.001  315 305 077  2076          CALL    RNT
051.004  011          2077          DB      CT.EQ           REQUIRE =
051.005  315 244 055  2078          CALL    EVAL            (ACCX) = VALUE
051.010  321          2079          POP     D               (DE) = VALUE INDEX
051.011  315 366 072  2080          CALL    CSA             (DE) = ABSOLUTE ADDRESS INTO SYMTAB
051.014  361          2081          POP     PSW             (A) = TYPE
051.015  303 202 071  2082          JMP     AVV             ASSIGN VALUE TO VARIABLE
```

```
                                2084  **      LIST - PROCESS LIST COMMAND.
                                2085  *
                                2086  *       LIST                    LIST ALL
                                2087  *       LIST NNN                LIST NNN
                                2088  *       LIST NNN,MMM            LIST NNN TO MMM
                                2089  *
                                2090  *       LIST [#CHAN][,NNNN[,MMMM]] ETC.                 /78.10.GC/
                                2091
051.020                         2092  LIST    EQU     *
                                2093
051.020  315 253 073            2094          CALL    DCN             DECODE CHANNEL NUMBER   /78.10.GC/
                                2095
                                2096  *       DECODE RANGE.
                                2097
051.023  021 000 000            2098  LIST.   LXI     D,0
051.026  325                    2099          PUSH    D               SET DEFAULT NN
051.027  033                    2100          DCX     D
051.030  033                    2101          DCX     D               (DE) = 377376A
051.031  315 072 076            2102          CALL    PNT             PEEK AT NEXT TOKEN
000.000                         2103          ERRNZ   CT.FIN
051.034  247                    2104          ANA     A
051.035  312 065 051            2105          JZ      LIST1           IS LIST 0,377376A
051.040  315 036 057            2106          CALL    EVALI           (DE) = NNN
051.043  341                    2107          POP     H               DISARED DEFAULT FIRST
051.044  325                    2108          PUSH    D               SET 1ST = LAST = NNN
051.045  315 072 076            2109          CALL    PNT             PEEK AT NEXT TOKEN
000.000                         2110          ERRNZ   CT.FIN
051.050  247                    2111          ANA     A
051.051  312 065 051            2112          JZ      LIST1           IS NNN
051.054  315 223 072            2113          CALL    CMA             REQUIRE ','
051.057  315 036 057            2114          CALL    EVALI           IS NNN,MMM
051.062  315 242 074            2115          CALL    FLN             CHECK VALIDITY OF LAST LINE NUMBER  /78.10.GC/
                                2116
                                2117  *       LIST TEXT
                                2118
051.065  341                    2119  LIST1   POP     H               (HL) = START
051.066  325                    2120          PUSH    D               SAVE END
051.067  353                    2121          XCHG                    (DE) = 1ST, ((SP)) = LAST
051.070  315 242 074            2122          CALL    FLN             FIND LINE BY NUMBER
                                2123
                                2124  *       SEE IF OFF THE END
                                2125
051.073  116                    2126  LIST2   MOV     C,M
051.074  043                    2127          INX     H
051.075  106                    2128          MOV     B,M             (BC) = LINE NUMBER OF NEXT LINE
051.076  043                    2129          INX     H
051.077  343                    2130          XTHL                    (HL) = LIMIT
051.100  175                    2131          MOV     A,L
051.101  221                    2132          SUB     C               COMPARE TO CURRENT
051.102  174                    2133          MOV     A,H
051.103  230                    2134          SBB     B
051.104  332 170 051            2135          JC      LIST6           ALL DONE
051.107  343                    2136          XTHL                    RESTORE LIMIT
051.110  345                    2137          PUSH    H               SAVE LINE ADDRESS
051.111  041 273 113            2138          LXI     H,LINE2
051.114  076 005                2139          MVI     A,5
```

```
051.116  315 157 031  2140          CALL    $UDD            UNPACK DECIMAL DIGITS
051.121  066 040      2141          MVI     M,' '           ADD BLANK
051.123  043          2142          INX     H
051.124  353          2143          XCHG                    (DE) = LINE ADDRESS
051.125  341          2144          POP     H               (HL) = PROGRAM TEXT ADDRESS
051.126  176          2145  LIST3   MOV     A,M             (A) = NEXT CHARACTER
051.127  043          2146          INX     H
051.130  247          2147          ANA     A
051.131  374 374 073  2148          CM      EKA             EXPAND KEYWORD TO ASCII
051.134  022          2149          STAX    D               STORE IN LISTING LINE
051.135  023          2150          INX     D
051.136  247          2151          ANA     A
051.137  302 126 051  2152          JNZ     LIST3           MORE TO GO
                      2153
                      2154  *       SEE IF TO WRITE TO FILE, OR TO CONSOLE
                      2155
051.142  345          2156          PUSH    H               SAVE PROGRAM TEXT ADDRESS
051.143  353          2157          XCHG                    (HL) = LINE NEXT ADDRESS
051.144  053          2158          DCX     H               BACKUP OVER END OF LINE
051.145  066 012      2159          MVI     M,NL
051.147  043          2160          INX     H
051.150  066 000      2161          MVI     M,0             ADD END OF LINE
051.152  315 242 100  2162          CALL    WLF             WRITE LINE TO FILE
051.155  341          2163          POP     H               (HL) = TEXT FWA
051.156  072 142 112  2164          LDA     CTLFLAG
000.000               2165          ERRNZ   CFCTLC-1
051.161  037          2166          RAR
051.162  332 106 070  2167          JC      ERR.CC          CTL-C STRUCK
051.165  303 073 051  2168          JMP     LIST2           DO NEXT
                      2169
                      2170  *       ALL DONE.
                      2171
051.170  341          2172  LIST6   POP     H
051.171  001 345 114  2173          LXI     B,ZERO          END OF COMMAND LINE
051.174  311          2174          RET


                      2176  **      LOCK - LOCK OUT DATA CHANGE
                      2177  *
                      2178  *       LOCK PREVENTS ANY DATA OR LINES OF TEXT TO BE
                      2179  *       CHANGED
                      2180  *
                      2181
                      2182  **      UNLOCK - ENABLE DATA CHANGE
                      2183  *
                      2184  *       UNLOCK CLEARS THE LOCK FLAG ENABLING
                      2185  *       DATA CHANGES
                      2186  *
                      2187  *       UNLOCK
                      2188  *       LOCK
                      2189
                      2190
                      2191  ********************************************************************
                      2192  *                                                                  *
```

```
                                2193  *       LOCK USES THE XRA  A  OPCODE (257) AS THE VALUE TO PUT IN      *
                                2194  *       LCKFLG THROUGH THE USE OF THE MVI INSTRUCTION IMPLEMENTED      *
                                2195  *       TO USE THE XRA  A  OPCODE AS THE SECOND BYTE OF THE MVI        *
                                2196  *       INSTRUCTION.                                                   *
                                2197  *                                                                      *
                                2198  ************************************************************************
                                2199
                                2200
                                2201
051.175  076                   2202  LOCK    DB      MI.MVIA         MVI OPCODE
                                2203
051.176  257                   2204  UNLOCK  XRA     A
051.177  062 137 112            2205          STA     LCKFLG          STORE EITHER 0 OR 257 (FROM XRA OPCODE)
051.202  311                   2206          RET                     EXIT



                                2208  **      NEXT - PROCESS NEXT.
                                2209  *
                                2210  *       NEXT VAR
                                2211  *
                                2212  *       PERFORM LOOPING FOR REQUESTED VARIABLE. IF NOT THE MOST
                                2213  *       RECENT, DISCARD 'FORTAB' ENTRYS UNTIL IS FOUND.
                                2214
                                2215
051.203                         2216  NEXT    EQU     *
051.203  315 352 077            2217          CALL    SFS.            SEARCH 'FOR' STACK
051.206  302 133 070            2218          JNZ     ERR.NV          NEXT MISSING VARIABLE
051.211  345                   2219          PUSH    H               SAVE FORTAB INDEX
                                2220  **      CALL    CSA             (DE) = ABS. ADDR. OF VARIABLE   /80.01.GC/
051.212  315 210 073            2221          CALL    CVX             COPY VALUE TO ACCX
051.215  315 000 073            2222          CALL    CSI             (DE) = INDEX INTO SYMTAB OF VARIBLE
051.220  341                   2223          POP     H               (HL) = FORTAB INDEX
051.221  325                   2224          PUSH    D               SAVE INDEX ADDRESS
051.222  353                   2225          XCHG
051.223  041 012 000            2226          LXI     H,12-2
051.226  031                   2227          DAD     D               (HL) = NEW TABLE LENGTH
051.227  042 073 112            2228          SHLD    FORTAB+MT.LEN   DISCARD ANY MORE INNER ENTRYS
051.232  052 071 112            2229          LHLD    FORTAB+MT.FWA
051.235  031                   2230          DAD     D               (HL) = TABLE ADDRESS
051.236  353                   2231          XCHG
051.237  315 352 104            2232          CALL    FPADD           ADD STEP TO INDEX
051.242  353                   2233          XCHG                    (HL) = ADDRESS OF STEP VALUE
051.243  321                   2234          POP     D               (DE) = ADDRESS OF INDEX
051.244  315 366 072            2235          CALL    CSA             (DE) = ABS. ADDR. OF VARIBLE
051.247  315 237 073            2236          CALL    CXV             COPY ACCX TO VALUE
                                2237
                                2238  *       COMPARE RESULT TO LIMIT.
                                2239  *
                                2240  *       IF INC >= 0, VAL-LIM>0 => TERMINATE
                                2241  *       IF INC < 0, LIM-VAL>0 => TERMINATE
                                2242
051.252  043                   2243          INX     H
051.253  043                   2244          INX     H
051.254  176                   2245          MOV     A,M             (A) = SIGN OF INCREMENT
```

```
051.255  043            2246          INX     H
051.256  043            2247          INX     H
051.257  345            2248          PUSH    H               SAVE ADDRESS OF LIMIT
051.260  247            2249          ANA     A
051.261  353            2250          XCHG                    (DE) = ADDRESS OF LIMIT
051.262  372 271 051     2251          JM      NXT1            IS < 0
                        2252
                        2253  *       COMPUTE VALUE-LIMIT
                        2254
051.265  315 210 073     2255          CALL    CVX             (ACCX) = LIMIT
051.270  353            2256          XCHG                    (DE) = ADDRESS OF VALUE
                        2257
                        2258  *       COMPUTE LIMIT-VALUE
                        2259
051.271  315 166 105     2260  NXT1    CALL    FPSUB           COMPARE
051.274  341            2261          POP     H               (HL) = ADDRESS OF LIMIT
051.275  072 204 042     2262          LDA     ACCX+2
051.300  247            2263          ANA     A
051.301  312 307 051     2264          JZ      NXT1.5          IS MATCH
051.304  362 317 051     2265          JP      NXT2            ALL DONE
                        2266
                        2267  *       LOOP TO AFTER 'FOR' STATEMENT
                        2268
051.307  043            2269  NXT1.5  INX     H
051.310  043            2270          INX     H
051.311  043            2271          INX     H
051.312  043            2272          INX     H
051.313  116            2273          MOV     C,M
051.314  043            2274          INX     H
051.315  106            2275          MOV     B,M
051.316  311            2276          RET
                        2277
                        2278  *       DONE. COLLAPSE 'FOR' OUT OF TABLE.
                        2279
051.317  052 073 112     2280  NXT2    LHLD    FORTAB+MT.LEN
051.322  021 364 377     2281          LXI     D,-12
051.325  031            2282          DAD     D
051.326  042 073 112     2283          SHLD    FORTAB+MT.LEN
051.331  311            2284          RET
```

```
                        2286  ***     OLD - GET NEW PROGRAM.
                        2287  *
                        2288  *       OLD     <STRING>
                        2289  *
                        2290  *       OLD CLEARS ALL THE TABLE, THEN LOADS A PROGRAM.
                        2291
                        2292
051.332                 2293  OLD     EQU     *
051.332  315 053 072     2294          CALL    CFN             COPY FILE NAME
051.335  041 141 112     2295          LXI     H,OVLMAN
051.340  176            2296          MOV     A,M             (A) = CURRENT VALUE
051.341  064            2297          INR     M               MAKE NON-ZERO
051.342  345            2298          PUSH    H
```

```
051.343  365          2299          PUSH    PSW
051.344  315 360 044  2300          CALL    SCR.            CLEAR PROGRAM
051.347  361          2301          POP     PSW
051.350  341          2302          POP     H
051.351  167          2303          MOV     M,A             RESTORE FLAG
051.352  303 206 077  2304          JMP     RNP             READ NEW PROGRAM AND RETURN


                      2306  **      ON - PROCESS 'ON' STATEMENT.
                      2307  *
                      2308  *       ON EXPR GOTO EXP1,...,EXPN
                      2309  *       ON EXPR GOSUB EXP1,...,EXPN
                      2310  *
                      2311  *       IF EXPR < 0, FLAG ERROR
                      2312  *       IF EXPR = 1,...,N TAKE EXP1,...,EXPN
                      2313  *       IF EXPR>N TAKE EXPN
                      2314  *       IF EXPR=0 TAKE EXPN
                      2315
                      2316
051.355  315 036 057  2317  ON      CALL    EVALI           EVALUATE INTEGER
051.360  353          2318          XCHG                    (HL) = INDEX
051.361  315 056 071  2319          CALL    ANT             ACCEPT NEXT TOKEN
051.364  376 225      2320          CPI     CT.GOT
051.366  312 001 052  2321          JE      ON1             GOTO
051.371  376 224      2322          CPI     CT.GOS
051.373  302 152 070  2323          JNE     ERR.SY          NOT GOSUB
051.376  315 143 100  2324          CALL    SRA             SET RETURN ADDRESS
052.001               2325  ON1     EQU     *
                      2326
                      2327  *       SKIP DOWN LIST UNTIL INDEX FOUND.
                      2328
052.001  315 033 074  2329  ON2     CALL    ELN             EVALUATE LINE NUMBER
052.004  315 056 071  2330          CALL    ANT             GET DELIMITER
052.007  062 035 052  2331          STA     ONA             SAVE FOR LATER EXAM
052.012  053          2332          DCX     H
052.013  174          2333          MOV     A,H
052.014  265          2334          ORA     L
052.015  312 034 050  2335          JZ      GOTO1           HAVE PROPER LABEL
052.020  072 035 052  2336          LDA     ONA
052.023  247          2337          ANA     A
000.000               2338          ERRNZ   CT.FIN
052.024  310          2339          RZ                      END OF LINE
052.025  376 026      2340          CPI     CT.CMA
052.027  302 152 070  2341          JNE     ERR.SY
052.032  303 001 052  2342          JMP     ON2
                      2343
052.035  000          2344  ONA     DB      0               TEMP AREA
```

```
                        2346  ***     OPEN - OPEN FILE
                        2347  *
                        2348  *       OPEN <STRING> FOR <VERB> AS FILE #N
                        2349  *
                        2350  *       <STRING> = STRING CONTAINING FILE NAME
                        2351  *
                        2352  *       <VERB> = READ           OPEN FILE FOR READ ACCESS
                        2353  *       <VERB> = WRITE          OPEN FILE FOR WRITE ACCESS
                        2354
                        2355
052.036                 2356  OPEN    EQU     *
052.036  315 053 072    2357          CALL    CFN             CRACK FILE NAME
052.041  315 305 077    2358          CALL    RNT
052.044  221            2359          DB      CT.FOR          REQUIRE FOR
052.045  315 056 071    2360          CALL    ANT
052.050  376 241        2361          CPI     CT.REA
052.052  312 062 052    2362          JE      OPEN1           VERB IS OK
052.055  376 313        2363          CPI     CT.WRI
052.057  302 152 070    2364          JNE     ERR.SY          NOT A GOOD VERB
052.062  365            2365  OPEN1   PUSH    PSW             SAVE VERB TOKEN
052.063  315 305 077    2366          CALL    RNT
052.066  311            2367          DB      CT.AS
052.067  315 305 077    2368          CALL    RNT
052.072  312            2369          DB      CT.FIL          FILE
052.073  315 273 073    2370          CALL    DCN,            DECODE CHANNEL NUMBER, NO COMMA
052.076  315 302 075    2371          CALL    LCC             LOCATE CHANNEL COLUMN NUMBER
052.101  066 001        2372          MVI     M,1             SET AT FRONT OF LINE
052.103  361            2373          POP     PSW             (A) = FUNCTION KEYWORD
052.104  305            2374          PUSH    B               SAVE TEXT POINTER
052.105  365            2375          PUSH    PSW             SAVE FUNCTION KEYWORD
052.106  072 140 112    2376          LDA     IOCHAN
052.111  075            2377          DCR     A               (A) = CHANNEL NUMBER
                        2378
                        2379  *       FIND THE FILE BLOCK, CREATE IT IF NECESSARY
                        2380
052.112  365            2381  OPEN2   PUSH    PSW             SAVE CHANNEL/BLOCK NUMBER
052.113  315 005 072    2382          CALL    CFA             COMPUTE FILEBLOCK ADDRESS
052.116  322 144 052    2383          JNC     OPEN3           GOTIT
                        2384
052.121  315 374 071    2385          CALL    CEF             CREATE EMPTY FILE BLOCK
052.124  072 125 112    2386          LDA     FILTAB+MT.LEN+1 A  = BUFFER JUST ADDED          /80.01.GC/
052.127  315 005 072    2387          CALL    CFA             HL = FILE-BLOCK ADDRESS         /80.01.GC/
052.132  332 160 070    2388          JC      ERR.TO          SHOULD NOT HAPPEN ! (NOT FOUND) /80.01.GC/
052.135  043            2389          INX     H                                               /80.01.GC/
000.000                 2390          ERRNZ   FB.FLG-1                                       /80.01.GC/
052.136  066 000        2391          MVI     M,0             ZERO THE FLAG                   /80.01.GC/
052.140  361            2392          POP     PSW
052.141  303 112 052    2393          JMP     OPEN2           SEE IF WE'VE CREATED ENOUGH
                        2394
                        2395  *       GOT THE FILE BLOCK,
                        2396  *       (HL) = FB FWA (ABS)
                        2397
052.144  043            2398  OPEN3   INX     H               (HL) = #FB.FLG
000.000                 2399          ERRNZ   FB.FLG-1
052.145  176            2400          MOV     A,M                                             /80.01.GC/
052.146  247            2401          ANA     A                                               /80.01.GC/
```

```
052.147  302 221 070  2402           JNZ     ERR.CIU         CHANNEL ALREADY IN USE          /80.01.GC/
                      2403
052.152  001 011 000  2404           LXI     B,FB.NAM-FB.FLG
052.155  011         2405           DAD     B               (HL) = ADDRESS FOR NAME IN FILE BLOCK
052.156  021 242 042  2406           LXI     D,FBLIST+FB.NAM (DE) = ADDRESS OF NAME IN SYSTEM FILE BLOCK
377.012              2407           ERRPL   FB.NAM-256      CODE ASSUMES 1 BYTE VALUE
377.021              2408           ERRPL   FB.NAML-256     CODE ASSUMES 1 BYTE VALUE
052.161  016 021     2409           MVI     C,FB.NAML       (BC) = #FB.NAML
052.163  315 252 030  2410           CALL    $MOVE           MOVE NAME TO PROPER BLOCK
052.166  315 217 074  2411           CALL    FOP             FILE OPEN PRESET
052.171  361         2412           POP     PSW             (A) = CHANNEL NUMBER
052.172  315 005 072  2413           CALL    CFA             COMPUTE FILE BLOCK ADDRESS
052.175  361         2414           POP     PSW             (A) = CT.REA OR CT.WRI
052.176  021 100 043  2415           LXI     D,DEFALTD       USE DATA DEFAULTS
052.201  315 210 052  2416           CALL    OPEN4           CALL OPEN ROUTINE
052.204  301         2417           POP     B               RESTORE TEXT POINTER
052.205  303 115 074  2418           JMP     FOC             FILE OPEN CLEANUP AND EXIT
                      2419
052.210  376 241     2420  OPEN4    CPI     CT.REA
052.212  312 021 101  2421           JE      $FOPER
052.215  303 030 101  2422           JMP     $FOPEW          OPEN FOR READ OR WRITE


                      2424  **       OUT - OUTPUT TO PORT.
                      2425  *
                      2426  *        OUT PORT,VALUE
                      2427
                      2428
052.220  315 235 052  2429  OUT      CALL    OUT1            EVALUATE PORT AND VALUE
052.223  145         2430           MOV     H,L             (H) = PORT
052.224  056 323     2431           MVI     L,MI.OUT
052.226  042 002 040  2432           SHLD    .IOWRK          SET VALUE
052.231  173         2433           MOV     A,E             (A) = VALUE
052.232  303 002 040  2434           JMP     .IOWRK          OUTPUT AND RETURN
                      2435
                      2436
                      2437  **       OUT1 - EVALUATE ADDRESS,VALUE
                      2438
052.235  315 036 057  2439  OUT1     CALL    EVAL1
052.240  325         2440           PUSH    D               SAVE ADDRESS
052.241  315 223 072  2441           CALL    CMA             REQUIRE ','
052.244  315 036 057  2442           CALL    EVAL1           (E) = VALUE
052.247  341         2443           POP     H               (HL) = ADDRESS
052.250  311         2444           RET
```

```
                                2446  **      PAUSE - PAUSE FOR TIME INTERVAL.
                                2447  *
                                2448  *       PAUSE <IEXP>
                                2449  *
                                2450  *       PAUSE FOR <IEXP>*2 MILLISECONDS. IF NO TIME IS
                                2451  *       SPECIFIED, PAUSE UNTIL A KEY IS STRUCK.
                                2452  *
                                2453  *       METHOD OF CALCULATION: (IF IEXP GIVEN)
                                2454  *
                                2455  *       AT EXAMINE TIME:
                                2456  *
                                2457  *       IF TARGET => TICCNT
                                2458  *                          THEN
                                2459  *                                IF TAR. - TIC <> 0
                                2460  *                                OR
                                2461  *                                IF TAR. - TIC. < 377 000A  TIME UP
                                2462  *                                ELSE WAIT
                                2463  *
                                2464  *       IF TARGET < TICCNT
                                2465  *                             THEN
                                2466  *                                IF TIC. - TAR. < 000 377A , TIME UP
                                2467  *                                ELSE WAIT
                                2468
                                2469
052.251                         2470  PAUSE   EQU     *
052.251  315 072 076            2471          CALL    PNT             CHECK NEXT TOKEN
000.000                         2472          ERRNZ   CT.FIN
052.254  247                    2473          ANA     A
052.255  312 233 103            2474          JZ      $RCHAR          NO PARAMETERS, JUST WAIT
                                2475
052.260  315 036 057            2476          CALL    EVALI           DECODE PAUSE INTERVAL
052.263  172                    2477          MOV     A,D             (A) = HIGH ORDER BYTE OF IEXP
052.264  074                    2478          INR     A
052.265  312 122 070            2479          JZ      ERR.IN          NUMBER TOO LARGE
052.270  052 033 040            2480          LHLD    .TICCNT
052.273  031                    2481          DAD     D               (HL) = TICCNT FINAL VALUE
052.274  353                    2482          XCHG
052.275  052 033 040            2483  PAUSE1  LHLD    .TICCNT         (HL) = TIC COUNTER
052.300  315 224 030            2484          CALL    $CHL            INVERT IT
052.303  031                    2485          DAD     D               TAR. - TIC.
052.304  332 320 052            2486          JC      PAUSE2          TAR. - TIC. => 0
                                2487
                                2488  *       TAR. < TIC.
                                2489
052.307  315 224 030            2490          CALL    $CHL            (HL) = TIC. - TAR.
052.312  174                    2491          MOV     A,H
052.313  247                    2492          ANA     A
052.314  310                    2493          RZ                      DONE
052.315  303 326 052            2494          JMP     PAUSE3          WAIT
                                2495
                                2496  *       TAR. => TIC.
                                2497
052.320  174                    2498  PAUSE2  MOV     A,H             CHECK FOR TAR. = TIC.
052.321  265                    2499          ORA     L
052.322  310                    2500          RZ                      DONE
                                2501
```

```
052.323 174          2502          MOV   A,H
052,324 074          2503          INR   A
052.325 310          2504          RZ                    DONE
                     2505
052.326 072 142 112  2506  PAUSE3   LDA   CTLFLAG
052,331 247          2507          ANA   A               SEE IF ANY CTL CHARACTERS HIT
052.332 300          2508          RNZ                   CONTROL CHARACTER HIT
052.333 303 275 052  2509          JMP   PAUSE1          CONTINUE WAITING


                     2511  **      POKE - WRITE VALUE INTO MEMORY.
                     2512  *
                     2513  *       POKE ADDR,VALUE
                     2514
                     2515
052,336              2516  POKE    EQU   *
052.336 315 235 052  2517          CALL  OUT1            READ ADDRESS AND VALUE
052,341 163          2518          MOV   M,E             SET VALUE
052.342 311          2519          RET


                     2521  ***     POSITION - SET FILE POSITION.
                     2522  *
                     2523  *       POSITION #N,IEXP
                     2524  *
                     2525  *       POSITION FILE #N AT BLOCK IEXP. FILE MUST BE OPEN FOR READ.
                     2526
                     2527


                     2529  **      PRINT - PROCESS PRINT STATEMENT.
                     2530  *
                     2531  *       PRINT VARLIST
                     2532  *
                     2533  *       IF VARIABLE SEPERATOR IS ',', TAB TO NEXT FIELD.
                     2534  *       IF SEPERATOR IS ';', DONT TAB.
                     2535  *       IF THE LAST TOKEN IN THE STATEMNET IS ',' OR ';', DONT
                     2536  *       CRLF AFTER LINE.
                     2537
                     2538
052.343              2539  PRINT   EQU   *
052,343 257          2540          XRA   A
052.344 062 144 053  2541          STA   PRIA            CLEAR '; OR ,' FLAG
052,347 315 253 073  2542          CALL  DCN             DECODE CHANNEL NUMBER
052,352 041 352 052  2543  PRI1    LXI   H,PRI1
052,355 345          2544          PUSH  H               SET 'RETURN ADDRESS'
052.356 315 072 076  2545          CALL  PNT             PREVIEW NEXT TOKEN
000.000              2546          ERRNZ CT.FIN
052.361 247          2547          ANA   A
052.362 312 142 053  2548          JZ    PRI7            END OF STATEMENT
```

```
052.365  062 144 053  2549          STA     PRIA            SAVE TYPE
052.370  376 346      2550          CPI     CT.TAB
052.372  312 105 053  2551          JE      PRI6            TAB FUNCTION
052.375  376 343      2552          CPI     CT.SPC
052.377  312 105 053  2553          JE      PRI6            SPC FUNCTION
053.002  376 027      2554          CPI     CT.SEM
053.004  312 056 071  2555          JE      ANT             ACCEPT ; AND GO TO PRI1
053.007  376 026      2556          CPI     CT.CMA
053.011  312 040 053  2557          JE      PRI3
                      2558
                      2559  *       MUST BE EXPRESSION.
                      2560
053.014  315 244 055  2561          CALL    EVAL            EVALUTE EXPRESSION
053.017  033          2562          DCX     D
053.020  032          2563          LDAX    D               (A) = TYPE
053.021  023          2564          INX     D
053.022  346 001      2565          ANI     CF.STR
053.024  302 200 100  2566          JNZ     TCS             IS STRING: TYPE CHARACTER STRING
                      2567
                      2568  *       HAVE NUMERIC VALUE.
                      2569
053.027  041 273 113  2570  PRI2    LXI     H,LINE2         USE SCRATCH AREA
053.032  315 237 110  2571          CALL    FTA             CONVERT FLOATING TO ASCII
053.035  303 251 100  2572          JMP     WLF.            WRITE LINE TO FILE AND RETURN TO PRI1
                      2573
                      2574  *       HAVE COMMA - SKIP TO NEXT FIELD
                      2575
053.040  315 056 071  2576  PRI3    CALL    ANT             ACCEPT ,
053.043  315 302 075  2577          CALL    LCC             LOCATE CHANNEL COLUMN COUNTER
053.046  176          2578          MOV     A,M             (A) = COLUMN COUNTER
053.047  376 072      2579          CPI     58
053.050               2580  PRIB    EQU     *-1             TAB LIMITS
053.051  322 225 100  2581          JNC     WEL             OVERFLOW - A NEW LINE
                      2582
                      2583  *       COMPUTE REQUIRED SPACES
                      2584
053.054  305          2585          PUSH    B                                               /80.01.GC/
                      2586
053.055  117          2587          MOV     C,A                                             /80.01.GC/
053.056  006 000      2588          MVI     B,0             BC = COLUMN COUNTER             /80.01.GC/
053.060  013          2589          DCX     B               ON RANGE [0,N]                  /80.01.GC/
053.061  021 016 000  2590          LXI     D,14            DE = FIELD SIZE                 /80.01.GC/
053.062               2591  PRIC    EQU     *-2                                             /80.01.GC/
053.064  315 106 030  2592          CALL    $DU66           DE = REMAINDER                  /80.01.GC/
053.067  301          2593          POP     B                                               /80.01.GC/
053.070  072 062 053  2594          LDA     PRIC            A  = FIELD SIZE                 /80.01.GC/
053.073  223          2595          SUB     E               A  = NUMBER OF SPACES REQUIRED  /80.01.GC/
                      2596
053.074  247          2597  PRI5    ANA     A
053.075  310          2598          RZ                      NO MORE SPACES                  /80.01.GC/
053.076  315 156 053  2599          CALL    PRI8            OUTPUT A SPACE                  /80.01.GC/
053.101  075          2600          DCR     A
053.102  303 074 053  2601          JMP     PRI5
                      2602
                      2603  *       HAVE TAB OR SPC FUNCTION
                      2604
```

```
053.105  315 056 071  2605  PRI6    CALL    ANT         ACCEPT TAB OR SPC
053.110  365         2606          PUSH    PSW         SAVE FUNCTION TYPE
053.111  315 044 057  2607          CALL    EVALI8      EVALUTE COUNT
053.114  315 305 077  2608          CALL    RNT
053.117  020         2609          DB      CT.PAR      REQUIRE ')'
053.120  361         2610          POP     PSW
053.121  376 343     2611          CPI     CT.SPC
053.123  173         2612          MOV     A,E         (A) = COUNT IF SPACE
053.124  312 074 053  2613          JE      PRI5        IS SPC                          /80.01.GC/
053.127  315 302 075  2614          CALL    LCC         LOCATE CHANNEL COLUMN COUNTER
053.132  176         2615          MOV     A,M         (A) = COLUMN
053.133  223         2616          SUB     E
053.134  057         2617          CMA
053.135  074         2618          INR     A                                           /78.10.GC/
053.136  362 074 053  2619          JP      PRI5        NOT PAST IT
053.141  311         2620          RET                 ALREADY PAST - DO NOTHING
                     2621
                     2622  *       HAVE END OF LINE
                     2623
053.142  341         2624  PRI7    POP     H           DISCARD 'RETURN' ADDRESS
053.143  076 000     2625          MVI     A,0
053.144              2626  PRIA    EQU     *-1
053.145  376 026     2627          CPI     CT.CMA      CHECK TEYP OF LAST TOKEN
053.147  310         2628          RE                  COMMA
053.150  376 027     2629          CPI     CT.SEM
053.152  310         2630          RE                  ;
053.153  303 225 100  2631          JMP     WEL         END LINE
                     2632
                     2633  *       OUTPUT A SPACE                                  /80.01.GC/
                     2634
053.156  365         2635  PRI8    PUSH    PSW                                     /80.01.GC/
053.157  041 207 112  2636          LXI     H,SPACE                                 /80.01.GC/
053.162  076 001     2637          MVI     A,1         COUNT = 1                   /80.01.GC/
053.164  315 251 100  2638          CALL    WLF.        WRITE CHARACTER TO THE FILE /80.01.GC/
053.167  361         2639          POP     PSW
053.170  311         2640          RET
```

```
                     2642  **      READ - READ FROM DATA STATEMENT.
                     2643  *
                     2644  *       READ PERFORMS READS FROM DATA STATEMENTS.
                     2645  *
                     2646  *       THE 1ST DATA STATEMENT IS FOUND AND USED, THEN THE 2ND,
                     2647  *       ETC.
                     2648
                     2649
053.171              2650  READ    EQU     *
053.171  052 303 114  2651          LHLD    DATPTR      (HL) = DATA STATEMENT POINTER
053.174  315 135 076  2652  REA1    CALL    PVI         PERFORM VALUE INPUT
053.177  042 303 114  2653          SHLD    DATPTR      SAVE FOR NEXT TIME
053.202  310         2654          RE                  NO MORE DATA NEEDED
                     2655
                     2656  *       SCAN FOR NEXT DATA STATEMENT
                     2657
```

```
053.203  176          2658  REA2    MOV     A,M
053.204  043          2659          INX     H
053.205  247          2660          ANA     A
053.206  302 203 053  2661          JNZ     REA2            NOT AT END OF STATEMENT
053.211  176          2662          MOV     A,M
053.212  043          2663          INX     H
053.213  246          2664          ANA     M
053.214  043          2665          INX     H
053.215  074          2666          INR     A
053.216  312 114 070  2667          JZ      ERR.DE          DATA EXHAUSED AT LINE 377377A
053.221  176          2668          MOV     A,M
053.222  376 251      2669          CPI     CT.DAT
053.224  302 203 053  2670          JNE     REA2            NOT DATA
053.227  043          2671          INX     H
053.230  303 174 053  2672          JMP     REA1            READ NEW DATA STATEMENT


                      2674  **      REPLACE - SAVE PROGRAM OVERTOP ANY EXISTING PROGRAM.
                      2675  *
                      2676  *       REPLACE <STRING>
                      2677  *
                      2678  *       SAME AS SAVE, BUT DOESNT SQUAK IF ALREADY EXISTS.
                      2679
                      2680
053.233               2681  REPLACE EQU     *
053.233  315 041 072  2682          CALL    CFN.            COPY FILE NAME AND FILE OPEN PRESET
053.236  305          2683          PUSH    B               SAVE (BC)
053.237  303 324 053  2684          JMP     SAVE1           SAVE IT


                      2686  **      RETURN - RETURN FROM GOSUB.
                      2687  *
                      2688
                      2689
053.242  052 076 112  2690  RETURN  LHLD    GOSTAB+MT.FWA
053.245  353          2691          XCHG                    (DE) = FWA
053.246  052 100 112  2692          LHLD    GOSTAB+MT.LEN
053.251  174          2693          MOV     A,H
053.252  265          2694          ORA     L
053.253  312 141 070  2695          JZ      ERR.RE          RETURN ERROR
053.256  053          2696          DCX     H
053.257  053          2697          DCX     H
053.260  053          2698          DCX     H
053.261  053          2699          DCX     H
053.262  042 100 112  2700          SHLD    GOSTAB+MT.LEN   SET REDUCED SIZE
053.265  031          2701          DAD     D               (HL) = ABS ADDRESS OF ENTRY
053.266  116          2702          MOV     C,M
053.267  043          2703          INX     H
053.270  106          2704          MOV     B,M             (BC) = RETURN ADDRESS
053.271  043          2705          INX     H
053.272  176          2706          MOV     A,M
053.273  043          2707          INX     H
```

```
053.274  146          2708          MOV     H,M
053.275  157          2709          MOV     L,A
053.276  042 133 112  2710          SHLD    CURNUM          (HL) = OLD LINE NUMBER
053.301  311          2711          RET
```

```
                      2713  ***     SAVE - SAVE PROGRAM ON DISK.
                      2714  *
                      2715  *       SAVE <FNAME>
                      2716  *
                      2717  *       WILL COMPLAIN IF FILE ALREADY EXISTS.
                      2718
                      2719
053.302               2720  SAVE    EQU     *
053.302  315 041 072  2721          CALL    CFN.            COPY FILE NAME AND FILE OPEN PRESET
053.305  305          2722          PUSH    B               SAVE (BC)
053.306  021 072 043  2723          LXI     D,DEFALTP       PROGRAM DEFAULT
053.311  315 046 101  2724          CALL    $FOPER.         OPEN FILE
053.314  322 177 070  2725          JNC     ERR.FAE         FILE ALREADY EXISTS
053.317  376 014      2726          CPI     EC.FNF
053.321  302 223 070  2727          JNE     $FERROR         NON-EXPECTED ERROR
                      2728
                      2729  *       ENTERED HERE FROM 'REPLACE' TO SAVE FILE REGARDLESS
                      2730
053.324  021 072 043  2731  SAVE1   LXI     D,DEFALTP       (DE) = DEFAULTS FOR SAVE
053.327  315 030 101  2732          CALL    $FOPEW          OPEN FOR WRITE, THEN
053.332  301          2733          POP     B               RESTORE TEXT POINTER
                      2734
                      2735  *       FILE IS OPEN, LIST PROGRAM TO IT
                      2736
053.333  076 001      2737          MVI     A,1
053.335  062 140 112  2738          STA     IOCHAN          SET I/O CHANNEL
053.340  345          2739          PUSH    H               SAVE ADDRESS OF BUFFER
053.341  315 023 051  2740          CALL    LIST.           LIST TO FILE                    /78.10.GC/
053.344  341          2741          POP     H
053.345  315 335 102  2742          CALL    $FCLO           CLOSE IT
053.350  001 345 114  2743          LXI     B,ZERO
053.353  303 115 074  2744          JMP     FOC             FILE OPEN CLEANUP, AND EXIT
```

```
                      2746  **      STEP - PERFORM SINGLE STEPPING.
                      2747  *
                      2748  *       STEP I
                      2749  *
                      2750
                      2751
053.356  315 072 076  2752  STEP    CALL    PNT             PREVIEW NEXT TOKEN
053.361  127          2753          MOV     D,A
053.362  137          2754          MOV     E,A
000.000               2755          ERRNZ   CT.FIN
053.363  247          2756          ANA     A
053.364  304 036 057  2757          CNZ     EVALI           EVALUATE COUNT
```

```
053.367  315 305 077  2758          CALL    RNT             FLUSH TOKEN PIPELINE
053.372  000          2759          DB      CT.FIN
053.373  033          2760          DCX     D
                      2761
053.374  325          2762  STEP1   PUSH    D               SAVE COUNT
053.375  076 001      2763          MVI     A,RM.STE
053.377  315 165 045  2764          CALL    CONT1           STEP 1
054.002  321          2765          POP     D
054.003  033          2766          DCX     D
054.004  172          2767          MOV     A,D
054.005  247          2768          ANA     A
054.006  362 374 053  2769          JP      STEP1           MORE TO GO
054.011  315 136 031  2770          CALL    $TYPTX
054.014  116 145 170  2771          DB      'Next','='+200Q
054.021  052 133 112  2772          LHLD    CURNUM
054.024  353          2773          XCHG
054.025  303 264 047  2774          JMP     TDI.            TYPE AS DECIMAL INTEGER


                      2776  **      STOP - STOP EXECUTION.
                      2777  *
                      2778
                      2779
054.030  315 201 044  2780  STOP    CALL    EXEC7           STORE BC
054.033  076 225      2781          MVI     A,BEC.ST
054.035  365          2782          PUSH    PSW
054.036  303 063 075  2783          JMP     ILM             ISSUE LINE MESSAGE


                      2785  ***     UNFREEZE - UNFREEZE FROZEN PROGRAM.
                      2786  *
                      2787  *       UNFREEZE <FNAME>
                      2788  *
                      2789  *       SAME AS "RUN SY0:FNAME.BAF"
                      2790
                      2791
054.041               2792  UNFREZ  EQU     *
054.041  315 313 075  2793          CALL    LFC             CHECK FOR DATA LOCK
054.044  315 103 054  2794          CALL    UNSAVE1         PRESET
054.047  021 057 054  2795          LXI     D,UNFREZA
054.052  377 040      2796          DB      SYSCALL,.LINK   LINK IT
054.054  303 223 070  2797          JMP     $FERROR         GOT PROBLEMS
                      2798
054.057  123 131 060  2799  UNFREZA DB      'SY0BAF'        DEFAULT BLOCK
```

```
                                2801  ***     UNSAVE - DELETE PROGRAM.
                                2802  *
                                2803  *       UNSAVE <FNAME>
                                2804
                                2805
                                2806
054.065                         2807  UNSAVE  EQU     *
054.065  315 103 054            2808          CALL    UNSAVE1         PRESET
054.070  021 072 043            2809          LXI     D,DEFALTP       PROGRAM DEFAULTS
054.073  305                    2810          PUSH    B
054.074  377 050                2811          DB      SYSCALL,.DELET  DELETE IT
054.076  301                    2812          POP     B               (BC) = TEXT POINTER
054.077  320                    2813          RNC                     NO ERROR
054.100  303 223 070            2814          JMP     $FERROR         FLAG ERROR
                                2815
                                2816
                                2817  **      GET READY FOR OPERATION
                                2818
054.103  315 053 072            2819  UNSAVE1 CALL    CFN             CRACK FILE NAME
054.106  021 012 000            2820          LXI     D,FB.NAM
054.111  031                    2821          DAD     D
054.112  353                    2822          XCHG
054.113  041 273 113            2823          LXI     H,LINE2
054.116  305                    2824          PUSH    B
054.117  001 021 000            2825          LXI     B,FB.NAML
054.122  345                    2826          PUSH    H               SAVE $FOPWRK
054.123  315 252 030            2827          CALL    $MOVE           MOVE IN NAME
054.126  341                    2828          POP     H
054.127  301                    2829          POP     B
054.130  311                    2830          RET
```

```
                                2834  **      LEXCAL - PERFORM LEXICAL ANALYSIS.
                                2835  *
                                2836  *       LEXCAL PARSES THE NEXT TOKEN FROM THE SOURCE LINE.
                                2837  *
                                2838  *       IF THE VARIABLE HAS NOT BEEN DEFINED, A SPECIAL ADDRESS,
                                2839  *       *LEXC* IS RETURNED. THIS ADDRESS CONTAINS A 0 OR A NULL STRING,
                                2840  *       DEPENDING UPON THE VARIABLE TYPE.
                                2841  *
                                2842  *       ENTRY   (BC) = SOURCE TEXT POINTER
                                2843  *       EXIT    (A) = TYPE (CT. CODE)
                                2844  *               (DE) = SYMTAB ENTRY ADDRESS+2 (IF SYMBOL)
                                2845  *               'C' SET IF VARIABLE AND NOT DEFINED
                                2846  *                (DE) = LEXC
                                2847  *                LEXA = VARIABLE NAM
                                2848  *       USES    A,F,B,C,D,E
                                2849
                                2850
054.131                         2851  LEXCAL  EQU     *
054.131  315 126 100            2852          CALL    SOB             SKIP OVER BLANKS
054.134  315 230 072            2853          CALL    CNC             CLASSIFY NEXT CHARACTER
000.000                         2854          ERRNZ   CT.FIN
054.137  247                   2855          ANA     A
054.140  310                   2856          RZ                      IS CT.FIN
054.141  003                   2857          INX     B               ACCEPT CHARACTER
054.142  370                   2858          RM                      IS KEYWORD
054.143  376 030               2859          CPI     CT.QUO
054.145  312 015 055           2860          JE      LEX12           HAVE STRING
054.150  376 014               2861          CPI     CT.LT
054.152  314 354 054           2862          CE      LEX10           IS <
054.155  376 012               2863          CPI     CT.GT
054.157  314 377 054           2864          CE      LEX11           HAVE >
054.162  376 001               2865          CPI     CT.ALP
054.164  312 231 054           2866          JE      LEX1            IS ALPHABETIC
054.167  376 002               2867          CPI     CT.NUM
054.171  312 202 054           2868          JE      LEX0            IS NUMERIC VALUE
054.174  376 003               2869          CPI     CT.SEP
054.176  300                   2870          RNE                     IS SOME KNOWN CHARACTER
054.177  303 174 070           2871          JMP     ERR.IC          ILLEGAL CHARACTER
                                2872
                                2873  *       IS NUMERIC VALUE. FLOAT IT.
                                2874
054.202  013                   2875  LEX0    DCX     B               (BC) = ADDRESS OF FIRST DIGIT
054.203  353                   2876          XCHG                    SAVE (HL) IN (DE)
054.204  140                   2877          MOV     H,B
054.205  151                   2878          MOV     L,C
054.206  315 323 107           2879          CALL    ATF             ASCII TO FLOATING
054.211  104                   2880          MOV     B,H
054.212  115                   2881          MOV     C,L
054.213  353                   2882          XCHG                    RESTORE (HL)
054.214  021 222 042           2883          LXI     D,LEXB
054.217  315 237 073           2884          CALL    CXV             COPY NUMBER INTO 'LEXB' HOLD AREA
054.222  076 300               2885          MVI     A,300Q          SET TYPE
054.224  247                   2886          ANA     A               CLEAR CARRY
054.225  062 221 042           2887          STA     LEXB-1          SET TYPE
054.230  311                   2888          RET
                                2889
```

```
                        2890  *       IS ALPHABETIC, MUST BE VARIABLE.
                        2891
054.231                 2892  LEX1    EQU     *
054.231  013           2893          DCX     B               POINT TO 1ST CHAR OF NAME
054.232  012           2894          LDAX    B               (A) = 1ST CHARACTER OF NAME
054.233  315 045 112   2895          CALL    $MCU            MAP CHARACTER TO UPPER CASE
054.236  127           2896          MOV     D,A
054.237  036 000       2897          MVI     E,0             (DE) = CARIABLE NAME
054.241  003           2898          INX     B
054.242  012           2899          LDAX    B
054.243  326 060       2900          SUI     '0'
054.245  332 264 054   2901          JC      LEX2            NOT NUMBER
054.250  376 012       2902          CPI     9+1
054.252  322 264 054   2903          JNC     LEX2            NOT NUMBER
054.255  074           2904          INR     A               DIFFERENTIATE BETWEEN X AND X0
054.256  007           2905          RLC
054.257  007           2906          RLC
054.260  007           2907          RLC
054.261  007           2908          RLC
054.262  137           2909          MOV     E,A             (E) = NUMBER INDEX
054.263  003           2910          INX     B               ADVANCE PAST NUMBER
                        2911
                        2912  *       HAVE VARIABLE NAME IN (DE), CHECK FOR $ AND (
                        2913
054.264  012           2914  LEX2    LDAX    B
054.265  376 044       2915          CPI     '$'
054.267  302 274 054   2916          JNE     LEX3            NOT $
054.272  003           2917          INX     B
000.000                 2918          ERRNZ   CF.STR-1
054.273  034           2919          INR     E               SET CF.STR
054.274  315 126 100   2920  LEX3    CALL    SOB             SKIP OVER BLANKS
054.277  012           2921          LDAX    B
054.300  376 050       2922          CPI     '('
054.302  302 312 054   2923          JNE     LEX3.5
054.305  003           2924          INX     B
054.306  076 002       2925          MVI     A,CF.VEC        SET VECTOR TYPE
054.310  263           2926          ORA     E               SET VECTOR TYPE
054.311  137           2927          MOV     E,A
                        2928
                        2929  *       (DE) = VARIABLE NAME AND TYPE, FIND IN SYMTAB
                        2930
054.312  345           2931  LEX3.5  PUSH    H                                       /80.01.GC/
054.313  325           2932          PUSH    D                                       /80.01.GC/
054.314  315 323 075   2933          CALL    LVS             DE = SYMTAB ADDRESS     /80.01.GC/
054.317  341           2934          POP     H               HL = SAVED TYPE         /80.01.GC/
054.320  023           2935          INX     D                                       /80.01.GC/
054.321  302 334 054   2936          JNZ     LEX7                                    /80.01.GC/
                        2937
                        2938  *       HAVE FOUND MATCH.
                        2939
054.324  032           2940          LDAX    D                                       /80.01.GC/
054.325  023           2941          INX     D                                       /80.01.GC/
054.326  346 007       2942          ANI     7               (A) = TYPE CODE
054.330  366 300       2943          ORI     300Q
054.332  341           2944          POP     H               RESTORE (HL)
054.333  311           2945          RET
```

```
                                2946
                                2947  *       ITEM NOT IN TABLE.
                                2948  *
                                2949  *       RETURN NULL OR ZERO VALUE.
                                2950
054.334  042 236 075            2951  LEX7    SHLD    LEXA            SAVE
054.337  175                    2952          MOV     A,L             (A) = FLAG FIELD OF NAME
054.340  021 214 042            2953          LXI     D,LEXC-1        (DE) = RESULT POINTER-1
054.343  346 007                2954          ANI     7               STRIP FLAGS
054.345  366 300                2955          ORI     300Q            SET VARIABLE TYPE
054.347  022                    2956          STAX    D               SET TYPE
054.350  023                    2957          INX     D               SET RESULT POINTER
054.351  341                    2958          POP     H               RESTORE (HL)
054.352  067                    2959          STC                     FLAG UNDEFINED
054.353  311                    2960          RET
                                2961
                                2962  *       HAVE <, SEE IF <= OR <>
                                2963
054.354  012                    2964  LEX10   LDAX    B
054.355  003                    2965          INX     B               ASSUME IS <= OR <>
054.356  376 075                2966          CPI     '='
054.360  076 015                2967          MVI     A,CT.LE         ASSUME <=
054.362  310                    2968          RE
054.363  013                    2969          DCX     B               GET TESTING CHARACTER
054.364  012                    2970          LDAX    B
054.365  003                    2971          INX     B               RESTORE (BC)
054.366  376 076                2972          CPI     '>'
054.370  076 016                2973          MVI     A,CT.NE         ASSUME <>
054.372  310                    2974          RE                      IS <>
054.373  076 014                2975          MVI     A,CT.LT         IS JUST <
054.375  013                    2976          DCX     B
054.376  311                    2977          RET
                                2978
                                2979  *       HAVE >, SEE IF >=
                                2980
054.377  012                    2981  LEX11   LDAX    B
055.000  003                    2982          INX     B               ASSUME IS >=
055.001  376 075                2983          CPI     '='
055.003  076 013                2984          MVI     A,CT.GE
055.005  310                    2985          RE                      IS <=
055.006  076 012                2986          MVI     A,CT.GT         IS >
055.010  013                    2987          DCX     B
055.011  311                    2988          RET
                                2989
                                2990
                                2991  **      LEX12 - PUT TEXT STRING INTO STRINGTABLE AS TEMP STRING.
                                2992  *
                                2993  *       ENTRY   (BC) = ADDRESS OF 1ST CHARACTER
                                2994  *       EXIT    LEXB = STRING HEADER
                                2995  *               (DE) = #LEXB
                                2996  *       USES    A,F,B,C,D,E
                                2997
055.012  076 000                2998  LEX11.5 MVI     A,0
                                2999
055.014  021                    3000          DB      MI.LXID         USE 'LXI,D' TO GOBBLE NEXT MVI
055.015  076 042                3001  LEX12   MVI     A,'"'
```

```
                                3002
055.017  062 033 055            3003            STA     LEXD            SET END OF STRING MATCH CHARACTER
055.022  345                    3004            PUSH    H               SAVE (HL)
055.023  305                    3005            PUSH    B               SAVE TEXT POINTER
055.024  041 377 377            3006            LXI     H,-1            (HL) = CHARACTER COUNTER
055.027  012                    3007  LEX13     LDAX    B
055.030  003                    3008            INX     B
055.031  043                    3009            INX     H
055.032  376 042                3010            CPI     '"'
055.033                         3011  LEXD      EQU     *-1             END OF STRING MATCH
055.034  312 044 055            3012            JE      LEX14           GOT END QUOTE
055.037  247                    3013            ANA     A
055.040  302 027 055            3014            JNZ     LEX13           NOT AT END OF LINE
055.043  053                    3015            DCX     H               RAN OFF END OF LINE, DONT COUNT 00 BYTE
                                3016
055.044  174                    3017  LEX14     MOV     A,H
055.045  247                    3018            ANA     A
055.046  302 144 070            3019            JNZ     ERR.SL          STRING LENGTH ERROR
055.051  345                    3020            PUSH    H               SAVE LENGTH IN (L)
055.052  042 222 042            3021            SHLD    LEXB            SET IN DESCRIPTOR
055.055  021 222 042            3022            LXI     D,LEXB
055.060  315 033 073            3023            CALL    CSE.            CREATE TEMP STRING TAB ENTRY
055.063  301                    3024            POP     B               (BC) = COUNT
055.064  321                    3025            POP     D               (DE) = FROM ADDRESS
055.065  315 252 030            3026            CALL    *MOVE           COPY STRING INTO TEMP
055.070  102                    3027            MOV     B,D
055.071  113                    3028            MOV     C,E
055.072  003                    3029            INX     B
055.073  076 301                3030            MVI     A,CT.SSV        SCALAR STRING VALUE
055.075  021 221 042            3031            LXI     D,LEXB-1
055.100  022                    3032            STAX    D               SET TYPE
055.101  023                    3033            INX     D               (DE) = DESCRIPTOR POINTER
055.102  341                    3034            POP     H               RESTORE (HL)
055.103  311                    3035            RET
```

```
                                3039  **      VARIAB - DECODE VARIABLE.
                                3040  *
                                3041  *       VARIAB IS CALLED TO EVALUATE A VARIABLE SPECIFICATION.
                                3042  *       VARIAB RESOLVES SUBSCRIPTS.
                                3043  *
                                3044  *       ENTRY   (BC) = TEXT POINTER
                                3045  *       EXIT    (BC) UPDATED
                                3046  *               (DE) = VARIABLE POINTER
                                3047  *       USES    A,F,B,C,D,E
                                3048
                                3049
      055.104                   3050  VARIAB  EQU     *
      055.104  315 056 071      3051          CALL    ANT             ACCEPT NEXT TOKEN
      055.107  365              3052  VARIAB. PUSH    PSW             SAVE TYPE
      055.110  346 002          3053          ANI     CF.VEC
      055.112  302 117 055      3054          JNZ     VAR2            IS VECTOR
      055.115  361              3055          POP     PSW
      055.116  311              3056          RET                     IS SIMPLE VARIABLE
                                3057
                                3058  *       HAVE SUBSCRIPT.
                                3059
      055.117  345              3060  VAR2    PUSH    H
      055.120  032              3061          LDAX    D               (A) = DIMENSION COUNT
      055.121  247              3062          ANA     A
      055.122  372 152 070      3063          JM      ERR.SY          IS FUNCTION
      055.125  312 171 070      3064          JZ      ERR.ND          NOT DECLARED
                                3065
                                3066  *       EVALUATE SUBSCRIPT.
                                3067
      055.130  353              3068          XCHG                    (HL) = SUBSCRIPT LIST-2
      055.131  043              3069          INX     H
      055.132  043              3070          INX     H
      055.133  021 000 000      3071          LXI     D,0             (DE) = INDEX
                                3072
      055.136  365              3073  VAR4    PUSH    PSW
      055.137  043              3074          INX     H
      055.140  043              3075          INX     H               POINT TO NEXT SUBSCRIPT LIMIT
      055.141  345              3076          PUSH    H               SAVE VECTAB POINTER
      055.142  305              3077          PUSH    B               SAVE (BC)
      055.143  116              3078          MOV     C,M
      055.144  043              3079          INX     H
      055.145  106              3080          MOV     B,M             (BC) = LIMIT
      055.146  305              3081          PUSH    B               SAVE LIMIT
      055.147  315 337 030      3082          CALL    $MU66           (HL) = NEW INDEX
      055.152  321              3083          POP     D               (DE) = NEW LIMIT
      055.153  301              3084          POP     B               (BC) = TEXT POINTER
      055.154  345              3085          PUSH    H               SAVE INDEX
      055.155  325              3086          PUSH    D               SAVE LIM
      055.156  315 036 057      3087          CALL    EVALI           EVALUATE SUBSCRIPT
      055.161  341              3088          POP     H               (HL) = LIM
      055.162  053              3089          DCX     H
      055.163  175              3090          MOV     A,L
      055.164  223              3091          SUB     E
      055.165  174              3092          MOV     A,H
      055.166  232              3093          SBB     D
      055.167  332 163 070      3094          JC      ERR.SR          SUBSCRIPT RANGE
```

```
055,172  341          3095          POP     H               (HL) = INDEX
055,173  031          3096          DAD     D               (HL) = NEW INDEX
055,174  353          3097          XCHG
055,175  341          3098          POP     H               (HL) = SYMTAB POINTER
055,176  361          3099          POP     PSW
055,177  075          3100          DCR     A
055,200  312 220 055  3101          JZ      VARS            NO MORE SUBSCRIPTS
                      3102
                      3103  *       EXPECT ,
                      3104
055,203  365          3105          PUSH    PSW
055,204  315 056 071  3106          CALL    ANT             ACCEPT NEXT TOKEN
055,207  376 026      3107          CPI     CT.CMA
055,211  302 166 070  3108          JNE     ERR.SC          NOT ENOUGH SUBSCRIPTS
055,214  361          3109          POP     PSW             (A) = REMAINING SUBSCRIPT COUNT
055,215  303 136 055  3110          JMP     VAR4            READ NEXT
                      3111
                      3112  *       EXPECT )
                      3113
055,220  315 056 071  3114  VARS    CALL    ANT             ACCEPT NEXT TOKEN
055,223  376 020      3115          CPI     CT.PAR
055,225  302 166 070  3116          JNE     ERR.SC          TOO MANY SUBSCRIPTS
                      3117
                      3118  *       SUBSCRIPT EVALUATED, (DE) = INDEX
                      3119
055,230  043          3120          INX     H
055,231  043          3121          INX     H
055,232  353          3122          XCHG
055,233  051          3123          DAD     H
055,234  051          3124          DAD     H
055,235  031          3125          DAD     D
055,236  353          3126          XCHG                    (DE) = ADDRESS OF ENTRY IN SYMTAB
055,237  341          3127          POP     H
055,240  361          3128          POP     PSW
055,241  346 375      3129          ANI     377Q-CF.VEC     CLEAR VECTOR TYPE
055,243  311          3130          RET
```

```
                                3133  **      EVAL - EVALUATES AN EXPRESSION.
                                3134  *
                                3135  *       EVAL EVALUATES EXPRESSIONS MADE UP OF OPERATORS AND
                                3136  *       SYMBOLS.
                                3137  *
                                3138  *       VALID OPERATORS ARE (IN ORDER OF PRECIDENCE)
                                3139  *
                                3140  *       - NOT   (UNARY MINUS, NOT)
                                3141  *       ^                       (EXPONENTIATION)
                                3142  *       *,/
                                3143  *       +,-
                                3144  *       < <= = <> >= >
                                3145  *       AND
                                3146  *       OR
                                3147  *
                                3148  *       EVAL PROCESSES EXPRESSIONS UNTIL AN INAPPROPRIATE TOKEN IS
                                3149  *       ENCOUNTERED.
                                3150  *
                                3151  *
                                3152  *       ENTRY   (BC) = TEXT POINTER
                                3153  *       EXIT    (BC) UPDATED
                                3154  *               (DE) = VALUE POINTER
                                3155  *       USES    A,F,B,C,D,E
                                3156
                                3157
 055.244                        3158  EVAL    EQU     *
 055.244  345                   3159          PUSH    H               SAVE (HL)
 055.245  315 255 055           3160          CALL    LEV1
 055.250  341                   3161          POP     H               RESTORE (HL)
 055.251  021 202 042           3162          LXI     D,ACCX          (DE) = RESULT ADDRESS
 055.254  311                   3163          RET


                                3165
                                3166  **      LEV1 - OR
                                3167
 055.255  315 304 055           3168  LEV1    CALL    LEV2
 055.260  376 315               3169  LEV11   CPI     CT.OR
 055.262  300                   3170          RNE                     NOT 'OR'
 055.263  315 030 077           3171          CALL    PSHX.           ACCEPT '-' AND SAVE ACCX
 055.266  315 304 055           3172          CALL    LEV2
 055.271  315 365 076           3173          CALL    POPY
 055.274  365                   3174          PUSH    PSW             SAVE TYPE
 055.275  315 323 061           3175          CALL    P.OR            PREFORM 'OR'
 055.300  361                   3176          POP     PSW
 055.301  303 260 055           3177          JMP     LEV11
                                3178
                                3179  *       LEV2 - AND
                                3180
 055.304  315 333 055           3181  LEV2    CALL    LEV3
 055.307  376 310               3182  LEV21   CPI     CT.AND
 055.311  300                   3183          RNE
 055.312  315 030 077           3184          CALL    PSHX.           ACCEPT 'AND' AND SAVE ACCX
 055.315  315 333 055           3185          CALL    LEV3
 055.320  315 365 076           3186          CALL    POPY
```

```
055.323  365            3187          PUSH    PSW
055.324  315 336 061    3188          CALL    P.AND           PERFORM AND
055.327  361            3189          POP     PSW
055.330  303 307 055    3190          JMP     LEV21
055.333                 3191  LEV3    EQU     *               NOT USED
                        3192
                        3193  *       LEV4 - COMPARE OPERATORS.
                        3194
055.333  315 367 055    3195  LEV4    CALL    LEV5
055.336  376 011        3196  LEV41   CPI     CT.EQ
055.340  330            3197          RC                      NOT COMPARE
055.341  376 017        3198          CPI     CT.NE+1
055.343  320            3199          RNC                     NOT COMPARE
055.344  365            3200          PUSH    PSW             SAVE TYPE
055.345  315 030 077    3201          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
055.350  315 367 055    3202          CALL    LEV5
055.353  315 365 076    3203          CALL    POPY
055.356  341            3204          POP     H               (H) = COMPARE TYPE
055.357  365            3205          PUSH    PSW             SAVE NEXT TYPE
055.360  315 375 061    3206          CALL    P.CMP           DO BOOLEAN
055.363  361            3207          POP     PSW
055.364  303 336 055    3208          JMP     LEV41
                        3209
                        3210  *       LEV5 - +, -
                        3211
055.367  315 025 056    3212  LEV5    CALL    LEV6
055.372  376 021        3213  LEV51   CPI     CT.PL
055.374  312 002 056    3214          JE      LEV52           IS +
055.377  376 022        3215          CPI     CT.MI
056.001  300            3216          RNE                     NOT + OR -
056.002  365            3217  LEV52   PUSH    PSW             SAVE TYPE
056.003  315 030 077    3218          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
056.006  315 025 056    3219          CALL    LEV6
056.011  315 365 076    3220          CALL    POPY
056.014  341            3221          POP     H               (H) = TYPE
056.015  365            3222          PUSH    PSW
056.016  315 134 062    3223          CALL    P.ADD
056.021  361            3224          POP     PSW
056.022  303 372 055    3225          JMP     LEV31
                        3226
                        3227  *       LEV6 - * /
                        3228
056.025  315 063 056    3229  LEV6    CALL    LEV7
056.030  376 023        3230  LEV61   CPI     CT.MU
056.032  312 040 056    3231          JE      LEV62           IS *
056.035  376 024        3232          CPI     CT.DI
056.037  300            3233          RNE                     NOT * /
056.040  365            3234  LEV62   PUSH    PSW
056.041  315 030 077    3235          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
056.044  315 063 056    3236          CALL    LEV7
056.047  315 365 076    3237          CALL    POPY
056.052  341            3238          POP     H               (HL) = TYPE
056.053  365            3239          PUSH    PSW
056.054  315 247 062    3240          CALL    P.MUL
056.057  361            3241          POP     PSW
056.060  303 030 056    3242          JMP     LEV61
```

```
                                3243
                                3244  *       LEV7 - ^
                                3245
056.063  315 112 056            3246  LEV7    CALL    LEV8
056.066  376 025                3247  LEV71   CPI     CT.EX
056.070  300                    3248          RNE                     NOT EXPONENTIAL
056.071  315 030 077            3249          CALL    PSHX.           ACCEPT ^ AND SAVE ACCX
056.074  315 112 056            3250          CALL    LEV8
056.077  315 365 076            3251          CALL    POPY
056.102  365                    3252          PUSH    PSW
056.103  315 270 062            3253          CALL    P.EXP
056.106  361                    3254          POP     PSW
056.107  303 066 056            3255          JMP     LEV71
                                3256
                                3257  *       LEV8 - UNARY  - NOT
                                3258
056.112  315 072 076            3259  LEV8    CALL    PNT             PEEK AT NEXT TOKEN
056.115  376 022                3260          CPI     CT.MI
056.117  312 127 056            3261          JE      LEV81           IS MINUS
056.122  376 314                3262          CPI     CT.NOT
056.124  302 170 056            3263          JNE     LEV9            NOT - OR NOT
056.127  315 056 071            3264  LEV81   CALL    ANT             READ '-' OR 'NOT'
056.132  365                    3265          PUSH    PSW             SAVE TYPE
056.133  315 170 056            3266          CALL    LEV9            PROCESS OPERAND
056.136  341                    3267          POP     H               (H) = TYPE
056.137  365                    3268          PUSH    PSW             SAVE NEXT TOKEN CODE
056.140  072 201 042            3269          LDA     ACCX-1
056.143  346 001                3270          ANI     CF.STR
056.145  302 155 070            3271          JNZ     ERR.TC          MUST BE NUMERIC
056.150  174                    3272          MOV     A,H
056.151  376 022                3273          CPI     CT.MI
056.153  302 163 056            3274          JNE     LEV82           IS NOT
                                3275
                                3276  *       IS -
                                3277
056.156  315 302 105            3278          CALL    FPNEG
056.161  361                    3279          POP     PSW             (A) = CODE FOR NEXT TOKEN
056.162  311                    3280          RET
                                3281
                                3282  *       IS NOT
                                3283
056.163  315 351 061            3284  LEV82   CALL    P.NOT
056.166  361                    3285          POP     PSW
056.167  311                    3286          RET
                                3287
                                3288  *       LEV9 - TOKEN
                                3289
056.170  315 072 076            3290  LEV9    CALL    PNT             PREVIEW NEXT TOKEN
056.173  376 300                3291          CPI     CT.VARL
056.175  332 234 056            3292          JC      LEV92           NOT VARIABLE
056.200  376 310                3293          CPI     CT.VARH+1
056.202  322 234 056            3294          JNC     LEV92           NOT VARIABLE
                                3295
                                3296  *       IS VARIABLE.
                                3297
056.205  315 104 055            3298          CALL    VARIAB          DECODE
```

```
056.210  041 201 042  3299          LXI     H,ACCX-1
056.213  167          3300          MOV     M,A
056.214  043          3301          INX     H
056.215  305          3302          PUSH    B               SAVE (BC)
056.216  006 004      3303          MVI     B,4             (B) = LOOP COUNT
056.220  032          3304  LEV95   LDAX    D
056.221  167          3305          MOV     M,A             COPY VALUE INTO ACCX
056.222  023          3306          INX     D
056.223  043          3307          INX     H
056.224  005          3308          DCR     B
056.225  302 220 056  3309          JNZ     LEV95
056.230  301          3310          POP     B               RESTORE (BC)
056.231  303 072 076  3311          JMP     PNT             PREVIEW NEXT TOKEN AND EXIT
                      3312
056.234  315 056 071  3313  LEV92   CALL    ANT             ACCEPT TOKEN
056.237  376 017      3314          CPI     CT.PAL
056.241  302 256 056  3315          JNE     LEV93           NOT (
056.244  315 244 055  3316          CALL    EVAL            IS PARENTHESISED EXPRESSION
                      3317
                      3318  *       FUNCTION COMPLETE, REQUIRE ')'
                      3319
056.247  315 305 077  3320  LEV94   CALL    RNT
056.252  020          3321          DB      CT.PAR          REQUIRE ')'
056.253  303 072 076  3322          JMP     PNT             READ NEXT TOKEN AND EXIT
                      3323
056.256  376 220      3324  LEV93   CPI     CT.FN
056.260  312 340 062  3325          JE      TXTFN           TEXT FUNCTION
                      3326
                      3327  *       IS NOT SIMPLE STRING OR VALUE. MUST BE FUNCTION
                      3328
056.263  326 320      3329          SUI     CT.FCN
056.265  332 152 070  3330          JC      ERR.SY          NOT FUNCTION
056.270  365          3331          PUSH    PSW
056.271  315 244 055  3332          CALL    EVAL            EVALUATE INNARDS
056.274  361          3333          POP     PSW
056.275  365          3334          PUSH    PSW             (A) = FUNCTION INDEX
056.276  376 030      3335          CPI     CT.SRA-CT.FCN
056.300  072 201 042  3336          LDA     ACCX-1          (A) = PARAMETER TYPE
056.303  332 307 056  3337          JC      LEV90           REQUIRE NUMERIC ARGUMENT
056.306  057          3338          CMA
056.307  346 001      3339  LEV90   ANI     CF.STR
056.311  302 155 070  3340          JNZ     ERR.TC          TYPE CONFLICT
056.314  361          3341          POP     PSW             (A) = FUNCTION CODE
056.315  041 247 056  3342          LXI     H,LEV94
056.320  345          3343          PUSH    H               SAVE 'LEV94' AS RETURN
                      3344
                      3345  *       IS SYSTEM FUNCTION
                      3346
056.321  315 061 031  3347          CALL    $TJMP           ENTER PROCESSOR
056.324  055 057      3348          DW      ABS
056.326  026 065      3349          DW      ATN
056.330  103 057      3350          DW      CHR$
056.332  140 057      3351          DW      CIN
056.334  125 064      3352          DW      COS
056.336  075 063      3353          DW      EXP
056.340  216 057      3354          DW      INT
```

```
056.342  064 057        3355            DW      LNO
056.344  225 063        3356            DW      LOG
056.346  317 060        3357            DW      MAX
056.350  320 060        3358            DW      MIN
056.352  006 061        3359            DW      PAD
056.354  014 061        3360            DW      PEEK
056.356  034 061        3361            DW      PIN
056.360  053 061        3362            DW      POS
056.362  074 061        3363            DW      RND
056.364  170 061        3364            DW      SEG
056.366  205 061        3365            DW      SGN
056.370  117 064        3366            DW      SIN
056.372  152 070        3367            DW      ERR.SY          SPC
056.374  360 063        3368            DW      SQR
056.376  231 061        3369            DW      STR$
057.000  152 070        3370            DW      ERR.SY          TAB
057.002  243 064        3371            DW      TAN
                        3372
                        3373  *         THESE FUNCTIONS REQUIRE STRING ARGUMENTS.
                        3374
057.004  065 057        3375            DW      ASC
057.006  314 057        3376            DW      LEFT$
057.010  306 057        3377            DW      LEN
057.012  111 060        3378            DW      MATCH
057.014  314 057        3379            DW      MID$
057.016  314 057        3380            DW      RIGHT$
057.020  270 061        3381            DW      VAL


                        3383  **        EVALN - EVALUATE NUMERIC EXPRESSION.
                        3384  *
                        3385  *         ENTRY   SAME AS EVAL.
                        3386  *         EXIT    SAME AS EVAL
                        3387  *         USES    A,F,B,C,D,E
                        3388
                        3389
057.022  315 244 055    3390  EVALN     CALL    EVAL
057.025  072 201 042    3391            LDA     ACCX-1
057.030  346 001        3392            ANI     CF.STR
057.032  302 155 070    3393            JNZ     ERR.TC          TYPE CONFLICT
057.035  311            3394            RET                     OK


                        3396  **        EVALI - EVALUATE INTEGER EXPRESSION.
                        3397  *
                        3398  *         ENTRY   SAME AS EVAL
                        3399  *         EXIT    (DE) = INTEGER VALUE
                        3400  *                 (BC) UPDATED
                        3401  *         USES    A,F,B,C,D,E
                        3402
                        3403
057.036  315 022 057    3404  EVALI     CALL    EVALN
```

```
057.041  303 002 075  3405          JMP     IFIX            FIX IT


                      3407  **      EVALI8 - EVALUATE 8 BIT INTEGER EXPRESSION.
                      3408  *
                      3409  *       ENTRY   SAME AS EVAL
                      3410  *       EXIT    (BC) UPDATED
                      3411  *               (E) = VALUE
                      3412  *       USES    A,F,B,C,D,E
                      3413
                      3414
057.044  315 036 057  3415  EVALI8  CALL    EVALI
057.047  172          3416          MOV     A,D
057.050  247          3417          ANA     A
057.051  310          3418          RZ                      OK
057.052  303 122 070  3419          JMP     ERR.IN          TOO LARGE
```

```
                                3423  **      ABS - ABSOLUTE VALUE.
                                3424  *
                                3425  *       Y=ABS(X)
                                3426
                                3427
057.055  072 204 042           3428  ABS     LDA     ACCX+2
057.060  247                   3429          ANA     A
057.061  374 302 105           3430          CM      FPNEG
                                3431
                                3432  *       IDENTIFY FUNCTION
                                3433
057.064  311                   3434  LNO     RET


                                3436  **      ASC - DECODE ASCII VALUE
                                3437  *
                                3438  *       X=ASC("C")
                                3439
                                3440
057.065  021 202 042           3441  ASC.    LXI     D,ACCX
057.070  315 315 074           3442          CALL    FSE             FIND STRING TABLE ENTRY
057.073  247                   3443          ANA     A
057.074  312 020 061           3444          JZ      PEEK1           NULL STRING YEILDS 0
057.077  176                   3445          MOV     A,M             GIVE VALUE
057.100  303 020 061           3446          JMP     PEEK1


                                3448  **      CHR$ - CONVERT VALUE INTO ASCII CHARACTER.
                                3449  *
                                3450  *       C$=CHR$(X)
                                3451
                                3452
057.103  315 002 075           3453  CHR$    CALL    IFIX            MAKE INTEGER
057.106  325                   3454          PUSH    D               SAVE VALUE
057.107  041 001 000           3455          LXI     H,1
057.112  042 202 042           3456          SHLD    ACCX            SET LENGTH
057.115  021 202 042           3457          LXI     D,ACCX
057.120  315 033 073           3458          CALL    CSE.            CREATE TEMP STRINGTAB ENTRY
057.123  315 262 061           3459          CALL    FRC             SET FUNCTION RETURNS CHARACTER
057.126  321                   3460          POP     D               (DE) = VALUE
057.127  173                   3461          MOV     A,E
057.130  346 177               3462          ANI     177Q            CLEAR BIT
057.132  167                   3463          MOV     M,A             STORE
057.133  300                   3464          RNZ                     IF NOT NULL
057.134  062 202 042           3465          STA     ACCX            NULL STRING IF 00
057.137  311                   3466          RET
```

```
                        3468  **      CIN - CHARACTER INPUT FUNCTION.
                        3469  *
                        3470  *       I=CIN(CHAN)
                        3471  *
                        3472  *       INPUT SINGLE CHARACTER, NO MAPPING OR PARITY ADJUSTMENT.
                        3473  *       I=-1 IF NO CHARACTER AVAILABLE
                        3474
                        3475
057.140                 3476  CIN     EQU     *
057.140  315 002 075   3477          CALL    IFIX            GET CHANNEL NUMBER
057.143  305           3478          PUSH    B               SAVE TEXT POINTER
057.144  172           3479          MOV     A,D
057.145  247           3480          ANA     A
057.146  302 122 070   3481          JNZ     ERR.IN          TOO LARGE A NUMBER
057.151  173           3482          MOV     A,E             (A) = CHANNEL NUMBER
057.152  247           3483          ANA     A
057.153  302 202 057   3484          JNZ     CIN2            FROM FILE
                        3485
                        3486  *       IS INPUT FROM CONSOLE
                        3487
057.156  377 001       3488          DB      SYSCALL,.SCIN   READ CHARACTER
057.160  137           3489  CIN0    MOV     E,A
057.161  026 000       3490          MVI     D,0
057.163  322 170 057   3491          JNC     CIN1            GOT CHARACTER
057.166  036 001       3492          MVI     E,1
057.170  365           3493  CIN1    PUSH    PSW
057.171  315 040 075   3494          CALL    IFLT            FLOAT IT
057.174  361           3495          POP     PSW
057.175  334 302 105   3496          CC      FPNEG           NEGATE IT, IF NO CHARACTER
057.200  301           3497          POP     B               RESTORE TEXT POINTER
057.201  311           3498          RET                     EXIT
                        3499
                        3500  *       READ CHARACTER FROM FILE
                        3501
057.202  315 005 072   3502  CIN2    CALL    CFA             COMPUTE FILE ADDRESS
057.205  332 210 070   3503          JC      ERR.FNO         FILE NOT OPEN
057.210  315 364 101   3504          CALL    $FREAD          READ CHARACTER
057.213  303 160 057   3505          JMP     CIN0            PROCESS VALUE ('C' SET IF EOF)



                        3507  **      INT - TRUNCATE TO NEAREST INTEGER.
                        3508  *
                        3509  *       Y=INT(X)
                        3510
                        3511
057.216                 3512  INT     EQU     *
057.216  041 204 042   3513          LXI     H,ACCX+2
057.221  176           3514          MOV     A,M             (A) = SIGN
057.222  247           3515          ANA     A
057.223  365           3516          PUSH    PSW             SAVE TEST RESULTS
057.224  374 302 105   3517          CM      FPNEG           MAKE POSITIVE
057.227  021 302 057   3518          LXI     D,INTA
057.232  315 352 104   3519          CALL    FPADD           ROUND UP
057.235  043           3520          INX     H               (HL) = #ACCX+3
```

```
057.236  305           3521          PUSH    B               SAVE (BC)
057.237  106           3522          MOV     B,M             (B) = EXPONENT
057.240  004           3523          INR     B
057.241  021 000 000   3524          LXI     D,0
057.244  112           3525          MOV     C,D             (C,D,E) = MASK
                       3526
                       3527  *       SHIFT IN ONE BITS TO CORRESPOND TO NON-FRACTIONAL BITS.
                       3528
057.245  005           3529  INT1    DCR     B
057.246  362 260 057   3530          JP      INT2            NO MORE
057.251  067           3531          STC
057.252  315 233 107   3532          CALL    SRS.            SHIFT (BCD) RIGHT THROUGH CARRY
057.255  303 245 057   3533          JMP     INT1
                       3534
057.260  053           3535  INT2    DCX     H
057.261  176           3536          MOV     A,M             MASK OFF VALUE
057.262  241           3537          ANA     C
057.263  167           3538          MOV     M,A
057.264  053           3539          DCX     H
057.265  176           3540          MOV     A,M
057.266  242           3541          ANA     D
057.267  167           3542          MOV     M,A
057.270  053           3543          DCX     H
057.271  176           3544          MOV     A,M
057.272  243           3545          ANA     E
057.273  167           3546          MOV     M,A
057.274  301           3547          POP     B               RESTORE (BC)
057.275  361           3548          POP     PSW             (A) = ORIGINAL SIGN TEST RESULTS
057.276  374 302 105   3549          CM      FPNEG           RE-INVERT IF NECESSARY
057.301  311           3550          RET
                       3551
057.302  000 000 101   3552  INTA    DB      0000,0000,1010,1570



                       3554  **      LEN - LENGTH OF STRING.
                       3555  *
                       3556  *       X=LEN(S$)
                       3557
                       3558
057.306  072 202 042   3559  LEN     LDA     ACCX            (A) = LENGTH
057.311  303 020 061   3560          JMP     PEEK1           FLOAT INTO ACCX



                       3562  **      LEFT$ - GET LEFTMOST CHARACTERS.
                       3563  *
                       3564  *       Y$=LEFT$(X$,CNT)
                       3565
                       3566
                       3567  **      RIGHT$ - GET RIGHTMOST CHARACTERS.
                       3568  *
                       3569  *       Y$=RIGHT$(X$,CNT)
                       3570
```

```
                                3571
                                3572
                                3573
                                3574  **      MID$ - GET SEGMENT OF CHARACTER STRING.
                                3575  *
                                3576  *       Y$=MID$(X$,POS,LEN)
                                3577
                                3578
057.314                         3579  LEFT$   EQU     *
057.314                         3580  RIGHT$  EQU     *
057.314                         3581  MID$    EQU     *
057.314  365                   3582          PUSH    PSW             SAVE TYPE CODE
057.315  315 223 072            3583          CALL    CMA             REQUIRE ','
057.320  315 033 077            3584          CALL    PSHX            SAVE X$ POINTER
057.323  315 044 057            3585          CALL    EVAL18          EVALUATE 8 BIT RESULT
057.326  123                   3586          MOV     D,E             (D) = LEN
057.327  036 001               3587          MVI     E,1             (E) = POS
057.331  315 370 076            3588          CALL    POPY.           (Y) = X$ POINTER
057.334  361                   3589          POP     PSW
000.003                         3590          ERRMI CT.MID-CT.LEF
057.335  376 070               3591          CPI     CT.MID-CT.FCN*2
057.337  312 365 057            3592          JE      MID$1           IS MID$
057.342  332 032 060            3593          JC      LEFT$1          IS LEFT$
377.377                         3594          ERRPL CT.MID-CT.RIG
                                3595
                                3596  *       IS RIGHT$
                                3597  *
                                3598  *       GENERATE MID$(X$,LENX$-MIN(LENX$,CNT),MIN(LENX$,CNT)
                                3599
057.345  072 210 042            3600          LDA     ACCY
057.350  137                   3601          MOV     E,A             (E) = LENX$
057.351  272                   3602          CMP     D
057.352  322 356 057            3603          JNC     RIGHT$1
057.355  127                   3604          MOV     D,A             (D) = MIN(LENX$,CNT)
057.356  173                   3605  RIGHT$1 MOV     A,E             (A) = LENX$
057.357  222                   3606          SUB     D               (A) = LENX$-MIN(LENX$,CNT)
057.360  137                   3607          MOV     E,A             (E) = LENX$-MIN(LENX$,CNT)
057.361  034                   3608          INR     E
057.362  303 032 060            3609          JMP     MID$2           MOVE
                                3610
                                3611  *       IS MID$
                                3612  *
                                3613  *       EVALUATE CNT
                                3614
057.365                         3615  MID$1   EQU     *
057.365  132                   3616          MOV     E,D             (E) = POS
057.366  172                   3617          MOV     A,D
057.367  247                   3618          ANA     A
057.370  312 122 070            3619          JZ      ERR.IN          0 ILLEGAL
057.373  026 377               3620          MVI     D,255           ASSUME NULL (LEN=255)
057.375  315 072 076            3621          CALL    PNT             PREVIEW NEXT TOKEN
060.000  376 020               3622          CPI     CT.PAR
060.002  312 032 060            3623          JE      MID$2           IS NULL
060.005  315 056 071            3624          CALL    ANT             ACCEPT ,
060.010  376 026               3625          CPI     CT.CMA
060.012  302 152 070            3626          JNE     ERR.SY
```

```
060.015  325          3627          PUSH   D              SAVE CURRENT POS
060.016  315 041 077  3628          CALL   PSHY           SAVE STRING
060.021  315 044 057  3629          CALL   EVALIB         EVALUATE LEN
060.024  315 370 076  3630          CALL   POPY.          (ACCY) = X$ POINTER
060.027  353          3631          XCHG
060.030  321          3632          POP    D
060.031  125          3633          MOV    D,L            SET NEW LEN
                      3634
                      3635  *       (ACCX) = X$ POINTER
                      3636  *       (D) = LEN
                      3637  *       (E) = POS
                      3638  *
                      3639  *       COMPUTE Y$ = MID$(X$,POS,LEN)
                      3640
060.032               3641  LEFT$1  EQU    *
060.032               3642  MID$2   EQU    *
                      3643
                      3644  *       COMPUTE LEN' = MIN(LEN,MAX(LENX$-POS+1,0))
                      3645
060.032  072 210 042  3646          LDA    ACCY           (A) = LENX$
060.035  223          3647          SUB    E
060.036  074          3648          INR    A              (A) = LENX$-POS+1
060.037  322 043 060  3649          JNC    MID$3          IS >= 0
060.042  257          3650          XRA    A              USE 0
060.043  272          3651  MID$3   CMP    D
060.044  322 050 060  3652          JNC    MID$4          (D) = MIN VALUE
060.047  127          3653          MOV    D,A            (D) = MIN VALUE
060.050  046 000       3654  MID$4   MVI    H,0
060.052  152          3655          MOV    L,D            (HL) = LEN
060.053  042 202 042  3656          SHLD   ACCX           SET IN BLOCK
060.056  325          3657          PUSH   D              SAVE LENGTH
060.057  021 202 042  3658          LXI    D,ACCX
060.062  315 033 073  3659          CALL   CSE.           CREATE TEMP STRINGTAB ENTRY
060.065  343          3660          XTHL                  SAVE ADDRESS
060.066  345          3661          PUSH   H              SAVE LEN, POS
060.067  021 210 042  3662          LXI    D,ACCY
060.072  315 315 074  3663          CALL   FSE            FIND STRING ENTRY
060.075  321          3664          POP    D              (E) = POS
060.076  173          3665          MOV    A,E
060.077  075          3666          DCR    A
060.100  315 072 030  3667          CALL   $DADA          (HL) = FROM ADDRESS
060.103  172          3668          MOV    A,D            (A) = LEN
060.104  353          3669          XCHG
060.105  341          3670          POP    H              (HL) = TO ADDRESS
060.106  303 257 061  3671          JMP    STR$1          MOVE, EXIT WITH TYPE = CT.SSV



                      3673  ***     MATCH - FIND SUBSTRING IN STRING.
                      3674  *
                      3675  *       I=MATCH$(S1$,S2$,J)
                      3676  *
                      3677  *       SCAN S1$ FOR OCCURANCE OF S2$, STARTING AT CHARACTER J
                      3678  *
                      3679  *       I=0 IF NOT FOUND
```

```
                                3680  *       I = CHARACTER NUMBER OF START OF MATCH IF FOUND
                                3681
                                3682
060,141                         3683  MATCH   EQU     *
060,111  041 307 060            3684          LXI     H,MATCHA
060,114  315 051 076            3685          CALL    MOV4            SAVE S1 DESCRIPTOR
060,117  315 223 072            3686          CALL    CMA             GOBBLE CMA
                                3687
060,122  315 244 055            3688          CALL    EVAL                                            /80.01.GC/
060,125  072 201 042            3689          LDA     AGCX-1                                          /80.01.GC/
060,130  346 001                3690          ANI     CF.STR                                          /80.01.GC/
060,132  312 155 070            3691          JZ      ERR.TC          REQUIRE A STRING                /80.01.GC/
060,135  041 313 060            3692          LXI     H,MATCHC
060,140  315 051 076            3693          CALL    MOV4            SAVE S2 DESCRIPTION
060,143  315 223 072            3694          CALL    CMA             GOBBLE ','
                                3695
060,146  315 044 057            3696          CALL    EVALI8          EVALUATE INDEX
060,151  305                    3697          PUSH    B               SAVE TEXT POINTER
060,152  103                    3698          MOV     B,E             (B) = J
060,153  021 313 060            3699          LXI     D,MATCHC
060,156  315 315 074            3700          CALL    FSE             FIND S2
060,161  345                    3701          PUSH    H               SAVE S2 ADDRESS
060,162  365                    3702          PUSH    PSW             SAVE S2 COUNT
060,163  021 307 060            3703          LXI     D,MATCHA
060,166  315 315 074            3704          CALL    FSE             FIND S1
060,171  365                    3705          PUSH    PSW             SAVE S1 LENGTH
060,172  175                    3706          MOV     A,L
060,173  062 276 060            3707          STA     MATCHB          SAVE ADDRESS (IN PAGE) OF START
060,176  170                    3708          MOV     A,B
060,177  247                    3709          ANA     A
060,200  312 122 070            3710          JZ      ERR.IN          ILLEGAL NUMBER
060,203  075                    3711          DCR     A
060,204  315 101 030            3712          CALL    $DADA.          (HL) = START OF SEARCH AREA
060,207  361                    3713          POP     PSW
060,210  220                    3714          SUB     B               (A) = LEN(S1)-J
060,211  332 257 060            3715          JC      MATC2,3         NOT ANYWHERE
060,214  074                    3716          INR     A
060,215  301                    3717          POP     B               (B) = S2 LENGTH
060,216  220                    3718          SUB     B               (A) = # OF TRYS -1
060,217  332 260 060            3719          JC      MATC2,5         NONE
060,222  321                    3720          POP     D               (DE) = S2 ADDRESS
060,223  074                    3721          INR     A
060,224  365                    3722          PUSH    PSW             SAVE TRY COUNT
                                3723
                                3724  *       SEE IF MATCH
                                3725
060,225  032                    3726  MATCH1  LDAX    D
060,226  276                    3727          CMP     M
060,227  302 247 060            3728          JNE     MATCH2          NOT THIS ONE
060,232  325                    3729          PUSH    D
060,233  345                    3730          PUSH    H
060,234  305                    3731          PUSH    B               SAVE ALL REGS
060,235  110                    3732          MOV     C,B             (C) = S2 LENGTH = COMPARE COUNT
060,236  315 060 030            3733          CALL    $COMP           COMPARE THE REST
060,241  301                    3734          POP     B
060,242  341                    3735          POP     H
```

```
060.243  321            3736           POP     D
060.244  312 273 060     3737           JE      MATCH3          GOT IT
                         3738
                         3739  *        NO MATCH AT THIS ONE
                         3740
060.247  043            3741  MATCH2   INX     H
060.250  361            3742           POP     PSW
060.251  075            3743           DCR     A
060.252  365            3744           PUSH    PSW             COUNT IT
060.253  302 225 060     3745           JNZ     MATCH1          MORE TO TRY
060.256  365            3746           PUSH    PSW             SET STACK PROPERLY
060.257  361            3747  MATC2.3  POP     PSW
060.260  361            3748  MATC2.5  POP     PSW
060.261  041 202 042     3749           LXI     H,ACCX
060.264  006 004         3750           MVI     B,4
060.266  315 212 031     3751           CALL    $ZERO           RETURN ZERO FOR ANSWER
060.271  301            3752           POP     B               RESTORE (BC)
060.272  311            3753           RET
                         3754
                         3755  *        GOT ONE
                         3756
060.273  361            3757  MATCH3   POP     PSW
060.274  175            3758           MOV     A,L             (A) = FWA OF STRING
060.275  326 000         3759           SUI     0               SUBTRTACT START ADDRESS
060.276                  3760  MATCHB   EQU     *-1             INDEX OF START OF S1
060.277  137            3761           MOV     E,A
060.300  026 000         3762           MVI     D,0
060.302  023            3763           INX     D               BIAS INTO 1 TO 256
060.303  301            3764           POP     B               RESTORE TEXT POINTER
060.304  303 040 075     3765           JMP     IFLT            FLOAT RESULT, AND EXIT
                         3766
060.307                  3767  MATCHA   DS      4               HOLD AREA FOR STRING DESCRIPTOR
060.313                  3768  MATCHC   DS      4               HOLD AREA FOR S2 DESCRIPTOR


                         3770  **       MAX - COMPUTE 'MAXIMUM' FUNCTION.
                         3771  *
                         3772  *        Y=MAX(X1,...,XN)
                         3773
                         3774
                         3775  **       MIN - COMPUTE 'MINIMUM' FUNCTION.
                         3776  *
                         3777  *        Y=MIN(X1,...,XN)
                         3778
                         3779
060.317  076            3780  MAX      DB      MI.MVIA
060.320  257            3781  MIN      XRA     A               (A) = MI.MVI IF MAX, 0 IF MIN
060.321  365            3782           PUSH    PSW             SAVE CODE
060.322  315 317 100     3783           CALL    XCY             (ACCY) = CURRENT CANDIDATE
                         3784
                         3785  *        (ACCY) = CURRENT CANDIDATE
                         3786
060.325  315 072 076     3787  MAX1     CALL    PNT             PEEK AT NEXT TOKEN
060.330  376 020         3788           CPI     CT.PAR
```

```
060.332  312 002 061  3789           JE      MAX2            IS )
060.335  315 223 072  3790           CALL    CMA             REQUIRE ','
060.340  315 041 077  3791           CALL    PSHY            SAVE CURRENT BEST
060.343  315 022 057  3792           CALL    EVALN           EVALUATE NUMBER
060.346  315 365 076  3793           CALL    POPY            RESTORE CURRENT BEST
060.351  315 033 077  3794           CALL    PSHX            SAVE LATEST
060.354  021 210 042  3795           LXI     D,ACCY
060.357  315 166 105  3796           CALL    FPSUB           COMPUTE (CANDIDATE-LATEST)
060.362  072 204 042  3797           LDA     ACCX+2
060.365  127          3798           MOV     D,A
060.366  315 357 076  3799           CALL    POPX            RESTORE LATEST TRY
060.371  361          3800           POP     PSW
060.372  365          3801           PUSH    PSW             (A) = MIN/MAX FLAG
060.373  252          3802           XRA     D               (A) = CODE
060.374  364 317 100  3803           CP      XCY             LATEST IS SUPERIOR
060.377  303 325 060  3804           JMP     MAX1
                      3805
                      3806  *        AT END OF LIST.
                      3807
061.002  361          3808  MAX2     POP     PSW
061.003  303 317 100  3809           JMP     XCY             (ACCX) = BEST FOUND


                      3811  **       PAD - READ KEYPAD.
                      3812  *
                      3813  *        Y=PAD(0)
                      3814
                      3815
061.006  315 260 003  3816  PAD      CALL    .RCK            ;READ VALUE
061.011  303 020 061  3817           JMP     PEEK1           RETURN VALUE


                      3819  **       PEEK - PEEK AT MEMORY.
                      3820  *
                      3821  *        X=PEEK(ADDR)
                      3822
                      3823
061.014  315 002 075  3824  PEEK     CALL    IFIX            EVAL TO 16 BITS
061.017  032          3825           LDAX    D
061.020  137          3826  PEEK1    MOV     E,A
061.021  026 000      3827           MVI     D,0             (DE) = VALUE
061.023  315 040 075  3828  PEEK1.5  CALL    IFLT            FLOAT INTO ACCX
061.026  076 300      3829  PEEK2    MVI     A,CT.SNV        SCALAR NUMERIC VALUE
061.030  062 201 042  3830           STA     ACCX-1
061.033  311          3831           RET
```

```
                                3833  **      PIN - PORT INPUT.
                                3834  *
                                3835  *       Y=PIN(PORT)
                                3836
                                3837
061.034  315 002 075           3838  PIN     CALL    IFIX
061.037  143                   3839          MOV     H,E
061.040  056 333               3840          MVI     L,MI.IN
061.042  042 002 040           3841          SHLD    .IOWRK
061.045  315 002 040           3842          CALL    .IOWRK          INPUT
061.050  303 020 061           3843          JMP     PEEK1           FLOAT AND EXIT


                                3845  **      POS - RETURN PRINT HEAD POSITION.
                                3846  *
                                3847  *       X=POS(PORT)
                                3848
                                3849
061.053  315 002 075           3850  POS     CALL    IFIX
061.056  041 253 112           3851          LXI     H,COLCNTS
061.061  173                   3852          MOV     A,E
061.062  247                   3853          ANA     A
061.063  312 067 061           3854          JZ      POS1            IS CHANNEL 0
061.066  023                   3855          INX     D               (DE) = INDEX INTO COLCNTS
061.067  031                   3856  POS1    DAD     D
061.070  176                   3857          MOV     A,M             (A) = POSITION
061.071  303 020 061           3858          JMP     PEEK1           FLOAT


                                3860  **      RND - COMPUTE TAUSWORTH 15 BIT RANDOM NUMBER.
                                3861  *
                                3862  *       X=RND(Y)
                                3863  *
                                3864  *       Y = 0, GIVE LAST RANDOM NUMBER
                                3865  *       Y > 0, GIVE NEXT RANDOM NUMBER
                                3866  *       Y < 0, SEED WITH Y
                                3867
                                3868
061.074                         3869  RND     EQU     *
061.074  072 204 042           3870          LDA     ACCX+2          EXAMINE SIGN
061.077  247                   3871          ANA     A
061.100  041 114 107           3872          LXI     H,'GL'          (HL) = SEED
061.101                         3873  RNDA    EQU     *-2             SEED
061.103  026 017               3874          MVI     D,15            (D) = BIT COUNT
061.105  312 150 061           3875          JZ      RND2            JUST RETURN SEED
061.110  362 116 061           3876          JP      RND1            GENERATE NEW NUMBER
061.113  052 203 042           3877          LHLD    ACCX+1          USE NEW SEED
061.116  174                   3878  RND1    MOV     A,H             SHIFT RIGHT ONE
061.117  247                   3879          ANA     A
061.120  037                   3880          RAR
061.121  147                   3881          MOV     H,A
061.122  175                   3882          MOV     A,L
```

```
061.123  037           3883           RAR
061.124  157           3884           MOV     L,A
061.125  027           3885           RAL                     'C' = 1
061.126  027           3886           RAL
061.127  027           3887           RAL
061.130  027           3888           RAL                     'C' = 100
061.131  255           3889           XRA     L               XOR WITH VALUE
061.132  027           3890           RAL
061.133  027           3891           RAL
061.134  027           3892           RAL
061.135  346 100       3893           ANI     100Q
061.137  264           3894           ORA     H               INSERT IN LEFT
061.140  147           3895           MOV     H,A
061.141  025           3896           DCR     D
061.142  302 116 061   3897           JNZ     RND1            MORE TO GO
061.145  042 101 061   3898           SHLD    RNDA            SAVE SEED
                       3899
061.150  353           3900  RND2     XCHG                    (DE) = VALUE
061.151  041 202 042   3901           LXI     H,ACCX
061.154  066 000       3902           MVI     M,0             ZERO LOW 8
061.156  043           3903           INX     H
061.157  163           3904           MOV     M,E
061.160  043           3905           INX     H
061.161  162           3906           MOV     M,D
061.162  043           3907           INX     H
061.163  066 200       3908           MVI     M,200Q          EXPONENT
061.165  303 202 105   3909           JMP     FPNRM           NORMALIZE AND EXIT


                       3911  **       SEG - DECODE SEGMENT VALUE.
                       3912  *
                       3913  *        Y=SEG(NUM)
                       3914  *
                       3915  *        DECODE VALUES 0-9.
                       3916  *        NUM = 10 GIVES BLANK.
                       3917
                       3918
061.170  315 002 075   3919  SEG      CALL    IFIX
061.173  041 356 003   3920           LXI     H,.DODA
061.176  031           3921           DAD     D
061.177  176           3922           MOV     A,M
061.200  366 200       3923           ORI     200Q            CLEAR DECIMAL
061.202  303 020 061   3924           JMP     PEEK1           DECODE VALUE


                       3926  **       SGN - RETURN SIGN OF NUMBER.
                       3927  *
                       3928  *        Y=SGN(X)
                       3929  *
                       3930  *        ; = -1 IF X<0, =0 IF X=0, =1 IF X > 0
                       3931
                       3932
```

```
061.205  021 000 000  3933  SGN     LXI     D,0
061.210  072 204 042  3934          LDA     ACCX+2
061.213  247          3935          ANA     A
061.214  312 040 075  3936          JZ      IFLT            IS 0
061.217  023          3937          INX     D
061.220  362 040 075  3938          JP      IFLT            IS POSITIVE
061.223  315 040 075  3939          CALL    IFLT
061.226  303 302 105  3940          JMP     FPNEG           MAKE -1


                      3942  **      STR$ - CONVERT FLOATING TO ASCII.
                      3943  *
                      3944  *       Y$=STR$(X)
                      3945
                      3946
061.231  041 273 113  3947  STR$    LXI     H,LINE2
061.234  315 237 110  3948          CALL    FTA             FLOATING TO ASCII
061.237  345          3949          PUSH    H               SAVE 'FROM'
061.240  157          3950          MOV     L,A
061.241  046 000      3951          MVI     H,0
061.243  365          3952          PUSH    PSW             SAVE LENTRH
061.244  042 202 042  3953          SHLD    ACCX            SET LENGTH
061.247  021 202 042  3954          LXI     D,ACCX
061.252  315 033 073  3955          CALL    CSE.            CREATE TEMP ENTRY
061.255  361          3956          POP     PSW             (A) = COUNT
061.256  321          3957          POP     D
061.257  315 015 071  3958  STR$1   CALL    MOV             MOVE IT
                      3959
                      3960  **      FRC - FUNCTION RETURNS CHARACTER VALUE.
                      3961
061.262  076 301      3962  FRC     MVI     A,CT.SSV        SCALAR STRING VALUE
061.264  062 201 042  3963          STA     ACCX-1
061.267  311          3964          RET


                      3966  **      VAL - CONVERT ASCII TO FLOATING POINT.
                      3967  *
                      3968  *       Y=VAL(X$)
                      3969
                      3970
061.270  021 202 042  3971  VAL     LXI     D,ACCX
061.273  315 315 074  3972          CALL    FSE             FIND STRINGTAB ENTRY
061.276  353          3973          XCHG
061.277  041 273 113  3974          LXI     H,LINE2
061.302  315 015 071  3975          CALL    MOV             MOVE TO *LINE*
061.305  066 000      3976          MVI     M,0             MAKE SURE TERMINATES
061.307  041 273 113  3977          LXI     H,LINE2
061.312  315 330 111  3978          CALL    $SOB            SKIP OVER BLANKS
061.315  315 323 107  3979          CALL    ATF             ASCII TO FLOATING
061.320  303 026 061  3980          JMP     PEEK2
```

```
                                3982  **      P.OR - PROCESS BOOLEAN 'OR'.
                                3983  *
                                3984  *       ENTRY   (ACCX) = 1ST VALUE
                                3985  *               (ACCY) = 1ND VALUE
                                3986  *       EXIT    (ACCX) = 1ST 'OR' 2ND
                                3987
                                3988
061.323  315 345 076            3989  P.OR    CALL    PBO             PROCESS BOOLEAN OPERATOR
061.326  172                    3990          MOV     A,D
061.327  264                    3991          ORA     H
061.330  127                    3992          MOV     D,A
061.331  173                    3993          MOV     A,E
061.332  265                    3994          ORA     L
061.333  303 371 061            3995          JMP     P.NOT1



                                3997  **      P.AND - PROCESS BOOLEAN AND.
                                3998  *
                                3999  *       ENTRY   NONE
                                4000  *       EXIT    (ACCX) = IFLT(IFIX(ACCX).AND.IFIX(ACCY))
                                4001
                                4002
061.336  315 345 076            4003  P.AND   CALL    PBO             PREPARE BOOLEAN
061.341  172                    4004          MOV     A,D
061.342  244                    4005          ANA     H
061.343  127                    4006          MOV     D,A
061.344  173                    4007          MOV     A,E
061.345  245                    4008          ANA     L
061.346  303 371 061            4009          JMP     P.NOT1



                                4011  **      P.NOT - PROCESS BOOLEAN 'NOT'.
                                4012  *
                                4013  *       ENTRY   NONE
                                4014  *       EXIT    (ACCX) = IFLT(.NOT.IFIX(ACCX))
                                4015
                                4016
061.351  072 201 042            4017  P.NOT   LDA     ACCX-1          (A) = TYPE OF ARGUMENT
061.354  376 300                4018          CPI     CT.SNV
061.356  302 155 070            4019          JNE     ERR.TC          WRONG TYPE
061.361  315 377 074            4020          CALL    IFIX.           (DE) = IFIX(ACCX)
061.364  172                    4021          MOV     A,D
061.365  057                    4022          CMA
061.366  127                    4023          MOV     D,A
061.367  173                    4024          MOV     A,E
061.370  057                    4025          CMA
061.371  137                    4026  P.NOT1  MOV     E,A             (DE) = RESULT
061.372  303 040 075            4027          JMP     IFLT            FLOAT AND EXIT
```

```
                                4029  **      P.CMP - PROCESS COMPARES.
                                4030  *
                                4031  *       Y= X1 'OP' X2
                                4032  *
                                4033  *       OP = < <= = <> >= >
                                4034  *
                                4035  *       THE TWO OPERANDS ARE COMPARED, AND THE RESULT GENERATES
                                4036  *       A BIT PATTERN:
                                4037  *
                                4038  *       001     EQUAL
                                4039  *       010     POSITIVE
                                4040  *       100     NEGATIVE
                                4041  *
                                4042  *       THIS PATTERN IS THEN MASKED WITH THE PATTERNS GENERATED BY THE
                                4043  *       OPERATORS:
                                4044  *
                                4045  *       =  001
                                4046  *       >  010
                                4047  *       >= 011
                                4048  *       <  100
                                4049  *       <= 101
                                4050  *       <> 110
                                4051  *
                                4052  *       ENTRY   (ACCX) = X2
                                4053  *               (ACCY) = X1
                                4054  *               (L) = OPERATOR 'CT.' CODE
                                4055  *       USES    A,F,D,E,H,L
                                4056
                                4057
061,375  345                   4058  P.CMP   PUSH    H               SAVE (H)
061,376  315 347 072           4059          CALL    CDT             CHECK OPERAND TYPE
062,001  302 047 062           4060          JNZ     P.CMP2          IS STRING
                                4061
                                4062  *       IS NUMERIC COMPARE.
                                4063
062,004  315 166 105           4064          CALL    FPSUB           (ACCX) = X1-X2
062.007  072 204 042           4065          LDA     ACCX+2          (A) = SIGN OF RESULT
062,012  247                   4066          ANA     A               SET FLAGS
062.013  341                   4067  P.CMP1  POP     H
062,014  067                   4068          STC
062.015  077                   4069          CMC                     CLEAR CARRY
062,016  076 001               4070          MVI     A,1             ASSUME IS ZERO
062.020  312 030 062           4071          JZ      P.CMP13         IS ZERO
062,023  027                   4072          RAL
062.024  362 030 062           4073          JP      P.CMP13         IS POSITIVE
062,027  027                   4074          RAL                     IS NEGATIVE
062.030  157                   4075  P.CMP13 MOV     L,A             (L) = RESULTS OF TEST
062,031  174                   4076          MOV     A,H             (A) = TYPE
062.032  326 010               4077          SUI     CT.EQ-1         (A) = CODE FOR DCONDITIONS
062.034  245                   4078          ANA     L
062.035  021 000 000           4079          LXI     D,0             ASSUME FALSE
062,040  312 023 061           4080          JZ      PEEK1.5         IS FALSE
062.043  033                   4081          DCX     D
062.044  303 023 061           4082          JMP     PEEK1.5         IS TRUE
```

```
                                4084  **      STRING COMPARES.
                                4085  *
                                4086  *       COMPARE CHARACTER FOR CHARACTER. IF A STRING RUNS OUT, ITS
                                4087  *       NEXT CHARACTER IS CONSIDERED TO BE '00'
                                4088
                                4089
062.047  305                   4090  P.CMP2  PUSH    B               SAVE (BC)
062.050  021 202 042           4091          LXI     D,ACCX
062.053  315 315 074           4092          CALL    FSE             FIND STRING ENTRY
062.056  107                   4093          MOV     B,A             (B) = LEN(X2)
062.057  345                   4094          PUSH    H               SAVE ADDRESS OF X2
062.060  021 210 042           4095          LXI     D,ACCY
062.063  315 315 074           4096          CALL    FSE             FIND ENTRY
062.066  117                   4097          MOV     C,A             (C) = LEN(X1)
062.067  321                   4098          POP     D               (DE) = ADR(X2), (HL) = ADR(X1)
062.070  353                   4099          XCHG
062.071  004                   4100          INR     B               (B) = LEN(X2)+1
062.072  014                   4101          INR     C               (C) = LEN(X1)+1
                                4102
                                4103  *       COMPARE STRINGS.
                                4104
062.073  005                   4105  P.CMP3  DCR     B
062.074  312 121 062           4106          JZ      P.CMP5          OUT OF X2
062.077  015                   4107          DCR     C
062.100  312 115 062           4108          JZ      P.CMP4          OUT OF X1, BUT NOT X2
062.103  032                   4109          LDAX    D
062.104  226                   4110          SUB     M
062.105  302 130 062           4111          JNZ     P.CMP6          HAVE RESULT
062.110  023                   4112          INX     D
062.111  043                   4113          INX     H
062.112  303 073 062           4114          JMP     P.CMP3          TOO EARLY TO TELL
                                4115
                                4116  *       RAN OUT OF X1, BUT NOT X2. RESULT IS X1 < X2
                                4117
062.115  015                   4118  P.CMP4  DCR     C               SET 'M' FLAG
062.116  303 130 062           4119          JMP     P.CMP6
                                4120
                                4121  *       RAN OUT OF X2, DONT KNOW ABOUT X1
                                4122
062.121  015                   4123  P.CMP5  DCR     C
062.122  312 130 062           4124          JZ      P.CMP6          OUT OF BOTH
062.125  076 001               4125          MVI     A,1
062.127  247                   4126          ANA     A               X2 > X1
                                4127
                                4128  *       HAVE COMPUARE RESULT IN PSW
                                4129
062.130  301                   4130  P.CMP6  POP     B               RESTORE (BC)
062.131  303 013 062           4131          JMP     P.CMP1
```

```
                          4133  **      P.ADD - PROCESS ADD AND SUBTRACT.
                          4134  *
                          4135
                          4136
062.134  076 021          4137  P.ADD   MVI     A,CT.PL
062.136  274              4138          CMP     H
062.137  302 241 062      4139          JNE     P.SUB           IS -
062.142  315 347 072      4140          CALL    COT             CHECK OPERAND TYPE
062.145  312 352 104      4141          JZ      FPADD           IS NUMERIC ADD
                          4142
                          4143  *       IS STRING CONCATINATE.
                          4144
062.150  072 202 042      4145          LDA     ACCX
062.153  157              4146          MOV     L,A             (L) = LEN(X2)
062.154  072 210 042      4147          LDA     ACCY
062.157  205              4148          ADD     L               (A) = RESULTANT LENGTH
062.160  332 144 070      4149          JC      ERR.SL          STRING LENGTH ERROR
062.163  157              4150          MOV     L,A             (HL) = LEN
062.164  046 000          4151          MVI     H,0
062.166  042 235 062      4152          SHLD    P.ADDA          SAVE INDEX IN BUILD BLOCK AREA
062.171  021 235 062      4153          LXI     D,P.ADDA
062.174  315 033 073      4154          CALL    CSE.            CREATE TEMP STRINGTAB ENTRY
062.177  345              4155          PUSH    H               SAVE 'TO'
062.200  021 210 042      4156          LXI     D,ACCY
062.203  315 315 074      4157          CALL    FSE             FIND ENTRY
062.206  353              4158          XCHG                    (DE) = FROM
062.207  341              4159          POP     H               (HL) = TO
062.210  315 015 071      4160          CALL    MOV             COPY 1ST
062.213  345              4161          PUSH    H               SAVE TO
062.214  021 202 042      4162          LXI     D,ACCX
062.217  315 315 074      4163          CALL    FSE
062.222  353              4164          XCHG                    (DE) = FROM
062.223  341              4165          POP     H               (HL) = TO
062.224  315 015 071      4166          CALL    MOV
062.227  021 235 062      4167          LXI     D,P.ADDA
062.232  303 210 073      4168          JMP     CVX             COPY BLOCK TO ACCX
                          4169
062.235                   4170  P.ADDA  DS      4



                          4172  *
                          4173  *       (ACCX) = (ACCY-ACCX)
                          4174
                          4175
062.241  315 177 077      4176  P.SUB   CALL    RNO             REQUIRE NUMERIC OPERANDS
062.244  303 166 105      4177          JMP     FPSUB
```

```
                                4179  **      M.MUL - PROCESS MULTIPLICATION AND DIVISION.
                                4180  *
                                4181  *       (ACCX) = (ACCX)*(ACCY)
                                4182  *
                                4183  *       ENTRY   (DE) = #ACCY
                                4184
                                4185
062.247  076 024               4186  P.MUL   MVI     A,CT.DI
062.251  274                   4187          CMP     H               SEE IF /
062.252  365                   4188          PUSH    PSW             SAVE RESULT
062.253  315 177 077           4189          CALL    RNO             REQUIRE NUMERIC OPERANDS
062.256  361                   4190          POP     PSW
062.257  302 323 105           4191          JNE     FPMUL           IS *
062.262  315 317 100           4192          CALL    XCY             INVERT
062.265  303 260 106           4193          JMP     FPDIV           DIVIDE


                                4195  **      P.EXP - EXPONENTIATION.
                                4196  *
                                4197  *       (ACCX) = (VAL)^(POWR)
                                4198  *
                                4199  *       IF (ACCY)>0, COMPUTE RSLT=EXP(X*LOG(Y))
                                4200  *
                                4201  *       ENTRY   (ACCY) = VAL
                                4202  *               (ACCX) = POWER
                                4203  *       EXIT    (ACCX) = RESULT
                                4204
                                4205
062.270                         4206  P.EXP   EQU     *
062.270  315 317 100           4207          CALL    XCY             (ACCX) = VAL
062.273  072 205 042           4208          LDA     ACCX+3          CHECK IF VAL IS 0
062.276  247                   4209          ANA     A
062.277  302 321 062           4210          JNZ     P.EXP1          VAL NON - ZERO
                                4211
                                4212  *       CHECK FOR 0^0
                                4213
062.302  072 213 042           4214          LDA     ACCY+3          CHECK IF POWER IS 0
062.305  247                   4215          ANA     A
062.306  300                   4216          RNZ                     EXIT; POWER NON - ZERO, VAL = 0
                                4217
062.307  021 147 112           4218          LXI     D,FP1.0         POWER = ZERO; RETURN RESULT OF 1
062.312  041 202 042           4219          LXI     H,ACCX
062.315  315 051 076           4220          CALL    MOV4            (ACCX) = 1
062.320  311                   4221          RET                     EXIT
                                4222
062.321  315 041 077           4223  P.EXP1  CALL    PSHY            SAVE EXPONENT
062.324  315 225 063           4224          CALL    LOG             (ACCX) = LOG(Y)
062.327  315 365 076           4225          CALL    POPY
062.332  315 323 105           4226          CALL    FPMUL
062.335  303 075 063           4227          JMP     EXP             (ACCX) = EXP(X*LOG(Y))
```

```
                        4231  **      TXTFN - PERFORM TEXT DEFINED FUNCTIONS.
                        4232  *
                        4233
                        4234
062.340                 4235  TXTFN   EQU     *
062.340  315 056 071    4236          CALL    ANT             ACCEPT NEXT TOKEN
062.343  346 002        4237          ANI     CF.VEC
062.345  312 152 070    4238          JZ      ERR.SY          NOT VECTOR TYPE
062.350  032            4239          LDAX    D
062.351  247            4240          ANA     A
062.352  362 216 070    4241          JP      ERR.UD          NOT DECLARED AS FUNCTION
062.355  023            4242          INX     D
062.356  353            4243          XCHG
062.357  136            4244          MOV     E,M
062.360  043            4245          INX     H
062.361  126            4246          MOV     D,M             (DE) = ADDRESS OF FUNCTION DEFINITION
062.362  353            4247          XCHG
062.363  305            4248  TXTF1   PUSH    B
062.364  343            4249          XTHL
062.365  301            4250          POP     B               (BC) = ADDRESS OF PARAMETER LIST
                        4251
                        4252  *       ASSIGN VALUES TO PARAMETER LIST.
                        4253
062.366  345            4254  TXTF2   PUSH    H               SAVE (HL)
062.367  315 136 075    4255          CALL    IST             INSERT SYMBOL IN TABLE
062.372  341            4256          POP     H
062.373  325            4257          PUSH    D               SAVE INDEX
062.374  365            4258          PUSH    PSW
062.375  315 056 071    4259          CALL    ANT             EXAMINE NEXT TOKEN
063.000  365            4260          PUSH    PSW             SAVE FOR LATER
063.001  376 026        4261          CPI     CT.CMA
063.003  312 013 063    4262          JE      TXTF3           IS ,
063.006  376 020        4263          CPI     CT.PAR
063.010  302 152 070    4264          JNE     ERR.SY          BAD SYNTAX
                        4265
                        4266  *       SWAP (BC) (HL) TO DECODE VALUE FOR VARIABLE
                        4267
063.013  305            4268  TXTF3   PUSH    B
063.014  343            4269          XTHL
063.015  301            4270          POP     B
063.016  315 244 055    4271          CALL    EVAL            EVALUATE PARAMETER VALUE
063.021  361            4272          POP     PSW             (A) = NEXT CHARACTER FROM *ANT*
063.022  062 041 063    4273          STA     TXTFNA          SAVE FOR COMPARISON
063.025  361            4274          POP     PSW             RESTORE TYPE
063.026  321            4275          POP     D               RESTORE FARM ADDRESS
063.027  315 366 072    4276          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL
063.032  315 202 071    4277          CALL    AVV             ASSIGN VALUE TO VARIABLE
063.035  315 056 071    4278          CALL    ANT             CHECK SEPERATOR
063.040  376 000        4279          CPI     0               MUST BE SAME AS FUNCTION LIST
063.041                 4280  TXTFNA  EQU     *-1
063.042  302 205 070    4281          JNE     ERR.AC          ARG COUNT ERROR
063.045  376 020        4282          CPI     CT.PAR
063.047  302 363 062    4283          JNE     TXTF1           MORE TO ASSIGN
063.052  305            4284          PUSH    B               EXCHANGE POINTERS
063.053  343            4285          XTHL
063.054  301            4286          POP     B
```

```
063.055  345           4287      PUSH    H          SAVE CALLER POINTER
063.056  315 305 077   4288      CALL    RNT
063.061  011           4289      DB      CT.EQ      REQUIRE =
063.062  315 244 055   4290      CALL    EVAL       EVALUATE FUNCTION
063.065  315 305 077   4291      CALL    RNT
063.070  000           4292      DB      CT.FIN     REQUIRE STATEMENT END
063.071  301           4293      POP     B
063.072  303 072 076   4294      JMP     PNT        EXIT WITH NEXT TOKEN PEEKED
```

```
                                4290  **      EXP - CALCULATE EXP(X).
                                4299  *
                                4300  *       Y=EXP(X)
                                4301  *
                                4302  *       VIA:
                                4303  *
                                4304  *       X1 = X * LN(2)^-1
                                4305  *
                                4306  *       Y = 2^(INT(X1)) * 2^(FRACT(X1))
                                4307  *
                                4308  *       FRACT(X) [0,1) = P5(X)
                                4309
                                4310
063.075                         4311  EXP     EQU     *
063.075  305                   4312          PUSH    B               SAVE TEXT POINTER
063.076  072 204 042           4313          LDA     ACCX+2          (A) = SIGN
063.101  247                   4314          ANA     A
063.102  365                   4315          PUSH    PSW             SAVE RESULTS
063.103  374 305 105           4316          CM      NEG             INSURE POSITIVE
063.106  041 221 063           4317          LXI     H,EXPA
063.111  315 327 105           4318          CALL    MUL             (ACCX) = X * LN(2)^-1
063.114  315 223 073           4319          CALL    CXY             SAVE IN ACCY
063.117  315 377 074           4320          CALL    IFIX.           (DE) = INT(X1)
063.122  172                   4321          MOV     A,D
063.123  247                   4322          ANA     A
063.124  312 133 063           4323          JZ      EXP1            EXPONENT NOT TOO BIG
063.127  074                   4324          INR     A
063.130  302 136 070           4325          JNZ     ERR.OV          EXPONENT TOO BIG
063.133  325                   4326  EXP1    PUSH    D               SAVE EXP
063.134  315 040 075           4327          CALL    IFLT            FLOAT INTO ACCX
063.137  041 210 042           4328          LXI     H,ACCY
063.142  315 172 105           4329          CALL    SUB             (ACCX) = FRACT(X1)
063.145  315 177 065           4330          CALL    POLY            EVALUATE P5(X)
063.150  006                   4331          DB      6
063.151  202 014 173           4332          DB      202Q,014Q,173Q,167Q     .001877576677
063.155  003 244 111           4333          DB      003Q,244Q,111Q,172Q     .008989340083
063.161  021 125 162           4334          DB      021Q,125Q,162Q,174Q     .055826431806
063.165  152 365 172           4335          DB      152Q,365Q,172Q,176Q     .2401536170
063.171  075 271 130           4336          DB      075Q,271Q,130Q,200Q     .6931530732
063.175  377 377 177           4337          DB      377Q,377Q,177Q,200Q     .9999999250
                                4338
063.201  321                   4339          POP     D               (DE) = EXP OF 2^(INT(X1))
063.202  173                   4340          MOV     A,E             (A) = EXPONENT ADJUSTMENT
063.203  041 205 042           4341          LXI     H,ACCX+3
063.206  206                   4342          ADD     M               ADJUST EXPONENT
063.207  167                   4343          MOV     M,A
063.210  332 136 070           4344          JC      ERR.OV          OVERFLOW
063.213  361                   4345          POP     PSW             (A) = RESULTS OF INITIAL SIGN TEST
063.214  374 312 065           4346          CM      RCX             EXP(X) = 1/EXP(-X)
063.217  301                   4347          POP     B               RESTORE BC
063.220  311                   4348          RET
                                4349
063.221  035 125 134           4350  EXPA    DB      035Q,125Q,134Q,201Q     1/LN(2)
```

```
                        4352  **      LOG - CALCULATE LOG BAS E
                        4353  *
                        4354  *       Y=LOG(X)
                        4355  *
                        4356  *       VIA:
                        4357  *
                        4358  *       LOGE(X) = LOGE(2)* LOG2(X)
                        4359  *
                        4360  *       LOG2(X) = EXPONENT(X) + LOG2(MANTISSA)
                        4361  *
                        4362  *       LOG2(N) [.5,1) = P3(X)/P2(X)
                        4363
                        4364
063.225                 4365  LOG     EQU     *
063.225  305           4366          PUSH    B               SAVE TEXT POINTER
063.226  041 204 042    4367          LXI     H,ACCX+2
063.231  176           4368          MOV     A,M             (A) = SIGN
063.232  247           4369          ANA     A
063.233  372 122 070    4370          JM      ERR.IN          MUST BE > 0                     /80.01.6C/
063.236  312 136 070    4371          JZ      ERR.OV
063.241  043           4372          INX     H
063.242  136           4373          MOV     E,M
063.243  026 000       4374          MVI     D,0             (DE) = EXPONENT
063.245  325           4375          PUSH    D               SAVE
063.246  066 200       4376          MVI     M,200Q          (ACCX) = MANTISSA
063.250  315 153 065    4377          CALL    POLYQ           COMPUTE P3(X)/P2(X)
063.253  004           4378          DB      4
063.254  000 000 100    4379          DB      000Q,000Q,100Q,201Q       1.0
063.260  160 330 146    4380          DB      160Q,330Q,146Q,203Q       6.4278 42090
063.264  005 271 110    4381          DB      005Q,271Q,110Q,203Q       4.5451 70876
063.270  172 202 132    4382          DB      172Q,202Q,132Q,177Q       .35355 34252
063.274  004           4383          DB      4
063.275  314 373 114    4384          DB      314Q,373Q,114Q,203Q       4.8114 74609
063.301  221 261 141    4385          DB      221Q,261Q,141Q,203Q       6.1058 51990
063.305  106 031 271    4386          DB      106Q,031Q,271Q,204Q      -8.8626 59939
063.311  054 100 276    4387          DB      054Q,100Q,276Q,202Q      -2.0546 66719
063.315  315 223 073    4388          CALL    CXY             (ACCY) = LOG2(MANTISSA)
063.320  321           4389          POP     D               (DE) = EXPONENT
063.321  315 040 075    4390          CALL    IFLT
063.324  041 350 063    4391          LXI     H,LOGA
063.327  315 356 104    4392          CALL    ADD             REMOVE EXPONENT BIAS
063.332  041 210 042    4393          LXI     H,ACCY
063.335  315 356 104    4394          CALL    ADD             (ACCX) = EXPONENT+LOG2(MANTISSA)
063.340  041 354 063    4395          LXI     H,LOGB
063.343  315 327 105    4396          CALL    MUL             (ACCX) = LOGE(2)*LOG2(X)
063.346  301           4397          POP     B
063.347  311           4398          RET
                        4399
063.350  000 000 200    4400  LOGA    DB      000Q,000Q,200Q,207Q       -128.
063.354  014 271 130    4401  LOGB    DB      014Q,271Q,130Q,200Q       LOGE(2)
```

```
                                4403  **      SQRT - SQUARE ROOT.
                                4404  *
                                4405  *       Y=SQRT(X)
                                4406  *
                                4407  *       VIA:
                                4408  *
                                4409  *       SQRT(X) = 2^B * SQRT(X*2^(-2*B))
                                4410  *
                                4411  *       SQRT(X1) [.25,1) = P2(X)/P2(X)
                                4412
                                4413
063,360                         4414  SQR     EQU     *
063.360  305                    4415          PUSH    B
063,361  315 033 077            4416          CALL    PSHX            SAVE X
063.364  041 204 042            4417          LXI     H,ACCX+2
063,367  176                    4418          MOV     A,M             (A) = SIGN
063.370  247                    4419          ANA     A
063,371  372 122 070            4420          JM      ERR.IN          MUST BE >= 0
063.374  312 112 064            4421          JZ      SQRT3           IS ZERO
063,377  043                    4422          INX     H
064.000  176                    4423          MOV     A,M             (A) = EXPONENT
                                4424
                                4425  *       EXPONENT >= 200Q. SCALE TO 177 OR 200
                                4426
064.001  326 177                4427          SUI     177Q
064,003  037                    4428          RAR                     (A) = B (SCALE FACTOR)
064.004  365                    4429          PUSH    PSW             SAVE FACTOR
064,005  077                    4430          CMC
064.006  303 015 064            4431          JMP     SQRT2
                                4432
                                4433  *       EXPONENT < 200Q. SCLAE TO 177 OR 200
                                4434
064.011  326 177                4435  SQRT1   SUI     177Q
064,013  037                    4436          RAR                     (A) = B (SCALE FACTOR)
064.014  365                    4437          PUSH    PSW             SAVE SCALE FACTOR
064,015  076 200                4438  SQRT2   MVI     A,200Q
064.017  336 000                4439          SBI     0               (A) = 200Q OR 177Q
064,021  167                    4440          MOV     M,A             (ACCX) = SCALED VALUE
064.022  315 177 065            4441          CALL    POLY            EVALUTE POLY
064,025  005                    4442          DB      5
064.026  053 017 255            4443          DB      053Q,017Q,255Q,177Q      -.32398 73450
064,032  327 005 104            4444          DB      327Q,005Q,104Q,201Q       1.062856525
064.036  153 213 241            4445          DB      153Q,213Q,241Q,201Q      -1.4758 65807
064,042  170 366 142            4446          DB      170Q,366Q,142Q,201Q       1.546293465
064.046  362 202 141            4447          DB      362Q,202Q,141Q,176Q       .19045 21794
                                4448
064.052  361                    4449          POP     PSW             (A) = EXPONENT ADJUST
064,053  041 205 042            4450          LXI     H,ACCX+3
064.056  206                    4451          ADD     M               ADJUST EXPONENT
064,057  167                    4452          MOV     M,A
                                4453
                                4454  *       APPLY HERON'S ITTERATION ONCE.
                                4455
064,060  315 223 073            4456          CALL    CXY             ACCY  = GUESS
064.063  315 357 076            4457          CALL    POPX            ACCX = X
064,066  041 210 042            4458          LXI     H,ACCY
```

```
064.071 345           4459          PUSH    H
064.072 315 264 106   4460          CALL    DIV
064.075 341           4461          POP     H               (HL) = #ACCY
064.076 315 356 104   4462          CALL    ADD             (ACCX) = GUESS+X/GUESS
064.101 041 205 042   4463          LXI     H,ACCX+3
064.104 176           4464          MOV     A,M
064.105 326 001       4465          SUI     1               DIVIDE BY 2
064.107 167           4466          MOV     M,A
064.110 301           4467          POP     B               RESTORE (BC)
064.111 311           4468          RET
                      4469
064.112 315 357 076   4470  SQRT3   CALL    POPX            RESTORE STACK
064.115 301           4471          POP     B               RESTORE (BC)
064.116 311           4472          RET                     EXIT


                      4474  **      SINCOS - SIN AND COSIN.
                      4475  *
                      4476  *       Y=SIN(X)
                      4477  *       Y=COS(X)
                      4478  *
                      4479  *       REDUCE RANGE FROM 0 TO PI/2, APPROXIMATE WITH
                      4480  *
                      4481  *       COS(X) = P4(X)
                      4482
                      4483
064.117               4484  SIN     EQU     *
064.117 021 163 112   4485          LXI     D,NPI.2
064.122 315 352 104   4486          CALL    FPADD           SIN(X) = COS(X-PI/2)
                      4487
064.125               4488  COS     EQU     *
064.125 305           4489          PUSH    B
064.126 315 252 065   4490          CALL    PTS             PERFORM TRIG SCALEING
064.131 072 204 042   4491          LDA     ACCX+2
064.134 247           4492          ANA     A
064.135 374 305 105   4493          CM      NEG             COS(-X) = COS(X)
                      4494
                      4495  *       REDUCE RANGE TO 0<=X<=2*PI
                      4496
064.140 041 167 112   4497          LXI     H,NPI2          POINT TO -2PI
064.143 315 331 065   4498          CALL    RAR             REDUCE ARGUMENT RANGE
                      4499
                      4500  *       REDUCE RANGE TO 0<=X<=PI/2
                      4501
064.146 041 163 112   4502          LXI     H,NPI.2
064.151 315 331 065   4503          CALL    RAR
064.154 074           4504          INR     A
064.155 037           4505          RAR
064.156 062 234 064   4506          STA     COSA            (COSA) = ODD IF TO INVERT SIGN
064.161 332 175 064   4507          JC      COS1            IF < PI/2
064.164 041 163 112   4508          LXI     H,NPI.2         (X) = -(X-PI/2)
064.167 315 356 104   4509          CALL    ADD
064.172 315 305 105   4510          CALL    NEG
064.175 041 202 042   4511  COS1    LXI     H,ACCX
```

```
064.200  315 327 105  4512          CALL    MUL             (ACCX) = X**X
064.203  315 177 065  4513          CALL    POLY
064.206  005          4514          DB      5
064.207  130 035 141  4515          DB      130Q,035Q,141Q,161Q     .00002315393167
064.213  130 065 245  4516          DB      130Q,065Q,245Q,167Q     -.00138 53704 276
064.217  267 123 125  4517          DB      267Q,123Q,125Q,174Q     .04166358467
064.223  020 000 200  4518          DB      020Q,000Q,200Q,177Q     -.49999 90534
064.227  000 000 100  4519          DB      000Q,000Q,100Q,201Q     .9999999534
                      4520
                      4521  *       NEGATE SIGN OF RESULT, IF NECESSARY
                      4522
064.233  076 000      4523  COS2    MVI     A,0
064.234               4524  COSA    EQU     *-1             ODD IF TO TOGGLE SIGN
064.235  037          4525          RAR
064.236  334 305 105  4526          CC      NEG
064.241  301          4527          POP     B
064.242  311          4528          RET


                      4530  **      TAN - COMPUTE TANGENT FUNCTION.
                      4531  *
                      4532
                      4533
064.243               4534  TAN     EQU     *
064.243  315 252 065  4535          CALL    PTS             PERFORM TRIG SCALING
064.246  072 204 042  4536          LDA     ACCX+2
064.251  247          4537          ANA     A
064.252  310          4538          RZ                      TAN(0) = 0
064.253  305          4539          PUSH    B
064.254  007          4540          RLC
064.255  062 234 064  4541          STA     COSA            SET NEGATION FLAG
064.260  334 305 105  4542          CC      NEG             TAN(-X) = -TAN(X)
064.263  041 173 112  4543          LXI     H,NPI           REDUCE RANGE BY PI
064.266  315 331 065  4544          CALL    RAR             REDUCE ARGUMENT RANGE
                      4545
                      4546  *       REDUCE IT BY PI/2
                      4547
064.271  041 163 112  4548          LXI     H,NPI.2
064.274  315 331 065  4549          CALL    RAR
064.277  247          4550          ANA     A
064.300  312 321 064  4551          JZ      TAN1            WAS IN RANGE 0 - PI/2
064.303  041 234 064  4552          LXI     H,COSA
064.306  256          4553          XRA     M
064.307  167          4554          MOV     M,A             TAN(X) = -TAN(PI-X)
064.310  041 163 112  4555          LXI     H,NPI.2
064.313  315 356 104  4556          CALL    ADD
064.316  315 305 105  4557          CALL    NEG             ACCX = -(X-PI)
                      4558
                      4559  *       SCALE TO PI/4
                      4560
064.321  041 177 112  4561  TAN1    LXI     H,NPI.4
064.324  315 331 065  4562          CALL    RAR             REDUCE ARGUMENT RANGE
064.327  062 016 065  4563          STA     TANA            SAVE COUNT
064.332  247          4564          ANA     A
```

```
064.333  312 347 064  4565          JZ      TAN2
                      4566
                      4567  *       TAN(X) = 1/TAN(PI/2-X)
                      4568
064.336  041 177 112  4569          LXI     H,NPI.4
064.341  315 356 104  4570          CALL    ADD
064.344  315 305 105  4571          CALL    NEG             (ACCX) = -(X-PI/2)
                      4572
064.347  041 203 112  4573  TAN2    LXI     H,PI.4
064.352  315 264 106  4574          CALL    DIV             (ACCX) = X/(PI/4)
064.355  315 142 065  4575          CALL    XPOLYQ          COMPUTE P1(X^2)/P2(X^2)
064.360  003          4576          DB      3
064.361  000 000 100  4577          DB      000Q,000Q,100Q,201Q     1.
064.365  151 147 270  4578          DB      151Q,147Q,270Q,207Q     71.59606050
064.371  346 235 103  4579          DB      346Q,235Q,103Q,211Q     270.4672235
064.375  002          4580          DB      2
064.376  331 222 233  4581          DB      331Q,222Q,233Q,204Q     -12.55329742
065.002  124 066 152  4582          DB      124Q,066Q,152Q,210Q     212.42445758
                      4583
065.006  315 365 076  4584          CALL    POPY            (ACCY) = X
065.011  353          4585          XCHG
065.012  315 327 105  4586          CALL    MUL             X*P1/P2
065.015  076 000      4587          MVI     A,0
065.016               4588  TANA    EQU     *-1
065.017  037          4589          RAR
065.020  334 312 065  4590          CC      RCX             TAKE RECIPRICAL OF ACCX
065.023  303 233 064  4591          JMP     COS2            NEGATE RESULT, IF NECESSARY


                      4593  **      ATAN = ATAN(X)
                      4594  *
                      4595
                      4596
065.026               4597  ATN     EQU     *
065.026  305          4598          PUSH    B
065.027  072 204 042  4599          LDA     ACCX+2
065.032  007          4600          RLC
065.033  062 234 064  4601          STA     COSA            SET NEGATE FLAG
065.036  334 305 105  4602          CC      NEG             ATAN(-X) = -ATAN(X)
065.041  072 205 042  4603          LDA     ACCX+3
065.044  326 201      4604          SUI     201Q
065.046  365          4605          PUSH    PSW             SAVE RANGE
065.047  324 312 065  4606          CNC     RCX             IF VALUE > 1, TAKE RECIPROCAL
                      4607
065.052  315 142 065  4608  ATAN1   CALL    XPOLYQ          =X*P3(X^2)/P2(X^2)
065.055  003          4609          DB      3
065.056  000 000 100  4610          DB      000Q,000Q,100Q,201Q     1.
065.062  156 132 103  4611          DB      156Q,132Q,103Q,203Q     4.2095 84416
065.066  332 176 164  4612          DB      332Q,176Q,164Q,202Q     3.640485264
065.072  004          4613          DB      4
065.073  156 000 252  4614          DB      156Q,000Q,252Q,172Q     -.01049 78419 9
065.077  104 042 123  4615          DB      104Q,042Q,123Q,177Q     .32474 16032
065.103  013 340 137  4616          DB      013Q,340Q,137Q,202Q     2.996099356
065.107  332 176 164  4617          DB      332Q,176Q,164Q,202Q     3.640485163
```

```
                                4618
065.113  315 365 076  4619          CALL    POPY
065.116  353          4620          XCHG
065.117  315 327 105  4621          CALL    MUL             MULTIPLY RESULT BY X
065.122  361          4622          POP     PSW             RESTORE INVERT CODE
065.123  332 137 065  4623          JC      ATAN2           NOT INVERTED
065.126  041 163 112  4624          LXI     H,NPI.2         PI/2-ATAN(1/X)= ATAN(X)
065.131  315 356 104  4625          CALL    ADD
065.134  315 305 105  4626          CALL    NEG             ACCX = -(X-PI/2)
065.137  303 233 064  4627  ATAN2   JMP     COS2            NEGATE IF NECESSARY

                                4629  **      XPOLYQ - EVALUATE X*P(X^2)/Q(X^2)
                                4630  *
                                4631  *       ENTRY   (ACCX) = VALUE
                                4632  *               (RET) = QUOTIENT LIST, NUMERATOR FIRST
                                4633  *       EXIT    TO AFTER LIST
                                4634  *       USES    ALL
                                4635
                                4636
065.142  315 033 077  4637  XPOLYQ  CALL    PSHX            SAVE X
065.145  041 202 042  4638          LXI     H,ACCX
065.150  315 327 105  4639          CALL    MUL             X=X^2

                                4641  **      POLYQ - EVALUATE P(X)/Q(X)
                                4642  *
                                4643  *       ENTRY   (ACCX) = X
                                4644  *               (RET) = QUOTIENT LIST, NUMERATOR FIRST
                                4645  *       EXIT    TO AFTER LIST
                                4646  *       USES    ALL
                                4647
                                4648
065.153  341          4649  POLYQ   POP     H               (HL) = LIST ADDRESS
065.154  315 204 065  4650          CALL    PLY             COMPUTE DENOMINATOR
065.157  345          4651          PUSH    H               SAVE (HL)
065.160  315 033 077  4652          CALL    PSHX            SAVE QUOTIENT
065.163  341          4653          POP     H               RESTORE (HL)
065.164  315 207 065  4654          CALL    PLYQ            COMPUTE NUMERATOR
065.167  345          4655          PUSH    H               SAVE RETURN ADDRESS
065.170  315 365 076  4656          CALL    POPY            (ACCY) = DENIMINATOR
065.173  353          4657          XCHG                    (HL) = $ACCY
065.174  303 264 106  4658          JMP     DIV             DIVIDE AND RETURN
```

```
                                4660  **      POLY - EVALUATE POLYNOMIAL.
                                4661  *
                                4662  *       ENTRY   ACCX = X
                                4663  *               (RET) = COEFFICIENT LIST
                                4664  *       EXIT    TO AFTER LIST
                                4665  *       USES    ALL
                                4666
                                4667
065.177  341                   4668  POLY    POP     H               (HL) = RETURN ADDRESS
065.200  315 204 065           4669          CALL    PLY             COMPUTE
065.203  351                   4670          PCHL


                                4672  **      PLY - COMPUTE POLYNOMIAL.
                                4673  *
                                4674  *       ACCX = PN(X)
                                4675  *
                                4676  *       COMPUTE A + X(B + X(C + X(D,...))))
                                4677
                                4678
065.204  315 223 073           4679  PLY     CALL    CXY             (ACCY) = ACCX VALUE
065.207  176                   4680  PLY0    MOV     A,M             (A) = COUNT
065.210  365                   4681          PUSH    PSW
065.211  043                   4682          INX     H
065.212  353                   4683          XCHG                    (DE) = ADDRESS
065.213  315 210 073           4684          CALL    CVX             (ACCX) = D
065.216  353                   4685          XCHG                    (HL) = ADDRESS OF D
065.217  303 240 065           4686          JMP     PLY2
                                4687
065.222  365                   4688  PLY1    PUSH    PSW             SAVE COUNT
065.223  345                   4689          PUSH    H               SAVE ADDRESS
065.224  041 210 042           4690          LXI     H,ACCY
065.227  315 327 105           4691          CALL    MUL             COMPUTE X(...)
065.232  341                   4692          POP     H
065.233  345                   4693          PUSH    H
065.234  315 356 104           4694          CALL    ADD             COMPUTE A + X(...)
065.237  341                   4695          POP     H
065.240  361                   4696  PLY2    POP     PSW
065.241  043                   4697          INX     H
065.242  043                   4698          INX     H
065.243  043                   4699          INX     H
065.244  043                   4700          INX     H
065.245  075                   4701  POLY2   DCR     A
065.246  302 222 065           4702          JNZ     PLY1            IF MORE TO GO
065.251  311                   4703          RET                     DONE
```

```
                                4705  **      PTS - PERFORM TRIG SCALING.
                                4706  *
                                4707  *       PTS SCALES A VALUE INTO THE RANGE -2*PI <= X <= 2*PI
                                4708  *       ONLY IF -10*PI <= X <= 10*PI.
                                4709  *
                                4710  *       FOR VALUES WITHIN THIS RANGE, THE ADDITIVE SCALING OF THE
                                4711  *       FUNCTIONS THEMSELVES IS MORE EFFICIENT.
                                4712  *
                                4713  *       ENTRY   (ACCX) = X
                                4714  *       EXIT    (ACCX) = SCALED VALUE
                                4715  *       USES    A,F,D,E,H,L
                                4716
                                4717
065.252  072 205 042           4718  PTS     LDA     ACCX+3          (A) = EXPONENT
065.255  376 206               4719          CPI     206Q
065.257  330                   4720          RC                      DOSENT NEED IT
065.260  305                   4721          PUSH    B               SAVE (BC)
                                4722
                                4723  *       COMPUTE SCALED = X - INT(X/2*PI) * 2*PI
                                4724
065.261  315 223 073           4725          CALL    CXY             (ACCY) = X
065.264  041 167 112           4726          LXI     H,NPI2
065.267  345                   4727          PUSH    H               SAVE ADDRESS OF NPI2
065.270  315 264 106           4728          CALL    DIV
065.273  315 216 057           4729          CALL    INT             FIX
065.276  341                   4730          POP     H
065.277  315 327 105           4731          CALL    MUL
065.302  041 210 042           4732          LXI     H,ACCY
065.305  315 172 105           4733          CALL    SUB             TAKE DIFFERENCE
065.310  301                   4734          POP     B
065.311  311                   4735          RET


                                4737  **      RCX - TAKE RECIPROCAL OF (ACCX).
                                4738  *
                                4739  *       (ACCX) = 1/(ACCX)
                                4740  *
                                4741  *       ENTRY   NONE
                                4742  *       EXIT    NONE
                                4743  *       USES    ALL
                                4744
                                4745
065.312  315 223 073           4746  RCX     CALL    CXY             (ACCY) = X
065.315  021 147 112           4747          LXI     D,FP1.0
065.320  315 210 073           4748          CALL    CVX             COPY VALUE INTO ACCX
065.323  041 210 042           4749          LXI     H,ACCY
065.326  303 264 106           4750          JMP     DIV             ACCX = 1/(ACCX)
```

```
                                4752  **      RAR - REDUCE ARGUMENT RANGE.
                                4753  *
                                4754  *       RAR REDUCES THE ARGUMENT RANGE OF A VALUE BY REPEATED
                                4755  *       ADDITION WITH A NEGATIVE CONSTANT, UNTIL THE NUMBER IS
                                4756  *       SMALLER THAN ABS(CONSTANT)
                                4757  *
                                4758  *       ENTRY   (HL) = CONSTANT
                                4759  *       EXIT    (A) = ADDITION COUNT
                                4760  *               (HL) UNCHANGED
                                4761  *       USES    A,F,B,C,D,E
                                4762
                                4763
065.331  257                   4764  RAR     XRA     A
                                4765
065.332  365                   4766  RAR1    PUSH    PSW             SAVE COUNT
065.333  315 223 073           4767          CALL    CXY             SAVE VALUE IN ACCY
065.336  345                   4768          PUSH    H
065.337  315 356 104           4769          CALL    ADD             SUBTRACE
065.342  341                   4770          POP     H
065.343  072 204 042           4771          LDA     ACCX+2
065.346  247                   4772          ANA     A
065.347  372 357 065           4773          JM      RAR2            DONE
065.352  361                   4774          POP     PSW
065.353  074                   4775          INR     A
065.354  303 332 065           4776          JMP     RAR1
                                4777
065.357  315 317 100           4778  RAR2    CALL    XCY             COPY LAST VALUE INTO ACCX
065.362  361                   4779          POP     PSW
065.363  311                   4780          RET
```

```
                                4784  **      ICL - INPUT COMMAND LINE.
                                4785  *
                                4786  *       ICL INPUTS A COMMAND INTO *LINE*.
                                4787  *
                                4788  *       KEYWORDS ARE EXPANDED UNLESS
                                4789  *               1) THEY FOLLOW A 'REM' KEYWORD
                                4790  *               2) THEY APPEAR IN QUOTES
                                4791  *
                                4792  *       ICL MAKES (AND ENFORCES) CERTAIN ASSUMTIONS ABOUT
                                4793  *       LEXICAL SYNTAX
                                4794  *               1) A PAIR OF ALPHA CHARACTERS MAY ONLY APPEAR IN A
                                4795  *                   KEYWORD, OR WITHIN A 'REM' OR QUOTED STRING.
                                4796  *               2) ALL KEYWORDS ARE UNIQUE WITHIN THE 1ST 3 CHARACTERS.
                                4797  *
                                4798  *       IF A CTL-C IS ENTERED, ICL EXITS WITH NO TEXT.
                                4799  *
                                4800  *       ENTRY   (A) = PROMPT CHARACTER
                                4801  *       EXIT    LINE READ
                                4802  *               'C' SET IF CTL-C ENTERED, NO TEXT.
                                4803  *               'C' CLEAR IF HAVE LINE
                                4804  *                'Z' SET IF NO ERROR IN LINE
                                4805  *       USES    ALL
                                4806
                                4807
065.364                         4808  ICL     EQU     *
065.364  041 266 112           4809          LXI     H,LINE+1
065.367  315 075 077           4810          CALL    RIL             READ INPUT LINE
065.372  330                   4811          RC                      CTL-C
                                4812
                                4813  **      ICL. - ENTRY FOR PRE-READ LINE
                                4814  *
                                4815  *               (HL) = LINE FWA (BUFFER ADDRESS +1)
                                4816
065.373  053                   4817  ICL.    DCX     H               PRE-DECREMENT H
065.374  345                   4818          PUSH    H               SAVE BUFFER FWA
065.375  104                   4819          MOV     B,H
065.376  115                   4820          MOV     C,L             (BC) = TO, (HL) = FROM
065.377  013                   4821          DCX     B               PREDECREMENT (BC)
                                4822
                                4823  *       COPY ANOTHER CHARACTER
                                4824
066.000  003                   4825  ICL1    INX     B
066.001  043                   4826  ICL1.5  INX     H
066.002  176                   4827          MOV     A,M
066.003  002                   4828          STAX    B               COPY CHARACTER
066.004  127                   4829          MOV     D,A             (D) = 0 IFF END OF LINE
066.005  247                   4830          ANA     A
066.006  312 234 066           4831          JZ      ICL10           ALL DONE
066.011  315 045 112           4832          CALL    $MCU            MAP CHARACTER TO UPPER CASE
066.014  376 042               4833          CPI     '"'
066.016  312 176 066           4834          JE      ICL7            GOT QUOTES
066.021  376 101               4835          CPI     'A'
066.023  332 000 066           4836          JC      ICL1            NOT ALPHA
066.026  376 133               4837          CPI     'Z'+1
066.030  322 000 066           4838          JNC     ICL1            NOT ALPHA
                                4839
```

```
                                4840  *       HAVE AN ALPHA CHARACTER. SEE IF WE HAVE 2 IN A ROW
                                4841
066.033  043                   4842          INX     H
066.034  176                   4843          MOV     A,M
066.035  053                   4844          DCX     H
066.036  315 045 112           4845          CALL    $MCU            MAP CHARACTER TO UPPER CASE
066.041  376 101               4846          CPI     'A'
066.043  332 000 066           4847          JC      ICL1            NOT TWO ALPHA
066.046  376 133               4848          CPI     'Z'+1
066.050  322 000 066           4849          JNC     ICL1            NOT TWO ALPHA
                                4850
                                4851  *       HAVE TWO ALPHA IN A ROW. MUST BE A KEYWORD. FIND IT IN LIST
                                4852
066.053  021 240 066           4853          LXI     D,KEYTAB
066.056  345                   4854  ICL2    PUSH    H               SAVE *FROM* ADDRESS
066.057  032                   4855          LDAX    D
066.060  002                   4856          STAX    B               ASSUME IS THIS KEYWORD
066.061  023                   4857          INX     D
066.062  032                   4858  ICL3    LDAX    D               COMPARE LINE AGAINST TABLE ENTRY
066.063  247                   4859          ANA     A
066.064  372 122 066           4860          JM      ICL5            GOT MATCH
066.067  353                   4861          XCHG                    (DE) = LINE, (HL) = KEYTAB ADDR
066.070  032                   4862          LDAX    D               (A) = CHARACTER
066.071  315 045 112           4863          CALL    $MCU            MAP TO UPPER
066.074  276                   4864          CMP     M
066.075  353                   4865          XCHG                    RESTORE (DE) AND (HL)
066.076  023                   4866          INX     D
066.077  043                   4867          INX     H
066.100  312 062 066           4868          JE      ICL3            STILL MATCHING
066.103  033                   4869          DCX     D               PRE-DECREMENT KEYTAB POINTER
                                4870
                                4871  *       NOT THIS KEYWORD, SCAN TO NEXT ONE AND RETRY
                                4872
066.104  023                   4873  ICL4    INX     D
066.105  032                   4874          LDAX    D
066.106  247                   4875          ANA     A
066.107  362 104 066           4876          JP      ICL4            NOT AT START OF NEXT
066.112  341                   4877          POP     H               (HL) = FWA OF UNKNOWN KEYWORD
066.113  074                   4878          INR     A               SEE IF AT END OF LIST
066.114  302 056 066           4879          JNZ     ICL2            NOT AT END OF LIST
066.117  303 216 066           4880          JMP     ICL8            INVALID KEYWORD
                                4881
                                4882  *       HAVE FOUND THE KEYWORD. SEE IF '(' FOLLOWS
                                4883  *
                                4884  *       (HL) POINTS JUST PAST THE KEYWORD ON THE LINE
                                4885
066.122  321                   4886  ICL5    POP     D               DISCARD KEYWORD FWA
066.123  012                   4887          LDAX    B               (A) = KEYWORD VALUE
066.124  003                   4888          INX     B
066.125  376 320               4889          CPI     CT.FCN
066.127  322 163 066           4890          JNC     ICL6            IS FUNCTION
066.132  365                   4891          PUSH    PSW             SAVE CODE
066.133  176                   4892          MOV     A,M
066.134  376 040               4893          CPI     ' '
066.136  302 142 066           4894          JNE     ICL5.5          NO BLANK FOLLOWING
066.141  043                   4895          INX     H
```

```
066.142 361           4896  ICL5.5  POP     PSW             (A) = KEYWORD CODE
066.143 026 000       4897          MVI     D,0             NO ERROR WHEN REACH END OF LINE
066.145 376 242       4898          CPI     CT.REM
066.147 312 223 066   4899          JE      ICL9            COPY REST OF LINE
066.152 376 251       4900          CPI     CT.DAT
066.154 312 223 066   4901          JE      ICL9            COPY REST OF LINE
066.157 053           4902          DCX     H               PRESET (HL) FOR INCREMENT
066.160 303 001 066   4903          JMP     ICL1.5
                      4904
                      4905  *       IS FUNCTION, REQUIRE '('
                      4906
066.163 315 330 111   4907  ICL6    CALL    $SOB            SKIP OVER BLANKS
066.166 376 050       4908          CPI     '('
066.170 312 001 066   4909          JE      ICL1.5          OK, GOBBLE '('
066.173 303 216 066   4910          JMP     ICL8            ERROR
                      4911
                      4912  *       GOT QUOTE, SCAN TO CLOSE QUOTE
                      4913
066.176 003           4914  ICL7    INX     B
066.177 043           4915          INX     H
066.200 176           4916          MOV     A,M
066.201 002           4917          STAX    B               STORE CHARACTER
066.202 247           4918          ANA     A
066.203 312 216 066   4919          JZ      ICL8            ERROR
066.206 376 042       4920          CPI     '"'
066.210 302 176 066   4921          JNE     ICL7            NOT CLOSE QUOTE
066.213 303 000 066   4922          JMP     ICL1            GOT CLOSE
                      4923
                      4924  *       ERROR IN LINE, FLAG IT, AND COPY THE REST VERBATIM
                      4925  *       (A) = 0
                      4926
066.216 076 212       4927  ICL8    MVI     A,CT.SYE
066.220 002           4928          STAX    B               SET ERROR
066.221 003           4929          INX     B
066.222 127           4930          MOV     D,A             (D) <> 0 INDICATING ERROR
                      4931
                      4932  *       COPY REST OF LINE VERBATIM
                      4933  *       (D) <> 0 IF ERROR
                      4934
066.223 176           4935  ICL9    MOV     A,M
066.224 002           4936          STAX    B
066.225 043           4937          INX     H
066.226 003           4938          INX     B
066.227 247           4939          ANA     A
066.230 302 223 066   4940          JNZ     ICL9            NOT DONE
066.233 013           4941          DCX     B
                      4942
                      4943  *       ALL DONE.
                      4944  *
                      4945  *       (BC) = LINE LWA
                      4946  *       (D) <> 0 IFF ERROR
                      4947
066.234 341           4948  ICL10   POP     H               (HL) = FWA
066.235 172           4949          MOV     A,D             (A) = ERROR FLAG
066.236 247           4950          ANA     A
066.237 311           4951          RET
```

```
                                4953  **      KEYTAB - KEYWORD TABLE.
                                4954  *
                                4955
066,240                         4956  KEYTAB  EQU     *
066,240  320 101 102           4957          DB      CT.ABS,'ABS'
066,244  310 101 116           4958          DB      CT.AND,'AND'
066,250  350 101 123           4959          DB      CT.ASC,'ASC'
066,254  311 101 123           4960          DB      CT.AS,'AS'
066,257  321 101 124           4961          DB      CT.ATN,'ATN'
066,263  200 102 125           4962          DB      CT.BLD,'BUILD'
066,271  201 102 131           4963          DB      CT.BYE,'BYE'
066,275  213 103 110           4964          DB      CT.CHA,'CHAIN'
066,303  322 103 110           4965          DB      CT.CHR,'CHR$'
066,310  323 103 111           4966          DB      CT.CIN,'CIN'
066,314  214 103 114           4967          DB      CT.CLR,'CLEAR'
066,322  215 103 114           4968          DB      CT.CLO,'CLOSE'
066,330  216 103 116           4969          DB      CT.CTL,'CNTRL'
066,336  202 103 117           4970          DB      CT.CNT,'CONTINUE'
066,347  324 103 117           4971          DB      CT.COS,'COS'
066,353  251 104 101           4972          DB      CT.DAT,'DATA'
066,360  252 104 105           4973          DB      CT.DEF,'DEF'
066,364  203 104 105           4974          DB      CT.DEL,'DELETE'
066,373  217 104 111           4975          DB      CT.DIM,'DIM'
066,377  253 105 116           4976          DB      CT.END,'END'
067,003  325 105 130           4977          DB      CT.EXP,'EXP'
067,007  312 106 111           4978          DB      CT.FIL,'FILE'
067,014  220 106 116           4979          DB      CT.FN,'FN'
067,017  221 106 117           4980          DB      CT.FOR,'FOR'
067,023  223 106 122           4981          DB      CT.FRZ,'FREEZE'         MUST APPEAR BEFORE 'FREE'
067,032  222 106 122           4982          DB      CT.FRE,'FREE'
067,037  224 107 117           4983          DB      CT.GOS,'GOSUB'
067,045  225 107 117           4984          DB      CT.GOT,'GOTO'
067,052  226 111 106           4985          DB      CT.IF,'IF'
067,055  254 111 116           4986          DB      CT.INP,'INPUT'
067,063  326 111 116           4987          DB      CT.INT,'INT'
067,067  352 114 105           4988          DB      CT.LEN,'LEN'
067,073  351 114 105           4989          DB      CT.LEF,'LEFT$'
067,101  227 114 105           4990          DB      CT.LET,'LET'
067,105  250 114 111           4991          DB      CT.LIN,'LINE'
067,112  204 114 111           4992          DB      CT.LIS,'LIST'
067,117  230 114 117           4993          DB      CT.LCK,'LOCK'
067,124  327 114 116           4994          DB      CT.LNO,'LNO'
067,130  330 114 117           4995          DB      CT.LOG,'LOG'
067,134  353 115 101           4996          DB      CT.MAT,'MATCH'
067,142  331 115 101           4997          DB      CT.MAX,'MAX'
067,146  354 115 111           4998          DB      CT.MID,'MID$'
067,153  332 115 111           4999          DB      CT.MIN,'MIN'
067,157  231 116 105           5000          DB      CT.NXT,'NEXT'
067,164  314 116 117           5001          DB      CT.NOT,'NOT'
067,170  232 117 114           5002          DB      CT.OLD,'OLD'
067,174  233 117 116           5003          DB      CT.ON,'ON'
067,177  234 117 120           5004          DB      CT.OPE,'OPEN'
067,204  315 117 122           5005          DB      CT.OR,'OR'
067,207  235 117 125           5006          DB      CT.OUT,'OUT'
067,213  333 120 101           5007          DB      CT.PAD,'PAD'
067,217  236 120 101           5008          DB      CT.PAU,'PAUSE'
```

```
067.225  334 120 105  5009          DB      CT.PEK,'PEEK'
067.232  335 120 111  5010          DB      CT.PIN,'PIN'
067.236  237 120 117  5011          DB      CT.POK,'POKE'
067.243  336 120 117  5012          DB      CT.POS,'POS'
067.247  240 120 122  5013          DB      CT.PRT,'PRINT'
067.255  241 122 105  5014          DB      CT.REA,'READ'
067.262  242 122 105  5015          DB      CT.REM,'REM'
067.266  205 122 105  5016          DB      CT.REP,'REPLACE'
067.276  243 122 105  5017          DB      CT.RES,'RESTORE'
067.306  244 122 105  5018          DB      CT.RET,'RETURN'
067.315  355 122 111  5019          DB      CT.RIG,'RIGHT$'
067.324  337 122 116  5020          DB      CT.RND,'RND'
067.330  206 122 125  5021          DB      CT.RUN,'RUN'
067.334  207 123 101  5022          DB      CT.SAV,'SAVE'
067.341  210 123 103  5023          DB      CT.SCR,'SCRATCH'
067.351  340 123 105  5024          DB      CT.SEG,'SEG'
067.355  341 123 107  5025          DB      CT.SGN,'SGN'
067.361  342 123 111  5026          DB      CT.SIN,'SIN'
067.365  343 123 120  5027          DB      CT.SPC,'SPC'
067.371  344 123 121  5028          DB      CT.SQR,'SQR'
067.375  345 123 124  5029          DB      CT.STR,'STR$'
070.002  211 123 124  5030          DB      CT.STE,'STEP'
070.007  255 123 124  5031          DB      CT.STP,'STOP'
070.014  346 124 101  5032          DB      CT.TAB,'TAB'
070.020  347 124 101  5033          DB      CT.TAN,'TAN'
070.024  316 124 110  5034          DB      CT.THN,'THEN'
070.031  317 124 117  5035          DB      CT.TO,'TO'
070.034  245 125 116  5036          DB      CT.UNF,'UNFREEZE'
070.045  246 125 116  5037          DB      CT.UNL,'UNLOCK'
070.054  247 125 116  5038          DB      CT.UNS,'UNSAVE'
070.063  356 126 101  5039          DB      CT.VAL,'VAL'
070.067  313 127 122  5040          DB      CT.WRI,'WRITE'
070.075  212 007 052  5041          DB      CT.SYE,BELL,'*ERR* '    HERE FOR LISTING VIA *EKA*, CANNOT BE MATCHED
070.105  377          5042          DB      377Q                    377Q = END OF TABLE
```

```
                        5046  **      ERROR PROCESSING.
                        5047  *
                        5048  *       THESE ERROR PROCESSORS ARE ENTERED WHEN AN ERROR IS DETECTED.
                        5049  *
                        5050  *       SINCE ALL TRACK OF CONTROL HAS BEEN LOST, EXECUTING CANNOT
                        5051  *       BE RESUMED.
                        5052  *
                        5053  *       THE USER MAY DISPLAY VARIABLES WHEN AN ERROR OCCURS, BUT MAY
                        5054  *       NOT 'CONTINUE'.
                        5055
                        5056
                        5057
                        5058
                        5059
070.106  076 200        5060  ERR.CC  MVI     A,BEC.CC                CONTROL-C
070.110  001            5061          DB      MI.LXIB
                        5062
070.111  076 201        5063  ERR.CB  MVI     A,BEC.CB                CTL-B
070.113  001            5064          DB      MI.LXIB
                        5065
070.114  076 202        5066  ERR.DE  MVI     A,BEC.DE                DATA EXHAUSTED
070.116  001            5067          DB      MI.LXIB
                        5068
070.117  076 203        5069  ERR.DO  MVI     A,BEC.DO                /0
070.121  001            5070          DB      MI.LXIB
                        5071
070.122  076 204        5072  ERR.IN  MVI     A,BEC.IN                ILLEGAL NUMBER
070.124  001            5073          DB      MI.LXIB
                        5074
070.125  076 205        5075  ERR.IU  MVI     A,BEC.IU                ILLEGAL USAGE
070.127  001            5076          DB      MI.LXIB
                        5077
070.130  076 206        5078  ERR.LK  MVI     A,BEC.LK                DATA LOCK ENGAGED
070.132  001            5079          DB      MI.LXIB
                        5080
070.133  076 207        5081  ERR.NV  MVI     A,BEC.NV                NEXT VARIABLE MISSING
070.135  001            5082          DB      MI.LXIB
                        5083
070.136  076 210        5084  ERR.OV  MVI     A,BEC.OV                OVERFLOW
070.140  001            5085          DB      MI.LXIB
                        5086
070.141  076 211        5087  ERR.RE  MVI     A,BEC.RE                RETURN ERROR
070.143  001            5088          DB      MI.LXIB
                        5089
070.144  076 212        5090  ERR.SL  MVI     A,BEC.SL                STRING LENGTH
070.146  001            5091          DB      MI.LXIB
                        5092
070.147  076 213        5093  ERR.SN  MVI     A,BEC.SN                STATEMENT NUMBER
070.151  001            5094          DB      MI.LXIB
                        5095
070.152  076 214        5096  ERR.SY  MVI     A,BEC.SY                SYNTAX ERROR
070.154  001            5097          DB      MI.LXIB
                        5098
070.155  076 215        5099  ERR.TC  MVI     A,BEC.TC                TYPE CONFLICH
070.157  001            5100          DB      MI.LXIB
                        5101
```

```
070.160  076 216        5102  ERR.TO  MVI     A,BEC.TO                TABLE OVERFLOW
070.162  001            5103          DB      MI.LXIB
                        5104
070.163  076 217        5105  ERR.SR  MVI     A,BEC.SR                SUBSCRIPT RANGE
070.165  001            5106          DB      MI.LXIB
                        5107
070.166  076 220        5108  ERR.SC  MVI     A,BEC.SC                SUBSCRIPT COUNT
070.170  001            5109          DB      MI.LXIB
                        5110
070.171  076 221        5111  ERR.ND  MVI     A,BEC.ND                NOT DIMENSIONED
070.173  001            5112          DB      MI.LXIB
                        5113
070.174  076 222        5114  ERR.IC  MVI     A,BEC.IC                ILLEGAL CHARACTER
070.176  001            5115          DB      MI.LXIB
                        5116
070.177  076 226        5117  ERR.FAE MVI     A,BEC.FAE       FILE ALREADY EXISTS
070.201  001            5118          DB      MI.LXIB
                        5119
070.202  076 227        5120  ERR.ILF MVI     A,BEC.ILF       ILLEGAL FILE NAME
070.204  001            5121          DB      MI.LXIB
                        5122
070.205  076 230        5123  ERR.AC  MVI     A,BEC.AC
070.207  001            5124          DB      MI.LXIB                 ARG COUNT
                        5125
070.210  076 231        5126  ERR.FNO MVI     A,BEC.FNO               FILE NOT OPEN
070.212  001            5127          DB      MI.LXIB
                        5128
070.213  076 001        5129  ERR.EOF MVI     A,EC.EOF                END OF FILE
070.215  001            5130          DB      MI.LXIB
                        5131
070.216  076 223        5132  ERR.UD  MVI     A,BEC.UD                UNDEFINED FUNCTION
070.220  001            5133          DB      MI.LXIB                                         /80.01.GC/
                        5134
070.221  076 233        5135  ERR.CIU MVI     A,BEC.CIU               CHANNEL IN USE          /80.01.GC/
                        5136
070.223                 5137  $ERROR  EQU     *
070.223                 5138  $FERROR EQU     *
                        5139
070.223  365            5140          PUSH    PSW                     SAVE ERROR CODE
070.224  041 061 112    5141          LXI     H,MTABIND+MT.LEN        (HL) = #LENGTH OF TXTTAB
070.227  136            5142          MOV     E,M
070.230  043            5143          INX     H
070.231  176            5144          MOV     A,M                     (AE) = LENGTH OF TABLE
070.232  247            5145          ANA     A
070.233  302 244 070    5146          JNZ     ERROR1          TABLE LENGTH > 3
070.236  173            5147          MOV     A,E
070.237  376 004        5148          CPI     4
070.241  334 320 077    5149          CC      SCRA            TABLE LENGTH < 3
                        5150
070.244                 5151  ERROR1  EQU     *
                        5152  *       CALL    FOP             MAKE OVL RESIDENT
                        5153  *       CALL    CLF             CLEAR FILE OPERATIONS
070.244  377 007        5154          DB      SYSCALL+.CLRCO  CLEAR CONSOLE
070.246  315 136 031    5155          CALL    $TYPTX
070.251  012 007 041    5156          DB      NL,BELL,'! ERROR -',' '+200Q
                        5157
```

```
                          5158  *       TYPE MESSAGE
                          5159
     070.265  303 063 075  5160          JMP     ILM             ISSUE LINE MESSAGE
```

```
                                5163  **      MTL - MANAGE TEXT LINE.
                                5164  *
                                5165  *       MTL IS CALLED TI INSERT/REPLACE/DELETE A TEXT LINE FROM
                                5166  *       THE TEXT BUFFER.
                                5167  *
                                5168  *       ENTRY   *LINE* = TEXT LINE
                                5169  *       EXIT    LINE INSERTED/DELETED/REPLACED
                                5170  *               'CI' FLAG SET
                                5171  *       USES    ALL
                                5172
                                5173
 070.270                        5174  MTL     EQU     *
 070.270  315 313 075           5175          CALL    LFC             CHECK FOR DATA LOCK
 070.273  041 265 112           5176          LXI     H,LINE          CRACK NUMBER FROM LINE
 070.276  315 171 111           5177          CALL    DDN             DECODE DECIMAL NUMBER
 070.301  315 206 072           5178          CALL    CLN             CHECK FOR LEGAL NUMBER
                                5179
                                5180  *       DELETE LEADING BLANKS.
                                5181
 070.304  053                  5182  MTL0    DCX     H
 070.305  076 040               5183          MVI     A,' '
 070.307  043                  5184  MTL1    INX     H               SKIP LEADING BLANKS
 070.310  276                  5185          CMP     M
 070.311  312 307 070           5186          JE      MTL1            STILL BLANKS
 070.314  315 273 111           5187          CALL    $CLL            COMPUTE LINE LENGTH
 070.317  075                  5188          DCR     A               REMOVE END COUNT
 070.320  312 325 070           5189          JZ      MTL1.5          AM TO DELETE
 070.323  306 003               5190          ADI     3               LINE NUMBER + END-OF-LINE
 070.325  117                  5191  MTL1.5  MOV     C,A             (C) = NEW LENGTH
 070.326  053                  5192          DCX     H
 070.327  162                  5193          MOV     M,D
 070.330  053                  5194          DCX     H
 070.331  163                  5195          MOV     M,E
 070.332  345                  5196          PUSH    H               SAVE 'FROM' ADDRESS
 070.333  052 057 112           5197          LHLD    TXTTAB+MT.FWA
 070.336  345                  5198          PUSH    H
 070.337  315 242 074           5199          CALL    FLN             FIND LINE BY NUMBER
 070.342  006 000               5200          MVI     B,0             (B) = OLD LENGTH
 070.344  332 361 070           5201          JC      MTL2            IS INSERT
 070.347  043                  5202          INX     H
 070.350  043                  5203          INX     H
 070.351  315 273 111           5204          CALL    $CLL
 070.354  306 002               5205          ADI     2
 070.356  107                  5206          MOV     B,A             (B) = OLD LENGTH
 070.357  053                  5207          DCX     H
 070.360  053                  5208          DCX     H               (HL) = ADDRESS TO INSERT
 070.361  321                  5209  MTL2    POP     D               (DE) = TABLE FWA
 070.362  175                  5210          MOV     A,L
 070.363  223                  5211          SUB     E
 070.364  157                  5212          MOV     L,A
 070.365  174                  5213          MOV     A,H
 070.366  232                  5214          SBB     D
 070.367  147                  5215          MOV     H,A             (HL) = INDEX
 070.370  171                  5216          MOV     A,C             (A) = LEW LENGTH
 070.371  220                  5217          SUB     B               (A) = NEW LENGTH - OLD
 070.372  137                  5218          MOV     E,A
```

```
071.373  237          5219          SBB     A
071.374  127          5220          MOV     D,A             (DE) = NEEDED BYTES COUNT
071.375  315 213 104  5221          CALL    $IBT            MAKE OR DESTROY ROOM
071.000  057 112      5222          DW      TXTTAB+1        TABLE POINTER
071.002  353          5223          XCHG
071.003  052 057 112  5224          LHLD    TXTTAB+MT.FWA
071.006  031          5225          DAD     D               (HL) = *TO* ADDRESS
071.007  321          5226          POP     D               (DE) = *FROM* ADDRSS
071.010  006 000      5227          MVI     B,0             (BC) = NEW LENGTH
071.012  303 252 030  5228          JMP     $MOVE           COPY TEXT INTO BUFFER AND RETURN


                      5230  **      MOV - MOVE A BLOCK OF DATA.
                      5231  *
                      5232  *       MOV MOVES A BLOCK OF DATA IN MEMORY.
                      5233  *
                      5234  *       ENTRY   (DE) = FROM
                      5235  *               (HL) = TO
                      5236  *               (A) = COUNT
                      5237  *       EXIT    MOVED
                      5238  *               (DE) = FROM + COUNT
                      5239  *               (HL) = TO + COUNT
                      5240  *       USES    A,F
                      5241
                      5242
071.015  305          5243  MOV     PUSH    B
071.016  117          5244          MOV     C,A
071.017  006 000      5245          MVI     B,0
071.021  315 252 030  5246          CALL    $MOVE
071.024  301          5247          POP     B
071.025  311          5248          RET
```

```
                                5252  **      AMB - ALLOCATE MEMORY BYTES.
                                5253  *
                                5254  *       AMB ALLOCATES A BLOCK OF MEMORY TO THE END OF A TABLE.
                                5255  *       AND RETURNS THE FWA OF THE BLOCK.
                                5256  *
                                5257  *       ENTRY   (DE) = TABLE ADDRESS+1
                                5258  *               (HL) = BYTES WANTED
                                5259  *       EXIT    (DE) = TABLE ADDRESS+1
                                5260  *               (HL) = FWA (ABS) OF BLOCK
                                5261  *       USES    A,F,H,L
                                5262
                                5263
071.026                         5264  AMB     EQU     *
071.026  345                    5265          PUSH    H               SAVE COUNT
071.027  325                    5266          PUSH    D               SAVE TABLE ADDRESS
071.030  023                    5267          INX     D
071.031  023                    5268          INX     D
071.032  032                    5269          LDAX    D
071.033  157                    5270          MOV     L,A             (HL) = TABLE LENGTH
071.034  023                    5271          INX     D
071.035  032                    5272          LDAX    D
071.036  321                    5273          POP     D
071.037  147                    5274          MOV     H,A
071.040  343                    5275          XTHL                    (HL) = COUNT
071.041  315 244 103            5276          CALL    $ATS            ALLOCATE SPACE
071.044  341                    5277          POP     H               (HL) = ORIGINAL LENGTH
071.045  032                    5278          LDAX    D
071.046  205                    5279          ADD     L
071.047  157                    5280          MOV     L,A
071.050  023                    5281          INX     D
071.051  032                    5282          LDAX    D
071.052  214                    5283          ADC     H
071.053  147                    5284          MOV     H,A             (HL) = FWA OF BLOCK
071.054  033                    5285          DCX     D
071.055  311                    5286          RET


                                5288  **      ANT - ACCEPT NEXT TOKEN.
                                5289  *
                                5290  *       ANT ACCEPTS THE NEXT TEXT TOKEN.
                                5291  *
                                5292  *       ENTRY   (BC) = TEXT POINTER
                                5293  *       EXIT    (A) = TYPE
                                5294  *               (DE) = INDEX (IF VARIABLE)
                                5295  *       USES    A,F, (D,E IF VARIABLE)
                                5296
                                5297
071.056  315 072 076            5298  ANT     CALL    PNT             PEEK AT NEXT TOKEN
071.061  365                    5299          PUSH    PSW             SAVE TYPE
000.000                         5300          ERRNZ   MI.NOP
071.062  257                    5301          XRA     A
071.063  062 073 076            5302          STA     PNTA            CLEAR TYPE
071.066  303 130 076            5303          JMP     PNT1            CLEAR 'TOKEN ALREADY READ' FLAG
```

```
                                5305  **      ATP - ADJUST TABLE POINTERS.
                                5306  *
                                5307  *       $ATP IS CALLED BY THE MANAGED TABLE PACKAGE WHENEVER THE TABLES
                                5308  *       HAVE BEEN SHUFFLED. $ATP IS TO ADJUST ANY ABS POINTERS THAT MAY
                                5309  *       EXIST.
                                5310  *
                                5311  *       THE ONLY ABS POINTES ARE THE ONES IN THE FILE BUFFERS IN
                                5312  *       THE FILTAB TABLE.
                                5313  *
                                5314  *       SINCE THE FILE BUFFER IMMEDIATELY FOLLOWS THE FILE BLOCK, THE
                                5315  *       DISPLACEMENT FOR THE TABLE CAN BE COMPUTED  BY SUBTRACTING THE
                                5316  *       OLD BUFFER FWA (IN FB.FWA) FROM THE NEW ONE (FILTAB ENTRY + FB.NAM +
                                5317  *       FB.NAML)
                                5318  *
                                5319  *       NOTE THAT THE LAST BUFFER IN THE TABLE MAY NOT HAVE IT'S POINTERS SETUP
                                5320  *       CORRECTLY, IN WHICH CASE THE GARBAGE THERE JUST GETS STIRED UP A LITTLE.
                                5321  *
                                5322  *       ENTRY   NONE
                                5323  *       EXIT    NONE
                                5324  *       USES    ALL
                                5325
                                5326
071.071  052 226 042           5327  $ATP    LHLD    FBUFAD          (HL) = OLD FILTAB MT.FWA
071.074  353                   5328          XCHG
071.075  052 122 112           5329          LHLD    FILTAB+MT.FWA   (HL) = NEW FILTAB MT.FWA
071.100  042 226 042           5330          SHLD    FBUFAD          SAVE FOR NEXT TIME
071.103  175                   5331          MOV     A,L
071.104  223                   5332          SUB     E
071.105  137                   5333          MOV     E,A             (DE) = TABLE DISPLACEMENT
071.106  174                   5334          MOV     A,H
071.107  232                   5335          SBB     D
071.110  127                   5336          MOV     D,A
071.111  006 005               5337          MVI     B,CHANMAX       (B) = TABLES TO ADJUST
071.113  041 265 042           5338          LXI     H,FBLIST+FBENL+FB.FWA   START AT FIRST USER BLOCK
071.116  016 004               5339  ATP1    MVI     C,4             4 ADDRESSES IN EACH BLOCK
071.120  176                   5340  ATP2    MOV     A,M             RELOCATE ADDRESS
071.121  203                   5341          ADD     E
071.122  167                   5342          MOV     M,A
071.123  043                   5343          INX     H
071.124  176                   5344          MOV     A,M
071.125  212                   5345          ADC     D
071.126  167                   5346          MOV     M,A
071.127  043                   5347          INX     H
071.130  015                   5348          DCR     C
071.131  302 120 071           5349          JNZ     ATP2            RELOCATE ALL 4 ADDRESSES
071.134  076 023               5350          MVI     A,FBENL-8
071.136  315 101 030           5351          CALL    $DADA.          POINT TO NEXT BLOCK
071.141  005                   5352          DCR     B
071.142  302 116 071           5353          JNZ     ATP1            RELOCATE ALL BLOCKS
071.145  311                   5354          RET                     EXIT
```

```
                                5356  **      AYS - ASK 'ARE YOU SURE?'
                                5357  *
                                5358  *       AYS ASKS THE USER IF HE IS SURE. A LINE LINE ANSWER IS
                                5359  *       RECEIVED, AND ITS FIRST CHARACTER IS CHECKED.
                                5360  *
                                5361  *       ENTRY   NONE
                                5362  *       EXIT    'Z' SET IF REPLY STARTED WITH 'Y'
                                5363  *               (BC) = $ZERO
                                5364  *       USES    ALL
                                5365
                                5366
071.146  315 136 031            5367  AYS     CALL    $TYPTX
071.151  007 123 145            5368          DB      BELL,'Sure','?'+200Q
071.157  041 265 112            5369          LXI     H,LINE
071.162  315 075 077            5370          CALL    RIL
071.165  332 106 070            5371          JC      ERR.CC          CTL-C STRUCK
071.170  176                    5372          MOV     A,M             (A) = REPLY
071.171  315 045 112            5373          CALL    $MCU
071.174  376 131                5374          CPI     'Y'
071.176  001 345 114            5375          LXI     B,ZERO          POINT TO END OF LINE
071.201  311                    5376          RET


                                5378  **      AVV - ASSIGN VALUE TO VARIABLE.
                                5379  *
                                5380  *       AVV ASSIGNS THE VALUE IN (ACCX) TO A VARIABLE POINTED TO
                                5381  *       BY      (DE).
                                5382  *
                                5383  *       IF THE TYPES DO NOT MATCH, FLAG AN ERROR.
                                5384  *
                                5385  *
                                5386  *       ENTRY   (ACCX) = VALUE
                                5387  *               (A) = TARGET TYPE
                                5388  *               (DE) = TARGET POINTER
                                5389  *       EXIT    TO 'RET' IF OK
                                5390  *               TO ERR,TC IF MISMATCH.
                                5391  *       USES    A,F,D,E
                                5392
                                5393
071.202  345                    5394  AVV     PUSH    H               SAVE (HL)
071.203  147                    5395          MOV     H,A
                                5396
                                5397  *       DETERMINE ABSOLUTE ADDRESS OF TARGET.
                                5398
071.204  072 201 042            5399          LDA     ACCX-1
071.207  057                    5400          CMA
071.210  254                    5401          XRA     H
071.211  346 001                5402          ANI     CF.STR
071.213  312 155 070            5403          JZ      ERR.TC          MISMATCH
071.216  244                    5404          ANA     H
071.217  312 240 073            5405          JZ      CXV.
                                5406
                                5407  *       HAVE STRING
                                5408  *       (DE) = ADDRESS OF BLOCK
```

```
                                5409
071,222  315 000 073   5410  AVV1    CALL    CSI             (DE) = INDEX INTO SYMBOL
071,225  325           5411          PUSH    D               SAVE 'TO' DESCRIPTOR ADDRESS
071,226  315 366 072   5412          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL
071,231  315 315 074   5413          CALL    FSE             (HL) = TO ABS, (A) = TO LEN
071,234  127           5414          MOV     D,A
071,235  072 202 042   5415          LDA     ACCX
071,240  222           5416          SUB     D               (A) = NEWLEN-OLDLEN
071,241  137           5417          MOV     E,A
071,242  237           5418          SBB     A
071,243  127           5419          MOV     D,A             (DE) = COUNT CHANGE
071,244  325           5420          PUSH    D               SAVE TABLE DELTA
071,245  353           5421          XCHG                    (DE) = 'TO' ABS ADDRESS
071,246  052 110 112   5422          LHLD    STRTAB+MT.FWA
071,251  173           5423          MOV     A,E             COMPUTE INDEX OF 'TO'
071,252  225           5424          SUB     L
071,253  157           5425          MOV     L,A
071,254  172           5426          MOV     A,D
071,255  234           5427          SBB     H
071,256  147           5428          MOV     H,A
071,257  321           5429          POP     D
071,260  172           5430          MOV     A,D
071,261  027           5431          RAL                     MOVE SIGN BIT INTO CARRY
071,262  315 213 104   5432          CALL    $IBT
071,265  110 112       5433          DW      STRTAB+1
071,267  325           5434          PUSH    D               SAVE COUNT
071,270  353           5435          XCHG
071,271  052 110 112   5436          LHLD    STRTAB+MT.FWA
071,274  031           5437          DAD     D               (HL ) ABS ADDRESS OF ADDITION
071,275  321           5438          POP     D               (DE) = COUNT
071,276  172           5439          MOV     A,D
071,277  247           5440          ANA     A
071,300  364 351 100   5441          CP      ZRO             CLEAR IF WAS ADDITION
071,303  321           5442          POP     D               (DE) = 'TO' DESCRIPTOR ADDRESS(INDEX)
071,304  315 366 072   5443          CALL    CSA             (DE) = 'TO' DESCRIPTOR ADDRESS(ABS.)
071,307  072 202 042   5444          LDA     ACCX
071,312  022           5445          STAX    D               SET NEW COUNT
071,313  315 315 074   5446          CALL    FSE
071,316  345           5447          PUSH    H
071,317  021 202 042   5448          LXI     D,ACCX
071,322  315 315 074   5449          CALL    FSE             FIND 'FROM'
071,325  353           5450          XCHG                    (DE) = 'FROM' ADDRESS
071,326  341           5451          POP     H               (HL) = 'TO' ADDRESS
071,327  315 015 071   5452          CALL    MOV             MOVE STRING
071,332  341           5453          POP     H               RESTORE (HL)
071,333  311           5454          RET
```

```
                                5456  **      CAS - CALCULATE ARRAY SIZE.
                                5457  *
                                5458  *       CAS COMPUTES THE NUMBER OF ENTRYS IN AN ARRAY.
                                5459  *
                                5460  *       ENTRY   (DE) = HEADER POINTER
                                5461  *       EXIT    (HL) = COUNT
                                5462  *       USES    A;F;D;E;H;L
                                5463
                                5464
071.334  305                    5465  CAS     PUSH    B               SAVE (BC)
071.335  032                    5466          LDAX    D               (A) = SUBSCRIPT COUNT
071.336  023                    5467          INX     D
071.337  023                    5468          INX     D
071.340  023                    5469          INX     D
071.341  023                    5470          INX     D
071.342  325                    5471          PUSH    D               SAVE SYMTAB ADDRESS
071.343  041 001 000            5472          LXI     H,1             (HL) = ACCUMULATOR
                                5473
071.346                         5474  CAS1    EQU     *
071.346  343                    5475          XTHL                    (HL) = ADDRESS
071.347  136                    5476          MOV     E,M
071.350  043                    5477          INX     H
071.351  126                    5478          MOV     D,M             (DE) = BOUND
071.352  043                    5479          INX     H
071.353  301                    5480          POP     B               (BC) = ACCUMULATOR
071.354  345                    5481          PUSH    H               SAVE ADDRESS
071.355  365                    5482          PUSH    PSW             SAVE COUNT
071.356  315 337 030            5483          CALL    $MU66           (HL) = ACCUMULATION
071.361  302 122 070            5484          JNZ     ERR.IN          OVERFLOW
071.364  361                    5485          POP     PSW
071.365  075                    5486          DCR     A               DECREMENT COUNT
071.366  302 346 071            5487          JNZ     CAS1            IF MORE
071.371  321                    5488          POP     D               DISCARD ADDRESS
071.372  301                    5489          POP     B               RESTORE BC
071.373  311                    5490          RET


                                5492  **      CEF - CREATE EMPTY FILE BUFFER.
                                5493  *
                                5494  *       CEF CREATES AN EMPTY FILE BUFFER
                                5495  *       ON THE END OF FILTAB.
                                5496  *
                                5497  *       ENTRY   NONE
                                5498  *       EXIT    NONE
                                5499  *       USES    ALL
                                5500
                                5501
071.374  021 122 112            5502  CEF     LXI     D,FILTAB+1
071.377  041 000 001            5503          LXI     H,256
072.002  303 244 103            5504          JMP     $ATS            ALLOCATE TABLE SPACE            /80.01.GC/
```

```
                                5506  **      CFA - COMPUTE FILE BLOCK ADDRESS.
                                5507  *
                                5508  *       CFA COMPUTES THE ABS ADDRESS OF A FILE BLOCK.
                                5509  *
                                5510  *       ENTRY   (A) = FILE BLOCK NUMBER (IOCHAN-1)
                                5511  *       EXIT    TO ERR.SY IF NUMBER TOO LARGE
                                5512  *                'C' CLEAR IF OK
                                5513  *                 (HL) = ABS ADDRESS OF FILE BLOCK
                                5514  *                'C' SET IF NOT THERE
                                5515  *       USES    A,F,D,E,H,L
                                5516
                                5517
072.005  376 007               5518  CFA     CPI     CHANMAX+2       +1 FOR TEST, +1 FOR SKEWED ENTRY
072.007  322 122 070           5519          JNC     ERR.IN          TOO LARGE
072.012  127                   5520          MOV     D,A             (D) = CHANNEL NUMBER
072.013  036 000               5521          MVI     E,0             (DE) = 256*CHANNEL
                                5522  *       ANA     A                                       /80.20.GC/
                                5523  *       JNZ     CFA1            IS USER CHANNEL         /80.02.GC/
                                5524
                                5525  *       IS SYSTEM BUFFER, SETUP WRITE-ACCESS TO PROTECTED H17 RAM
                                5526
                                5527  *       ERRNZ   M.SYSM                                  /80.02.GC/
                                5528  *       LHLD    S.DLINK                                 /80.02.GC/
                                5529  *       MVI     M,1             SET M.SYSM NON-ZERO     /80.02.GC/
                                5530  *       CALL    $WER            WRITE ENABLE RAM        /80.02.GC/
072.015  052 124 112           5531  CFA1    LHLD    FILTAB+MT,LEN
072.020  175                   5532          MOV     A,L             SEE IF WE HAVE THAT MANY
072.021  223                   5533          SUB     E
072.022  174                   5534          MOV     A,H
072.023  232                   5535          SBB     D
072.024  330                   5536          RC                      FILE BLOCK NOT IN TABLE
072.025  172                   5537          MOV     A,D             (A) = CHANNEL NUMBER
072.026  021 033 000           5538          LXI     D,FBENL
072.031  315 007 031           5539          CALL    $MU86
072.034  021 230 042           5540          LXI     D,FBLIST
072.037  031                   5541          DAD     D               (HL) = ABS ADDRESS OF BLOCK
072.040  311                   5542          RET


                                5544  **      CFN - CRACK FILE NAME.
                                5545  *
                                5546  *       CFN DECODES A STRING FROM THE TEXT LINE INTO THE FILE
                                5547  *       NAME AREA OF THE SYSTEM FILE BLOCK.
                                5548  *
                                5549  *       ENTRY   (BC) = LINE POINTER
                                5550  *       EXIT    (BC) ADVANCED
                                5551  *               (HL) = FWA OF FILE BLOCK
                                5552  *       USES    ALL
                                5553
                                5554
072.041  315 053 072           5555  CFN.    CALL    CFN             CFN WITH FOP
072.044  315 217 074           5556          CALL    FOP             FILE OPEN PRESET
072.047  041 230 042           5557          LXI     H,FBLIST
072.052  311                   5558          RET
```

```
                          5559
072.053  315 244 055    5560  CFN     CALL    EVAL                                    /78.10.GC/
072.056  033            5561          DCX     D                                       /78.10.GC/
072.057  032            5562          LDAX    D               (A) = TYPE              /78.10.GC/
072.060  023            5563          INX     D                                       /78.10.GC/
072.061  346 001        5564          ANI     CF.STR                                  /78.10.GC/
072.063  312 152 070    5565          JZ      ERR.SY                                  /78.10.GC/
072.066  315 315 074    5566          CALL    FSE             FIND STRING TABLE ENTRY
072.071  247            5567          ANA     A
072.072  312 202 070    5568          JZ      ERR.ILF         ILLEGAL FILE NAME
072.075  353            5569          XCHG                    (DE) = STRING ADDRESS
072.076  376 021        5570          CPI     FB.NAML
072.100  322 202 070    5571          JNC     ERR.ILF         TOO LONG A NAME
072.103  305            5572          PUSH    B               SAVE (BC)
072.104  041 242 042    5573          LXI     H,FBLIST+FB.NAM
072.107  117            5574          MOV     C,A
072.110  006 000        5575          MVI     B,0             (BC) = LEN
072.112  315 252 030    5576          CALL    $MOVE           MOVE IN NAME
072.115  257            5577          XRA     A
072.116  167            5578          MOV     M,A             TERMINATE NAME
072.117  062 231 042    5579          STA     FBLIST+FB.FLG   CLEAR STATUS
072.122  041 230 042    5580          LXI     H,FBLIST
072.125  301            5581          POP     B               RESTORE REGS
072.126  311            5582          RET                     EXIT


                          5584  **      $CFS - CALCULATE FREE SPACE.
                          5585  *
                          5586  *       $CFS COUNTS THE FREE SPACE AVAILABLE TO MANAGED TABLES.
                          5587  *
                          5588  *       ENTRY   NONE
                          5589  *       EXIT    (HL) = COUNT
                          5590  *       USES    A,F,B,E,H,L
                          5591
                          5592
072.127  041 061 112    5593  $CFS    LXI     H,MTABIND+MT.LEN
072.132  345            5594          PUSH    H               SAVE POINTER ON STACK
072.133  041 346 114    5595          LXI     H,MTAREA        (HL) = ACCUMULATOR
072.136  076 010        5596          MVI     A,MTABL         (A) = NUMBER OF TABLES
072.140  343            5597  CFS1    XTHL                    (HL) = ADDRESS OF NEXT TABLE
072.141  136            5598          MOV     E,M
072.142  043            5599          INX     H
072.143  126            5600          MOV     D,M
072.144  043            5601          INX     H
072.145  043            5602          INX     H
072.146  043            5603          INX     H
072.147  043            5604          INX     H
072.150  343            5605          XTHL
072.151  031            5606          DAD     D               (HL) = LENGTH
072.152  075            5607          DCR     A
072.153  302 140 072    5608          JNZ     CFS1            MORE TABLES TO ADD
                          5609
                          5610  *       (HL) = TABLE BYTE COUNT + TABLE FWA
                          5611
```

```
072.156 321           5612          POP     D               (DE) = ADDRESS OF MEML+2
072.157 033           5613          DCX     D
072.160 033           5614          DCX     D
072.161 032           5615          LDAX    D
072.162 225           5616          SUB     L
072.163 157           5617          MOV     L,A
072.164 023           5618          INX     D
072.165 032           5619          LDAX    D
072.166 234           5620          SBB     H
072.167 147           5621          MOV     H,A
072.170 311           5622          RET


                      5624  **      CLF - CLEAR FILE OPERATIONS.
                      5625  *
                      5626  *       CLF IS CALLED TO CLEAR FILE JUNK.
                      5627
                      5628
072.171               5629  CLF     EQU     *
072.171 041 000 000   5630          LXI     H,0
072.174 042 124 112   5631          SHLD    FILTAB+MT.LEN   EMPTY ALL BUT ONE FILE
072.177 377 056       5632          DB      SYSCALL+.CLEARA CLEAR ALL CHANNELS (BUT OVERLAY CHANNEL)
072.201 257           5633          XRA     A
072.202 062 231 042   5634          STA     FBLIST+FB.FLG   CLEAR STATUS OF INTERNAL BUFFER
072.205 311           5635          RET


                      5637  **      CLN - CHECK FOR LEGAL NUMBER.
                      5638  *
                      5639  *       CLN EXAMINES A LINE NUMBER TO SEE IF IT OCCURS IN THE
                      5640  *       LEGAL RANGE.
                      5641  *
                      5642  *       ENTRY   (DE) = LINE NUMBER
                      5643  *       EXIT    TO *RET* IF BAD
                      5644  *               TO ERR.SR IF BAD
                      5645  *       USES    A,F
                      5646
                      5647
072.206 172           5648  CLN     MOV     A,D
072.207 263           5649          ORA     E
072.210 312 122 070   5650          JZ      ERR.IN          IS 0
072.213 023           5651          INX     D
072.214 172           5652          MOV     A,D
072.215 263           5653          ORA     E
072.216 033           5654          DCX     D
072.217 312 122 070   5655          JZ      ERR.IN          IS 377377A
072.222 311           5656          RET                     IS OK
```

```
                                5658  **      CMA - CHECK FOR COMMA.
                                5659  *
                                5660  *       CMA REQUIRES A COMMA IN THE TEXT STREAM.
                                5661  *
                                5662  *       ENTRY   NONE
                                5663  *       EXIT    (BC) ADVANCED
                                5664  *       USES    A,F,B,C,D,E
                                5665
                                5666
072,223  315 305 077           5667  CMA     CALL    RNT
072,226  026                   5668          DB      CT.CMA
072,227  311                   5669          RET



                                5671  **      CNC - CLASSIFY NEXT CHARACTER.
                                5672  *
                                5673  *       CNC CLASSIFYS THE NEXT TEXT CHARACTER.
                                5674  *
                                5675  *       ENTRY   (BC) = TEXT POINTER
                                5676  *       EXIT    (A) = 'CT.' CODE
                                5677  *       USES    A,F
                                5678
                                5679
072,230  012                   5680  CNC     LDAX    B               (A) = CODE
072,231  247                   5681          ANA     A
072,232  370                   5682          RM                      IS KEYWORD
000.000                         5683          ERRNZ   CT.FIN
072,233  310                   5684          RZ                      IS FIN
072,234  376 060               5685          CPI     '0'
072,236  332 265 072           5686          JC      CNC1            NOT NUMERIC OR ALPHA
072,241  376 072               5687          CPI     '9'+1
072,243  076 002               5688          MVI     A,CT.NUM
072,245  330                   5689          RC                      IS NUMERIC
072,246  012                   5690          LDAX    B
072,247  315 045 112           5691          CALL    $MCU            MAP CHARACTER TO UPPER CASE
072,252  376 101               5692          CPI     'A'
072,254  332 265 072           5693          JC      CNC1            NOT ALPHA
072,257  376 133               5694          CPI     'Z'+1
072,261  076 001               5695          MVI     A,CT.ALP
072,263  330                   5696          RC                      IS ALPHABETIC
072,264  012                   5697          LDAX    B
                                5698
                                5699  *       NOT ALPHABETIC OR NUMERIC. FIND IN TABLE.
                                5700
072,265  345                   5701  CNC1    PUSH    H
072,266  041 302 072           5702          LXI     H,CNCA
072,271  315 371 111           5703          CALL    $TBLS           SEARCH TABLE
072,274  176                   5704          MOV     A,M             (A) = INDEX
072,275  341                   5705          POP     H               RESOTRE (HL)
072,276  310                   5706          RZ                      FOUND
072,277  076 003               5707          MVI     A,CT.SEP        SEPERATOR
072,301  311                   5708          RET
                                5709
                                5710  **      TABLE OF SPECIAL TERMINATORS.
```

```
                                5711
072,302                         5712  CNCA    EQU     *
072,302  053 021               5713          DB      '+',CT.PL
072,304  055 022               5714          DB      '-',CT.MI
072,306  050 017               5715          DB      '(',CT.PAL
072,310  051 020               5716          DB      ')',CT.PAR
072,312  052 023               5717          DB      '*',CT.MU
072,314  057 024               5718          DB      '/',CT.DI
072,316  136 025               5719          DB      '^',CT.EX
072,320  072 000               5720          DB      ':',CT.FIN
072,322  056 002               5721          DB      '.',CT.NUM
072,324  054 026               5722          DB      ',',CT.CMA
072,326  074 014               5723          DB      '<',CT.LT
072,330  075 011               5724          DB      '=',CT.EQ
072,332  076 012               5725          DB      '>',CT.GT
072,334  073 027               5726          DB      ';',CT.SEM
072,336  042 030               5727          DB      '"',CT.QUO
072,340  133 017               5728          DB      '[',CT.PAL
072,342  135 020               5729          DB      ']',CT.PAR
072,344  043 031               5730          DB      '#',CT.PS
072,346  000                   5731          DB      0               END OF TABLE


                                5733  **      CQT - CHECK OPERAND TYPES.
                                5734  *
                                5735  *       CQT CHECKS THE OPERANDS TO SEE IF THE TYPE IS CONSISTANT.
                                5736  *
                                5737  *       EXIT    (ACCX), (ACCY) = 2 OPERANDS
                                5738  *       EXIT    TO *RET* IF BOTH SAME TYPE
                                5739  *                'Z' SET IF NUMERIC
                                5740  *                TO ERR.TE IF OF DIFFERING TYPES
                                5741  *       USES    A,F,H,L
                                5742
                                5743
072,347  072 201 042           5744  CQT     LDA     ACCX-1
072,352  041 207 042           5745          LXI     H,ACCY-1
072,355  057                   5746          CMA
072,356  256                   5747          XRA     M
072,357  346 001               5748          ANI     CF.STR
072,361  312 155 070           5749          JZ      ERR.TC          DIFFERENT TYPES
072,364  246                   5750          ANA     M               (A) = CODE
072,365  311                   5751          RET                     RETURN WITH CODE


                                5753  **      CSA - CALCULATE SYMTAB ABSOLUTE ADDR.
                                5754  *
                                5755  *       CSA CALCULATES AN ABSOLUTE ADDRESS FOR A GIVEN
                                5756  *       INDEX
                                5757  *
                                5758  *       ENTRY   (DE) = INDEX INTO SYMTAB
                                5759  *       EXIT    (DE) = ABSOLUTE ADDRESS
                                5760  *       USES    D,E
```

```
                                5761
                                5762
072.366                         5763  CSA     EQU     *
                                5764
072.366  365                   5765          PUSH    PSW             SAVE (A)
072.367  345                   5766          PUSH    H               SAVE (HL)
072.370  052 064 112           5767          LHLD    SYMTAB+MT.FWA   (HL) = #FWA OF SYMTAB
072.373  031                   5768          DAD     D                                   /80.01.GC/
072.374  353                   5769          XCHG                    DE = ABSOLUTE ADDRESS IN SYMTAB /80.01.GC/
072.375  341                   5770          POP     H               RESTORE (HL)
072.376  361                   5771          POP     PSW
072.377  311                   5772          RET                     EXIT


                                5774  **      CSI - CALCULATE SYMTAB INDEX
                                5775  *
                                5776  *       CSI CALCULATES AN INDEX INTO THE SYMTAB
                                5777  *       FROM A GIVEN ABSOLUTE ADDRESS
                                5778  *
                                5779  *       ENTRY   (DE) = ABSOLUTE ADDRESS INTO SYMBOL
                                5780  *       EXIT    (DE) = INDEX INTO SYMTAB
                                5781  *       USES    D,E
                                5782
                                5783
073.000                         5784  CSI     EQU     *
                                5785
073.000  365                   5786          PUSH    PSW             SAVE (A)
073.001  345                   5787          PUSH    H               SAVE (HL)
073.002  052 064 112           5788          LHLD    SYMTAB+MT.FWA
073.005  173                   5789          MOV     A,E
073.006  225                   5790          SUB     L
073.007  137                   5791          MOV     E,A
073.010  172                   5792          MOV     A,D
073.011  234                   5793          SBB     H
073.012  127                   5794          MOV     D,A             (DE) = INDEX INTO SYMBOL TABLE
073.013  341                   5795          POP     H               RESTORE (HL)
073.014  361                   5796          POP     PSW             RESTORE (A)
073.015  311                   5797          RET                     EXIT


                                5799  **      CSE - CREATE STRING TABLE ENTRY.
                                5800  *
                                5801  *       CSE CREATES A STRING TABLE ENTRY.
                                5802  *
                                5803  *       ENTRY   (DE) = POINTER BLOCK ADDRESS.
                                5804  *       EXIT    DESCRIPTOR SET IN BLOCK
                                5805  *               (DE) = POINTER BLOCK ADDRESS
                                5806  *               (HL) = ABS STRING ADDRESS
                                5807  *       USES    A,F,D,E,H,L
                                5808
                                5809
073.016  305                    5810  CSE     PUSH    B
```

```
073.017 041 143 112  5811          LXI   H,STRVI
073.022 315 056 073  5812          CALL  CSE..
073.025 021 110 112  5813          LXI   D,STRTAB+1
073.030 303 045 073  5814          JMP   CSE1
                     5815
073.033 305          5816  CSE.    PUSH  B
073.034 041 145 112  5817          LXI   H,STRTI
073.037 315 056 073  5818          CALL  CSE..
073.042 021 115 112  5819          LXI   D,TSTTAB+1
                     5820
073.045 315 026 071  5821  CSE1    CALL  AMB             MAKE ROOM
073.050 160          5822          MOV   M,B
073.051 043          5823          INX   H
073.052 161          5824          MOV   M,C             SET NUMBER IN STRING
073.053 043          5825          INX   H               (HL) = ABS ADDRESS
073.054 301          5826          POP   B
073.055 311          5827          RET
                     5828
073.056 043          5829  CSE..   INX   H
073.057 064          5830          INR   M               INCREMENT INDEX
073.060 053          5831          DCX   H
073.061 302 065 073  5832          JNZ   CSE2            NOT OVERFLOW
073.064 064          5833          INR   M
073.065 106          5834  CSE2    MOV   B,M
073.066 043          5835          INX   H
073.067 116          5836          MOV   C,M             (BC) = STRING NAME
073.070 353          5837          XCHG                  (HL) = BLOCK ADDRESS + 2
073.071 136          5838          MOV   E,M
073.072 043          5839          INX   H
073.073 126          5840          MOV   D,M             (DE) = STRING LENGTH
073.074 043          5841          INX   H
073.075 160          5842          MOV   M,B
073.076 043          5843          INX   H
073.077 161          5844          MOV   M,C             STORE IN HEADER
073.100 023          5845          INX   D
073.101 023          5846          INX   D               +2 FOR HEADER
073.102 353          5847          XCHG
073.103 311          5848          RET


                     5850  **      CUF     -  CLEAR USER FUNCTION
                     5851  *
                     5852  *       CUF CLEARS THE USER-DEFINED FUNCTIONS FROM THE FUNCTION TABLE
                     5853  *       BY REMOVING THE ENTRIES FROM  *SYMTAB*.
                     5854  *
                     5855  *       ENTRY:  NONE
                     5856  *
                     5857  *       EXIT:   USER-DEFINED FUNCTIONS OUT OF THE SYMBOL TABLE ENTRY
                     5858  *
                     5859  *       USES:   ALL
                     5860  *
                     5861
073.104              5862  CUF     EQU     *
                     5863
```

```
073.104  021 000 000  5864          LXI     D,0             SET THE INDEX TO ZERO
                      5865
073.107  052 066 112  5866  CUF1    LHLD    SYMTAB+MT.LEN
073.112  023          5867          INX     D
073.113  315 346 111  5868          CALL    HLCPDE
073.116  033          5869          DCX     D
073.117  330          5870          RC                      ALL FINISHED (LENGTH <= INDEX+1)
                      5871
073.120  315 126 073  5872          CALL    CUF2            PROCESS THE ENTRY
                      5873
073.123  303 107 073  5874          JMP     CUF1
                      5875
                      5876  *       PROCESS A SYMBOL TABLE ENTRY
                      5877
073.126               5878  CUF2    EQU     *
                      5879
073.126  052 064 112  5880          LHLD    SYMTAB+MT.FWA
073.131  031          5881          DAD     D               HL = FWA OF SYMBOL TABLE ENTRY
073.132  043          5882          INX     H
073.133  176          5883          MOV     A,M             A  FLAG BYTE
073.134  043          5884          INX     H
073.135  346 002      5885          ANI     CF.VEC
073.137  312 163 073  5886          JZ      CUF3            NOT A VECTOR
                      5887
073.142  176          5888          MOV     A,M
073.143  247          5889          ANA     A
073.144  362 172 073  5890          JP      CUF4            IS A VECTOR
                      5891
                      5892  *       DELETE A FUNCTION
                      5893
073.147  325          5894          PUSH    D
073.150  353          5895          XCHG                    HL = INDEX INTO TABLE
073.151  021 006 000  5896          LXI     D,6             COUNT = 6
073.154  315 203 104  5897          CALL    $DBT            DELETE THE BYTES FROM THE TABLE
073.157  064 112      5898          DW      SYMTAB+1
073.161  321          5899          POP     D
073.162  311          5900          RET
                      5901
                      5902  *       PASS OVER A SCALAR
                      5903
073.163  001 006 000  5904  CUF3    LXI     B,6
073.166  353          5905          XCHG
073.167  011          5906          DAD     B
073.170  353          5907          XCHG                    INDEX = INDEX + 6
073.171  311          5908          RET
                      5909
                      5910  *       PASS OVER A VECTOR
                      5911
073.172  043          5912  CUF4    INX     H
073.173  043          5913          INX     H               SKIP 'DIM' AND '0' BYTES
                      5914
073.174  116          5915          MOV     C,M
073.175  043          5916          INX     H
073.176  106          5917          MOV     B,M
073.177  043          5918          INX     H               BC = ARRAY SIZE FOR VECTOR ENTRYS
                      5919
```

```
073.200  353            5920          XCHG
073.201  011            5921          DAD     B               INDEX = INDEX + SIZE
073.202  001 006 000     5922          LXI     B,6
073.205  011            5923          DAD     B               INDEX = INDEX + 6 (BYTES SKIPPED AT START)
073.206  353            5924          XCHG
073.207  311            5925          RET


                        5927  **      CVX - COPY VALUE INTO 'X' ACCUMULATOR.
                        5928  *
                        5929  *       CVX COPIES A 4 BYTE VALUE INTO THE X ACCUMULATOR.
                        5930  *
                        5931  *       ENTRY   (DE) = ADDRESS OF VALUE
                        5932  *       EXIT    COPIED
                        5933  *       USES    A,F
                        5934
                        5935
073.210  345            5936  CVX     PUSH    H
073.211  325            5937          PUSH    D
073.212  041 202 042    5938          LXI     H,ACCX
073.215  315 051 076    5939          CALL    MOV4            MOVE
073.220  321            5940          POP     D
073.221  341            5941          POP     H
073.222  311            5942          RET


                        5944  **      CXY - COPY (ACCX) INTO (ACCY)
                        5945  *
                        5946  *       ENTRY   NONE
                        5947  *       EXIT    NONE
                        5948  *       USES    A,F,D,E
                        5949
                        5950
073.223                 5951  CXY     EQU     *
073.223  345            5952          PUSH    H               SAVE (HL)
073.224  021 201 042    5953          LXI     D,ACCX-1        SOURCE
073.227  041 207 042    5954          LXI     H,ACCY-1        DESTINATION
073.232  315 045 076    5955          CALL    MOV5            MOVE (ACCX) TO (ACCY)
073.235  341            5956          POP     H               RESTORE (HL)
073.236  311            5957          RET                     EXIT


                        5959  **      CXV - COPY X TO VALUE.
                        5960  *
                        5961  *       CXV COPIES THE CONTENTS OF THE 'X' ACCUMULATOR INTO A MEMORY
                        5962  *       LOCATION.
                        5963  *
                        5964  *       ENTRY   (DE) = TARGET ADDRESS
                        5965  *       EXIT    COPIED
                        5966  *       USES    A,F
```

```
                                5967
                                5968
073.237  345                   5969  CXV       PUSH    H
073.240  325                   5970  CXV.      PUSH    D
073.241  353                   5971            XCHG
073.242  021 202 042           5972            LXI     D,ACCX
073.245  315 051 076           5973            CALL    MOV4            MOVE
073.250  321                   5974            POP     D               RESTORE DE
073.251  341                   5975            POP     H
073.252  311                   5976            RET


                                5978  **        DCN - DECODE CHANNEL NUMBER.
                                5979  *
                                5980  *         DCN DECODES A CHANNEL SPECIFICATION OF THE FORM:
                                5981  *
                                5982  *         #N                OR
                                5983  *         #LNO(EXPR) ARCHAIC(THIS IS TACKY!)                      /80.01.GC/
                                5984  *
                                5985  *         IF THE CHANNEL EXPRESSION IS OMITTED, IOCHAN IS SETUP TO INDICATE THE
                                5986  *         SYSTEM CONSOLE.
                                5987  *
                                5988  *         ENTRY   (BC) = TEXT POINTER
                                5989  *         EXIT    (BC) ADVANCED
                                5990  *                  IOCHAN = 0 IF CONSOLE, = N+1 IF FILE
                                5991  *                 (A) = (IOCHAN)
                                5992  *         USES    ALL
                                5993
                                5994
073.253  257                   5995  DCN       XRA     A
073.254  062 140 112           5996            STA     IOCHAN          ASSUME NONE
073.257  315 072 076           5997            CALL    FNT
073.262  376 031               5998            CPI     CT.PS
073.264  300                   5999            RNE                     NONE
073.265  315 273 073           6000            CALL    DCN.            DECODE CHANNEL NUMBER
073.270  303 223 072           6001            JMP     CMA             REQUIRE COMMA AND EXIT
                                6002
                                6003  **        DCN. - DECODE CHANNEL NUMBER.
                                6004  *
                                6005  *         SAME AS DCN, BUT REQUIRES CHANNEL
                                6006  *         AND DOESNT CHECK FOR TRAILING COMMA
                                6007  *
                                6008
073.273  315 305 077           6009  DCN.      CALL    FNT
073.276  031                   6010            DB      CT.PS
073.277  315 036 057           6011            CALL    EVALI           EVALUATE AN EXPRESSION          /80.01.GC/
                                6012
                                6013  **        DCN.. - CHECK CHANNEL NUMBER.
                                6014  *
                                6015  *         CHECH (DE) FOR VALID CHANNEL NUMBER
                                6016  *         EXIT    (A) = CHANNEL VALUE
                                6017  *
                                6018
073.302  172                   6019  DCN..     MOV     A,D
```

```
073.303  267            6020          ORA     A
073.304  302 122 070     6021          JNZ     ERR.IN          TOO LARGE
073.307  173            6022          MOV     A,E
073.310  376 006        6023          CPI     CHANMAX+1                       /78.10.6C/
073.312  322 122 070     6024          JNC     ERR.IN          TOO LARGE
073.315  247            6025          ANA     A
073.316  312 322 073     6026          JZ      DCN1
073.321  074            6027          INR     A               (A) = 2+N
073.322  062 140 112     6028  DCN1    STA     IOCHAN
073.325  311            6029          RET                     EXIT


                        6031  **      DNF - DELETE NON-OPEN FILE BLOCKS.
                        6032  *
                        6033  *       DNF DELETES ALL THE NON-OPEN FILE BLOCKS THAT ARE AT THE
                        6034  *       END OF THE FILTAB, AS SOON AS AN OPEN FILE BLOCK BECOMES THE END ONE.
                        6035  *       NO MORE WILL BE DELETED. THUS, IF #5 IS OPEN, AND #4, #3, AND #2  ARE
                        6036  *       CLOSED, THEY WILL REMAIN UNRECOVERED UNTIL #5 IS CLOSED, AND THEN BE
                        6037  *       CLEANED OUT IN ONE SWOOP.
                        6038  *
                        6039  *       ENTRY   NONE
                        6040  *       EXIT    NONE
                        6041  *       USES    ALL
                        6042
                        6043
073.326  072 125 112     6044  DNF     LDA     FILTAB+MT.LEN+1 (A) = # OF BUFFERS
073.331  247            6045          ANA     A
073.332  310            6046          RZ                      NONE ELIGIBLE
073.333  021 033 000     6047          LXI     D,FBENL
073.336  315 007 031     6048          CALL    $MU86
073.341  021 231 042     6049          LXI     D,FBLIST+FB.FLG
073.344  031            6050          DAD     D               (HL) = ADDRESS OF FB.STA FOR LAST BLOCK W/BUFFER
073.345  176            6051          MOV     A,M
073.346  247            6052          ANA     A
073.347  300            6053          RNZ                     IS OPEN
073.350  041 125 112     6054          LXI     H,FILTAB+MT.LEN+1
073.353  065            6055          DCR     M               SHORTEN TABLE
073.354  303 326 073     6056          JMP     DNF             TRY AGAIN


                        6058  **      DTS - DELETE TEMP STRINGS.
                        6059  *
                        6060  *       DTS DELETES ANY TEMP STRINGS WHICH MAY HAVE BUILT UP
                        6061  *       IN THE STRING TABLE.
                        6062  *
                        6063  *       ENTRY   NONE
                        6064  *       EXIT    TABLE PARED.
                        6065  *       USES    H,L
                        6066
                        6067
073.357  041 000 000     6068  DTS     LXI     H,0
073.362  042 117 112     6069          SHLD    TSTTAB+MT.LEN
```

```
073.365  041 300 000  6070          LXI     H,300Q
073.366               6071  DTSA    EQU     *-2
073.370  042 145 112  6072          SHLD    STRTI           RESET STRING TEMP INDEX
073.373  311          6073          RET


                      6075  **      EKA - EXPAND KEYWORD INTO ASCII EQUIVALENT.
                      6076  *
                      6077  *       EKA EXPANDS A KEYWORD BYTE INTO THE ASCII EQUIVALENT.
                      6078  *
                      6079  *       ENTRY   (A) = TOKEN
                      6080  *               (DE) = ADDRESS FOR STRING
                      6081  *       EXIT    (A) = LAST CHARACTER OF ASCII
                      6082  *               (DE) = ADDRESS FOR LAST CHARACTER OF ASCII
                      6083  *       USES    A,F,B,C,D,E
                      6084
                      6085
073.374  001 240 066  6086  EKA     LXI     B,KEYTAB
073.377  325          6087          PUSH    D               SAVE ADDRESS
074.000  127          6088          MOV     D,A             (D) = PATTERN
074.001  012          6089  EKA1    LDAX    B
074.002  272          6090          CMP     D
074.003  003          6091          INX     B
074.004  302 001 074  6092          JNE     EKA1            NOT THERE, YET
074.007  321          6093          POP     D               (DE) = ADDRESS
074.010  365          6094          PUSH    PSW             SAVE KEYWORD BYTE
                      6095
                      6096  *       EXPAND IT.
                      6097
074.011  012          6098  EKA2    LDAX    B
074.012  022          6099          STAX    D
074.013  003          6100          INX     B
074.014  023          6101          INX     D
074.015  247          6102          ANA     A
074.016  362 011 074  6103          JP      EKA2            MORE TO GO
074.021  033          6104          DCX     D               REPLACE EXTRA BYTE WITH ' ' OR '('
074.022  361          6105          POP     PSW
074.023  376 320      6106          CPI     CT.FCN
074.025  076 040      6107          MVI     A,' '           ASSUME NOT FUNCTION
074.027  330          6108          RC                      NOT FUNCTION
074.030  076 050      6109          MVI     A,'('           IS FUNCTION
074.032  311          6110          RET


                      6112  **      ELN - EVALUATE LINE NUMBER.
                      6113  *
                      6114  *       ELN IS CALLED WHEN A LINE NUMBER IS TO BE EVALUATED.
                      6115  *
                      6116  *       THE LINE NUMBER CAN EITHER BE A DECIMAL INTEGER, OR
                      6117  *       THE EXPRESSION LNO(EXPR)
                      6118  *
                      6119  *       ENTRY   (BC) = LINE POINTER
```

```
                          6120  *       EXIT    (BC) UPDATED
                          6121  *               (DE) = LINE NUMBER
                          6122  *       USES    A,F,B,C,D,E
                          6123
                          6124
074.033  315 126 100     6125  ELN     CALL    SOB             SKIP BLANKS
                          6126
                          6127  *       MUST HAVE DECIMAL INTEGER, OR LNO(
                          6128
074.036  012             6129          LDAX    B
074.037  376 327         6130          CPI     CT.LNO
074.041  312 102 074     6131          JE      ELN2            IS LNO(EXPR)
074.044  315 321 111     6132          CALL    $CVD.           SEE IF DIGIT
074.047  332 152 070     6133          JC      ERR.SY
074.052  021 000 000     6134          LXI     D,0             (DE) =ACCUM
                          6135
                          6136  *       HAVE DECIMAL INTERGER
                          6137
074.055  012             6138  ELN1    LDAX    B
074.056  315 321 111     6139          CALL    $CVD.
074.061  330             6140          RC                      END OF NUMBER
074.062  345             6141          PUSH    H               SAVE (HL)
074.063  315 324 030     6142          CALL    $MU10           (HL) = 10*ACCUM
074.066  315 072 030     6143          CALL    $DADA           (HL) = 10*ACCUM+DIGIT
074.071  353             6144          XCHG
074.072  003             6145          INX     B
074.073  341             6146          POP     H               RESTORE (HL)
074.074  332 122 070     6147          JC      ERR.IN          ILLEGAL NUMBER IF OVERFLOW
074.077  303 055 074     6148          JMP     ELN1            TRY FOR MORE
                          6149
                          6150  *       IS LNO(EXPR)
                          6151
074.102  003             6152  ELN2    INX     B               SKIP LNO( KEYWORD
074.103  315 036 057     6153          CALL    EVALI
074.106  325             6154          PUSH    D               SAVE VALUE
074.107  315 305 077     6155          CALL    RNT
074.112  020             6156          DB      CT.PAR          REQURE ')'
074.113  321             6157          POP     D
074.114  311             6158          RET


                          6160  **      FOC - FILE OPEN CLEANUP.
                          6161  *
                          6162  *       FOC IS CALLED TO CLEANUP AFTER FOP. FOC RESTORES AS MUCH
                          6163  *       MEMORY AS THE SYSTEM IS NOT USING (ALSO DEPENDS UPON CNTRL OPTION)
                          6164  *
                          6165  *       ENTRY   NONE
                          6166  *       EXIT    NONE
                          6167  *       USES    NONE
                          6168
                          6169
074.115  315 054 031     6170  FOC     CALL    $SAVALL         SAVE ALL
                          6171  *       LHLD    S.DLINK                                 /80.02.GC/
                          6172  *       ERRNZ   M.SYSM                                  /80.02.GC/
```

```
                        6173  *       MVI     M,0             CLEAR ARTIFICAL SYSTEM MODE (ALLOW H17 RAM TO
                        6174  *                                WRITE DISABLE))                 /80.02.GC/
074,120  052 320 040    6175          LHLD    S.SYSM
074,123  021 360 377    6176          LXI     D,-16
074,126  031            6177          DAD     D               (HL) = LWA USABLE
074,127  072 141 112    6178          LDA     OVLMAN          (A) = OVERLAY MANGAEMENT FLAG
074,132  247            6179          ANA     A
074,133  312 146 074    6180          JZ      FOC1            LEAVE OVERLAY OUT
074,136  353            6181          XCHG
074,137  052 324 040    6182          LHLD    S.OMAX
074,142  315 224 030    6183          CALL    $CHL            (HL) = -OVLMAX
074,145  031            6184          DAD     D               (HL) = LIMIT TO ALLOW OVL RESIDENT
074,146  353            6185  FOC1    XCHG                    (DE) = PROSPECTIVE NEW MEML
074,147  052 127 112    6186          LHLD    MEML            (HL) = CURRENET MEML
074,152  173            6187  FOC1.3  MOV     A,E                                     /80.01.GC/
074,153  225            6188          SUB     L               MAKE SURE GETTING LARGER, NOT SMALLER!
074,154  172            6189          MOV     A,D
074,155  234            6190          SBB     H
074,156  332 210 074    6191          JC      FOC3            NOT ENOUGH MEMORY
074,161  353            6192          XCHG                    (HL) = NEW LIMIT
074,162  042 127 112    6193  FOC1.5  SHLD    MEML            NOTIFY TABLES
074,165  353            6194          XCHG                    (DE) = NEW MEML
074,166  052 322 040    6195          LHLD    S.USRM
074,171  315 216 030    6196          CALL    $CDENL
074,174  312 205 074    6197          JE      FOC2            NO NEED TO REQUEST, ALREADY GOT IT
074,177  353            6198          XCHG                    (HL) = REQUEST
074,200  377 052        6199          DB      SYSCALL,.SETTP  SET TOP
074,202  332 223 070    6200          JC      SERROR          ERROR
074,205  303 047 031    6201  FOC2    JMP     $RSTALL         RESTORE ALL AND EXIT
                        6202
                        6203  *       NOT ENOUGH MEMORY TO RESIDE OVERLAY
                        6204
074,210  257            6205  FOC3    XRA     A
074,211  062 141 112    6206          STA     OVLMAN          CLEAR RESIDE
074,214  303 160 070    6207          JMP     ERR.TO          TABLE OVERFLOW


                        6209  **      FOP - FILE OPEN PRESET.
                        6210  *
                        6211  *       FOP IS CALLED BEFORE FILE OPENING AND CLOSING IS TO TAKE
                        6212  *       PLACE. SINCE THE SYSTEM WILL LOAD AN OVERLAY, AND MAY
                        6213  *       NEED TO LOAD A DEVICE DRIVER, FOP WILL SQUEEZE THE TABLES
                        6214  *       UP TO TAKE AS LITTLE SPACE AS POSSIBLE. LATER ON, FOC WILL BE
                        6215  *       USED TO RESTORE THE TABLES INTO ANY OPEN SPACE
                        6216  *       LEFT AFTER THE OPERATIONS.
                        6217  *
                        6218  *       ENTRY   NONE
                        6219  *       EXIT    NONE
                        6220  *       USES    NONE
                        6221
                        6222
074,217  315 054 031    6223  FOP     CALL    $SAVALL         SAVE REGS
074,222  315 230 074    6224          CALL    FOP.                                    /80.01.GC/
074,225  303 163 074    6225          JMP     FOC1.5                                  /80.01.GC/
```

```
                          6226
074.230 315 127 104       6227 FOP.    CALL   MTD            MOVE TABLES DOWN
074.233 325               6228         PUSH   D              SAVE LWA
074.234 315 071 071       6229         CALL   $ATP           ADJUST TABLE POINTERS
074.237 341               6230         POP    H              (HL) = LWA
074.240 043               6231         INX    H
074.241 311               6232         RET                                        /80.01.GC/


                          6234 **      FLN - FIND LINE BY NUMBER.
                          6235 *
                          6236 *       FLN SEARCHES THE TEXT BUFFER FOR THE SPECIFIED LINE.
                          6237 *
                          6238 *       ENTRY  (DE) = LINE NUMBER
                          6239 *       EXIT   TO ERR.SN IF LINE NUMBER = 65535
                          6240 *              /C/ SET IF NOT FOUND
                          6241 *              (HL) = ADDRESS OF LINE IF FOUND, ADDRESS IF LINE+1 IF NOT
                          6242 *       USES   A,F,H,L
                          6243
                          6244
074.242 305               6245 FLN     PUSH   B
074.243 041 346 114       6246         LXI    H,MTAREA
                          6247
                          6248 *       CHECK IF LINE NUMBER = 65535
                          6249
074.246 172               6250         MOV    A,D
074.247 074               6251         INR    A
074.250 302 260 074       6252         JNZ    FLN1           HIGH ORDER BYTE <> 377
074.253 173               6253         MOV    A,E
074.254 074               6254         INR    A
074.255 312 147 070       6255         JZ     ERR.SN         LINE NUMBER = 65535; ERROR
                          6256
074.260 173               6257 FLN1    MOV    A,E
074.261 226               6258         SUB    M
074.262 107               6259         MOV    B,A            (B) = LOW LETER
074.263 172               6260         MOV    A,D
074.264 043               6261         INX    H
074.265 236               6262         SBB    M
074.266 332 312 074       6263         JC     FLN3           RAN PAST
074.271 302 300 074       6264         JNZ    FLN1.5         NOT THERE YET
074.274 260               6265         ORA    B
074.275 312 312 074       6266         JZ     FLN3           FOUND IT
074.300 043               6267 FLN1.5  INX    H
074.301 176               6268 FLN2    MOV    A,M            SKIP THIS LINE
074.302 043               6269         INX    H
074.303 247               6270         ANA    A
074.304 302 301 074       6271         JNZ    FLN2           NOT END OF LINE
074.307 303 260 074       6272         JMP    FLN1
                          6273
                          6274 *       FOUND LINE, /C/ CLEAR IF FOUND
                          6275
074.313 053               6276 FLN3    DCX    H
074.313 301               6277         POP    B
074.314 311               6278         RET
```

```
                        6280  **      FSE - FIND STRINGTAB ENTRY.
                        6281  *
                        6282  *       FSE FINDS A SPECIFED STRING IN THE TABLE.
                        6283  *
                        6284  *       ENTRY   (DE) = DESCRIPTOR ADDRESS
                        6285  *       EXIT    (HL) = ABS ADDRESS
                        6286  *               (A) = LENGTH
                        6287  *       USES    A,F,D,E,H,L
                        6288
                        6289
074.315  032           6290  FSE     LDAX    D               (A) = LENGTH
074.316  365           6291          PUSH    PSW
074.317  023           6292          INX     D
074.320  023           6293          INX     D
074.321  353           6294          XCHG
074.322  126           6295          MOV     D,M
074.323  043           6296          INX     H
074.324  136           6297          MOV     E,M             (DE) = INDEX
                        6298
                        6299  *       CHECK FOR WHICH STRING TABLE
                        6300
074.325  172           6301          MOV     A,D
074.326  346 100       6302          ANI     100Q
074.330  312 341 074   6303          JZ      FSE0
074.333  052 115 112   6304          LHLD    TSTTAB+MT.FWA
074.336  303 344 074   6305          JMP     FSE1
                        6306
074.341  052 110 112   6307  FSE0    LHLD    STRTAB+MT.FWA
000.000               6308          ERRNZ   *-FSE1
                        6309
                        6310  *       SEE IF WE HAVE IT YET
                        6311
074.344  172           6312  FSE1    MOV     A,D
074.345  247           6313          ANA     A
074.346  362 374 074   6314          JP      FSE3            NOT LOOKING FOR A VALID STRING ID
074.351  276           6315          CMP     M
074.352  043           6316          INX     H
074.353  302 363 074   6317          JNE     FSE2            NO MATCH
074.356  173           6318          MOV     A,E
074.357  276           6319          CMP     M
074.360  312 374 074   6320          JE      FSE3            FOUND
074.363  043           6321  FSE2    INX     H               NOT FOUND
074.364  176           6322          MOV     A,M
074.365  247           6323          ANA     A
074.366  362 363 074   6324          JP      FSE2            SKIP TO NEXT INDEX
074.371  303 344 074   6325          JMP     FSE1            TRY AGAIN
                        6326
                        6327  *       FOUND IT.
                        6328
074.374  043           6329  FSE3    INX     H
074.375  361           6330          POP     PSW             (A) = LEN
074.376  311           6331          RET
```

```
                                6333  **      IFIX - SPLIT NUMBER INTO INTEGER AND FRACTION.
                                6334  *
                                6335  *       IFIX FIXES ((DE)) INTO AN INTEGER.
                                6336  *
                                6337  *       ENTRY   (DE) = ADDRESS OF NUMBER
                                6338  *       EXIT    (DE) = INTEGERAL PART OF 0<=N<=65535
                                6339  *                      TO ERR.IN OTHERWISE
                                6340
                                6341
074.377  021 202 042            6342  IFIX.   LXI     D,ACCX          USE ACCX
075.002  305                    6343  IFIX    PUSH    B
075.003  345                    6344          PUSH    H
075.004  353                    6345          XCHG
075.005  315 250 107            6346          CALL    LDD             (BCDE) = NUMBER
075.010  171                    6347          MOV     A,C
075.011  247                    6348          ANA     A
075.012  372 122 070            6349          JM      ERR.IN          TOO LARGE
075.015  076 220                6350          MVI     A,220Q
075.017  220                    6351          SUB     B
075.020  332 122 070            6352          JC      ERR.IN          TOO LARGE
075.023  306 007                6353          ADI     7               (A) = SHIFT COUNT
075.025  107                    6354          MOV     B,A
075.026  315 231 107            6355  IFIX1   CALL    SRS
075.031  005                    6356          DCR     B
075.032  302 026 075            6357          JNZ     IFIX1           NOT DONE YET
075.035  341                    6358          POP     H
075.036  301                    6359          POP     B
075.037  311                    6360          RET


                                6362  **      IFLT - FLOAT NUMBER.
                                6363  *
                                6364  *       ENTRY   (DE) = VALUE
                                6365  *       EXIT    (ACCX) = NUMBER VALUE
                                6366  *                       (DE) = $ACCX-1
                                6367
                                6368
075.040  305                    6369  IFLT    PUSH    B
075.041  345                    6370          PUSH    H
075.042  001 000 227            6371          LXI     B,200Q+23*256
075.045  315 213 105            6372          CALL    NRM             NORMALIZE
075.050  315 245 106            6373          CALL    STX             STORE IN ACCX
075.053  076 300                6374          MVI     A,CT.SNV        SCALAR NUMERIC VALUE
075.055  062 201 042            6375          STA     ACCX-1          SET TYPE
075.060  341                    6376          POP     H
075.061  301                    6377          POP     B
075.062  311                    6378          RET
```

```
                                6380  **      ILM - ISSUE LINE MESSAGE.
                                6381  *
                                6382  *       ILM ISSUES A MESSAGE OF THE FORM
                                6383  *
                                6384  *       XXXXXX AT LINE NNNNN
                                6385  *
                                6386  *       WHERE XXXXXX = SUPPLIED TEXT,
                                6387  *              NNNNN = (CURNUM)
                                6388  *
                                6389  *       NOTE THAT ILM ALSO CLEARS THE BASIC WORKING CHANNEL
                                6390  *       (CHANNEL 0). THIS IS A KLUDGE SO THAT THE CHANNEL DOESNT REMAIN
                                6391  *       OPEN IF AN ERROR OCCOURS WHILE USING IT.
                                6392  *
                                6393  *       ENTRY   (SP+0) = ERROR CODE
                                6394  *       EXIT    NONE
                                6395  *       USES    A,F,H,L
                                6396
                                6397
075.063  315 217 074           6398  ILM     CALL    FOP             ALLOW OVERLAY TO RESIDE
075.066  257                   6399          XRA     A
075.067  377 055               6400          DB      SYSCALL,.CLEAR
075.071  361                   6401          POP     PSW             (A) = CODE
075.072  046 040               6402          MVI     H,' '
075.074  377 057               6403          DB      SYSCALL,.ERROR  LOOKUP ERROR
                                6404
075.076  041 124 043           6405          LXI     H,RESTART       (HL) = EXIT PROCESSOR ADDRESS
075.077                        6406  ILMA    EQU     *-2             SET BY CALLER
075.101  345                   6407          PUSH    H               SET AS 'RETURN ADDRESS'
075.102  041 301 114           6408          LXI     H,RUNMOD
075.105  176                   6409          MOV     A,M             (A) = OLD RUN MODE
075.106  366 200               6410          ORI     RM.HLT          SET HALT FLAG
075.110  167                   6411          MOV     M,A
075.111  376 200               6412          CPI     RM.IMM+RM.HLT
075.113  310                   6413          RE                      DONT PRINT LINE NUMBER IF IMMEDIATE
075.114  315 136 031           6414          CALL    $TYPTX
075.117  101 164 040           6415          DB      'At Line',' '+200Q
075.127  052 133 112           6416          LHLD    CURNUM
075.132  353                   6417          XCHG
075.133  303 206 100           6418          JMP     TDI             TYPE LINE NUMBER


                                6420  **      IST - INSERT IN SYMBOL TABLE.
                                6421  *
                                6422  *       IST LOOKS UP THE ADDRESS HOLDING THE VALUE FOR
                                6423  *       A VARIABLE. IF THE VARIABLE IS A MATRIX OR VECTOR,
                                6424  *       THE SUBSCRIPT IS EVALUATED AND THE PARTICULAR ENTRY
                                6425  *       IS RETURED AS A SCALAR VALUE.
                                6426  *
                                6427  *       IF THE VARIABLE IS NOT YET DEFINED, AND IT IS NOT A
                                6428  *       VECTOR, IT IS DEFINED WITH A VALUE OF 0 (OR A NULL STRING)
                                6429  *
                                6430  *       ENTRY   (BC) = TEXT POINTER
                                6431  *       EXIT    (BC) UPDATED
                                6432  *               (DE) = INDEX OF SYMBOL ADDRESS
```

```
                          6433  *             (A) = TYPE
                          6434  *       USES  ALL
                          6435
                          6436
075.136  315 172 075     6437  IST     CALL  IST0          INSERT SYMBOL IN TABLE
075.141  365             6438          PUSH  PSW           SAVE TYPE
075.142  315 000 073     6439          CALL  CSI           (DE) = INDEX INTO SYMBOL TABLE
075.145  346 001         6440          ANI   CF.STR
075.147  312 170 075     6441          JZ    IST00         IS NOT STRING TYPE
075.152  325             6442          PUSH  D             SAVE INDEX
075.153  315 366 072     6443          CALL  CSA           (DE) = ABS. ADDR. INTO SYMBOL
075.156  325             6444          PUSH  D
075.157  023             6445          INX   D
075.160  023             6446          INX   D
075.161  032             6447          LDAX  D             (A) = STRING ID
075.162  321             6448          POP   D             RESTORE DE
075.163  247             6449          ANA   A
075.164  314 016 073     6450          CZ    CSE           CREATE STRING TABLE ENTRY IF NOT THERE
075.167  321             6451          POP   D
075.170  361             6452  IST00   POP   PSW           RESTORE TYPE
075.171  311             6453          RET
                         6454
075.172                  6455  IST0    EQU   *
075.172  315 056 071     6456          CALL  ANT           ACCEPT NEXT TOKEN
075.175  376 300         6457          CPI   CT.VARL       SEE IF HAVE VARIABLE
075.177  332 152 070     6458          JC    ERR.SY        NOT VARIABLE
075.202  376 310         6459          CPI   CT.VARH+1
075.204  322 152 070     6460          JNC   ERR.SY        NOT VARIABLE
075.207  041 151 335     6461          LXI   H,-LEXLIM-1
075.212  031             6462          DAD   D
075.213  332 257 075     6463          JC    IST2          IS PRE-DEFINED
075.216  365             6464  IST1    PUSH  PSW           SAVE TYPE
                         6465
                         6466  *       NEVER BEFORE DEFINED.
                         6467
075.217  346 002         6468          ANI   CF.VEC
075.221  302 171 070     6469          JNZ   ERR.ND        NOT DECLARED
075.224  021 064 112     6470          LXI   D,SYMTAB+1
075.227  041 006 000     6471          LXI   H,6
075.232  315 026 071     6472          CALL  AMB           ALLOCATE 6 BYTES
075.235  021 000 000     6473          LXI   D,0           (DE) = NAME
075.236                  6474  LEXA    EQU   *-2           UNDEFINED NAME
075.240  162             6475          MOV   M,D
075.241  043             6476          INX   H
075.242  163             6477          MOV   M,E           STORE NAME
075.243  043             6478          INX   H
075.244  124             6479          MOV   D,H
075.245  135             6480          MOV   E,L           (DE) = ADDRESS OF VALUE
075.246  257             6481          XRA   A             (A) = 0
075.247  167             6482          MOV   M,A           CLEAR VALUE
075.250  043             6483          INX   H
075.251  167             6484          MOV   M,A
075.252  043             6485          INX   H
075.253  167             6486          MOV   M,A
075.254  043             6487          INX   H
075.255  167             6488          MOV   M,A           000 000 000 000
```

```
075.256  361              6489          POP     PSW             RESTORE TYPE
                          6490
                          6491  *       IS NOW DEFINED.
                          6492
075.257  334 107 055      6493  IST2    CC      VARIAB.         PROCESS VARIABLE IF NOT JUST DEFINED
075.262  311              6494          RET
```

```
                          6496  **      IVT - INSERT VECTOR IN TABLE.
                          6497  *
                          6498  *       IVT INSERTS A SYMBOL OF TYPE VECTOR IN THE SYMBOL TABLE.
                          6499  *       THE SYMBOL MUST NOT BE PREVIOUSLY DEFINED.
                          6500  *
                          6501  *       ENTRY   (BC) = TEXT POINTER
                          6502  *       EXIT    (BC) UPDATED
                          6503  *               (DE) = SYMBOL ADDRESS
                          6504  *               (A) = TYPE - CF.VEC
                          6505  *       USES    A,F,B,C,D,E
                          6506
                          6507
075.263  315 056 071      6508  IVT     CALL    ANT             ACCEPT NEXT TOKEN
075.266  041 151 335      6509          LXI     H,-LEXLIM-1
075.271  031              6510          DAD     D
075.272  332 125 070      6511          JC      ERR.IU          ALREADY DEFINED
075.275  356 002          6512          XRI     CF.VEC          TOGGLE VECTOR FLAG
075.277  303 216 075      6513          JMP     IST1            PROCESS AS IST
```

```
                          6515  **      LCC - LOCATE CHANNEL COLUMN COUNTER.
                          6516  *
                          6517  *       LCC IS CALLED TO LOCATE THE BYTE CONTAINING THE COLUMN COUNTER
                          6518  *       FOR THIS CHANNEL ((IOCHAN)). SINCE PRINTING
                          6519  *       CAN BE IN PROGRESS ON SEVERAL CHANNELS AT ONCE, A SEPERATE COLUNE
                          6520  *       COUNTER IS KEPT FOR EACH ONE.
                          6521  *
                          6522  *       ENTRY   NONE
                          6523  *       EXIT    (HL) = ADDRESS OF RIGHT ENTRY IN *COLCNTS*
                          6524  *       USES    A,F,H,L
                          6525
                          6526
075.302  041 253 112      6527  LCC     LXI     H,COLCNTS
075.305  072 140 112      6528          LDA     IOCHAN
075.310  303 101 030      6529          JMP     $DADA.          (HL) = ADDRESS
```

```
                                6531  **      LFC - LOCK FLAG CHECK
                                6532  *
                                6533  *       LFC CHECKS IF THE DATA LOCK IS INVOKED
                                6534  *
                                6535  *       ENTRY   NONE
                                6536  *       EXIT    TO ERR.LK IF DATA LOCK IN FORCE
                                6537  *               TO (RET) IF NORMAL
                                6538  *       USES    A,F
                                6539
                                6540
075.313  072 137 112           6541  LFC     LDA     LCKFLG          (A) = DATA LOCK FLAG
075.316  247                   6542          ANA     A
075.317  302 130 070           6543          JNZ     ERR.LK          DATA LOCK IN FORCE
075.322  311                   6544          RET                     EXIT


                                6546  **      LVS     -  LOOK-UP VARIABLE IN SYMBOL TABLE
                                6547  *
                                6548  *       LVS LOOKS UP THE SPECIFIED VARIABLE IN THE SYMBOL TABLE.
                                6549  *       THE VARIABLE IS SPECIFIED BY THE VARIABLE NAME AND TYPE
                                6550  *       IN THE  *DE*  REGISTER PAIR AS PER THE  *SYMTAB*  FORMAT.
                                6551  *
                                6552  *       ENTRY:  DE      =  SYMTAB KEY
                                6553  *
                                6554  *       EXIT:   PSW     =  'Z' CLEAR IF NOT FOUND
                                6555  *                          'Z' SET   IF     FOUND
                                6556  *                           DE  =  SYMTAB ADDRESS
                                6557  *
                                6558  *       USES:   PSW,DE
                                6559  *
                                6560
075.323                         6561  LVS     EQU     *
                                6562
075.323  345                   6563          PUSH    H
075.324  305                   6564          PUSH    B
075.325  052 066 112           6565          LHLD    SYMTAB+MT.LEN
075.330  104                   6566          MOV     B,H
075.331  115                   6567          MOV     C,L             BC = SYMTAB LENGTH
075.332  052 064 112           6568          LHLD    SYMTAB+MT.FWA   HL = SYMTAB FWA
                                6569
075.335  170                   6570  LVS1    MOV     A,B
075.336  261                   6571          ORA     C
075.337  312 040 076           6572          JZ      LVS4            NOT FOUND
                                6573
075.342  176                   6574          MOV     A,M
075.343  043                   6575          INX     H
075.344  272                   6576          CMP     D
075.345  176                   6577          MOV     A,M
075.346  302 363 075           6578          JNE     LVS2            NO MATCH
075.351  273                   6579          CMP     E
075.352  302 363 075           6580          JNE     LVS2            NO MATCH
                                6581
                                6582  *       HAVE A MATCH
                                6583
```

```
075.355  053            6584          DCX   H
075.356  353            6585          XCHG                DE = SYMTAB ADDRESS
075.357  257            6586          XRA   A             SET THE ZERO FLAG
075.360  301            6587          POP   B
075.361  341            6588          POP   H
075.362  311            6589          RET
                        6590
                        6591  *       HAVE NO MATCH
                        6592
075.363  013            6593  LVS2    DCX   B
075.364  013            6594          DCX   B
075.365  013            6595          DCX   B
075.366  013            6596          DCX   B
075.367  013            6597          DCX   B
075.370  013            6598          DCX   B             BC = BC - 6
                        6599
075.371  176            6600          MOV   A,M
075.372  043            6601          INX   H
075.373  346 002        6602          ANI   CF.VEC
075.375  312 027 076    6603          JZ    LVS3          IS NOT A VECTOR
                        6604
076.000  176            6605          MOV   A,M
076.001  247            6606          ANA   A
076.002  372 027 076    6607          JM    LVS3          IS JUST A FUNCTION
                        6608
                        6609  *       PROCESS A VECTOR
                        6610
076.005  043            6611          INX   H
076.006  043            6612          INX   H             SKIP 'DIM' AND '0' BYTES
                        6613
076.007  325            6614          PUSH  D
076.010  136            6615          MOV   E,M
076.011  043            6616          INX   H
076.012  126            6617          MOV   D,M
076.013  043            6618          INX   H             DE = ARRAY SIZE
                        6619
076.014  171            6620          MOV   A,C
076.015  223            6621          SUB   E
076.016  117            6622          MOV   C,A
076.017  170            6623          MOV   A,B
076.020  232            6624          SBB   D
076.021  107            6625          MOV   B,A           BC = BC - SIZE
                        6626
076.022  031            6627          DAD   D             HL = HL + ARRAY SIZE
076.023  321            6628          POP   D
                        6629
076.024  303 335 075    6630          JMP   LVS1
                        6631
                        6632  *       PROCESS A NORMAL SCALAR
                        6633
076.027  305            6634  LVS3    PUSH  B
076.030  001 004 000    6635          LXI   B,6-2
076.033  011            6636          DAD   B             HL = HL + 6-2
076.034  301            6637          POP   B
                        6638
076.035  303 335 075    6639          JMP   LVS1          DO IT AGAIN
```

```
                        6640
                        6641 *      PROCESS AN UNFOUND VARIABLE
                        6642
076.040 366 001         6643 LVS4   ORI     1               CLEAR ZERO FLAG
076.042 301             6644        POP     B
076.043 341             6645        POP     H
076.044 311             6646        RET

                        6648 **     MOVX - MOVE X BYTES OF DATA
                        6649 *
                        6650 *      MOVX CONSISTS OF TWO ROUTINES
                        6651 *
                        6652 *      MOV4 MOVES 4 BYTES OF DATA
                        6653 *
                        6654 *      MOV5 MOVES 5 BYTES OF DATA
                        6655 *
                        6656 *      ENTRY   (HL) = DESTINATION ADDRESS
                        6657 *              (DE) = SOURCE ADDRESS
                        6658 *      EXIT    (HL) = (HL) + COUNT
                        6659 *              (DE) = (DE) + COUNT
                        6660 *      USES    A,F,D,E,H,L
                        6661
                        6662
076.045                 6663 MOV5   EQU     *               ENTRY POINT TO MOVE 4 BYTES
076.045 032             6664        LDAX    D
076.046 167             6665        MOV     M,A
076.047 023             6666        INX     D
076.050 043             6667        INX     H
076.051 032             6668 MOV4   LDAX    D
076.052 167             6669        MOV     M,A
076.053 023             6670        INX     D
076.054 043             6671        INX     H
076.055 032             6672        LDAX    D
076.056 167             6673        MOV     M,A
076.057 023             6674        INX     D
076.060 043             6675        INX     H
076.061 032             6676        LDAX    D
076.062 167             6677        MOV     M,A
076.063 023             6678        INX     D
076.064 043             6679        INX     H
076.065 032             6680        LDAX    D
076.066 167             6681        MOV     M,A
076.067 023             6682        INX     D
076.070 043             6683        INX     H
076.071 311             6684        RET
```

```
                                6686  **      PNT - PREVIEW NEXT TOKEN.
                                6687  *
                                6688  *       PNT READS THE NEXT TEXT TOKEN. HOWEVER, THE TOKEN POINTER
                                6689  *       IS NOT ADVANCED, SO THAT IT CAN BE PREVIEWED OVER
                                6690  *       AND OVER, (AND ACCEPTED ONCE).
                                6691  *
                                6692  *       ENTRY   (BC) = TEXT POINTER
                                6693  *       EXIT    (BC) UPDATED
                                6694  *               (A) = TYPE
                                6695  *               (DE) = CODE (IF VARIABLE)
                                6696  *       USES    A,F (D,E IF VARIABLE)
                                6697
                                6698
076.072  076 000                6699  PNT     MVI     A,0             (A) = TYPE
076.073                         6700  PNTA    EQU     *-1             TYPE OF CURRENT TOKEN
076.074  376 300                6701          CPI     CT.VARL
076.076  332 111 076            6702          JC      PNT2            IS NOT VARIABLE
076.101  376 310                6703          CPI     CT.VARH+1
076.103  322 111 076            6704          JNC     PNT2            IS NOT VARIABLE
076.106  021 000 000            6705          LXI     D,0             (DE) = INDEX
076.107                         6706  PNTB    EQU     *-2
076.111                         6707  PNT2    EQU     *
076.111  000                    6708  PNTC    NOP                     'RET' IF VALUE IN PNTA, PNTB
076.112  315 131 054            6709          CALL    LEXCAL
076.115  353                    6710          XCHG
076.116  042 107 076            6711          SHLD    PNTB            SET INDEX
076.121  353                    6712          XCHG
076.122  062 073 076            6713          STA     PNTA            SET TYPE
076.125  365                    6714          PUSH    PSW
076.126  076 311                6715          MVI     A,MI.RET        VALUE IS IN PNTA, PNTB
076.130  062 111 076            6716  PNT1    STA     PNTC            SET FLAG
076.133  361                    6717          POP     PSW
076.134  311                    6718          RET


                                6720  **      PVI - PERFORM VALUE INPUT.
                                6721  *
                                6722  *       PVI READS A LIST OF VARIABLES FROM THE TEXT AT (BC), AND
                                6723  *       ASSIGNS THEM THE VALUES OF THE EXPRESSIONS AT (HL).
                                6724  *
                                6725  *       ENTRY   (BC) = VARIABLE LIST
                                6726  *               (HL) = TEXT LIST
                                6727  *       EXIT    (BC) UPDATED
                                6728  *               (HL) UPDATED
                                6729  *               'Z' SET IF VARIABLE LIST SATISFIED
                                6730  *       USES    ALL
                                6731
                                6732
076.135  315 072 076            6733  PVI     CALL    PNT             PEEK AT NEXT TOKEN
000.000                         6734          ERRNZ   CT.FIN
076.140  247                    6735          ANA     A
076.141  310                    6736          RZ                      NO MORE VARIABLES
076.142  326 020                6737          SUI     CT.PAR          SEE IF )
076.144  310                    6738          RZ                      NO MORE VARIABLES
```

```
076.145 315 330 111  6739          CALL    $SOB            SKIP BLANKS
076.150 176          6740          MOV     A,M             (A) = NEXT NON-BLANK
076.151 247          6741          ANA     A
076.152 312 236 076  6742          JZ      PVI3            NO DATA
                     6743
                     6744  *       WE KNOW WE HAVE DATA (OR A SPECIFED NULL VALUE)
                     6745
076.155 345          6746          PUSH    H
076.156 315 136 075  6747          CALL    IST             INSERT SYMBOL IN TABLE
076.161 341          6748          POP     H
076.162 365          6749          PUSH    PSW             SAVE TYPE
076.163 315 062 077  6750          CALL    RCE             REQUIRE DELIMITER, CLEAR RNT
076.166 361          6751          POP     PSW             (A) = TYPE OF VARIABLE
076.167 305          6752          PUSH    B
076.170 325          6753          PUSH    D               SAVE INDEX
076.171 365          6754          PUSH    PSW             SAVE TYPE
076.172 104          6755          MOV     B,H
076.173 115          6756          MOV     C,L             (BC) = VALUE LIST ADDRESS
076.174 012          6757          LDAX    B
076.175 376 054      6758          CPI     ','
076.177 302 210 076  6759          JNE     PVI1            IS NOT NULL VALUE
076.202 321          6760          POP     D
076.203 361          6761          POP     PSW             SKIP ASSIGNING VALUE
076.204 003          6762          INX     B
076.205 303 230 076  6763          JMP     PVI2
                     6764
                     6765  *       STORE VALUE.
                     6766
076.210 361          6767  PVI1    POP     PSW             (A) = TYPE
076.211 365          6768          PUSH    PSW
076.212 315 240 076  6769          CALL    PVI5            EVALUATE VALUE
076.215 315 062 077  6770          CALL    RCE             REQUIRE COMMA OR END
076.220 361          6771          POP     PSW             RESTORE TYPE
076.221 321          6772          POP     D               (DE) = VARIABLE POINTER
076.222 315 366 072  6773          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL TABLE
076.225 315 202 071  6774          CALL    AVV             ASSIGN VALUE TO VARIABLE
076.230              6775  PVI2    EQU     *
076.230 140          6776          MOV     H,B
076.231 151          6777          MOV     L,C             (BC) = VALUE LIST ADDRESS
076.232 301          6778          POP     B
076.233 303 135 076  6779          JMP     PVI             PROCESS ANOTHER
                     6780
                     6781  *       RAN OUT OF VALUES.
                     6782
076.236 264          6783  PVI3    ORA     H               CLEAR 'Z'
076.237 311          6784          RET
                     6785
                     6786  *       CRACK VALUE.
                     6787  *
                     6788  *       (A) = TYPE OF INPUT VARIABLE
                     6789
000.000              6790          ERRNZ   CF.STR-1        ASSUME 1 BIT FOR STRING
076.240 037          6791  PVI5    RAR
076.241 332 261 076  6792          JC      PVI7            IS STRING VARIABLE
                     6793
                     6794  *       REQUIRE NUMBER.
```

```
                                6795  *       IF NOT VALID NUMBER, *ATF* WONT LEAVE POINTER AT DELIMITER
                                6796
076.244  140                   6797  PVI6    MOV     H,B
076.245  151                   6798          MOV     L,C
076.246  315 323 107           6799          CALL    ATF             ASCII TO FLOATING
076.251  076 300               6800          MVI     A,CT.SNV
076.253  062 201 042           6801          STA     ACCX-1          SET SCALAR NUMBERIC VALUE
076.256  104                   6802          MOV     B,H
076.257  115                   6803          MOV     C,L             UPDATE (BC)
076.260  311                   6804          RET
                                6805
                                6806  *       MUST BE STRING. IF QUOTES, GOBBLE IT ALL. IF NONE, GO TO COMMA
                                6807
076.261  012                   6808  PVI7    LDAX    B               (A) = FIRST DATA CHARACTER
076.262  376 042               6809          CPI     '"'
076.264  003                   6810          INX     B               ASSUME HAVE QUOTE
076.265  312 325 076           6811          JE      PVI10           INPUT AS STRING
                                6812
                                6813  *       DOSENT HAVE QUOTES. COPY INTO LINE2, AND ADD THE QUOTES
                                6814
076.270  041 273 113           6815          LXI     H,LINE2
076.273  013                   6816          DCX     B               POINT TO 1ST CHARACTER
076.274  012                   6817  PVI8    LDAX    B
076.275  167                   6818          MOV     M,A
076.276  003                   6819          INX     B
076.277  043                   6820          INX     H
076.300  012                   6821          LDAX    B               CHECK NEXT CHARACTER
076.301  376 054               6822          CPI     ','
076.303  312 312 076           6823          JE      PVI9            GOT THE END
000.000                        6824          ERRNZ   CT.FIN
076.306  247                   6825          ANA     A
076.307  302 274 076           6826          JNZ     PVI8            NOT AT END OF LINE
                                6827
                                6828  *       ALL DONE COPYING STRING. ADD CLOSE QUOTE
                                6829
076.312  066 042               6830  PVI9    MVI     M,'"'
076.314  305                   6831          PUSH    B               SAVE (BC)
076.315  001 273 113           6832          LXI     B,LINE2
076.320  315 325 076           6833          CALL    PVI10           BUILD STRING
076.323  301                   6834          POP     B               RESTORE (BC)
076.324  311                   6835          RET
                                6836
076.325  315 015 055           6837  PVI10   CALL    LEX12           READ STRING TYPE
076.330  305                   6838          PUSH    B
076.331  001 005 000           6839          LXI     B,5
076.334  033                   6840          DCX     D               POINT TO VALUE-1
076.335  041 201 042           6841          LXI     H,ACCX-1
076.340  315 252 030           6842          CALL    $MOVE           MOVE DESCRIPTOR INTO ACCX
076.343  301                   6843          POP     B
076.344  311                   6844          RET
```

```
                                6846  **      PBO - PRESET BOOLEAN OPERATORS
                                6847  *
                                6848  *       PBO INSURES THAT BOTH VALUES ARE NUMERIC, AND THEN
                                6849  *       FIXES BOTH TO INTEGERS.
                                6850  *
                                6851  *       ENTRY   (ACCX) = VALUE 1
                                6852  *               (ACCY) = VALUE 2
                                6853  *       EXIT    (HL) = IFIX(ACCX)
                                6854  *               (DE) = IFIX(ACCY)
                                6855  *       USES    A,F,D,E,H,L
                                6856
                                6857
076.345  315 177 077           6858  PBO     CALL    RNO             REQUIRE NUMERIC OPERANDS
076.350  315 002 075           6859          CALL    IFIX            (DE) = IFIX(ACCX)
076.353  353                   6860          XCHG
076.354  303 377 074           6861          JMP     IFIX.           (DE) = IFIX(ACCX)


                                6863  **      POPX - POP VALUE INTO *X* ACCUMULATOR.
                                6864  *
                                6865  *       ENTRY   NONE
                                6866  *       EXIT    NONE
                                6867  *       USES    H,L
                                6868
                                6869
076.357  041 201 042           6870  POPX    LXI     H,ACCX-1
076.362  303 373 076           6871          JMP     POP


                                6873  **      POPY - POP VALUE INTO *Y* ACCUMULATOR.
                                6874  *
                                6875  *       ENTRY   NONE
                                6876  *       EXIT    (DE) = #ACCY
                                6877  *       USES    A,F,D,E,H,L
                                6878
                                6879
                                6880  **      POPY. - POP VALUE INTO *Y* ACCUMULATOR.
                                6881  *
                                6882  *       ENTRY   NONE
                                6883  *       EXIT    NONE
                                6884  *       USES    D,E,H,L
                                6885
                                6886
076.365  021 210 042           6887  POPY    LXI     D,ACCY
076.370  041 207 042           6888  POPY.   LXI     H,ACCY-1        STORE AREA
```

```
                                6890  **      POP - POP VALUE FROM WRKTAB.
                                6891  *
                                6892  *       ENTRY   (HL) = ADDRESS OF 5 BYTE AREA
                                6893  *       EXIT    DATA IN AREA
                                6894  *       USES    H,L
                                6895
                                6896
076,373  365                   6897  POP     PUSH    PSW             SAVE PSW
076,374  325                   6898          PUSH    D
076,375  345                   6899          PUSH    H
076,376  052 105 112           6900          LHLD    WRKTAB+MT.LEN
077,001  021 373 377           6901          LXI     D,-5
077,004  031                   6902          DAD     D
077,005  042 105 112           6903          SHLD    WRKTAB+MT.LEN   DECREASE SIZE
077,010  322 117 070           6904          JNC     ERR.DO          SHOULD NOT OCCUR
077,013  353                   6905          XCHG
077,014  052 103 112           6906          LHLD    WRKTAB+MT.FWA
077,017  031                   6907          DAD     D               (HL) = ABS ADDRESS OF 5 BYTES
077,020  353                   6908          XCHG
077,021  341                   6909          POP     H               (HL) = TO
077,022  315 045 076           6910          CALL    MOV5            MOVE DATA
077,025  321                   6911          POP     D
077,026  361                   6912          POP     PSW
077,027  311                   6913          RET


                                6915  **      PSHX - PUSH (ACCX) ONTO STACK.
                                6916  *
                                6917  *       ENTRY   NONE
                                6918  *       USES    A,F,D,E,H,L
                                6919
                                6920
077,030  315 056 071           6921  PSHX.   CALL    ANT             ACCEPT OPERATION
077,033  041 201 042           6922  PSHX    LXI     H,ACCX-1
077,036  303 044 077           6923          JMP     PSH             PUSH ACCX
                                6924
                                6925
                                6926  **      PSHY - PUSH (ACCY) ONTO WORK STACK.
                                6927  *
                                6928  *       ENTRY   NONE
                                6929  *       EXIT    (ACCY) ON STACK
                                6930  *       USES    A,F,D,E,H,L
                                6931
                                6932
077,041  041 207 042           6933  PSHY    LXI     H,ACCY-1
                                6934
                                6935
                                6936  **      PSH - PUSH MEMORY VALUE INTO WORK STAC.
                                6937  *
                                6938  *       ENTRY   (HL) = ADDRESS OF 5 BYTES
                                6939  *       EXIT    ON WRKTAB
                                6940  *       USES    A,F,D,E,H,L
                                6941
                                6942
```

```
077.044  345            6943  PSH     PUSH    H
077.045  041 005 000    6944          LXI     H,5
077.050  021 103 112    6945          LXI     D,WRKTAB+1
077.053  315 026 071    6946          CALL    AMB             ALLOCATE 5 BYTES
077.056  321            6947          POP     D               (DE) = FROM
077.057  303 045 076    6948          JMP     MOV5            COPY AND EXIT


                        6950  **      RCE - REQUIRE COMMA OR END.
                        6951  *
                        6952  *       RCE REQUIRES EITHER A COMMA, OR END OF STATEMENT.
                        6953  *
                        6954  *       ENTRY   (BC) = TEXT POINTER
                        6955  *       EXIT    TO *RET* IF OK
                        6956  *                (BC) UPDATED
                        6957  *                (A) = TYPE CODE
                        6958  *                TO ERR.SY IF NOT ',' OR CT.FIN
                        6959  *       USES    A,F,B,C
                        6960
                        6961
077.062  315 056 071    6962  RCE     CALL    ANT             ACCEPT NEXT TOKEN
077.065  247            6963          ANA     A
000.000                 6964          ERRNZ   CT.FIN
077.066  310            6965          RZ                      IS FIN
077.067  376 026        6966          CPI     CT.CMA
077.071  310            6967          RE                      COMMA
077.072  303 152 070    6968          JMP     ERR.SY          SYNTAX ERROR


                        6970  **      RIL - READ INPUT LINE.
                        6971  *
                        6972  *       RIL READS A LINE FROM THE SYSTEM CONSOLE.
                        6973  *
                        6974  *       ENTRY   (HL) = LINE FWA
                        6975  *       EXIT    'C' SETE IF CTL-C STRUCK
                        6976  *               'C' CLEAR IF GOT LINE
                        6977  *                (A) = LINE LENGTH
                        6978  *       USES    A,F,D,E
                        6979
                        6980
077.075  345            6981  RIL     PUSH    H               SAVE START ADDRESS
077.076  072 142 112    6982  RIL1    LDA     CTLFLAG
000.000                 6983          ERRNZ   CFCTLC-1
077.101  037            6984          RAR
077.102  332 137 077    6985          JC      RIL3            CTL-C
077.105  377 001        6986          DB      SYSCALL,.SCIN
077.107  332 076 077    6987          JC      RIL1
077.112  376 012        6988          CPI     NL
077.114  302 120 077    6989          JNE     RIL2            NOT END OF LINE
077.117  257            6990          XRA     A               USE 00 AS END OF LINE
077.120  167            6991  RIL2    MOV     M,A
077.121  043            6992          INX     H
```

```
077.122  302 076 077  6993          JNZ     RIL1            MORE TO GO
077.125  074          6994          INR     A               (A) = 1
077.126  062 253 112  6995          STA     COLCNTS         SET CONSOLE COLUMN AT FRONT OF LINE
077.131  321          6996          POP     D               (DE) = LINE FWA
077.132  175          6997          MOV     A,L             (A) = LENGTH OF LINE
077.133  223          6998          SUB     E
077.134  247          6999          ANA     A               CLEAR CARRY
077.135  353          7000          XCHG                    (HL) = LINE FWA
077.136  311          7001          RET
                      7002
                      7003  *       CTL-C HIT
                      7004
077.137  341          7005  RIL3    POP     H
077.140  311          7006          RET


                      7008  **      RLF - READ LINE FROM FILE.
                      7009  *
                      7010  *       RLF READS A LINE FROM THE FILE NUMBER SPECIFIED IN IOCHAN.
                      7011  *       THIS MAY BE THE CONSOLE, OR IT MAY BE A FILE BLOCK.
                      7012  *
                      7013  *       ENTRY   (HL) = LINE ADDRESS
                      7014  *       EXIT    'C' SET IF CTL-C (WAS CONSOLE INPUT)
                      7015  *       USES    A,F,D,E
                      7016
                      7017
077.141  072 140 112  7018  RLF     LDA     IOCHAN
077.144  247          7019          ANA     A
077.145  312 075 077  7020          JZ      RIL             IS CONSOLE, READ LINE
                      7021
                      7022  *       IS FROM FILE
                      7023
077.150  345          7024          PUSH    H               SAVE TEXT FWA
077.151  075          7025          DCR     A
077.152  315 005 072  7026          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
077.155  332 210 070  7027          JC      ERR.FNO         FILE NOT OPEN
077.160  321          7028          POP     D               (DE) = LINE FWA
077.161  305          7029          PUSH    B               SAVE BC
077.162  001 005 001  7030          LXI     B,LINEL+6       MAX.CHAR.TO.READ+6  FOR LINE #  /78.10.GC/
077.165  325          7031          PUSH    D               SAVE LINE FWA
077.166  315 161 101  7032          CALL    $FREAL          READ LINE
077.171  332 213 070  7033          JC      ERR.EOF         EOF ON DEVICE
077.174  341          7034          POP     H               (HL) = LINE FWA
077.175  301          7035          POP     B               RESTORE (BC)
077.176  311          7036          RET
```

```
                                7038  **      RND - REQUIRE NUMERIC OPERANDS.
                                7039  *
                                7040  *       RND REQUIRES THAT (ACCX) AND (ACCY) ARE BOTH NUMERIC.
                                7041  *
                                7042  *       ENTRY   NONE
                                7043  *       EXIT    TO *RET* IF NUMERIC
                                7044  *               TO ERR.TE IF NOT
                                7045  *       USES    A,F,H,L
                                7046
                                7047
077.177  315 347 072            7048  RND     CALL    COT
077.202  310                    7049          RZ                      NUMERIC
077.203  303 155 070            7050          JMP     ERR.TC          TYPE ERROR



                                7052  **      RNP - READ NEW PROGRAM.
                                7053  *
                                7054  *       RNP IS CALLED TO READ A NEW SOURCE PROGRAM INTO THE TEXT TABLE (TXTTAB)
                                7055  *
                                7056  *       ALL TXTTAB DEPENDANT TABLES ARE CLEARED.
                                7057  *
                                7058  *       RNP EXPECTS THAT THE PROPER FILE NAME IS ALREADY INSTALLED
                                7059  *       IN THE FIRST FILE BLOCK IN FILTAB. NOTE THAT AS THE PROGRAM TEXT
                                7060  *       IS INSERTED, FILTAB MAY MOVE BETWEEN LINES.  THUS, RNP RE-LOADS
                                7061  *       THIS ADDRESS BEFORE EVERY OPERATION.
                                7062  *
                                7063  *       ENTRY   NONE
                                7064  *       EXIT    (BC) = $ZERO
                                7065  *       USES    ALL
                                7066
                                7067
077.206  315 320 077            7068  RNP     CALL    SCRA            CLEAR TEXT TABLE
077.211  315 104 073            7069          CALL    CUF             CLEAR USER-DEFINED FUNCTIONS FROM SYMTAB
077.214  315 021 045            7070          CALL    CLR1            CLEAR TEXT TABLE REFERENCES
077.217  315 217 074            7071          CALL    FOP             FILE OPEN PRESET
077.222  041 230 042            7072          LXI     H,FBLIST        (HL) = TABLE FWA
077.225  021 072 043            7073          LXI     D,DEFALTF
077.230  315 021 101            7074          CALL    $FOPER          OPEN FOR READ
077.233  315 115 074            7075          CALL    FDC             RESTORE MEMORY SPACE
077.236  001 005 001            7076  RNP1    LXI     B,LINEL+6
077.241  021 266 112            7077          LXI     D,LINE+1
077.244  041 230 042            7078          LXI     H,FBLIST        (HL) = FB FWA
077.247  325                    7079          PUSH    D               SAVE ADDRESS
077.250  315 161 101            7080          CALL    $FREAL          READ LINE INTO BUFFER+1
077.253  341                    7081          POP     H               (HL) = $BUFFER+1
077.254  332 273 077            7082          JC      RNP2            ALL DONE
077.257  315 373 065            7083          CALL    ICL.            COMPRESS LINE INTO BUFFER+0
077.262  332 273 077            7084          JC      RNP2            CTL-C HIT
077.265  315 270 070            7085          CALL    MTL             MERGE TEXT LINEE
077.270  303 236 077            7086          JMP     RNP1            GET NEXT
                                7087
                                7088  *       END OF TEXT
                                7089
077.273  041 230 042            7090  RNP2    LXI     H,FBLIST        (HL) = FB FWA
```

```
077.276  315 335 102  7091          CALL    *FCLO           CLOSE INPUT FILE
077.301  001 345 114  7092          LXI     B,ZERO          (BC) = TEXT POINTER = NO MORE
077.304  311          7093          RET                     EXIT


                      7095  **      RNT - REQUIRE NEXT TOKEN.
                      7096  *
                      7097  *       RNT CHECSK TO SEE IF THE NEXT TOKEN IS THE REQUIRED VALUE,
                      7098  *       AND FLAGS A SYNTAX ERROR IF NOTL
                      7099  *
                      7100  *       ENTRY   (RET) = REQUIRED TOKEN
                      7101  *       EXIT    TO (RET)+1 IF MATCH
                      7102  *                (DE) = SYMBOL POINTER (IF VARIABLE)
                      7103  *                (A) = VARIABLE TYPE
                      7104  *               TO ERR.SY IF NOT
                      7105  *       USES    A,F, (DE, IF VARIABLE)
                      7106
                      7107
077.305  315 056 071  7108  RNT     CALL    ANT             ACCEPT NEXT TOKEN
077.310  343          7109          XTHL
077.311  276          7110          CMP     M
077.312  043          7111          INX     H
077.313  343          7112          XTHL
077.314  310          7113          RE                      OK
077.315  303 152 070  7114          JMP     ERR.SY          NO GOOD


                      7116  **      SCRA - SCRATCH TEXT BUFFER
                      7117  *
                      7118  *       SCRA INSERTS THE DUMMY LAST LINE
                      7119  *       AT THE BEGINNING OF THE BUFFER
                      7120  *
                      7121  *       ENTRY   NONE
                      7122  *       EXIT    (BC) = $0 BYTE
                      7123  *       USES     A,F,B,C,H,L
                      7124
                      7125
077.320  041 003 000  7126  SCRA    LXI     H,3             SCRATCH STORE
077.323  042 061 112  7127          SHLD    TXTTAB+MT.LEN           LENGTH = 3
077.326  041 377 377  7128          LXI     H,377377A
077.331  042 346 114  7129          SHLD    MTAREA
077.334  001 350 114  7130          LXI     B,MTAREA+2
077.337  257          7131          XRA     A
077.340  002          7132          STAX    B               CLEAR TEXT, SET ((BC)) = 0
                      7133
077.341  311          7134          RET                     EXIT
```

```
                                7136  **      SES - SKIP TO END OF STATEMENT.
                                7137  *
                                7138  *       SES SKIPS OVER TEXT UNTIL AN END-OF-LINE IS DETECTED
                                7139  *
                                7140  *       ENTRY   (BC) = TEXT POINTER
                                7141  *       EXIT    (BC) UPATED
                                7142  *       USES    A,F,B,C
                                7143
                                7144
077,342  315 056 071           7145  SES     CALL    ANT             ACCEPT NEXT TOKEN
000.000                         7146          ERRNZ   CT.FIN
077,345  247                   7147          ANA     A
077,346  302 342 077           7148          JNZ     SES             NOT YET
077,351  311                   7149          RET


                                7151  **      SFS - SEARCH 'FOR' STACK.
                                7152  *
                                7153  *       SFS SEARCHES A FOR STACK FOR AN ENTRY MATCHEDING A
                                7154  *       SUPPLIED ONE.
                                7155  *
                                7156  *
                                7157  *       ENTRY   (DE) = INDEX VARIABLE INDEX
                                7158  *
                                7159  *       EXIT    'Z' SET IF FOUND
                                7160  *               (HL) = INDEX OF IND+2 (IF ANY)
                                7161  *               (DE) = ABS. ADDRESS OF INDEX IN SYMTAB
                                7162  *
                                7163  *       USES    A,F,H,L
                                7164  *
                                7165
077,352  315 072 076           7166  SFS.    CALL    PNT             ALLOW NULL AS THE LAST IN STACK
000.000                         7167          ERRNZ   CT.FIN
077,355  127                   7168          MOV     D,A             ASSUME (A) = 0 = CT.FIN
077,356  137                   7169          MOV     E,A
077,357  312 375 077           7170          JZ      SFS0            (DE) = 0 = INDEX
                                7171
077,362  315 136 075           7172  SFS     CALL    IST             INSERT SYMBOL IN TABLE
077,365  376 300                7173          CPI     CT.SNV
077,367  302 152 070           7174          JNE     ERR.SY          MUST BE SCALAR NUMERIC VARIABLE
077,372  315 366 072           7175          CALL    CSA             DE = ABS. SYMTAB ADDRESS        /80.01.GC/
                                7176
077,375  305                   7177  SFS0    PUSH    B               SAVE TEXT POINTER
077,376  052 073 112           7178          LHLD    FORTAB+MT.LEN
100.001  104                   7179          MOV     B,H
100.002  115                   7180          MOV     C,L
100.003  052 071 112           7181          LHLD    FORTAB+MT.FWA   (HL) = TABLE FWA
100.006  345                   7182          PUSH    H
100.007  011                   7183          DAD     B               (HL) = LWA+1
                                7184
100.010  172                   7185          MOV     A,D                                     /80.01.GC/
100.011  263                   7186          ORA     E                                       /80.01.GC/
100.012  312 025 100           7187          JZ      SFS1                                    /80.01.GC/
                                7188
```

```
100.015 353          7189          XCHG
100.016 053          7190          DCX   H                                    /80.01.GC/
100.017 053          7191          DCX   H                                    /80.01.GC/
100.020 176          7192          MOV   A,M                                  /80.01.GC/
100.021 043          7193          INX   H                                    /80.01.GC/
100.022 156          7194          MOV   L,M                                  /80.01.GC/
100.023 147          7195          MOV   H,A                                  /80.01.GC/
100.024 353          7196          XCHG                 DE = SYMBOL KEY       /80.01.GC/
                     7197
100.025 170          7198  SFS1    MOV   A,B
100.026 261          7199          ORA   C              CHECK COUNT
100.027 312 073 100  7200          JZ    SFS3           NOT FOUND
100.032 305          7201          PUSH  B
100.033 001 365 377  7202          LXI   B,-11
100.036 011          7203          DAD   B              (HL) = ADDRESS OF LAST ELEMENT
100.037 301          7204          POP   B
100.040 172          7205          MOV   A,D
100.041 263          7206          ORA   E
100.042 176          7207          MOV   A,M
100.043 053          7208          DCX   H
100.044 312 120 100  7209          JZ    SFS6           WILL TAKE THE LAST
100.047 273          7210          CMP   E              SEE IF FOUND          /80.01.GC/
100.050 302 060 100  7211          JNE   SFS2           NOT FOUND
100.053 176          7212          MOV   A,M
100.054 272          7213          CMP   D                                    /80.01.GC/
100.055 312 074 100  7214          JE    SFS4           FOUND
100.060 171          7215  SFS2    MOV   A,C
100.061 326 014      7216          SUI   12
100.063 117          7217          MOV   C,A
100.064 170          7218          MOV   A,B            COUNT = COUNT-12
100.065 336 000      7219          SBI   0
100.067 107          7220          MOV   B,A
100.070 303 025 100  7221          JMP   SFS1           TRY AGAIN
                     7222
                     7223  *       NOT FOUND.
                     7224
100.073 264          7225  SFS3    ORA   H
                     7226
                     7227  *       FOUND. COMPUTE FORTAB INDEX FROM ABS.
                     7228
100.074 043          7229  SFS4    INX   H
100.075 043          7230  SFS5    INX   H
100.076 104          7231          MOV   B,H
100.077 115          7232          MOV   C,L            (BC) = ABS ADDRESS
100.100 341          7233          POP   H              (HL) = FORTAB FWA
100.101 365          7234          PUSH  PSW            SAVE CODE
100.102 171          7235          MOV   A,C
100.103 225          7236          SUB   L
100.104 157          7237          MOV   L,A
100.105 170          7238          MOV   A,B
100.106 234          7239          SBB   H
100.107 147          7240          MOV   H,A
100.110 315 323 075  7241          CALL  LVS            GET AN ABS. ADDRESS   /80.01.GC/
100.113 023          7242          INX   D                                    /80.01.GC/
100.114 023          7243          INX   D                                    /80.01.GC/
100.115 361          7244          POP   PSW            RESTORE CODE
```

```
100.116  301            7245          POP     B               RESTORE (BC)
100.117  311            7246          RET
                        7247
                        7248  *       NO VARIABLE SUPPLIED, JUST TAKE THE LAST ONE.
                        7249
100.120  126            7250  SFS6    MOV     D,M                                     /80.01.GC/
100.121  043            7251          INX     H
100.122  136            7252          MOV     E,M             (DE) = VARIABLE INDEX   /80.01.GC/
100.123  303 075 100     7253          JMP     SFS5



                        7255  **      SOB - SKIP OVER BLANKS.
                        7256  *
                        7257  *       ENTRY   (BC) = TEXT POINTER
                        7258  *       EXIT    (BC) = ADDRESS OF NEXT NON-BLANK CHARACTER
                        7259  *       USES    A,F,B,C
                        7260
                        7261
100.126  012            7262  SOB     LDAX    B
100.127  376 040        7263          CPI     ' '
100.131  312 137 100     7264          JE      SOB1            IS BLANK
100.134  376 011        7265          CPI     TAB
100.136  300            7266          RNE                     NOT TAB, EITHER
100.137  003            7267  SOB1    INX     B
100.140  303 126 100     7268          JMP     SOB



                        7270  **      SRA - STACK RETURN ADDRESS.
                        7271  *
                        7272  *       SRA STACKS THE TEXT RETURN ADDRESS (END OF CURRENT STATEMENT)
                        7273  *       AND THE CURRENT LINE NUMBER ON STACK 'GOSTAB'.
                        7274  *
                        7275  *       ENTRY   (BC) = TEXT POINTER
                        7276  *       EXIT    (BC) UNCHANGED.
                        7277  *       USES    A,F,D,E
                        7278
                        7279
100.143  305            7280  SRA     PUSH    B               SAVE TEXT ADDRESS
100.144  345            7281          PUSH    H               SAVE (HL)
100.145  315 342 077     7282          CALL    SES             SKIP TO END OF STATEMENT
100.150  041 004 000     7283          LXI     H,4
100.153  021 076 112     7284          LXI     D,GOSTAB+1
100.156  315 026 071     7285          CALL    AMB             ALLOCATE ROOM
100.161  353            7286          XCHG
100.162  052 133 112     7287          LHLD    CURNUM          (HL) = CURRENT LINE NUMBER
100.165  353            7288          XCHG
100.166  161            7289          MOV     M,C             SAVE RETURN ADDRESS
100.167  043            7290          INX     H
100.170  160            7291          MOV     M,B
100.171  043            7292          INX     H
100.172  163            7293          MOV     M,E             SET CURNUM
100.173  043            7294          INX     H
```

```
100.174  162              7295          MOV     M,D
100.175  341              7296          POP     H
100.176  301              7297          POP     B               RESTORE (BC)
100.177  311              7298          RET


                          7300  **      TCS - TYPE CHARACTER STRING.
                          7301  *
                          7302  *       TCS TYPES A CHARACTER STRING ON THE CONSOLE.
                          7303  *
                          7304  *       ENTRY   (DE) = STRING DESCRIPTOR
                          7305  *       EXIT    TYPED
                          7306  *       USES    A,F,D,E,H,L
                          7307
                          7308
100.200                   7309  TCS     EQU     *
100.200  315 315 074     7310          CALL    FSE             FIND STRINGTAB ENTRY
100.203  303 251 100     7311          JMP     WLF.            WRITE LINE TO FILE


                          7313  **      TDI - TYPE DECIMAL INTEGER.
                          7314  *
                          7315  *       TDI TYPES AN INTEGER AS A 5 PLACE NUMBER. LEADING ZEROS ARE
                          7316  *       SUPPRESSED.
                          7317  *
                          7318  *       ENTRY   (DE) = NUMBER
                          7319  *       EXIT    TYPED
                          7320  *       USES    A,F,D,E
                          7321
                          7322
100.206  345              7323  TDI     PUSH    H
100.207  315 040 075     7324          CALL    IFLT            FLOAT IT
100.212  041 273 113     7325          LXI     H,LINE2
100.215  315 237 110     7326          CALL    FTA             FLOATING TO ASCII
100.220  315 217 103     7327          CALL    $TYPCC          TYPE NUMBER MINUS .
100.223  341              7328          POP     H
100.224  311              7329          RET


                          7331  **      WEL - WRITE END OF LINE.
                          7332  *
                          7333  *       WEL WRITES AN END OF LINE CHARACTER TO THE CURRENT OUTPUT FILE.
                          7334  *
                          7335  *       ENTRY   NONE
                          7336  *       EXIT    NONE
                          7337  *       USES    A,F,D,E,H,L
                          7338
                          7339
100.225  315 302 075     7340  WEL     CALL    LCC
100.230  257              7341          XRA     A
```

```
100.231  167            7342          MOV     M,A             SET COLUMN # = 1(-1 FOR THE NL CHARACTER)
100.232  041 241 100    7343          LXI     H,WELA
100.235  074            7344          INR     A               (A) = 1
100.236  303 251 100    7345          JMP     WLF.            WRITE CHARACTER AND EXIT
                        7346
100.241  012            7347  WELA    DB      NL


                        7349  **      WLF - WRITE LINE TO FILE.
                        7350  *
                        7351  *       WLF WRITES A LINE IN 'LINE2' TO THE INDICATED (IOCHAN) FILE.
                        7352  *       THIS MAY BE THE SYSTEM CONSOLE, AND THE 'LINE' MAY NOT
                        7353  *       BE A COMPLETE LINE (I.E., HAVE NO NL CHARACTER)
                        7354  *
                        7355  *       ENTRY   (LINE2) = TEXT
                        7356  *       EXIT    NONE
                        7357  *       USES    A,F,D,E,H,L
                        7358
                        7359
100.242  041 273 113    7360  WLF     LXI     H,LINE2
100.245  315 273 111    7361          CALL    $CLL            COMPUTE LINE LENGTH
100.250  075            7362          DCR     A               REMOVE 00 BYTE COUNT
                        7363
                        7364  **      WLF. - WRITE LINE TO FILE.
                        7365  *
                        7366  *       ENTRY   (A) = LINE LENGTH
                        7367  *               (HL) = LINE FWA
                        7368  *       EXIT    NONE
                        7369  *       USES    A,F,D,E,H,L
                        7370
                        7371
100.251  305            7372  WLF.    PUSH    B               SAVE (BC)
100.252  117            7373          MOV     C,A
100.253  006 000        7374          MVI     B,0             (BC) = COUNT
100.255  353            7375          XCHG                    (DE) = TEXT ADDRESS
100.256  315 302 075    7376          CALL    LCC             LOCATE COLCNT FOR THIS CHANNEL
100.261  176            7377          MOV     A,M
100.262  201            7378          ADD     C
100.263  167            7379          MOV     M,A             UPDATE COLUME NUMBER
100.264  353            7380          XCHG                    (HL) = TEXT ADDRESS
100.265  072 140 112    7381          LDA     IOCHAN
100.270  247            7382          ANA     A
100.271  302 301 100    7383          JNZ     WLF2            WRITE TO DISK
                        7384
                        7385  *       TO CONSOLE
                        7386
100.274  171            7387          MOV     A,C
100.275  301            7388          POP     B               RESTORE (BC)
100.276  303 217 103    7389          JMP     $TYPCC          WRITE TO CONSOLE AND EXIT
                        7390
                        7391  *       WRITE TO DISK BUFFER. LOCATE FILE BLOCK
                        7392
100.301  345            7393  WLF2    PUSH    H               SAVE TEXT FWA
100.302  075            7394          DCR     A               (A) = FILE BLOCK NUMBER
```

```
100.303  315 005 072  7395          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
100.306  332 210 070  7396          JC      ERR,FNO         FILE NOT OPEN
100.311  321          7397          POP     D               (DE) = DATA FWA
100.312  315 047 102  7398          CALL    $FWRIB          WRITE DATA TO BUFFER
100.315  301          7399          POP     B               RESTORE (BC)
100.316  311          7400          RET


                      7402  **      XCY - EXCHANGE (ACCX) WITH (ACCY)
                      7403  *
                      7404  *       ENTRY   NONE
                      7405  *       EXIT    NONE
                      7406  *       USES    A,F
                      7407
                      7408
100.317  305          7409  XCY     PUSH    B
100.320  325          7410          PUSH    D
100.321  345          7411          PUSH    H
100.322  021 201 042  7412          LXI     D,ACCX-1
100.325  041 207 042  7413          LXI     H,ACCY-1
100.330  016 005      7414          MVI     C,5             (C) = BYTE COUNT
100.332  032          7415  XCY1    LDAX    D
100.333  106          7416          MOV     B,M
100.334  167          7417          MOV     M,A             (A) = NEXT BYTE IN LIST
100.335  170          7418          MOV     A,B
100.336  022          7419          STAX    D
100.337  023          7420          INX     D
100.340  043          7421          INX     H
100.341  015          7422          DCR     C
100.342  302 332 100  7423          JNZ     XCY1            MORE TO GO
100.345  341          7424          POP     H
100.346  321          7425          POP     D
100.347  301          7426          POP     B
100.350  311          7427          RET


                      7429  **      ZRO - ZERO MEMORY.
                      7430  *
                      7431  *       ZRO ZEROS A FIELD OF MEMORY.
                      7432  *
                      7433  *       ENTRY   (HL) = ADDRESS
                      7434  *               (DE) = COUNT
                      7435  *       EXIT    NONE
                      7436  *       USES    A,F,D,E,H,L
                      7437
                      7438
100.351  172          7439  ZRO     MOV     A,D
100.352  263          7440          ORA     E
100.353  310          7441          RZ                      DONE
100.354  066 000      7442          MVI     M,0
100.356  043          7443          INX     H
100.357  033          7444          DCX     D
```

```
100.360  303 351 100  7445          JMP     ZRO
```

```
                                7448  ***     THE FOLLOWING SUBROUTINES ARE ENVOKED BY INTERRUPTS.
                                7449  *
                                7450


                                7452  **      CCINT - PROCESS CTL-C INTERRUPT.
                                7453  *
                                7454  *       ENTRY   NONE
                                7455  *       EXIT    TO CALLER (INTERRUPT)
                                7456  *       USES    A,F
                                7457
                                7458
      100.363  076 001          7459  CCINT   MVI     A,CFCTLC
      100.365  303 372 100      7460          JMP     CBINT1          PROCESS AS CTL-B


                                7462  **      CBINT - CONTROL-B INTERRUPT.
                                7463  *
                                7464  *       ENTRY   NONE
                                7465  *       EXIT    NONE
                                7466  *       USES    A,F
                                7467
                                7468
      100.370  076 002          7469  CBINT   MVI     A,CFCTLB
      100.372  345              7470  CBINT1  PUSH    H
      100.373  365              7471          PUSH    PSW             SAVE FLAG BIT
      100.374  315 136 031      7472          CALL    $TYPTX
      100.377  336              7473          DB      '^'+200Q
      101.000  361              7474          POP     PSW
      101.001  365              7475          PUSH    PSW             (A) = CODE
      000.000                   7476          ERRNZ   CFCTLB-2        02 IF CTL-B
      000.000                   7477          ERRNZ   CFCTLC-1        01 IF CTL-C
      101.002  366 002          7478          ORI     2               =3 IF CTLC, =2 IF CTL-B
      101.004  306 100          7479          ADI     '@'
      101.006  315 241 103      7480          CALL    $WCHAR          ECHO CHARACTER
      101.011  361              7481          POP     PSW             (A) = CODE
      101.012  041 142 112      7482          LXI     H,CTLFLAG
      101.015  266              7483          ORA     M
      101.016  167              7484          MOV     M,A
      101.017  341              7485          POP     H
      101.020  311              7486          RET                     RETURN
```

```
101.021                 7489           XTEXT   FOPE


                        7491X **       $FOPEx - OPEN FILE BLOCK FOR I/O
                        7492X *
                        7493X *        $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                        7494X *        FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                        7495X *        THE FILE VIA *HDOS*.
                        7496X *
                        7497X *        ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                        7498X *                (HL) = ADDRESS OF FILE BLOCK
                        7499X *        EXIT    TO $FERROR IF ERROR
                        7500X *                TO CALLER IF OK
                        7501X *        USES    A,F,B,C,D,E
                        7502X
                        7503X
101.021  315 046 101   7504X $FOPER  CALL    $FOPER.
101.024  320           7505X         RNC
101.025  303 223 070   7506X         JMP     $FERROR         IN ERROR
                        7507X
101.030  315 051 101   7508X $FOPEW  CALL    $FOPEW.
101.033  320           7509X         RNC
101.034  303 223 070   7510X         JMP     $FERROR         IN ERROR
                        7511X
101.037  315 054 101   7512X $FOPEU  CALL    $FOPEU.
101.042  320           7513X         RNC
101.043  303 223 070   7514X         JMP     $FERROR         IN ERROR
                        7515X
                        7516X
101.046  076 002       7517X $FOPER. MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
101.050  001           7518X         DB      001Q            LXI,B TO SKIP NEXT MVI
101.051  076 004       7519X $FOPEW. MVI     A,FT.OW         OPEN FOR WRITE
101.053  001           7520X         DB      001Q            LXI,B TO SKIP NEXT MIV
101.054  076 006       7521X $FOPEU. MVI     A,FT.OR+FT.OW
                        7522X
                        7523X *        (A) = FILE FLAGS
                        7524X
101.056  345           7525X         PUSH    H               SAVE FILE BLOCK ADDRESS
101.057  365           7526X         PUSH    PSW             SAVE NEW FLAGS
000.000                 7527X         ERRNZ   FB.CHA
101.060  106           7528X         MOV     B,M             (B) = CHANNEL NUMBER
101.061  305           7529X         PUSH    B               SAVE HANNEL NUMBER
000.000                 7530X         ERRNZ   FB.FLG-FB.CHA-1
101.062  043           7531X         INX     H
101.063  117           7532X         MOV     C,A             (C) = NEW FILE FLAGS
101.064  176           7533X         MOV     A,M             (A) = CURRENT TYPE
101.065  247           7534X         ANA     A
101.066  171           7535X         MOV     A,C             (A) = NEW FLAGS TO BE SET
101.067  312 101 101   7536X         JZ      $FOPE1          NOT ALREADY OPEN
                        7537X
                        7538X *        ALREADY OPEN. SQUACK
                        7539X
101.072  301           7540X         POP     B               RESTORE (BC)
101.073  361           7541X         POP     PSW             DISCARD NEW FLAGS
```

```
101.074  341          7542X          POP     H               (HL) = FB ADDRESS
101.075  076 031      7543X          MVI     A,EC.FAO        FILE ALREADY OPEN
101.077  067          7544X          STC
101.100  311          7545X          RET
                       7546X
000.000               7547X          ERRNZ   FB.FWA-FB.FLG-1
101.101  043          7548X $FOPE1   INX     H               (HL) = $FB.FWA
101.102  116          7549X          MOV     C,M
101.103  043          7550X          INX     H
101.104  106          7551X          MOV     B,M             (BC) = FB.FWA
101.105  043          7552X          INX     H
000.000               7553X          ERRNZ   FB.PTR-FB.FWA-2
101.106  161          7554X          MOV     M,C             SET FB.PTR = FB.FWA
101.107  043          7555X          INX     H
101.110  160          7556X          MOV     M,B
101.111  043          7557X          INX     H
000.000               7558X          ERRNZ   FB.LIM-FB.PTR-2
101.112  161          7559X          MOV     M,C             SET FB.LIM = FB.FWA
101.113  043          7560X          INX     H
101.114  160          7561X          MOV     M,B
101.115  043          7562X          INX     H
000.000               7563X          ERRNZ   FB.NAM-FB.LIM-4
101.116  043          7564X          INX     H
101.117  043          7565X          INX     H               (HL) = $FB.NAM
                       7566X
                       7567X *        FILE BLOCK POINTERS SETUP, OPEN FILE
                       7568X
101.120  345          7569X          PUSH    H               SAVE NEW ADDRESS FOR NAME
101.121  041 152 101  7570X          LXI     H,$FOPEB
101.124  247          7571X          ANA     A                                       /78.10.GC/
101.125  312 134 101  7572X          JZ      $FOPE2
000.000               7573X          ERRNZ   .EXIT
101.130  315 371 111  7574X          CALL    $TBLS           FIND CODE
101.133  176          7575X          MOV     A,M
101.134  062 142 101  7576X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
101.137  341          7577X          POP     H               (HL) = $FB.NAM
101.140  361          7578X          POP     PSW             (A) = CHANNEL NUMBER
101.141  377 000      7579X          DB      SYSCALL,.EXIT
101.142               7580X $FOPEA   EQU     *-1             SYSCALL CODE
101.143  321          7581X          POP     D               (D) = NEW FLAG
101.144  341          7582X          POP     H               (HL) = FILE BLOCK ADDRESS
101.145  330          7583X          RC                      EXIT IF ERROR
101.146  043          7584X          INX     H
000.000               7585X          ERRNZ   FB.FLG-1
101.147  162          7586X          MOV     M,D             SET NEW FLAGS
101.150  053          7587X          DCX     H               RESTORE (HL)
101.151  311          7588X          RET
                       7589X
101.152  002 042      7590X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
101.154  004 043      7591X          DB      FT.OW,.OPENW
101.156  006 044      7592X          DB      FT.OR+FT.OW,.OPENU
101.160  000          7593X          DB      0               SHOULD NOT OCCUR
101.161               7594           XTEXT   FREAL
```

```
                                7596X **      $FREAL - READ BYTES FROM FILE BUFFER.
                                7597X *
                                7598X *       $FREAL IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                                7599X *
                                7600X *       ENTRY   (BC) = BYTE COUNT
                                7601X *               (DE) = FWA FOR BYTES
                                7602X *               (HL) = ADDRESS OF FILE BUFFER
                                7603X *       EXIT    TO *FERROR* IF ERROR
                                7604X *               TO CALLER IF OK
                                7605X *                (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                                7606X *                (DE) = ADDRESS OF FIRST UNUSED BYTE
                                7607X *                'C' SET IF EOF DURING READ
                                7608X *       USES    A,F,B,C,D,E
                                7609X
                                7610X
101.161  315 174 101            7611X $FREAL  CALL    $FREAL.
101.164  320                    7612X         RNC                     RETURN IF OK
101.165  376 001                7613X         CPI     EC.EOF
101.167  302 223 070            7614X         JNE     $FERROR         ERROR IS NOT EOF
101.172  067                    7615X         STC
101.173  311                    7616X         RET                     ERROR IS SIMPLY EOF
                                7617X
                                7618X
101.174                         7619X $FREAL. EQU     *
101.174  013                    7620X         DCX     B               (BC) = COUNT NOT ENCLUDING 00 BYTE
101.175  257                    7621X         XRA     A
101.176  062 216 103            7622X         STA     EOFFLG          CLEAR EOF FLAG
101.201  345                    7623X         PUSH    H
101.202  315 042 103            7624X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                                7625X
                                7626X *       COPY DATA FROM BUFFER TO TARGET
                                7627X
101.205  325                    7628X $REAL2  PUSH    D               SAVE TARGET ADDRESS
101.206  072 205 103            7629X         LDA     T.FLG
101.211  346 002                7630X         ANI     FT.OR
101.213  076 011                7631X         MVI     A,EC.FNO
101.215  067                    7632X         STC                     ASSUME FILE NOT OPEN
101.216  312 352 101            7633X         JZ      $REAL8          ERROR
101.221  170                    7634X         MOV     A,B
101.222  261                    7635X         ORA     C
101.223  312 352 101            7636X         JZ      $REAL8          ALL DONE
                                7637X
                                7638X *       COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                                7639X
101.226  052 210 103            7640X $REAL3  LHLD    T.PTR
101.231  353                    7641X         XCHG                    (DE) = (FB.PTR) = ADDRESS OF DATA
101.232  052 212 103            7642X         LHLD    T.LIM           (HL) = LIMIT ADDRESS
101.235  175                    7643X         MOV     A,L
101.236  223                    7644X         SUB     E
101.237  157                    7645X         MOV     L,A
101.240  174                    7646X         MOV     A,H
101.241  232                    7647X         SBB     D
101.242  147                    7648X         MOV     H,A             (HL) = NUMBER OF BYTES IN BUFFER
101.243  171                    7649X         MOV     A,C
101.244  225                    7650X         SUB     L               COMPARE TO REQUESTED COUNT
101.245  170                    7651X         MOV     A,B
```

```
101.246  234          7652X          SBB     H
101.247  322 254 101  7653X          JNC     $REAL4          LESS THAN REQUESTED COUNT
101.252  140          7654X          MOV     H,B
101.253  151          7655X          MOV     L,C             DONT TRANSFER MORE THAN LIMIT
101.254  174          7656X $REAL4   MOV     A,H
101.255  265          7657X          ORA     L
101.256  302 272 101  7658X          JNZ     $REAL6          SOME IN BUFFER
                      7659X
                      7660X *        BUFFER IS EMPTY. RE-FILL IT
                      7661X
101.261  315 122 103  7662X          CALL    $FFB            FILL FILE BUFFER
101.264  332 352 101  7663X          JC      $REAL8          ERROR CONDITION
101.267  303 226 101  7664X          JMP     $REAL3          COUNT THE DATA
                      7665X
                      7666X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                      7667X *
                      7668X *        (BC) = LIMIT COUNT
                      7669X *        (DE) = FROM
                      7670X *        (HL) = COUNT
                      7671X *        ((SP)) = TO
                      7672X
101.272  171          7673X $REAL6   MOV     A,C
101.273  225          7674X          SUB     L
101.274  117          7675X          MOV     C,A
101.275  170          7676X          MOV     A,B
101.276  234          7677X          SBB     H
101.277  107          7678X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
101.300  305          7679X          PUSH    B
101.301  343          7680X          XTHL                    (HL) = REMAINING REQUEST COUNT
101.302  301          7681X          POP     B               (BC) = COUNT FOR THIS COPY
101.303  343          7682X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
101.304  032          7683X $REAL7   LDAX    D
101.305  023          7684X          INX     D
101.306  167          7685X          MOV     M,A
101.307  043          7686X          INX     H
101.310  247          7687X          ANA     A               SEE IF 00 BYTE
101.311  302 320 101  7688X          JNZ     $REL7.3         NOT 00
                      7689X
                      7690X *        IS 00 BYTE. IGNORE IT
                      7691X
101.314  343          7692X          XTHL
101.315  043          7693X          INX     H               ADD ONE TO UNREQUITED COUNT
101.316  343          7694X          XTHL
101.317  053          7695X          DCX     H               BACKSPACE OVER CHARACTER
101.320  013          7696X $REL7.3  DCX     B
101.321  376 012      7697X          CPI     NL
101.323  312 343 101  7698X          JE      $REL7.5         IS END OF LINE
101.326  170          7699X          MOV     A,B
101.327  261          7700X          ORA     C
101.330  302 304 101  7701X          JNZ     $REAL7          MORE TO GO
101.333  353          7702X          XCHG
101.334  042 210 103  7703X          SHLD    T.PTR           UPDATE POINTER
101.337  301          7704X          POP     B               (BC) = REMAINING COUNT
101.340  303 205 101  7705X          JMP     $REAL2          SEE IF MORE IN BUFFER
                      7706X
                      7707X *        END OF CODED LINE
```

```
                                7708X
101,343  353                    7709X $REL7.5 XCHG
101,344  033                    7710X         DCX     D               BACK OVER NL CHARACTER
101,345  042 210 103            7711X         SHLD    T.PTR           UPDATE POINTER
101,350  301                    7712X         POP     B               (BC) = REMAINING COUNT
101,351  325                    7713X         PUSH    D               SAVE TARGET LWA
                                7714X
                                7715X *       READ COMPLETE.
                                7716X *
                                7717X *       (PSW) = COMPLETION FLAGS
                                7718X
101,352  321                    7719X $REAL8  POP     D               RESTORE TARGET ADDRESS
101,353  365                    7720X         PUSH    PSW             SAVE RETURN CODE
101,354  257                    7721X         XRA     A
101,355  022                    7722X         STAX    D               FLAG END OF LINE
101,356  361                    7723X         POP     PSW             RESTORE RESULT FLAGS
101,357  023                    7724X         INX     D               POINT TO NEXT FREE
101,360  341                    7725X $REAL9  POP     H
101,361  303 070 103            7726X         JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
101,364                         7727          XTEXT   FREAD



                                7729X **      $FREAD - READ ONE BYTE FROM FILE BUFFER.
                                7730X *
                                7731X *       $FREAD IS CALLED TO READ ONE BYTE FROM A FILE BUFFER.
                                7732X *
                                7733X *       ENTRY   (HL) = ADDRESS OF FILE BUFFER
                                7734X *       EXIT    TO *FERROR* IF ERROR
                                7735X *                 TO CALLER IF OK
                                7736X *                  (A) = CHARACTER
                                7737X *                  'C' SET IF EOF DURING READ
                                7738X *       USES    A,F,B,C,D,E
                                7739X
                                7740X
101,364  315 377 101            7741X $FREAD  CALL    $FREAD.
101,367  320                    7742X         RNC                     RETURN IF OK
101,370  376 001                7743X         CPI     EC.EOF
101,372  302 223 070            7744X         JNE     $FERROR         ERROR IS NOT EOF
101,375  067                    7745X         STC
101,376  311                    7746X         RET                     ERROR IS SIMPLY EOF
                                7747X
                                7748X
101,377                         7749X $FREAD. EQU     *
101,377  257                    7750X         XRA     A
102,000  062 216 103            7751X         STA     EOFFLG          CLEAR EOF FLAG
102,003  345                    7752X         PUSH    H
102,004  315 042 103            7753X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
102,007  052 212 103            7754X $READ1  LHLD    T.LIM
102,012  353                    7755X         XCHG
102,013  052 210 103            7756X         LHLD    T.PTR
102,016  315 216 030            7757X         CALL    $CDEHL          SEE IF ANY TO READ
102,021  302 035 102            7758X         JNE     $READ2          GOT DATA
102,024  315 122 103            7759X         CALL    $FFB            FILL FILE BUFFER
102,027  332 043 102            7760X         JC      $READ8          ERROR CONDITION
```

```
102.032  303 007 102  7761X          JMP     $READ1          TRY AGAIN
                      7762X
102.035  176          7763X $READ2   MOV     A,M             (A) = CHARACTER
102.036  247          7764X          ANA     A               CLEAR CARRY
102.037  043          7765X          INX     H
102.040  042 210 103  7766X          SHLD    T.PTR
                      7767X
                      7768X *        READ COMPLETE
                      7769X *
                      7770X *        (PSW) = COMPLETION FLAGS
                      7771X
102.043  341          7772X $READ8   POP     H
102.044  303 070 103  7773X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
102.047               7774           XTEXT   FWRIB


                      7776X **       $FWRIB - WRITE BYTES FROM FILE BUFFER.
                      7777X *
                      7778X *        $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                      7779X *
                      7780X *        ENTRY   (BC) = BYTE COUNT
                      7781X *                (DE) = FWA FOR BYTES
                      7782X *                (HL) = ADDRESS OF FILE BUFFER
                      7783X *        EXIT    TO *FERROR* IF ERROR
                      7784X *                TO CALLER IF OK
                      7785X *                 (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                      7786X *        USES    A,F,B,C,D,E
                      7787X
                      7788X
102.047  315 056 102  7789X $FWRIB   CALL    $FWRIB.
102.052  320          7790X          RNC                     RETURN IF OK
102.053  303 223 070  7791X          JMP     $FERROR         ERROR
                      7792X
                      7793X
102.056               7794X $FWRIB.  EQU     *
102.056  345          7795X          PUSH    H
102.057  315 042 103  7796X          CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                      7797X
                      7798X *        COPY DATA FROM USER AREA TO BUFFER
                      7799X
102.062  325          7800X $WRIB2   PUSH    D               SAVE AREA ADDRESS
102.063  072 205 103  7801X          LDA     T.FLG
102.066  346 004      7802X          ANI     FT.OW           SEE IF OPEN FOR WRITE
102.070  312 224 102  7803X          JZ      $WRIB8          FILE NOT OPEN FOR WRITE
102.073  170          7804X          MOV     A,B
102.074  261          7805X          ORA     C
102.075  312 224 102  7806X          JZ      $WRIB8          ALL DONE
                      7807X
                      7808X *        COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                      7809X
102.100  052 210 103  7810X $WRIB3   LHLD    T.PTR
102.103  353          7811X          XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
102.104  052 214 103  7812X          LHLD    T.LWA           (HL) = LIMIT ADDRESS
102.107  175          7813X          MOV     A,L
```

```
102.110  223           7814X          SUB     E
102.111  157           7815X          MOV     L,A
102.112  174           7816X          MOV     A,H
102.113  232           7817X          SBB     D
102.114  147           7818X          MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
102.115  171           7819X          MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
102.116  225           7820X          SUB     L
102.117  170           7821X          MOV     A,B
102.120  234           7822X          SBB     H
102.121  322 126 102   7823X          JNC     $WRIB4          MORE REQUESTED THEN ROOM
102.124  140           7824X          MOV     H,B
102.125  151           7825X          MOV     L,C             USE REQUESTED COUNT
102.126  174           7826X $WRIB4   MOV     A,H
102.127  265           7827X          ORA     L
102.130  302 170 102   7828X          JNZ     $WRIB6          SOME ROOM IN BUFFER
                       7829X
                       7830X *        BUFFER IS FULL. EMPTY IT
                       7831X
102.133  305           7832X          PUSH    B               SAVE COUNT
102.134  052 206 103   7833X          LHLD    T.FWA
102.137  042 210 103   7834X          SHLD    T.PTR           CLEAR REMOVAL POINTER
102.142  353           7835X          XCHG
102.143  052 214 103   7836X          LHLD    T.LWA
102.146  175           7837X          MOV     A,L
102.147  223           7838X          SUB     E
102.150  117           7839X          MOV     C,A
102.151  174           7840X          MOV     A,H
102.152  232           7841X          SBB     D
102.153  107           7842X          MOV     B,A             (BC) = DATA IN BUFFER
102.154  072 204 103   7843X          LDA     T.CHA
102.157  377 005       7844X          DB      SYSCALL,.WRITE  WRITE BUFFER
102.161  301           7845X          POP     B               (BC) = DESIRED COUNT
102.162  322 100 102   7846X          JNC     $WRIB3          GOT THE DATA
                       7847X
                       7848X *        ERROR ON WRITE.
                       7849X
102.165  303 224 102   7850X          JMP     $WRIB8          HAVE ERROR
                       7851X
                       7852X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                       7853X *
                       7854X *        (BC) = REQUEST COUNT
                       7855X *        (DE) = TO
                       7856X *        (HL) = COUNT
                       7857X *        ((SP)) = FROM
                       7858X
102.170  171           7859X $WRIB6   MOV     A,C
102.171  225           7860X          SUB     L
102.172  117           7861X          MOV     C,A
102.173  170           7862X          MOV     A,B
102.174  234           7863X          SBB     H
102.175  107           7864X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
102.176  305           7865X          PUSH    B
102.177  343           7866X          XTHL                    (HL) = REMAINING REQUEST COUNT
102.200  301           7867X          POP     B               (BC) = COUNT FOR THIS COPY
102.201  343           7868X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
102.202  176           7869X $WRIB7   MOV     A,M
```

```
102.203  022            7870X          STAX    D
102.204  023            7871X          INX     D
102.205  043            7872X          INX     H
102.206  013            7873X          DCX     B
102.207  170            7874X          MOV     A,B
102.210  261            7875X          ORA     C
102.211  302 202 102     7876X          JNZ     $WRIB7          MORE TO GO
102.214  353            7877X          XCHG
102.215  042 210 103     7878X          SHLD    T.PTR           UPDATE POINTER
102.220  301            7879X          POP     B               (BC) = REMAINING COUNT
102.221  303 062 102     7880X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                        7881X
                        7882X *        WRITE COMPLETE.
                        7883X *
                        7884X *        (PSW) = COMPLETION FLAGS
                        7885X
102.224  321            7886X $WRIBB   POP     D               RESTORE TARGET ADDRESS
102.225  341            7887X          POP     H
102.226  303 070 103     7888X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                        7890X **       $FWBRK  -  BREAKOUTPUT                                  /B0.02.GC/
                        7891X *
                        7892X *        $FWBRK empties the specified buffer by filling it with NULLs
                        7893X *        and then writing it.  Note this is used to insure that block
                        7894X *        mode I/O is output if it is not really a serial device (eg.
                        7895X *        writing to AT: from  *EDIT*.
                        7896X *
                        7897X *
                        7898X *        ENTRY:  HL      =  FILE BLOCK POINTER
                        7899X *
                        7900X *        EXIT:   HL      =  FILE BLOCK POINTER
                        7901X *                TO  $FERROR IF ERROR
                        7902X *
                        7903X *        USES:   PSW,BC,DE
                        7904X *
                        7905X
102.231  315 240 102     7906X $FWBRK   CALL    $FWBRK.
102.234  320            7907X          RNC                     NO ERROR
                        7908X
102.235  303 223 070     7909X          JMP     $FERROR
                        7910X
102.240  345            7911X $FWBRK.  PUSH    H
102.241  315 042 103     7912X          CALL    CBT             COPY BUFFER TO TEMPORARY
102.244  315 254 102     7913X          CALL    $FWBRK1
102.247  341            7914X          POP     H
102.250  315 070 103     7915X          CALL    CTB             COPY TEMPORARY TO BUFFER
102.253  311            7916X          RET
                        7917X
102.254  052 214 103     7918X $FWBRK1  LHLD    T.LWA
102.257  353            7919X          XCHG                    DE = BUFFER LWA
102.260  052 210 103     7920X          LHLD    T.PTR           HL = BUFFER PTR
102.263  173            7921X          MOV     A,E
102.264  225            7922X          SUB     L
```

```
102.265  117          7923X          MOV     C,A
102.266  172          7924X          MOV     A,D
102.267  234          7925X          SBB     H
102.270  107          7926X          MOV     B,A             BC = DE - HL
102.271  261          7927X          ORA     C
102.272  310          7928X          RZ                      THE BUFFER IS ALREADY FLUSHED
                      7929X
                      7930X *        FILL THE BUFFER WITH NULLS
                      7931X
102.273  170          7932X FWBRK2   MOV     A,B
102.274  261          7933X          ORA     C
102.275  312 307 102   7934X          JZ      FWBRK3          NO MORE LEFT TO FILL
                      7935X
102.300  066 000       7936X          MVI     M,0
102.302  043          7937X          INX     H
102.303  013          7938X          DCX     B
102.304  303 273 102   7939X          JMP     FWBRK2
                      7940X
102.307  052 206 103   7941X FWBRK3   LHLD    T.FWA
102.312  042 210 103   7942X          SHLD    T.PTR
102.315  353          7943X          XCHG                    DE = BUFFER FWA
102.316  052 214 103   7944X          LHLD    T.LWA           HL = BUFFER LWA
102.321  175          7945X          MOV     A,L
102.322  223          7946X          SUB     E
102.323  117          7947X          MOV     C,A
102.324  174          7948X          MOV     A,H
102.325  232          7949X          SBB     D
102.326  107          7950X          MOV     B,A             BC = HL - DE  ( BC = COUNT )
102.327  072 204 103   7951X          LDA     T.CHA
102.332  377 005       7952X          DB      SYSCALL,.WRITE
102.334  311          7953X          RET
102.335               7954           XTEXT   FCLO


                      7956X **       $FCLO - CLOSE FILE BLOCK.
                      7957X *
                      7958X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                      7959X *        BLOCK.
                      7960X *
                      7961X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      7962X *        EXIT    TO $FERROR IF ERROR
                      7963X *                 TO CALLER IF OK
                      7964X *        USES    A,F,B,C,D,E
                      7965X
                      7966X
102.335  315 344 102   7967X $FCLO    CALL    $FCLO.
102.340  320          7968X          RNC                     NO ERROR
102.341  303 223 070   7969X          JMP     $FERROR
                      7970X
102.344  345          7971X $FCLO.    PUSH    H               SAVE FILE BLOCK ADDRESS
000.000               7972X          ERRNZ   FB.FLG-1
102.345  043          7973X          INX     H               (HL) = $FB.FLG
102.346  176          7974X          MOV     A,M
102.347  066 000       7975X          MVI     M,0             CLEAR FLAG
```

```
102.351  247              7976X           ANA     A
102.352  312 040 103      7977X           JZ      $FCLO4          FILE NOT OPEN
102.355  346 004          7978X           ANI     FT.OW
102.357  312 032 103      7979X           JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                          7980X
                          7981X *         WAS OPEN FOR WRITE, SEE IF NEED FLUSH THE LAST SECTOR
                          7982X
102.362  315 234 030      7983X           CALL    $INDL
102.365  003 000          7984X           DW      FB.PTR-FB.FLG
102.367  325              7985X           PUSH    D               SAVE (FB.PTR)
102.370  315 234 030      7986X           CALL    $INDL           (DE) = (FB.FWA)
102.373  001 000          7987X           DW      FB.FWA-FB.FLG
102.375  341              7988X           POP     H               (HL) = (FB.PTR)
102.376  175              7989X           MOV     A,L
102.377  223              7990X           SUB     E
103.000  117              7991X           MOV     C,A
103.001  174              7992X           MOV     A,H
103.002  232              7993X           SBB     D
103.003  107              7994X           MOV     B,A             (BC) = AMOUNT IN BLOCK
103.004  261              7995X           ORA     C
103.005  312 032 103      7996X           JZ      $FCLO3          NONE TO FLUSH
                          7997X
                          7998X *         NEED TO FLUSH BUFFER
                          7999X *
                          8000X *         (BC) = DATA AMOUNT
                          8001X *         (DE) = FWA
                          8002X *         (HL) = LWA+1
                          8003X
103.010  171              8004X           MOV     A,C
103.011  247              8005X           ANA     A
103.012  312 025 103      8006X           JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                          8007X
                          8008X *         ZERO FILL PARTIAL SECTOR
                          8009X
103.015  066 000          8010X $FCLO1    MVI     M,0
103.017  043              8011X           INX     H
103.020  014              8012X           INR     C
103.021  302 015 103      8013X           JNZ     $FCLO1
103.024  004              8014X           INR     B               COUNT ANOTHER FULL SECTOR
103.025  341              8015X $FCLO2    POP     H               (HL) = FB.FWA
103.026  176              8016X           MOV     A,M             (A) = CHANNEL NUMBER
000.000                   8017X           ERRNZ   FB.CHA
103.027  345              8018X           PUSH    H
103.030  377 005          8019X           DB      SYSCALL+.WRITE          FLUSH
                          8020X
                          8021X *         READY TO CLOSE FILE.
                          8022X *
                          8023X *         'C' SET IF ERROR
                          8024X *          (A) = ERROR CODE
                          8025X
103.032  341              8026X $FCLO3    POP     H               (HL) = FILE BLOCK ADDRESS
103.033  330              8027X           RC                      ERROR
000.000                   8028X           ERRNZ   FB.CHA
103.034  176              8029X           MOV     A,M             (A) = CHANNEL NUMBER
103.035  345              8030X           PUSH    H
103.036  377 046          8031X           DB      SYSCALL+.CLOSE  CLOSE CHANNEL
```

```
103.040  341           8032X $FCLO4  POP     H               (HL) = FILE BLOCK ADDRESS
103.041  311           8033X         RET
103.042                8034          XTEXT   FUTIL


                       8036X **      $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                       8037X
                       8038X **      CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                       8039X *
                       8040X *       ENTRY   (HL) = FILE BLOK FWA
                       8041X *       EXIT    NONE
                       8042X *       USES    A,F,H,L
                       8043X
103.042  325           8044X CBT     PUSH    D
103.043  305           8045X         PUSH    B               SAVE REGISTERS
000.000                8046X         ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
103.044  021 204 103   8047X         LXI     D,T.CHA         (DE) = TARGET FOR MOVE
103.047  006 005       8048X         MVI     B,10/2
103.051  176           8049X CBT1    MOV     A,M             COPY FILE BUFFER INTO WORK AREA
103.052  022           8050X         STAX    D
103.053  043           8051X         INX     H
103.054  023           8052X         INX     D
103.055  176           8053X         MOV     A,M
103.056  022           8054X         STAX    D
103.057  043           8055X         INX     H
103.060  023           8056X         INX     D
103.061  005           8057X         DCR     B
103.062  302 051 103   8058X         JNZ     CBT1            MORE TO GO
103.065  301           8059X         POP     B
103.066  321           8060X         POP     D               (DE) = DATA TARGET ADDRESS
103.067  311           8061X         RET
                       8062X
                       8063X
                       8064X **      CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                       8065X *
                       8066X *       ENTRY   (HL) = FILE BLOCK ADDRESS
                       8067X *       EXIT    NONE
                       8068X *       USES    NONE
                       8069X
103.070  365           8070X CTB     PUSH    PSW
103.071  325           8071X         PUSH    D
103.072  305           8072X         PUSH    B
103.073  345           8073X         PUSH    H               SAVE REGISTERS
103.074  006 004       8074X         MVI     B,8/2
103.076  021 204 103   8075X         LXI     D,T.CHA
103.101  032           8076X CTB1    LDAX    D
103.102  167           8077X         MOV     M,A
103.103  023           8078X         INX     D
103.104  043           8079X         INX     H
103.105  032           8080X         LDAX    D
103.106  167           8081X         MOV     M,A
103.107  023           8082X         INX     D
103.110  043           8083X         INX     H
103.111  005           8084X         DCR     B
```

```
103.112  302 101 103  8085X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
103.115  341          8086X          POP     H
103.116  301          8087X          POP     B
103.117  321          8088X          POP     D
103.120  361          8089X          POP     PSW
103.121  311          8090X          RET


                      8092X **       $FFB - FILE FILE BUFFER.
                      8093X *
                      8094X *        $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                      8095X *
                      8096X *        ENTRY   NONE
                      8097X *        EXIT    'C' SET IF READ INCOMPLETE
                      8098X *                 (A) = ERROR CODE
                      8099X *                 'C' CLEAR IF READ COMPLETEE
                      8100X *                  DATA IN BUFFER
                      8101X *        USES    A,F,D,E,H,L
                      8102X
                      8103X
103.122  072 216 103  8104X $FFB     LDA     EOFFLG
103.125  037          8105X          RAR
103.126  330          8106X          RC                      EOF
                      8107X
                      8108X *        CAN READ MORE. DO SO
                      8109X
103.127  305          8110X          PUSH    B               SAVE COUNT
103.130  052 206 103  8111X          LHLD    T.FWA
103.133  042 210 103  8112X          SHLD    T.FTR           CLEAR REMOVAL POINTER
103.136  353          8113X          XCHG
103.137  052 214 103  8114X          LHLD    T.LWA
103.142  042 212 103  8115X          SHLD    T.LIM           SET DATA LIMIT
103.145  175          8116X          MOV     A,L
103.146  223          8117X          SUB     E
103.147  117          8118X          MOV     C,A
103.150  174          8119X          MOV     A,H
103.151  232          8120X          SBB     D
103.152  107          8121X          MOV     B,A             (BC) = ROOM IN BUFFER
103.153  072 204 103  8122X          LDA     T.CHA
103.156  377 004      8123X          DB      SYSCALL+.READ   READ BUFFER
103.160  120          8124X          MOV     D,B             (D) = SECTORS UNREAD
103.161  301          8125X          POP     B               (BC) = DESIRED COUNT
103.162  320          8126X          RNC                     GOT THE DATA
                      8127X
                      8128X *        ERROR ON READ. SEE IF EOF
                      8129X
103.163  027          8130X          RAL
103.164  062 216 103  8131X          STA     EOFFLG          SET EOF, WE HOPE
103.167  376 003      8132X          CPI     EC.EOF*2+1
103.171  037          8133X          RAR
103.172  300          8134X          RNE                     IS NOT EOF, RETURN NOW!
103.173  072 213 103  8135X          LDA     T.LIM+1
103.176  222          8136X          SUB     D
103.177  062 213 103  8137X          STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
```

```
103,202  247          8138X          ANA     A
103,203  311          8139X          RET                     EXIT WITH DATA
                      8140X
                      8141X
                      8142X **       TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                      8143X
000,000               8144X          ERRNZ   FB.CHA
103,204  000          8145X T.CHA    DB      0               CHANNEL NUMBER
000,000               8146X          ERRNZ   *-T.CHA-FB.FLG
103,205  000          8147X T.FLG    DB      0               FLAG BYTE
000,000               8148X          ERRNZ   *-T.CHA-FB.FWA
103,206  000 000      8149X T.FWA    DW      0
000,000               8150X          ERRNZ   *-T.CHA-FB.PTR
103,210  000 000      8151X T.PTR    DW      0
000,000               8152X          ERRNZ   *-T.CHA-FB.LIM
103,212  000 000      8153X T.LIM    DW      0
000,000               8154X          ERRNZ   *-T.CHA-FB.LWA
103,214  000 000      8155X T.LWA    DW      0
000,012               8156X TLEN     EQU     *-T.CHA         LENGTH OF TEMP CELLS
                      8157X
103,216  000          8158X EOFFLG   DB      0
103,217               8159           XTEXT   TYPCC



                      8161X **       $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                      8162X *
                      8163X *        $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                      8164X *        THE CHARACTER ADDRESS AND COUNT.
                      8165X *
                      8166X *        ENTRY   (HL) = ADDRESS
                      8167X *                (A) = COUNT
                      8168X *        EXIT    (HL) = LAST CHARACTER ADDRESS+1
                      8169X *        USES    A,F,H,L
                      8170X
                      8171X
103,217               8172X $TYPCC   EQU     *
103,217  247          8173X          ANA     A
103,220  310          8174X          RZ                      NOTHING TO TYPE
103,221  365          8175X          PUSH    PSW             SAVE COUNT
103,222  176          8176X          MOV     A,M             (A) = CHARACTER
103,223  043          8177X          INX     H
103,224  377 002      8178X          DB      SYSCALL,.SCOUT
103,226  361          8179X          POP     PSW
103,227  075          8180X          DCR     A
103,230  303 217 103  8181X          JMP     $TYPCC
103,233               8182           XTEXT   RCHAR
```

```
                                8184X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                8185X *
                                8186X *       ENTRY   NONE
                                8187X *       EXIT    (A) = CHARACTER
                                8188X *       USES    A,F
                                8189X
                                8190X
      103,233  377 001           8191X $RCHAR  DB      SYSCALL,.SCIN
      103,235  332 233 103       8192X         JC      $RCHAR          NOT READY
      103,240  311               8193X         RET
                                8194X
      103,241  377 002           8195X $WCHAR  DB      SYSCALL,.SCOUT
      103,243  311               8196X         RET
                                8197          LON     C
      103,244                   8198          XTEXT   ATS


                                8200X **      $ATS - ALLOCATE TABLE SPACE.
                                8201X *
                                8202X *       ATS IS CALLED TO ALLOCATE ADDITIONAL SPACE TO A MANAGED TABLE.
                                8203X *
                                8204X *       IF NO MOVING IS REQUIRED, $ATS REQUIRES ABOUT 150 MICROSECONDS.
                                8205X *
                                8206X *       ENTRY (HL) = BYTES TO ALLOCATE
                                8207X *              (DE) = ADDRESS OF TABLE INDEX+1
                                8208X *       EXIT    SPACE ALLOCATED (IF ENOUGH ROOM)
                                8209X *              TO *ERR.TO* IF NO MORE ROOM
                                8210X *       USES    A,F,H,L
                                8211X
                                8212X
      103,244                   8213X $ATS    EQU     *               ENTRY POINT
      103,244  305              8214X         PUSH    B               SAVE REGISTERS
      103,245  325              8215X         PUSH    D
      103,246  345              8216X         PUSH    H
      103,247  353              8217X         XCHG                    (DE) = BN (BYTES NEEDED)
      103,250  042 356 103      8218X         SHLD    ATSA            SAVE FOR LATER
      103,253  116              8219X         MOV     C,M
      103,254  043              8220X         INX     H
      103,255  106              8221X         MOV     B,M             (BC) = TFWA (TABLE FWA)
      103,256  043              8222X         INX     H
      103,257  305              8223X         PUSH    B               SAVE TFWA
      103,260  116              8224X         MOV     C,M
      103,261  043              8225X         INX     H
      103,262  106              8226X         MOV     B,M             (BC) = TL (TABLE LENGTH)
      103,263  043              8227X         INX     H
      103,264  353              8228X         XCHG                    (HL) = BN
      103,265  011              8229X         DAD     B               (HL) = NEW TABLE LENGTH
      103,266  104              8230X         MOV     B,H
      103,267  115              8231X         MOV     C,L             (BC) = NEW TABLE LENGTH
      103,270  341              8232X         POP     H
      103,271  011              8233X         DAD     B               (HL) = NEW TABLE LWA
      103,272  332 321 103      8234X         JC      ATS1
      103,275  353              8235X         XCHG                    (DE) = NEW LWA, (HL) = INDEX ENTRY ADDRESS
      103,276  043              8236X         INX     H               SPACE OVER ALLOCATION FACTOR
```

```
103.277  173           8237X          MOV     A,E
103.300  226           8238X          SUB     M               COMPARE NEW LWA WITH NEXT TABLE FWA
103.301  043           8239X          INX     H
103.302  172           8240X          MOV     A,D
103.303  236           8241X          SBB     M
103.304  322 321 103    8242X          JNC     ATS1            OVERFLOW
                       8243X
                       8244X *        HAVE ENOUGH ROOM WITHOUT TABLE MOVES. UPDATE INDEX
                       8245X
103.307  053           8246X          DCX     H
103.310  053           8247X          DCX     H
103.311  053           8248X          DCX     H
103.312  160           8249X          MOV     M,B             SET NEW LENGTH
103.313  053           8250X          DCX     H
103.314  161           8251X          MOV     M,C
103.315  341           8252X          POP     H               RESTORE REGISTERS
103.316  321           8253X          POP     D
103.317  301           8254X          POP     B
103.320  311           8255X          RET

                       8257X **       THE TABLE OVERFLOWED IT'S FREE SPACE. REALLOCATE FREE SPACE
                       8258X *        AMONG STACKS.
                       8259X *
                       8260X *        (ATSA) = TABLE INDEX FWA
                       8261X *        (STACK TOP) = BN (BYTES NEEDED)
                       8262X
103.321  315 127 104    8263X ATS1     CALL    MTD             MOVE TABLES DOWN
103.324  041 012 000    8264X          LXI     H,10
103.327  031           8265X          DAD     D
103.330  321           8266X          POP     D
103.331  325           8267X          PUSH    D               (DE) = BN
103.332  031           8268X          DAD     D
103.333  332 160 070    8269X          JC      ERR.TO          TABLE OVERFLOW
103.336  353           8270X          XCHG                    (DE) = FIRST FREE SPACE AFTER ALLOCATION
103.337  052 127 112    8271X          LHLD    MEML            (HL) = MEMORY LIMIT ADDRESS
103.342  173           8272X          MOV     A,E
103.343  225           8273X          SUB     L
103.344  157           8274X          MOV     L,A
103.345  172           8275X          MOV     A,D
103.346  234           8276X          SBB     H
103.347  147           8277X          MOV     H,A             (HL) = -SPACE LEFT
103.350  322 160 070    8278X          JNC     ERR.TO          TABLE OVERFLOW
                       8279X
                       8280X *        THE ROOM EXISTS. ADD REQESTED SPACE TO PROPER TABLE.
                       8281X
103.353  301           8282X          POP     B               (BC) = BN (BYTES NEEDED)
103.354  345           8283X          PUSH    H               SAVE -(SPACE LEFT)
103.355  041 000 000    8284X          LXI     H,0             (HL) = TABLE INDEX FWA
103.356                8285X ATSA     EQU     *-2
103.360  043           8286X          INX     H
103.361  043           8287X          INX     H
103.362  136           8288X          MOV     E,M
103.363  043           8289X          INX     H
103.364  126           8290X          MOV     D,M             (DE) = CURRENT SIZE
103.365  353           8291X          XCHG
```

```
103.366  011          8292X          DAD     B
103.367  353          8293X          XCHG                    (DE) = NEW SIZE
103.370  162          8294X          MOV     M,D
103.371  053          8295X          DCX     H
103.372  163          8296X          MOV     M,E             SET NEW SIZE
                      8297X
                      8298X *        TABLES ARE ALL PACKED TOGETHER AT THE BOTTOM OF THE TABLE
                      8299X *        AREA. DECIDE HOW MUCH SPACE IS TO BE GIVEN TO EACH TABLE,
                      8300X *        AND MOVE THEM ONE BY ONE INTO POSITION, STARTING WITH THE
                      8301X *        HIGHEST TABLE, WORKING DOWN TO TABLE 2.
                      8302X
103.373  301          8303X          POP     B               (B) = -(SPACE LEFT)
103.374  046 003      8304X          MVI     H,3             DIVIDE BY 8
                      8305X
                      8306X *        DIVIDE SPACE LEFT BY 8
                      8307X
103.376  067          8308X ATS2     STC
103.377  170          8309X          MOV     A,B
104.000  037          8310X          RAR                     SHIFT RIGHT WITH SIGN EXTEND
104.001  107          8311X          MOV     B,A
104.002  171          8312X          MOV     A,C
104.003  037          8313X          RAR
104.004  117          8314X          MOV     C,A
104.005  045          8315X          DCR     H
104.006  302 376 103   8316X          JNZ     ATS2
104.011  003          8317X          INX     B               (BC) = 1/8 FREE SPACE
104.012  170          8318X          MOV     A,B
104.013  247          8319X          ANA     A
104.014  362 160 070   8320X          JP      ERR.TO          TABLE OVERFLOW
                      8321X
                      8322X *        (BC) = 1/8 FREE SPACE AVAILABLE.
                      8323X *
                      8324X *        MOVE TABLES INTO FINAL POSITION.
                      8325X
104.017  072 124 104   8326X          LDA     ATSB            (A) = TABLE COUNT-1
104.022  041 130 112   8327X          LXI     H,MEML+1
104.025  126          8328X          MOV     D,M
104.026  053          8329X          DCX     H
104.027  136          8330X          MOV     E,M             (DE) = (MEML)
104.030  053          8331X          DCX     H
                      8332X
104.031  365          8333X ATS3     PUSH    PSW             SAVE COUNT
104.032  305          8334X          PUSH    B
104.033  053          8335X          DCX     H
104.034  106          8336X          MOV     B,M
104.035  053          8337X          DCX     H
104.036  116          8338X          MOV     C,M             (BC) = TABLE LENGTH
104.037  053          8339X          DCX     H
104.040  173          8340X          MOV     A,E
104.041  221          8341X          SUB     C
104.042  137          8342X          MOV     E,A
104.043  172          8343X          MOV     A,D
104.044  230          8344X          SBB     B
104.045  127          8345X          MOV     D,A             (DE) = MEM TOP - TABLE SIZE
104.046  053          8346X          DCX     H
104.047  053          8347X          DCX     H
```

```
104.050  176          8348X          MOV     A,M             (A) = NUMBER OF 1/8'S TO GIVE THIS TABLE
104.051  343          8349X          XTHL                    (HL) = 1/8TH -SPACE
104.052  353          8350X          XCHG                    (HL) = MEM ADDRESS
104.053  247          8351X          ANA     A
104.054  312 064 104  8352X          JZ      ATS5            NO SPACE FOR THIS TABLE
104.057  031          8353X ATS4     DAD     D               DECREMENT BY FREE SPACE AMOUNT
104.060  075          8354X          DCR     A
104.061  302 057 104  8355X          JNZ     ATS4            GIVE SPECIFIED NUMBER OF 1/8THS
104.064  353          8356X ATS5     XCHG                    (DE) = TARGET ADDRESS
104.065  343          8357X          XTHL                    (HL) = TABLE ENTRY ADDRESS
104.066  345          8358X          PUSH    H
104.067  043          8359X          INX     H
104.070  176          8360X          MOV     A,M
104.071  163          8361X          MOV     M,E             SET NEW ADDRESS
104.072  365          8362X          PUSH    PSW
104.073  043          8363X          INX     H
104.074  176          8364X          MOV     A,M
104.075  162          8365X          MOV     M,D
104.076  147          8366X          MOV     H,A
104.077  361          8367X          POP     PSW
104.100  157          8368X          MOV     L,A
104.101  353          8369X          XCHG                    (BC) = COUNT, (DE) = FROM, (HL) = TO
104.102  345          8370X          PUSH    H
104.103  315 252 030  8371X          CALL    $MOVE           MOVE TABLE
104.106  321          8372X          POP     D               (DE) = NEW MEMORY LIMIT
104.107  341          8373X          POP     H
104.110  301          8374X          POP     B
104.111  361          8375X          POP     PSW
104.112  075          8376X          DCR     A
104.113  302 031 104  8377X          JNZ     ATS3            IF MORE TABLES TO MOVE
104.116  315 071 071  8378X          CALL    $ATP            ADJUST TABLE POINTERS
104.121  321          8379X          POP     D
104.122  301          8380X          POP     B
104.123  311          8381X          RET                     RETURN
                      8382X
104.124  007          8383X ATSB     DB      MTABL-1         TABLE COUNT-1
104.125  056 112      8384X ATSC     DW      MTABIND         ADDRESS OF 1ST TABLE TO MANAGE


                      8386X **       MTD - MOVE TABLES DOWN.
                      8387X *
                      8388X *        MTD IS CALLED TO MOVE ALL THE MANAGED TABLES DOWN INTO THE LOW
                      8389X *        PART OF THE MEMORY AREA, SO THAT ALL OF THE FREE SPACE IS CONCEN
                      8390X *        AFTER THE LAST TABLE.
                      8391X *
                      8392X *        ENTRY NONE
                      8393X *        EXIT    (DE) = FIRST FREE BYTE (LAST TABLE LWA+1)
                      8394X *        USES    ALL
                      8395X
                      8396X
104.127  052 125 104  8397X MTD      LHLD    ATSC
104.132  072 124 104  8398X          LDA     ATSB
                      8399X
                      8400X *        WONT NEED TO MOVE FIRST TABLE, FIND ITS LWA.
```

```
                                8401X
104.135  043                   8402X           INX     H
104.136  116                   8403X           MOV     C,M
104.137  043                   8404X           INX     H
104.140  106                   8405X           MOV     B,M             (BC) = FWA
104.141  043                   8406X           INX     H
104.142  136                   8407X           MOV     E,M
104.143  043                   8408X           INX     H
104.144  126                   8409X           MOV     D,M             (DE) = TABLE LEN
104.145  043                   8410X           INX     H
104.146  353                   8411X           XCHG
104.147  011                   8412X           DAD     B
104.150  353                   8413X           XCHG                    (DE) = TABLE LWA+1
                               8414X
                               8415X *         MOVE NEXT TABLE DOWN.
                               8416X
104.151  365                   8417X MTD1      PUSH    PSW
104.152  043                   8418X           INX     H
104.153  116                   8419X           MOV     C,M
104.154  163                   8420X           MOV     M,E             SET NEW START ADDRESS
104.155  043                   8421X           INX     H
104.156  106                   8422X           MOV     B,M             (BC) = TABLE FWA
104.157  162                   8423X           MOV     M,D
104.160  043                   8424X           INX     H
104.161  305                   8425X           PUSH    B
104.162  116                   8426X           MOV     C,M
104.163  043                   8427X           INX     H
104.164  106                   8428X           MOV     B,M             (BC) = TABLE LENGTH
104.165  043                   8429X           INX     H
104.166  343                   8430X           XTHL                    (HL) = TABLE FWA
104.167  353                   8431X           XCHG                    (DE) = FWA, (HL) = NEW ADDRESS
104.170  315 252 030           8432X           CALL    $MOVE           MOVE DOWN
104.173  353                   8433X           XCHG
104.174  341                   8434X           POP     H               (DE) = NEXT FREE BYTE, (HL) = INDEX POINTER
104.175  361                   8435X           POP     PSW
104.176  075                   8436X           DCR     A
104.177  302 151 104           8437X           JNZ     MTD1
104.202  311                   8438X           RET                     EXIT
104.203                        8439            XTEXT   DBT


                               8441X **        $DBT - DELETE BYTES FROM TABLE.
                               8442X *
                               8443X *         DBT DELETES BYTES FROM A MANAGED TABLE.
                               8444X *
                               8445X *         ENTRY   (DE) = BYTES TO DELETE
                               8446X *                 (HL) = POINTER TO PLACE (IN TABLE) TO BEGIN DELETEING (PTR
                               8447X *                 (RET+1, RET+2) = TABLE INDEX ADDRESS+1
                               8448X *         EXIT    BYTES DELETED.
                               8449X *         USES    A,F
                               8450X
                               8451X
104.203                        8452X $DBT      EQU     *
104.203  173                   8453X           MOV     A,E
```

```
104.204  057          8454X          CMA
104.205  137          8455X          MOV     E,A
104.206  172          8456X          MOV     A,D
104.207  057          8457X          CMA
104.210  127          8458X          MOV     D,A
104.211  023          8459X          INX     D               (DE) = -(BYTES TO DELETE)
104.212  067          8460X          STC                     SET CARRY


                      8462X **       $IBT - INSERT BYTES INTO TABLE,
                      8463X *
                      8464X *        $IBT IS CALLED TO MAKE A FREE SPACE IN A MANAGED TABLE, THIS
                      8465X *        FREE SPACE MAY BE CREATED ANYWHERE IN A TABLE: AT THE FRONT,
                      8466X *        AT THE BACK, OR IN THE MIDDLE,
                      8467X *
                      8468X *        ENTRY   (DE) = BYTES NEEDED (BN)
                      8469X *                 (IF 'C' SET, DELETE BYTES)
                      8470X *                 (HL) = POINTER TO INSERT AREA IN TABLE (PTR)
                      8471X *                 (RET+1, RET+2) = TABLE ADDRESS
                      8472X *        EXIT    BYTES INSERTED
                      8473X *                 TO (RET)+2
                      8474X *        USES    A,F
                      8475X
                      8476X
104.213  042 244 104  8477X $IBT     SHLD    IBTA            SAVE PTR
104.216  353          8478X          XCHG
104.217  343          8479X          XTHL                    (HL) = RETURN ADDRES
104.220  136          8480X          MOV     E,M
104.221  043          8481X          INX     H
104.222  126          8482X          MOV     D,M             (DE) = TABLE ADDRESS
104.223  043          8483X          INX     H
104.224  343          8484X          XTHL                    (HL) = BYTES NEEDED (BN)
104.225  345          8485X          PUSH    H               SAVE ENTRY (DE)
104.226  305          8486X          PUSH    B
104.227  332 301 104  8487X          JC      IBT2            IF TO DELETE
104.232  345          8488X          PUSH    H               SAVE BN
104.233  315 244 103  8489X          CALL    $ATS            ALLOCATE TABLE SPACE
                      8490X
                      8491X *        MOVE (TL-PTR) BYTES FROM (TFWA+PTR) TO (TFWA+PTR+BN)
                      8492X *        MOVE (TL-PTR-BN) BYTES FROM (TFWA+PTR) TO (TFWA+PTR+BN)
                      8493X
104.236  353          8494X          XCHG                    (HL) = TABLE ADDRESS
104.237  136          8495X          MOV     E,M
104.240  043          8496X          INX     H
104.241  126          8497X          MOV     D,M             (DE) = TABLE FWA
104.242  043          8498X          INX     H
104.243  001 000 000  8499X          LXI     B,0             (BC) = POINTER
104.244                8500X IBTA     EQU     *-2
104.246  353          8501X          XCHG
104.247  011          8502X          DAD     B               (HL) = TFWA+PTR
104.250  353          8503X          XCHG                    (DE) = TFWA+PTR
104.251  176          8504X          MOV     A,M
104.252  221          8505X          SUB     C
104.253  117          8506X          MOV     C,A
```

```
104.254  043          8507X          INX     H
104.255  176          8508X          MOV     A,M
104.256  230          8509X          SBB     B
104.257  107          8510X          MOV     B,A             (BC) = TL-PTR
104.260  341          8511X          POP     H               (HL) = BN
104.261  171          8512X          MOV     A,C
104.262  225          8513X          SUB     L
104.263  117          8514X          MOV     C,A
104.264  170          8515X          MOV     A,B
104.265  234          8516X          SBB     H
104.266  107          8517X          MOV     B,A             (BC) = TL-PTR-BN
104.267  031          8518X          DAD     D               (HL) = TFWA+PTR+BN
104.270  315 252 030   8519X IBT1     CALL    $MOVE           MOVE BLOCK
104.273  301          8520X          POP     B               RESTORE REGISTERS
104.274  321          8521X          POP     D
104.275  052 244 104   8522X          LHLD    IBTA            RESTORE (HL)
104.300  311          8523X          RET

                      8525X **       DELETE BYTES FROM TABLE.
                      8526X
104.301  174          8527X IBT2     MOV     A,H
104.302  057          8528X          CMA
104.303  147          8529X          MOV     H,A
104.304  175          8530X          MOV     A,L
104.305  057          8531X          CMA
104.306  157          8532X          MOV     L,A
104.307  043          8533X          INX     H               (HL) = BYTES TO DELETE
                      8534X
                      8535X *        MOVE (TL-PTR-BN) BYTES FROM (PTR+BN+TFWA) TO (PTR+TFWA)
                      8536X
104.310  353          8537X          XCHG                    (HL) = ADDRESS, (DE) = BN
104.311  116          8538X          MOV     C,M
104.312  043          8539X          INX     H
104.313  106          8540X          MOV     B,M             (BC) = TFWA
104.314  305          8541X          PUSH    B
104.315  043          8542X          INX     H
104.316  176          8543X          MOV     A,M
104.317  223          8544X          SUB     E
104.320  117          8545X          MOV     C,A
104.321  167          8546X          MOV     M,A
104.322  043          8547X          INX     H
104.323  176          8548X          MOV     A,M
104.324  232          8549X          SBB     D
104.325  107          8550X          MOV     B,A             (BC) = TL-BN
104.326  167          8551X          MOV     M,A             SET NEW LEN IN TABLE
104.327  052 244 104   8552X          LHLD    IBTA            (HL) = PTR
104.332  171          8553X          MOV     A,C
104.333  225          8554X          SUB     L
104.334  117          8555X          MOV     C,A
104.335  170          8556X          MOV     A,B
104.336  234          8557X          SBB     H
104.337  107          8558X          MOV     B,A             (BC) = TL-PTR-BN
104.340  353          8559X          XCHG                    (DE) = PTR, (HL) = BN
104.341  343          8560X          XTHL
```

```
104,342  031          8561X          DAD     D               (HL) = PTR+TFWA
104,343  353          8562X          XCHG                    (DE) = PTR+TFWA
104,344  341          8563X          POP     H               (HL) = BN
104,345  031          8564X          DAD     D
                      8565X
                      8566X *        (BC) = TL-PTR-BN
                      8567X *        (DE) = BTF+TFWA
                      8568X *        (HL) = PTR+TFWA+BN
                      8569X
104,346  353          8570X          XCHG
104,347  303 270 104  8571X          JMP     IBT1            MOVE DATA AND EXIT
104,352               8572           XTEXT   FPP
```

```
                                8576X **      FPADD - FLOATING POINT ADD.
                                8577X *
                                8578X *       ACCX = ACCX + (DE)
                                8579X *
                                8580X *       ENTRY (DE) = POINTER TO 4 BYTE FP VALUE
                                8581X *       EXIT    ACCX = RESULT
                                8582X *               SUPPLIED VALUE UNCHANGED
                                8583X *       USES    A,F
                                8584X
                                8585X
     104.352  315 215 107      8586X FPADD   CALL    SPE             SETUP PACKAGE ENTRY
     104.355  353              8587X         XCHG                    (HL) = ADDRESS OF VALUE


                                8589X **      ADD - PERFORM FLOATING POINT ADD.
                                8590X *
                                8591X *       ACCX = ACCX + (HL)
                                8592X *
                                8593X *       ENTRY (HL) = POINTER TO 4 BYTE FP VALUE
                                8594X *               RESULT STORED IN ACCX
                                8595X *       USES    ALL
                                8596X
                                8597X
     104.356                    8598X ADD     EQU     *
     104.356  315 250 107      8599X         CALL    LDD             (BCDE) = Y
     104.361  041 205 042      8600X         LXI     H,ACCX+3
                                8601X
                                8602X *       CHECK FOR X+0, 0+Y
                                8603X
     104.364  170              8604X         MOV     A,B             (A) = EXP(Y)
     104.365  267              8605X         ORA     A
     104.366  310              8606X         RZ                      IF Y=0
                                8607X
     104.367  176              8608X ADD0    MOV     A,M             (A) = EXP(X)
     104.370  267              8609X         ORA     A
     104.371  312 160 105      8610X         JZ      ADD5            X = 0; RESULT = (BCDE)          /80.02,8C/
                                8611X
                                8612X *       COMPARE EXPONENTS, TO SEE IF SIGNIFICANT
                                8613X
     104.374  220              8614X         SUB     B
     104.375  322 022 105      8615X         JNC     ADD1            EXPX GT EXPY
     105.000  052 202 042      8616X         LHLD    ACCX            SWAP (BCDE) WITH ACCX
     105.003  353              8617X         XCHG
     105.004  042 202 042      8618X         SHLD    ACCX
     105.007  052 204 042      8619X         LHLD    ACCX+2
     105.012  305              8620X         PUSH    B
     105.013  343              8621X         XTHL
     105.014  301              8622X         POP     B
     105.015  042 204 042      8623X         SHLD    ACCX+2
     105.020  057              8624X         CMA
     105.021  074              8625X         INR     A               (A) = SHIFT COUNT
                                8626X
                                8627X *       (A) = SHIFT COUNT FOR JUSTIFICATION
                                8628X
```

```
105,022  312 074 105  8629X ADD1   JZ      ADD3            NONE TO SHIFT
105,025  376 030      8630X         CPI     24
105,027  332 057 105  8631X         JC      ADD2.5          IS LESS THAN 24
                      8632X
                      8633X *       WOULD NEED TO SHIFT INTO INSIGNIFACANCE, JUST ADD 0
                      8634X
105,032  021 000 000  8635X         LXI     D,0             (DE) = 0
105,035  112          8636X         MOV     C,D             (C) = 0
105,036  303 074 105  8637X         JMP     ADD3
                      8638X
                      8639X *       DO JUSTIFYING RIGHT SHIFT
                      8640X
105,041  132          8641X ADD2    MOV     E,D
105,042  121          8642X         MOV     D,C
105,043  171          8643X         MOV     A,C
105,044  027          8644X         RAL
105,045  076 000      8645X         MVI     A,0
105,047  237          8646X         SBB     A
105,050  117          8647X         MOV     C,A
105,051  174          8648X         MOV     A,H
105,052  326 010      8649X         SUI     8
105,054  312 074 105  8650X         JZ      ADD3            IF NO MORE
105,057  147          8651X ADD2.5  MOV     H,A             (H) = SHIFT COUNT
105,060  376 010      8652X         CPI     8
105,062  322 041 105  8653X         JNC     ADD2            IF MORE THAN 8
105,065  315 231 107  8654X ADD2.7  CALL    SRS             SHIFT RIGHT WITH SIGN EXTEND
105,070  045          8655X         DCR     H
105,071  302 065 105  8656X         JNZ     ADD2.7
                      8657X
                      8658X *       NUMBERS ALLIGNED, PERFORM ADD
                      8659X
105,074  041 202 042  8660X ADD3    LXI     H,ACCX
105,077  171          8661X         MOV     A,C
105,100  365          8662X         PUSH    PSW             SAVE OLD Y SIGN
105,101  176          8663X         MOV     A,M
105,102  213          8664X         ADC     E               ADD WITH ROUND
105,103  137          8665X         MOV     E,A
105,104  043          8666X         INX     H
105,105  176          8667X         MOV     A,M
105,106  212          8668X         ADC     D
105,107  127          8669X         MOV     D,A
105,110  043          8670X         INX     H
105,111  176          8671X         MOV     A,M
105,112  211          8672X         ADC     C
105,113  117          8673X         MOV     C,A             (CDE) = NEW SUM
105,114  176          8674X         MOV     A,M             (A) = X SIGN
105,115  043          8675X         INX     H
105,116  106          8676X         MOV     B,M             (B) = NEW EXPONENT
105,117  037          8677X         RAR
105,120  147          8678X         MOV     H,A             (H) 200 BIT = CARRY, 100 BIT = X SIGN
105,121  361          8679X         POP     PSW             (A) = Y SIGN
105,122  037          8680X         RAR
105,123  254          8681X         XRA     H               (A) 100 BIT = XSIGN XOR YSIGN
105,124  027          8682X         RAL
105,125  251          8683X         XRA     C               (A) 200 BIT = XSIGN XOR YSIGN XOR SUMSIGN
105,126  254          8684X         XRA     H               (A) = XSIGN XOR YSIGN XOR SUMSIGN XOR CARRY
```

```
105.127  362 142 105  8685X          JP      ADD4            IS NOT OVERFLOW
                      8686X
                      8687X *        IS OVERFLOW. SHIFT RIGHT 1
                      8688X
105.132  174          8689X          MOV     A,H
105.133  315 232 107  8690X          CALL    SRS.            SHIFT RIGHT
105.136  004          8691X          INR     B
105.137  312 136 070  8692X          JZ      ERR.OV          IF OVERFLOW
                      8693X
                      8694X *        RESULT IN (B,C,D,E)
                      8695X
105.142  305          8696X ADD4     PUSH    B               SAVE OLD EXPONENT
105.143  315 213 105  8697X          CALL    NRM
105.146  361          8698X          POP     PSW             (A) = OLD EXPONENT
105.147  220          8699X          SUB     B
105.150  376 025      8700X          CPI     21
105.152  324 221 105  8701X          CNC     NRM0            USE 0 IF HAVE LOST 21 BITS OF SIGNIFICANCE
105.155  303 245 106  8702X          JMP     STX             STORE AND EXIT
                      8703X
                      8704X *        NORMALIZE RESULT = (BCDE)                       /80.02.GC/
                      8705X
105.160  315 213 105  8706X ADD5     CALL    NRM             NORMALIZE               /80.02.GC/
105.163  303 245 106  8707X          JMP     STX             STORE AND EXIT          /80.02.GC/


                      8709X **       FPSUB - FLOATING POINT SUBTRACT.
                      8710X *
                      8711X *        FPSUB COMPUTES (DE) - ACCX
                      8712X *
                      8713X *        ENTRY (DE) = POINTER TO 4 BYTE FP VALUE
                      8714X *        EXIT    ACCX = RESULT
                      8715X *                SUPPLIED VALUE UNCHANGED
                      8716X *        USES    A,F
                      8717X
                      8718X
105.166  315 215 107  8719X FPSUB    CALL    SPE             SETUP PACKAGE ENTRY/EXIT
105.171  353          8720X          XCHG                    (HL) = ADDRESS
105.172  345          8721X SUB      PUSH    H               SAVE
105.173  315 305 105  8722X          CALL    NEG             NEGATE (ACCX)
105.176  341          8723X          POP     H               (HL) = ADDRESS OF VALUE
105.177  303 356 104  8724X          JMP     ADD             ADD, RESTORE, RETURN
```

```
                                8728X **      FPNRM - FLOATING POINT NORMALIZE.
                                8729X *
                                8730X *       FPNRM NORMALIZES THE CONTENTS OF (ACCX).
                                8731X *
                                8732X *       ENTRY NONE
                                8733X *       EXIT    (ACCX) NORMALIZED
                                8734X *       USES    A,F
                                8735X
                                8736X
105.202  315 215 107            8737X FPNRM   CALL    SPE             SETUP PACKAGE ENTRY
105.205  315 245 107            8738X NRM.    CALL    LDX             (BCDE) = (ACCX)
105.210  303 142 105            8739X         JMP     ADD4            NORMALIZE AND STORE


                                8741X **      NRM - NORMALIZE NUMBER.
                                8742X *
                                8743X *       ENTRY (B,C,D,E) = NUMBER
                                8744X *       EXIT    NORMAILLIZED
                                8745X *       USES    H,L
                                8746X
                                8747X
105.213                         8748X NRM     EQU     *
105.213  171                    8749X         MOV     A,C
105.214  262                    8750X         ORA     D
105.215  263                    8751X         ORA     E
105.216  302 242 105            8752X         JNZ     NRM2            IF NON-ZERO
                                8753X
                                8754X *       NUMBER IS ZERO
                                8755X
105.221  001 000 000            8756X NRM0    LXI     B,0
105.224  120                    8757X         MOV     D,B
105.225  130                    8758X         MOV     E,B             (BCDE) = 0
105.226  311                    8759X         RET
                                8760X
                                8761X *       NUMBER IS NON-ZERO
                                8762X
105.227  112                    8763X NRM1    MOV     C,D
105.230  123                    8764X         MOV     D,E
105.231  137                    8765X         MOV     E,A
105.232  170                    8766X         MOV     A,B
105.233  326 011                8767X         SUI     9
105.235  332 136 070            8768X         JC      ERR.OV          IF OVERFLOW
105.240  074                    8769X         INR     A
105.241  107                    8770X         MOV     B,A
                                8771X
105.242  171                    8772X NRM2    MOV     A,C
105.243  027                    8773X         RAL
105.244  251                    8774X         XRA     C
105.245  027                    8775X         RAL
105.246  330                    8776X         RC                      IF NORMALIZED
105.247  172                    8777X         MOV     A,D
105.250  027                    8778X         RAL
105.251  171                    8779X         MOV     A,C
105.252  027                    8780X         RAL
```

```
105.253  322 257 105  8781X          JNC     NRM3            IF PL
105.256  074         8782X          INR     A
105.257  247         8783X NRM3     ANA     A
105.260  312 227 105  8784X          JZ      NRM1            IF A FULL WORD TO SHIFT
                      8785X
                      8786X *        SHIFT LEFT UNTIL NORMALIZED
                      8787X
105.263  315 101 107  8788X NRM4     CALL    LSH             LSFT SHIFT
105.266  005         8789X          DCR     B
105.267  312 221 105  8790X          JZ      NRM0            UNDERFLOW
105.272  171         8791X          MOV     A,C
105.273  027         8792X          RAL
105.274  251         8793X          XRA     C
105.275  027         8794X          RAL
105.276  322 263 105  8795X          JNC     NRM4            IF MORE TO SHIT
105.301  311         8796X          RET                     EXIT
```

```
                                  8800X **      FPNEG - FLOATING POINT NEGATE.
                                  8801X *
                                  8802X *       FPNEG NEGATES THE CONTENTS OF ACCX.
                                  8803X *
                                  8804X *       ENTRY NONE
                                  8805X *       EXIT    (ACCX) = -(ACCX)
                                  8806X *       USES    A,F
                                  8807X
                                  8808X
     105.302  315 215 107         8809X FPNEG   CALL    SPE             SETUP PACKAGE ENTRY
     105.305  315 245 107         8810X NEG     CALL    LDX             (BCDE) = (ACCX)
     105.310  315 260 107         8811X         CALL    TCV             TWO'S COMPLEMENT IT
     105.313  303 245 106         8812X         JMP     STX             STORE AND RETURN


                                  8815X **      FPTST - FLOATING POINT TEST.
                                  8816X *
                                  8817X *       FPTST TESTS THE SIGN AND VALUE OF (ACCX).
                                  8818X *
                                  8819X *       ENTRY NONE
                                  8820X *       EXIT    'Z' SET IF (ACCX) = 0
                                  8821X *               'M' SET IF (ACCX) < 0
                                  8822X *       USES    A,F
                                  8823X
                                  8824X
     105.316  072 204 042         8825X FPTST   LDA     ACCX+2
     105.321  247                 8826X         ANA     A               SET CONDITION CODE
     105.322  311                 8827X         RET
```

```
                                8830X **      FPMUL - FLOATING POINT MULTIPLY.
                                8831X *
                                8832X *       ENTRY (DE) = ADDRESS OF Y
                                8833X *       EXIT    ACCX = ACCX * Y
                                8834X *       USES    A,F
                                8835X
                                8836X
105.323  315 215 107           8837X FPMUL   CALL    SPE             SETUP PACKAGE ENTRY
105.326  353                   8838X         XCHG                    (HL) = ADDRESS OF VALUE



                                8840X **      MUL - FLOATING POINT MULTIPLY.
                                8841X *
                                8842X
                                8843X
105.327                         8844X MUL     EQU     *
105.327  021 200 000           8845X         LXI     D,MI,ADDB       (DE) = 'ADD B', 'NOP'
105.332  315 114 107           8846X         CALL    PMD             PREPARE MULTIPLY
105.335  312 240 106           8847X         JZ      MUL5            IS ZERO
105.340  332 136 070           8848X         JC      ERR.OV          IS OVERFLOW
105.343  147                   8849X         MOV     H,A             SAVE NEW EXPONENT
105.344  345                   8850X         PUSH    H               SAVE NEW EXP AND SIGN
105.345  171                   8851X         MOV     A,C
105.346  062 034 106           8852X         STA     MULA            SETUP MULTIPLICAND
105.351  062 076 106           8853X         STA     MULD
105.354  062 143 106           8854X         STA     MULH
105.357  172                   8855X         MOV     A,D
105.360  062 072 106           8856X         STA     MULC
105.363  062 137 106           8857X         STA     MULG
105.366  173                   8858X         MOV     A,E
105.367  062 133 106           8859X         STA     MULF
105.372  041 204 042           8860X         LXI     H,ACCX+2
105.375  176                   8861X         MOV     A,M
105.376  062 121 106           8862X         STA     MULE
106.001  053                   8863X         DCX     H
106.002  176                   8864X         MOV     A,M
106.003  062 051 106           8865X         STA     MULB
106.006  053                   8866X         DCX     H
106.007  106                   8867X         MOV     B,M
106.010  046 007               8868X         MVI     H,7
106.012  154                   8869X         MOV     L,H
106.013  021 000 000           8870X         LXI     D,0
106.016  112                   8871X         MOV     C,D             ZERO ACCUMULATOR
106.017  170                   8872X         MOV     A,B
106.020  247                   8873X         ANA     A
106.021  312 047 106           8874X         JZ      MUL2.5
106.024  170                   8875X L1      MOV     A,B             (A) = MULTIPLICAND
106.025  037                   8876X         RAR
106.026  107                   8877X         MOV     B,A
106.027  171                   8878X         MOV     A,C
106.030  322 035 106           8879X         JNC     L2              BIT NOT PRESENT
106.033  306 000               8880X         ADI     0
106.034                         8881X MULA    EQU     *-1
106.035  037                   8882X L2      RAR
```

```
106.036  117          8883X          MOV     C,A
106.037  045          8884X          DCR     H
106.040  362 024 106  8885X          JP      L1
106.043  322 047 106  8886X          JNC     MUL2.5          NOT CARRY
106.046  014          8887X          INR     C
                      8888X
                      8889X *        2ND PARTIAL PRODUCT
                      8890X
106.047  145          8891X MUL2.5   MOV     H,L
106.050  006 000      8892X          MVI     B,0
106.051               8893X MULB     EQU     *-1
106.052  072 202 042  8894X          LDA     ACCX
106.055  260          8895X          ORA     B
106.056  312 120 106  8896X          JZ      L4.5            NONE IN LOW TWO BYTES
                      8897X
106.061  170          8898X MUL3     MOV     A,B
106.062  037          8899X          RAR
106.063  107          8900X          MOV     B,A
106.064  171          8901X          MOV     A,C
106.065  322 077 106  8902X          JNC     L4              NOT SET
106.070  172          8903X          MOV     A,D
106.071  306 000      8904X          ADI     0
106.072               8905X MULC     EQU     *-1
106.073  127          8906X          MOV     D,A
106.074  171          8907X          MOV     A,C
106.075  316 000      8908X          ACI     0
106.076               8909X MULD     EQU     *-1
106.077  037          8910X L4       RAR
106.100  117          8911X          MOV     C,A
106.101  172          8912X          MOV     A,D
106.102  037          8913X          RAR
106.103  127          8914X          MOV     D,A
106.104  045          8915X          DCR     H
106.105  362 061 106  8916X          JP      MUL3
106.110  322 120 106  8917X          JNC     L4.5            NOT CARRY
106.113  024          8918X          INR     D
106.114  302 120 106  8919X          JNZ     L4.5
106.117  014          8920X          INR     C
106.120  006 000      8921X L4.5     MVI     B,0
106.121               8922X MULE     EQU     *-1
                      8923X
106.122  170          8924X L5       MOV     A,B
106.123  037          8925X          RAR
106.124  107          8926X          MOV     B,A
106.125  171          8927X          MOV     A,C
106.126  322 144 106  8928X          JNC     L6              NOT SET
106.131  173          8929X          MOV     A,E
106.132  306 000      8930X          ADI     0
106.133               8931X MULF     EQU     *-1
106.134  137          8932X          MOV     E,A
106.135  172          8933X          MOV     A,D
106.136  316 000      8934X          ACI     0
106.137               8935X MULG     EQU     *-1
106.140  127          8936X          MOV     D,A
106.141  171          8937X          MOV     A,C
106.142  316 000      8938X          ACI     0
```

```
106.143                    8939X MULH    EQU     *-1
106.144  037              8940X L6      RAR
106.145  117              8941X         MOV     C,A
106.146  172              8942X         MOV     A,D
106.147  037              8943X         RAR
106.150  127              8944X         MOV     D,A
106.151  173              8945X         MOV     A,E
106.152  037              8946X         RAR
106.153  137              8947X         MOV     E,A
106.154  055              8948X         DCR     L
106.155  302 122 106       8949X         JNZ     L5
106.160  322 174 106       8950X         JNC     L7              NOT TO ROUND UP
106.163  034              8951X         INR     E
106.164  302 174 106       8952X         JNZ     L7
106.167  024              8953X         INR     D
106.170  302 174 106       8954X         JNZ     L7
106.173  014              8955X         INR     C
                           8956X *       NORMALIZE
                           8957X
106.174  171              8958X L7      MOV     A,C
106.175  341              8959X         POP     H               (HL) = EXPONENT AND SIGN
106.176  104              8960X         MOV     B,H
106.177  027              8961X         RAL
106.200  247              8962X         ANA     A
106.201  372 216 106       8963X         JM      MUL3.5          NORMALIZED
                           8964X
106.204  173              8965X         MOV     A,E
106.205  027              8966X         RAL
106.206  137              8967X         MOV     E,A
106.207  172              8968X         MOV     A,D
106.210  027              8969X         RAL
106.211  127              8970X         MOV     D,A
106.212  171              8971X         MOV     A,C
106.213  027              8972X         RAL
106.214  117              8973X         MOV     C,A
106.215  005              8974X         DCR     B               ADJUST EXPONENT
106.216  004              8975X MUL3.5  INR     B               ADJUST EXPONENT
106.217  312 136 070       8976X         JZ      ERR.OV
                           8977X
                           8978X *       NEGATE IF NECESSARY
                           8979X
106.222  175              8980X MUL4    MOV     A,L
106.223  247              8981X         ANA     A
106.224  374 260 107       8982X         CM      TCV             TWOS COMP VALUE
106.227  170              8983X         MOV     A,B
106.230  247              8984X         ANA     A
106.231  312 245 106       8985X         JZ      STX             VALUE IS 0
106.234  005              8986X         DCR     B
106.235  302 245 106       8987X         JNZ     STX             NOT UNDERFLOW
                           8988X
                           8989X *       RESULT = 0
                           8990X
106.240  001 000 000       8991X MUL5    LXI     B,0
106.243  120              8992X         MOV     D,B
106.244  130              8993X         MOV     E,B             (BCDE) = 0
                           8994X *       JMP     STX
```

```
                                8996X **        STX - STORE REGISTERS INTO X VALUE.
                                8997X *
                                8998X *         ENTRY (B,C,D,E) = VALUES
                                8999X *         EXIT    STORED IN REG.X
                                9000X
                                9001X
     106.245  041 202 042       9002X STX       LXI     H,ACCX
     106.250  163               9003X STO       MOV     M,E
     106.251  043               9004X           INX     H
     106.252  162               9005X           MOV     M,D
     106.253  043               9006X           INX     H
     106.254  161               9007X           MOV     M,C
     106.255  043               9008X           INX     H
     106.256  160               9009X           MOV     M,B
     106.257  311               9010X           RET
```

```
                                9013X **      FPDIV - FLOATING POINT DIVIDE.
                                9014X *
                                9015X *       ACCX = ACCX/Y
                                9016X *
                                9017X *       ENTRY (DE) = POINTER TO Y
                                9018X *       EXIT    (ACCX) = RESULT
                                9019X *       USES    A,F
                                9020X
                                9021X
      106.260  315 215 107      9022X FPDIV   CALL    SPE             SETUP PACKAGE ENTRY
      106.263  353              9023X         XCHG                    (HL) = ADDRESS OF VALUE


                                9025X **      DIV - FLOATING POINT DEVIDE.
                                9026X *
                                9027X *       X=Y/X
                                9028X
                                9029X
                                9030X
      106.264                   9031X DIV     EQU     *
      106.264  021 220 077      9032X         LXI     D,MI.CMC*256+MI.SUBB    (DE) = 'SUB B', 'CMC'
      106.267  315 114 107      9033X         CALL    PMD             PRESET FOR DEVICDE
      106.272  302 305 106      9034X         JNZ     DIV0            IF NEIGHER ZERO
      106.275  170              9035X         MOV     A,B
      106.276  247              9036X         ANA     A
      106.277  312 117 070      9037X         JZ      ERR.DO          (Y) = 0
      106.302  303 240 106      9038X         JMP     MULS            (X) = 0
                                9039X
      106.305  332 136 070      9040X DIV0    JC      ERR.OV          IF OVERFLOW
      106.310  074              9041X         INR     A
      106.311  312 136 070      9042X         JZ      ERR.OV          IF OVERFLOW
      106.314  147              9043X         MOV     H,A             (H) = RESULT EXP, (L) = RESULT SIGN
      106.315  345              9044X         PUSH    H
      106.316  173              9045X         MOV     A,E
      106.317  062 367 106      9046X         STA     DIVA
      106.322  172              9047X         MOV     A,D
      106.323  062 373 106      9048X         STA     DIVB
      106.326  171              9049X         MOV     A,C
      106.327  062 377 106      9050X         STA     DIVC
      106.332  171              9051X         MOV     A,C
      106.333  062 016 107      9052X         STA     PMAC+1
      106.336  172              9053X         MOV     A,D
      106.337  062 012 107      9054X         STA     PMAB+1
      106.342  173              9055X         MOV     A,E
      106.343  062 006 107      9056X         STA     PMAA+1
      106.346  315 245 107      9057X         CALL    LDX             (BCDE) = X VALUE
      106.351  053              9058X         DCX     H
      106.352  345              9059X         PUSH    H
      106.353  056 003          9060X         MVI     L,3             (L) = LOOP COUNT
      106.355  076 002          9061X DIV1    MVI     A,2
      106.357  275              9062X         CMP     L
      106.360  336 372          9063X         SBI     -6
      106.362  147              9064X         MOV     H,A             (H) = 7 IF FIRST, 8 IF 2ND OR 3RD
      106.363  006 000          9065X         MVI     B,0             (B) = RESULT
```

```
106.365  173          9066X DIV2   MOV     A,E
106.366  326 000      9067X        SUI     0
106.367               9068X DIVA   EQU     *-1
106.370  137          9069X        MOV     E,A
106.371  172          9070X        MOV     A,D
106.372  336 000      9071X        SBI     0
106.373               9072X DIVB   EQU     *-1
106.374  127          9073X        MOV     D,A
106.375  171          9074X        MOV     A,C
106.376  336 000      9075X        SBI     0
106.377               9076X DIVC   EQU     *-1
107.000  117          9077X        MOV     C,A
107.001  322 020 107  9078X        JNC     DIV3
107.004  173          9079X        MOV     A,E
107.005  306 000      9080X PMAA   ADI     0
107.007  137          9081X        MOV     E,A
107.010  172          9082X        MOV     A,D
107.011  316 000      9083X PMAB   ACI     0
107.013  127          9084X        MOV     D,A
107.014  171          9085X        MOV     A,C
107.015  316 000      9086X PMAC   ACI     0
107.017  117          9087X        MOV     C,A
107.020  077          9088X DIV3   CMC
                      9089X
                      9090X *      SET RESULT BIT IN ACCUMULATOR
                      9091X
107.021  170          9092X        MOV     A,B
107.022  027          9093X        RAL
107.023  107          9094X        MOV     B,A
                      9095X
                      9096X *      SHIFT REMAINDER VALUE LEFT 1
                      9097X
107.024  315 101 107  9098X        CALL    LSH
107.027  045          9099X        DCR     H
107.030  302 365 106  9100X        JNZ     DIV2
                      9101X
                      9102X *      STORE SUBVALUE
                      9103X
107.033  343          9104X        XTHL
107.034  160          9105X        MOV     M,B
107.035  053          9106X        DCX     H
107.036  343          9107X        XTHL
107.037  055          9108X        DCR     L
107.040  302 355 106  9109X        JNZ     DIV1
107.043  341          9110X        POP     H
107.044  043          9111X        INX     H
107.045  043          9112X        INX     H
107.046  130          9113X        MOV     E,B
107.047  126          9114X        MOV     D,M
107.050  043          9115X        INX     H
107.051  171          9116X        MOV     A,C          (A) = REMAINDER HIGH ORDER
107.052  116          9117X        MOV     C,M          (BCDE) = RESULT
107.053  341          9118X        POP     H
107.054  104          9119X        MOV     B,H          (B) = RESULTANT EXPONENT
107.055  147          9120X        MOV     H,A
107.056  072 377 106  9121X        LDA     DIVC
```

```
107.061  234          9122X  SUB   B        SEE IF NEXT RESULT BIT WOULD BE 1 (OR CLOSE
107.062  334 315 107  9123X  CC    RVU      ROUND VALUE UP IF SO
107.065  171          9124X  MOV   A,C
107.066  346 100      9125X  ANI   100Q
107.070  302 216 106  9126X  JNZ   MUL3.5   IF NOT TO NORMALIZE
107.073  315 101 107  9127X  CALL  LSH
107.076  303 222 106  9128X  JMP   MUL4
```

```
                                9132X **     LSH - LEFT SHIFT VALUE,
                                9133X *
                                9134X *      ENTRY (C,D,E) = VALUE
                                9135X *      EXIT    (C,D,E) SHIFTED RIGHT 1
                                9136X
                                9137X
107.101  247                   9138X LSH    ANA     A               CLEAR CARRY
107.102  173                   9139X        MOV     A,E
107.103  027                   9140X        RAL
107.104  137                   9141X        MOV     E,A
107.105  172                   9142X        MOV     A,D
107.106  027                   9143X        RAL
107.107  127                   9144X        MOV     D,A
107.110  171                   9145X        MOV     A,C
107.111  027                   9146X        RAL
107.112  117                   9147X        MOV     C,A
107.113  311                   9148X        RET


                                9150X **     PMD - PRESET MULTIPLY/DIVIED
                                9151X *
                                9152X *      ENTRY (DE) = EXPONENT INSTRUCTIONS (FOR MULTIPLY OR DIVIDE)
                                9153X *             (HL) = ADDRESS OF 'Y' VALUE
                                9154X *      EXIT    (C,D,E) = X VALUES
                                9155X *             'Z' SET IF VALUE ZERO
                                9156X *             'C' SET OF OVERFLOW
                                9157X *             'L' = NEW SIGN
                                9158X *             (A) = NEW EXPONENT
                                9159X
                                9160X
107.114                         9161X PMD    EQU     *
107.114  345                   9162X        PUSH    H
107.115  353                   9163X        XCHG                    (HL) = EXPONENT INSTRUCTIONS
107.116  042 164 107           9164X        SHLD    PMDB
107.121  041 202 042           9165X        LXI     H,ACCX
107.124  072 204 042           9166X        LDA     ACCX+2
107.127  062 152 107           9167X        STA     PMDA            SET SIGN
107.132  247                   9168X        ANA     A
107.133  374 204 107           9169X        CM      PMD2            IF MUST COMPLEMENT X
107.136  341                   9170X        POP     H               (HL) = ADDRESS OF Y
107.137  315 250 107           9171X        CALL    LDD             LOAD NUMBER
107.142  171                   9172X        MOV     A,C
107.143  157                   9173X        MOV     L,A             (L) = SIGN
107.144  247                   9174X        ANA     A
107.145  374 260 107           9175X        CM      TCV             IS NEGATIVE
107.150  175                   9176X        MOV     A,L             (A) = SIGN
107.151  356 000               9177X        XRI     0               COMPARE SIGNS
107.152                         9178X PMDA   EQU     *-1             SIGN OF X
107.153  157                   9179X        MOV     L,A
107.154  170                   9180X        MOV     A,B
107.155  247                   9181X        ANA     A
107.156  310                   9182X        RZ                      IF ZERO
107.157  072 205 042           9183X        LDA     ACCX+3
107.162  247                   9184X        ANA     A
```

```
107.163  310              9185X          RZ                      IF ZERO
107.164  200              9186X PMDB     ADD     B               IF DIVIDE, = 'SUB  B'
107.165  000              9187X          NOP                             = 'CMC'
107.166  107              9188X          MOV     B,A             (B) = SUM OF 2
                          9189X
                          9190X *        SEE IF EXPONENT OVERFLOW
                          9191X
107.167  037              9192X          RAR
107.170  250              9193X          XRA     B
107.171  170              9194X          MOV     A,B             (A) = SUM OF EXPONENTS
107.172  362 200 107      9195X          JP      PMD1            OVERFLOW OR UNDERFLOW
107.175  356 200          9196X          XRI     200Q
107.177  311              9197X          RET                     'Z' SET IF UNDERFLOW
                          9198X
                          9199X *        OVERFLOW OR UNDERFLOW.
                          9200X
107.200  007              9201X PMD1     RLC                     'C' IF OVERFLOW
107.201  330              9202X          RC
107.202  257              9203X          XRA     A               UNDERFLOW. SET *Z*
107.203  311              9204X          RET                     EXIT
                          9205X
                          9206X
                          9207X *        COMPLEMENT ACCX TO A POSITIVE NUMBER
                          9208X
107.204  315 245 107      9209X PMD2     CALL    LDX
107.207  315 260 107      9210X          CALL    TCV
107.212  303 245 106      9211X          JMP     STX             STORE AND RETURN


                          9213X **       SPE - SETUP PACKAGE ENTRY.
                          9214X *
                          9215X *        SPE IS CALLED UPON ENTRY TO THE FLOATING POINT PACKAGE.
                          9216X *
                          9217X *        IT SAVES THE REGISTERS ON THE STACK, SETS UP A RETURN ADDRESS
                          9218X *        TO A RESTORE REGISTER ROUTINE, AND THEN ENTERS THE SELECTED
                          9219X *        ROUTINE.
                          9220X *
                          9221X *        ENTRY (SP+0) = ADDRESS TO RETURN CONTROL TO
                          9222X *        EXIT    REGISTERS ON STACK, *SPEX* SET AS RETURN ADDRESS
                          9223X *        USES    B,C,H,L
                          9224X
                          9225X
107.215  343              9226X SPE      XTHL                    SAVE H
107.216  325              9227X          PUSH    D               SAVE D
107.217  305              9228X          PUSH    B               SAVE B
107.220  001 225 107      9229X          LXI     B,SPEX
107.223  305              9230X          PUSH    B               SET 'RETURN ADDRESS'
107.224  351              9231X          PCHL                    ENTER ROUTINE
                          9232X
                          9233X *        RETURN FROM ROUTINE, RESTORE REGISTERS AND RETURN TO CALLER.
                          9234X
107.225  301              9235X SPEX     POP     B
107.226  321              9236X          POP     D
107.227  341              9237X          POP     H
```

```
107,230  311          9238X          RET


                      9240X **       SRS - SHIFT RIGHT WITH SIGN EXTEND.
                      9241X *
                      9242X *        ENTRY (C,D,E) = VALUE
                      9243X *        EXIT   (C,D,E) SHIFTED RIGHT 1 BIT
                      9244X *        USES   A
                      9245X
107,231               9246X SRS      EQU    *
107,231  171         9247X          MOV    A,C
107,232  027         9248X SRS.     RAL
107,233  171         9249X SRS..    MOV    A,C
107,234  037         9250X          RAR
107,235  117         9251X          MOV    C,A
107,236  172         9252X          MOV    A,D
107,237  037         9253X          RAR
107,240  127         9254X          MOV    D,A
107,241  173         9255X          MOV    A,E
107,242  037         9256X          RAR
107,243  137         9257X          MOV    E,A
107,244  311         9258X          RET


                      9260X **       LDX - LOAD X VALUE INTO REGISTERS
                      9261X *
                      9262X *        ENTRY NONE
                      9263X *        EXIT   (BCDE) = (ACCX)
                      9264X *        USES   ALL
                      9265X
                      9266X
107,245  041 202 042  9267X LDX      LXI    H,ACCX


                      9269X **       LDD - LOAD VALUE INTO REGISTERS.
                      9270X *
                      9271X *        ENTRY (HL) = ADDRESS OF VALUE
                      9272X *        EXIT   (B,C,D,E) = X VALUE
                      9273X
                      9274X
107,250  136         9275X LDD      MOV    E,M
107,251  043         9276X          INX    H
107,252  126         9277X          MOV    D,M
107,253  043         9278X          INX    H
107,254  116         9279X          MOV    C,M
107,255  043         9280X          INX    H
107,256  106         9281X          MOV    B,M
107,257  311         9282X          RET
```

```
                                9284X **      TCV - TWOS COMPLEMENT VALUE.
                                9285X *
                                9286X *       ENTRY (BCDE) = VALUE
                                9287X
      107.260                   9288X TCV     EQU     *
      107.260  171             9289X         MOV     A,C
      107.261  057             9290X         CMA
      107.262  117             9291X         MOV     C,A
      107.263  172             9292X         MOV     A,D
      107.264  057             9293X         CMA
      107.265  127             9294X         MOV     D,A
      107.266  173             9295X         MOV     A,E
      107.267  057             9296X         CMA
      107.270  137             9297X         MOV     E,A
      107.271  034             9298X         INR     E
      107.272  300             9299X         RNZ
      107.273  024             9300X         INR     D
      107.274  300             9301X         RNZ
      107.275  171             9302X         MOV     A,C             (A) = SIGN
      107.276  014             9303X         INR     C
      107.277  247             9304X         ANA     A
      107.300  372 213 105     9305X         JM      NRM             IF POSITIVE TO NEGATIVE, NORMALIZE
      107.303  171             9306X         MOV     A,C             WAS NEGATIVE TO POSITIVE, MAY NEED RIGHT SH
      107.304  247             9307X         ANA     A
      107.305  360             9308X         RP                      DONT NEED SHIFT
      107.306  004             9309X         INR     B
      107.307  312 136 070     9310X         JZ      ERR.OV          IF OVERFLOW
      107.312  303 233 107     9311X         JMP     SRS.            SHIFT RIGHT AND EXIT


                                9313X **      RVU - ROUND VALUE UP.
                                9314X *
                                9315X *       RVU IS CALLED TO ADD ONE BIT TO THE VALUE.
                                9316X *
                                9317X *       ENTRY (BCDE) = VALUE
                                9318X *       EXIT    (BCDE) ADJUSTED
                                9319X *       USES    A,F,B,C,D,E
                                9320X
                                9321X
      107.315  034             9322X RVU     INR     E
      107.316  300             9323X         RNZ                     NO CARRY
      107.317  024             9324X         INR     D
      107.320  300             9325X         RNZ                     NO CARRY
      107.321  014             9326X         INR     C
      107.322  311             9327X         RET
      107.323                  9328          XTEXT   FPC
```

```
                                9331X **      ATF - ASCII TO FLOATING.
                                9332X *
                                9333X *       ATF CONVERTS AN ASCII STRING INTO A FLOATING POINT VALUE
                                9334X *       IN ACCX.
                                9335X *
                                9336X *       SYNTAX
                                9337X *
                                9338X *        NNNN [.NNN] [E [+-] NN]
                                9339X *
                                9340X *       NO LEADING BLANKS ALLOWED, A SINGLE LEADING
                                9341X *       '-' IS ALLOWED, AND PROCESSED.
                                9342X *
                                9343X *       ENTRY   (HL) = ADDRESS OF TEXT
                                9344X *       EXIT    (HL) UPDATED
                                9345X *               (ACCX) = VALUE
                                9346X *       USES    A,F,H,L
                                9347X
                                9348X
      107.323                   9349X ATF     EQU     *
      107.323  305             9350X         PUSH    B               SAVE REGISTERS
      107.324  325             9351X         PUSH    D
      107.325  176             9352X         MOV     A,M             SEE IF '-'
      107.326  376 055         9353X         CPI     '-'
      107.330  365             9354X         PUSH    PSW             SAVE RESULTS UNTIL THE VERY END
      107.331  302 335 107     9355X         JNE     ATF0            NOT -
      107.334  043             9356X         INX     H               SKIP '-'
      107.335  345             9357X ATF0    PUSH    H               SAVE TEXT POINTER
      107.336  006 006         9358X         MVI     B,6             DIGIT COUNT+2
                                9359X
                                9360X *       COUNT # OF SIGNIFICANT DIGITS
                                9361X
      107.340  005             9362X ATF1    DCR     B
      107.341  312 053 110     9363X         JZ      ATF3            TOO MANY DIGITS
      107.344  176             9364X         MOV     A,M
      107.345  043             9365X         INX     H
      107.346  376 056         9366X         CPI     '.'
      107.350  312 340 107     9367X         JE      ATF1            DONT COUNT DECIMAL POINT
      107.353  376 060         9368X         CPI     '0'
      107.355  332 365 107     9369X         JC      ATF1.5          NOT DIGIT
      107.360  376 072         9370X         CPI     '9'+1
      107.362  332 340 107     9371X         JC      ATF1            IS DIGIT
                                9372X
                                9373X *       WILL DECODE NUMBER AS DECIMAL INTEGER
                                9374X
      107.365  341             9375X ATF1.5  POP     H               (HL) = START OF NUMBER
      107.366  021 000 000     9376X         LXI     D,0
      107.371  315 202 111     9377X         CALL    DDN1            DECODE DECIMAL NUMBER
      107.374  006 000         9378X         MVI     B,0             ZERO AFTER-DECIMAL COUNT
      107.376  076 056         9379X         MVI     A,'.'
      110.000  276             9380X         CMP     M
      110.001  314 233 111     9381X         CE      DDN2            DECODE FRACTIONAL, IF ANY
      110.004  305             9382X         PUSH    B               SAVE DP COUNT
      110.005  112             9383X         MOV     C,D
      110.006  123             9384X         MOV     D,E
      110.007  257             9385X         XRA     A               CLEAR CARRY
      110.010  137             9386X         MOV     E,A             (E) = 0
```

```
                        9387X          MOV     B,E             (B) = 0
                        9388X          MOV     A,C
                        9389X          ORA     D
                        9390X          JZ      ATF2.5          IS 0
                        9391X          MVI     B,217Q
                        9392X
                        9393X *        NORMALIZE
                        9394X
                        9395X ATF2     MOV     A,D
                        9396X          RAL
                        9397X          MOV     D,A
                        9398X          MOV     A,C
                        9399X          RAL
                        9400X          MOV     C,A
                        9401X          DCR     B
                        9402X          ANI     100Q
                        9403X          JZ      ATF2            MORE TO GO
                        9404X ATF2.5   XCHG
                        9405X          SHLD    ACCX            SET LOW-ORDER
                        9406X          MOV     H,B
                        9407X          MOV     L,C
                        9408X          SHLD    ACCX+2          SET HIGH-ORDER
                        9409X          XCHG                    (HL) = NEXT BYTE ADDR
                        9410X          POP     B               (B) = SCALE COUNT
                        9411X          JMP     ATF5            CHECK FOR EXPONENT
                        9412X
                        9413X *        MUST DECODE VIA FLOATING NUMBERS.
                        9414X
                        9415X ATF3     CALL    MULS            CLEAR ACCX
                        9416X          MVI     B,227Q-16
                        9417X          LXI     H,ACCY
                        9418X          CALL    STO             SETUP Y
                        9419X          POP     H               (HL) = NUMBER START
                        9420X ATF4     CALL    DFD             DECODE FLOATING DECIMAL
                        9421X          MVI     B,0             CLEAR DP COUNT
                        9422X          MVI     A,'.'
                        9423X          CMP     M
                        9424X          CE      DFD1            IF FRACTIONAL PART
                        9425X
                        9426X *        HAVE FLOATING VALUE, LOOK FOR E+-NN
                        9427X *        (B) = DP SCALE COUNT
                        9428X
                        9429X ATF5     MVI     A,'E'
                        9430X          CMP     M
                        9431X          MVI     A,0             ASSUME HAVE ONE
                        9432X          JNE     ATF8            HAVE NONE
                        9433X          INX     H               INCREMENT PAST 'E'
                        9434X
                        9435X *        DECODE EXPONENT.
                        9436X
                        9437X          MOV     A,M             (A) = NEXT CHARACTER
                        9438X          SUI     '+'
                        9439X          JZ      ATF6            IS +
                        9440X          CPI     '-'-'+'
                        9441X          MVI     A,80H           ASSUME -
                        9442X          JE      ATF6            IS -
```

```
110.130  257           9443X          XRA     A               IS NONE, USE +
110.131  053           9444X          DCX     H
110.132  043           9445X ATF6     INX     H               ADVANCE PAST + OR -
110.133  365           9446X          PUSH    PSW             SAVE SIGN
110.134  110           9447X          MOV     C,B             (C) = DP COUNT
110.135  315 171 111   9448X          CALL    DDN             DECODE DECIMAL DIGITS
110.140  101           9449X          MOV     B,C             RESTORE DP COUNT
110.141  343           9450X          XTHL                    SAVE (HL), (H) = EXPONENT SIGN
110.142  172           9451X          MOV     A,D
110.143  247           9452X          ANA     A
110.144  302 122 070   9453X          JNZ     ERR.IN          IF TOO LARGE
110.147  174           9454X          MOV     A,H
110.150  027           9455X          RAL                     'C' SET IF NEGATIVE
110.151  173           9456X          MOV     A,E
110.152  322 157 110   9457X          JNC     ATF7            NOT NEGATIVE
110.155  057           9458X          CMA
110.156  074           9459X          INR     A
110.157  341           9460X ATF7     POP     H
110.160  200           9461X ATF8     ADD     B               (A) = SCALE COUNT
110.161  312 230 110   9462X          JZ      ATF11           NO SCALEING
110.164  345           9463X          PUSH    H               SAVE (HL)
110.165  041 327 105   9464X          LXI     H,MUL           ASSUME *                        /78.10.GC/
110.170  042 217 110   9465X          SHLD    ATFA                                            /78.10.GC/
110.173  041 153 112   9466X          LXI     H,FP10.         ASSUME *10
110.176  362 213 110   9467X          JP      ATF9            IS POSITIVE
110.201  345           9468X          PUSH    H                                               /78.10.GC/
110.202  041 264 106   9469X          LXI     H,DIV                                           /78.10.GC/
110.205  042 217 110   9470X          SHLD    ATFA                                            /78.10.GC/
110.210  341           9471X          POP     H                                               /78.10.GC/
110.211  057           9472X          CMA
110.212  074           9473X          INR     A               (A) = COUNT
110.213  117           9474X ATF9     MOV     C,A             (C) = SCALE COUNT
110.214  305           9475X ATF10    PUSH    B
110.215  345           9476X          PUSH    H
110.216  315 327 105   9477X          CALL    MUL             SCALE
110.217               9478X ATFA     EQU     *-2
110.221  341           9479X          POP     H
110.222  301           9480X          POP     B
110.223  015           9481X          DCR     C
110.224  302 214 110   9482X          JNZ     ATF10           IF MORE TO GO
110.227  341           9483X          POP     H               RESTORE (HL)
                       9484X
                       9485X *        DONE.
                       9486X
110.230  361           9487X ATF11    POP     PSW             (PSW) = RESULTS OF EARLY '=' CHECK
110.231  314 302 105   9488X          CE      FPNEG           MUST NEGATE
110.234  321           9489X          POP     D
110.235  301           9490X          POP     B
110.236  311           9491X          RET
```

```
                                9494X **      FTA - FLOATING TO ASCII.
                                9495X *
                                9496X *       FTA CONVERTS A FLOATING POINT NUMBER INTO AN ASCII
                                9497X *       REPRESENTATION..
                                9498X *
                                9499X *       ENTRY   (ACCX) = VALUE
                                9500X *               (HL) = ADDRESS TO STORE TEXT
                                9501X *       EXIT    (A) = LENGTH OF STRING DECODED
                                9502X *               (DE) = ADDRESS OF LAST BYTE
                                9503X *       USES    A,F,D,E
                                9504X
                                9505X
     110.237                    9506X FTA     EQU     *
     110.237  305               9507X         PUSH    B
     110.240  345               9508X         PUSH    H
     110.241  066 040           9509X         MVI     M,' '           INSURE LEADING BLANK
     110.243  072 204 042       9510X         LDA     ACCX+2
     110.246  247               9511X         ANA     A               TEST VALUE
     110.247  362 261 110       9512X         JP      FTA1
     110.252  043               9513X         INX     H               ADD MINUS SIGN
     110.253  066 055           9514X         MVI     M,'-'
     110.255  315 302 105       9515X         CALL    FPNEG           INVERT IT
     110.260  264               9516X         ORA     H               CLEAR 'Z'
     110.261  043               9517X FTA1    INX     H
     110.262  006 001           9518X         MVI     B,1             (B) = EXPONENT
     110.264  312 356 110       9519X         JZ      FTA2.7          IS 0
                                9520X
                                9521X *       SCALE NUMBER
                                9522X
     110.267  021 267 110       9523X FTA2    LXI     D,FTA2
     110.272  325               9524X         PUSH    D               SET 'RETURN ADDRESS'
     110.273  021 153 112       9525X         LXI     D,FP10.
     110.276  072 205 042       9526X         LDA     ACCX+3
     110.301  005               9527X         DCR     B
     110.302  376 201           9528X         CPI     201Q
     110.304  332 323 105       9529X         JC      FPMUL           ACCX = ACCX * 10
     110.307  004               9530X         INR     B
     110.310  004               9531X         INR     B
     110.311  326 205           9532X         SUI     205Q
     110.313  322 260 106       9533X         JNC     FPDIV           ACCX = ACCX / 10
     110.316  074               9534X         INR     A
     110.317  372 332 110       9535X         JM      FTA2.5          IS SCALED
     110.322  072 204 042       9536X         LDA     ACCX+2
     110.325  376 120           9537X         CPI     120Q
     110.327  322 260 106       9538X         JNC     FPDIV
     110.332  005               9539X FTA2.5  DCR     B
     110.333  321               9540X         POP     D               DISCARD 'RETURN ADDRESS'
                                9541X
                                9542X *       ROUND NUMBER
                                9543X
     110.334  072 204 042       9544X         LDA     ACCX+2
     110.337  365               9545X         PUSH    PSW             SAVE HIGH ORDER PART
     110.340  021 165 111       9546X         LXI     D,FTAA
     110.343  315 352 104       9547X         CALL    FPADD           ROUND UP
     110.346  321               9548X         POP     D               (D) = OLD MANTISSA
     110.347  072 204 042       9549X         LDA     ACCX+2
```

```
110.352  272            9550X          CMP     D
110.353  302 267 110     9551X          JNE     FTA2            CAUSED MAJOR CHANGE, ROUND AGAIN
                         9552X
                         9553X *        SCALED, (B) = DECIMAL PLACE
                         9554X
110.356                  9555X FTA2.7   EQU     *
110.356  170            9556X          MOV     A,B
110.357  376 007        9557X          CPI     7
110.360                  9558X FTAC     EQU     *-1             SCIENTIFIC/FIXED FLAG
110.361  365            9559X          PUSH    PSW
110.362  332 367 110     9560X          JC      FTA3            USED FIXED NOTATION
110.365  006 001        9561X          MVI     B,1             USE SCIENTIFIC NOATATION
110.367  016 007        9562X FTA3     MVI     C,7             (C) = DIGIT COUNT (+1 FOR .)
110.370                  9563X FTAD     EQU     *-1
110.371  004            9564X          INR     B               (B) = DIGITS BEFORE DP (+1)
                         9565X
                         9566X *        SEE IF TO PLACE DECIMAL POINT
                         9567X
110.372  005            9568X FTA4     DCR     B
110.373  302 001 111     9569X          JNZ     FTA4.5          NOT TIME FOR DECIMAL POINT
110.376  066 056        9570X          MVI     M,'.'
111.000  043            9571X          INX     H
111.001  015            9572X FTA4.5   DCR     C
111.002  312 066 111     9573X          JZ      FTA8.5          IF ROOM FOR NO MORE DIGITS
                         9574X
                         9575X *        DECODE DIGIT
                         9576X
111.005  345            9577X FTA5     PUSH    H
111.006  041 205 042     9578X          LXI     H,ACCX+3
111.011  126            9579X          MOV     D,M
111.012  172            9580X          MOV     A,D
111.013  376 201        9581X          CPI     2010
111.015  076 000        9582X          MVI     A,0
111.017  332 045 111     9583X          JC      FTA7.5          IF NO DIGIT FOR THIS PLACE
111.022  257            9584X          XRA     A
111.023  067            9585X FTA6     STC
111.024  037            9586X          RAR
111.025  025            9587X          DCR     D
111.026  372 023 111     9588X          JM      FTA6            GENERATE MASK OF 1S FOR SIG BITS
111.031  126            9589X          MOV     D,M             (D) = EXPONENT
111.032  053            9590X          DCX     H
111.033  246            9591X          ANA     M               (A) = VALUE
111.034  137            9592X          MOV     E,A
111.035  256            9593X          XRA     M
111.036  167            9594X          MOV     M,A
111.037  173            9595X          MOV     A,E
111.040  007            9596X FTA7     RLC
111.041  025            9597X          DCR     D
111.042  372 040 111     9598X          JM      FTA7            ROTATE VALUE LOW ORDER
111.045  306 060        9599X FTA7.5   ADI     '0'             (A) = DIGIT
111.047  341            9600X          POP     H
111.050  167            9601X          MOV     M,A
111.051  043            9602X          INX     H
111.052  315 202 105     9603X          CALL    FPNRM           NORMALIZE
111.055                  9604X FTA8     EQU     *
111.055  021 153 112     9605X          LXI     D,FF10,
```

```
111.080 315 323 105  9606X          CALL    FPMUL
111.083 303 372 110  9607X          JMP     FTA4            HAVE NOT PRINTED DECIMAL YET
                     9608X
                     9609X *        ADD EXPONENT, IF NECESSARY
                     9610X
111.086 361          9611X FTA8.5   POP     PSW
111.087 332 135 111  9612X          JC      FTA12           NOT SCIENTIFIC
                     9613X
                     9614X *        ADD E+-NN
                     9615X
111.092 066 105      9616X          MVI     M,'E'
111.094 043          9617X          INX     H
111.095 066 053      9618X          MVI     M,'+'
111.097 075          9619X          DCR     A
111.100 362 107 111  9620X          JP      FTA9
111.103 066 055      9621X          MVI     M,'-'
111.105 057          9622X          CMA
111.106 074          9623X          INR     A
111.107 043          9624X FTA9     INX     H
111.110 066 057      9625X          MVI     M,'0'-1
111.112 064          9626X FTA10    INR     M               DECODE 10S DIGIT
111.113 326 012      9627X          SUI     10
111.115 362 112 111  9628X          JP      FTA10
111.120 306 012      9629X          ADI     10
111.122 043          9630X          INX     H
111.123 066 057      9631X          MVI     M,'0'-1
111.125 064          9632X FTA11    INR     M
111.126 075          9633X          DCR     A
111.127 362 125 111  9634X          JP      FTA11
111.132 303 152 111  9635X          JMP     FTA13           DONT TRIM TRAILING THINGS
                     9636X
                     9637X *        DONE STRIP TRAILING ZEROS.
                     9638X
111.135 053          9639X FTA12    DCX     H
111.136 176          9640X          MOV     A,M
111.137 376 060      9641X          CPI     '0'
111.141 312 135 111  9642X          JE      FTA12
111.144 376 056      9643X          CPI     '.'
111.146 302 152 111  9644X          JNE     FTA13           NOT .
111.151 053          9645X          DCX     H
111.152 043          9646X FTA13    INX     H
111.153 066 040      9647X          MVI     M,' '           ADD TRAILING BLANK
111.155 043          9648X          INX     H
                     9649X
111.156 321          9650X          POP     D               (DE) = NUMBER FWA
111.157 175          9651X          MOV     A,L
111.160 223          9652X          SUB     E
111.161 353          9653X          XCHG
111.162 033          9654X          DCX     D
111.163 301          9655X          POP     B
111.164 311          9656X          RET
                     9657X
111.165 051 000 000  9658X FTAA     DB      51Q,0,0,2000Q   5.E-7
```

```
                                 9660X **      DDN - DECODE DECIMAL NUMBER.
                                 9661X *
                                 9662X *       ENTRY   (HL) = TEXT POINTER
                                 9663X *       EXIT    (DE) = VALUE (IF NON-NULL)
                                 9664X *               (HL) UPDATED
                                 9665X *               TO 'DDNERR' IF NULL
                                 9666X *       USES    ALL
                                 9667X
                                 9668X
111,171                          9669X DDN     EQU     *
111,171  315 320 111             9670X         CALL    $CVD            CHECK DECIMAL VALUE
111,174  332 122 070             9671X         JC      DDNERR          HAVE NO DECIMAL DIGITS
111,177  021 000 000             9672X         LXI     D,0             (DE) = ACCUMULATOR
111,202  315 320 111             9673X DDN1    CALL    $CVD            CHECK DECIMAL VALUE
111,205  330                     9674X         RC                      NO MORE DIGITS
111,206  345                     9675X         PUSH    H               SAVE TEXT POINTER
111,207  353                     9676X         XCHG            (HL) =MULTIPLIER
111,210  051                     9677X         DAD     H               (HL) = X*2
111,211  124                     9678X         MOV     D,H
111,212  135                     9679X         MOV     E,L
111,213  051                     9680X         DAD     H               (HL) = X*4
111,214  051                     9681X         DAD     H               (HL) = X*8
111,215  031                     9682X         DAD     D               (HL) = X*10
111,216  332 122 070             9683X         JC      DDNERR          OVERFLOW
111,221  137                     9684X         MOV     E,A
111,222  026 000                 9685X         MVI     D,0             (DE) = DIGIT VALUE
111,224  031                     9686X         DAD     D
111,225  332 122 070             9687X         JC      DDNERR          NO GOOD
111,230  353                     9688X         XCHG                    (DE) = VALUE
111,231  341                     9689X         POP     H
111,232  005                     9690X         DCR     B               COUNT DF
111,233  043                     9691X DDN2    INX     H
111,234  303 202 111             9692X         JMP     DDN1            ACCEPT ANOTHER


                                 9694X **      DFD - DECODE FLOATING DECIMAL.
                                 9695X *
                                 9696X *       DFD PERFORMS THE EQUIVALENT TO DDN, BUT DOES IT IN
                                 9697X *       THE FLOATING POINT ACCUMULATOR.
                                 9698X *
                                 9699X
                                 9700X
111,237  315 320 111             9701X DFD     CALL    $CVD            CHECK VALID DEC
111,242  330                     9702X         RC                      NO GOOD
111,243  062 212 042             9703X         STA     ACCY+2
111,246  345                     9704X         PUSH    H
111,247  305                     9705X         PUSH    B               SAVE (B)
111,250  041 153 112             9706X         LXI     H,FP10.
111,253  315 327 105             9707X         CALL    MUL             SCALE ACCUM
111,256  041 210 042             9708X         LXI     H,ACCY
111,261  315 356 104             9709X         CALL    ADD             ADD VALUE
111,264  301                     9710X         POP     B
111,265  341                     9711X         POP     H
111,266  005                     9712X         DCR     B               COUNT DIGIT
```

```
111.267  043            9713X DFD1   INX    H
111.270  303 237 111    9714X        JMP    DFD             ANOTHER DIGIT
                        9715         LOF    C
111.273                 9716         XTEXT  WER


                        9718X **     $WER - WRITE ENABLE RAM.
                        9719X *
                        9720X *      $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                        9721X *      RAM AREA.
                        9722X *
                        9723X *      ENTRY  NONE
                        9724X *      EXIT   NONE
                        9725X *      USES   NONE
                        9726X
                        9727X
031.241                 9728X $WER   EQU    31241A          IN H17 ROM


                        9730X **     $WDR - WRITE DISABLE RAM.
                        9731X *
                        9732X *      $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                        9733X *      RAM AREA.
                        9734X *
                        9735X *      ENTRY  NONE
                        9736X *      EXIT   NONE
                        9737X *      USES   NONE
                        9738X
                        9739X
031.222                 9740X $WDR   EQU    31222A          IN H17 ROM
070.122                 9741  DDNERR EQU    ERR.IN
111.273                 9742         XTEXT  CLL


                        9744X **     CLL - COMPUTE LINE LENGTH.
                        9745X *
                        9746X *      CLL COUNTS THE NUMBER OF CHARACTERS IN A SOURCE LINE.
                        9747X *      THE LINE IS TERMINATED BY A 00 BYTE; THE 00 BYTE IS ENCLUDED
                        9748X *      IN THE COUNT.
                        9749X *
                        9750X *      ENTRY  (HL) = FWA OF LINE
                        9751X *      EXIT   (HL) UNCHANGED
                        9752X *             (A) = LENGTH OF LINE
                        9753X *      USES   A,F
                        9754X
                        9755X
111.273  345            9756X $CLL   PUSH   H               SAVE STARTING ADDRESS
111.274  325            9757X        PUSH   D
111.275  026 000        9758X        MVI    D,0
                        9759X
```

```
111.277  176          9760X CLL1   MOV    A,M
111.300  024          9761X        INR    D
111.301  247          9762X        ANA    A
111.302  043          9763X        INX    H
111.303  302 277 111   9764X        JNZ    CLL1          SCAN FOR END
111.306  172          9765X        MOV    A,D
111.307  321          9766X        POP    D
111.310  341          9767X        POP    H
111.311  311          9768X        RET
111.312               9769         XTEXT  CRLF


                      9771X **     $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                      9772X *
                      9773X *      $CRLF IS USED TO GENERATE PADDED CRLF'S.
                      9774X *
                      9775X *      ENTRY  NONE
                      9776X *      EXIT   (A) = 0
                      9777X *      USES   A,F
                      9778X
                      9779X
111.312  076 012      9780X $CRLF  MVI    A,NL
111.314  377 002      9781X        DB     SYSCALL,.SCOUT
111.316  257          9782X        XRA    A
111.317  311          9783X        RET
111.320               9784         XTEXT  CVD


                      9786X **     $CVD - CHECK FOR VALID DIGIT.
                      9787X *
                      9788X *      CVD EXAMINES A DIGIT TO SEE IF IT IS A VALID DECIMAL DIGIT.
                      9789X *
                      9790X *      ENTRY  (HL) = ADDRESS OF CHARACTER
                      9791X *      EXIT   'C' SET IF ILLEGAL
                      9792X *               (A) = VALUE
                      9793X *      USES   A,F
                      9794X
                      9795X
111.320  176          9796X $CVD   MOV    A,M           (A) = CHARACTER
111.321  326 060      9797X $CVD.  SUI    '0'
111.323  330          9798X        RC                   ILLEGAL
111.324  376 012      9799X        CPI    9+1
111.326  077          9800X        CMC
111.327  311          9801X        RET
111.330               9802         XTEXT  ZERO
```

```
                        9804X **      $ZERO - ZERO MEMORY
                        9805X *
                        9806X *       $ZERO ZEROS A BLOCK OF MEMORY.
                        9807X *
                        9808X *       ENTRY   (HL) = ADDRESS
                        9809X *               (B) = COUNT
                        9810X *       EXIT    (A) = 0
                        9811X *       USES    A,B,F,H,L
                        9812X
                        9813X
031.212                 9814X $ZERO   EQU     31212A          IN H17 ROM
111.330                 9815          XTEXT   SOB



                        9817X **      $SOB - SKIP OVER BLANKS.
                        9818X *
                        9819X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                        9820X *
                        9821X *       ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                        9822X *       EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                        9823X *               (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                        9824X *       USES    A,F,H,L
                        9825X
                        9826X
111.330  053           9827X $SOB     DCX     H               PRE-DECREMENT
111.331  043           9828X $SOB1    INX     H
111.332  176           9829X          MOV     A,M
111.333  376 040       9830X          CPI     ' '
111.335  312 331 111   9831X          JE      $SOB1           GOT BLANK
111.340  376 011       9832X          CPI     TAB
111.342  312 331 111   9833X          JE      $SOB1           GOT TAB
111.345  311           9834X          RET
111.346                9835          XTEXT   CDEHL



                        9837X **      $CDEHL - COMPARE (DE) TO (HL)
                        9838X *
                        9839X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                        9840X *
                        9841X *       ENTRY   NONE
                        9842X *       EXIT    'Z' SET IF (DE) = (HL)
                        9843X *       USES    A,F
                        9844X
                        9845X
030.216                 9846X $CDEHL   EQU     30216A          IN H17 ROM
111.346                 9847          XTEXT   HLCPDE
                        9848X **      HLCPDE  - (HL) COMPARED TO (DE)
                        9849X *
                        9850X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                        9851X *
                        9852X *       ENTRY:  (HL)&(DE) SET UP
                        9853X *
```

```
                        9854X *       EXIT:   (PSW)   =
                        9855X *                       'Z' SET   IF (HL) =  (DE)
                        9856X *                       'C' SET   IF (HL) <  (DE)
                        9857X *                       'C' CLEAR IF (HL) >= (DE)
                        9858X *
                        9859X *
                        9860X *       USES:   (PSW)
                        9861X *
                        9862X
    111.346  174        9863X HLCPDE  MOV     A,H
    111.347  272        9864X         CMP     D       'C' SET => (A) < (D)
    111.350  300        9865X         RNZ
    111.351  175        9866X         MOV     A,L
    111.352  273        9867X         CMP     E       'C' SET => (L) < (E)
    111.353  311        9868X         RET
    111.354             9869          XTEXT   DU66


                        9871X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        9872X *
                        9873X *       (HL) = (BC)/(DE)
                        9874X *
                        9875X *       ENTRY   (BC), (DE) PRESET
                        9876X *       EXIT    (HL) = RESULT
                        9877X *               (DE) = REMAINDER
                        9878X *       USES    ALL
                        9879X
                        9880X
    030.106             9881X $DU66   EQU     30106A          IN H17 ROM
    111.354             9882          XTEXT   MU86


                        9884X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                        9885X *
                        9886X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                        9887X *       BIT VALUE.
                        9888X *
                        9889X *       ENTRY   (A) = MULTIPLIER
                        9890X *               (DE) = MULTIPLICAND
                        9891X *       EXIT    (HL) = RESULT
                        9892X *               'Z' SET IF NOT OVERFLOW
                        9893X *       USES    A,F,H,L
                        9894X
                        9895X
    031.007             9896X $MU86   EQU     31007A          IN H17 ROM
    111.354             9897          XTEXT   ZEROS
```

```
                        9899X **      8 CONSTANT ZERO BYTES.
                        9900X
031.320                 9901X $ZEROS  EQU     31320A          IN H17 ROM
111.354                 9902          XTEXT   UDD


                        9904X **      $UDD - UNPACK DECIMAL DIGITS.
                        9905X *
                        9906X *       UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                        9907X *       DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                        9908X *
                        9909X *       ENTRY   (B,C) = ADDRESS VALUE
                        9910X *               (A) = DIGIT COUNT
                        9911X *               (H,L) = MEMORY ADDRESS
                        9912X *       EXIT    (HL) = (HL) + (A)
                        9913X *       USES    ALL
                        9914X
                        9915X
031.157                 9916X $UDD    EQU     31157A          IN H17 ROM
111.354                 9917          XTEXT   CCO


                        9919X **      $CCO - CLEAR CONTROL-O
                        9920X *
                        9921X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                        9922X *
                        9923X *       ENTRY   NONE
                        9924X *       EXIT    NONE
                        9925X *       USES    NONE
                        9926X
                        9927X
111.354 315 054 031     9928X $CCO    CALL    $SAVALL         SAVE REGISTERS
111.357 076 004         9929X         MVI     A,I.CONFL
111.361 001 001 000     9930X         LXI     B,CO.FLG        CLEAR CO.FLG
111.364 377 006         9931X         DB      SYSCALL+.CONSL
111.366 303 047 031     9932X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
111.371                 9933          XTEXT   DADA


                        9935X **      $DADA - PERFORM (H,L) = (H,L) + (Q,A)
                        9936X *
                        9937X *       ENTRY   (H,L) = BEFORE VALUE
                        9938X *               (A) = BEFORE VALUE
                        9939X *       EXIT    (H,L) = (H,L) + (Q,A)
                        9940X *               'C' SET IF OVERFLOW
                        9941X *       USES    F,H,L
                        9942X
                        9943X
030.072                 9944X $DADA   EQU     30072A          IN H17 ROM
111.371                 9945          XTEXT   DADA2
```

```
                        9947X **      $DADA. - ADD (0,A) TO (H,L)
                        9948X *
                        9949X *       ENTRY   NONE
                        9950X *       EXIT    (HL) = (HL) + (0A)
                        9951X *       USES    A,F,H,L
                        9952X
                        9953X
030.101                 9954X $DADA.  EQU     30101A          IN H17 ROM
111.371                 9955          XTEXT   MOVE



                        9957X **      $MOVE - MOVE DATA
                        9958X *
                        9959X *       $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        9960X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        9961X *       FIRST TO LAST.
                        9962X *
                        9963X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        9964X *       LAST TO FIRST.
                        9965X *
                        9966X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        9967X *
                        9968X *       ENTRY   (BC) = COUNT
                        9969X *               (DE) = FROM
                        9970X *               (HL) = TO
                        9971X *       EXIT    MOVED
                        9972X *               (DE) = ADDRESS OF NEXT FROM BYTE
                        9973X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                        9974X *               'C' CLEAR
                        9975X *       USES    ALL
                        9976X
                        9977X
030.252                 9978X $MOVE   EQU     30252A          IN H17 ROM
111.371                 9979          XTEXT   MU66



                        9981X **      $MU66 - UNSIGNED 16X16 MULTIPLY.
                        9982X *
                        9983X *       ENTRY (BC) = MULTIPLICAND
                        9984X *               (DE) = MULTIPLIER
                        9985X *       EXIT    (HL) = RESULT
                        9986X *               'Z' SET IF NOT OVERFLOW
                        9987X *       USES    ALL
                        9988X
                        9989X
030.337                 9990X $MU66   EQU     30337A          IN H17 ROM
111.371                 9991          XTEXT   TBLS
```

```
                                9993X **      $TBLS - TABLE SEARCH
                                9994X *
                                9995X *       TABLE FORMAT
                                9996X *
                                9997X *       DB      KEY1,VAL1,
                                9998X *       .       .
                                9999X *       .       .
                               10000X *       DB      KEYN,VALN
                               10001X *       DB      0
                               10002X *
                               10003X *       ENTRY   (A) = PATTERN
                               10004X *               (H,L) = TABLE FWA
                               10005X *       EXIT    (A) = PATTERN IF FOUND
                               10006X *               'Z' SET IF FOUND
                               10007X *               'Z' CLEAR IF NOT FOUND OR PATTERN=0          /78.10.GC/
                               10008X *       USES    A,F,H,L
                               10009X
                               10010X
111,371  305                   10011X $TBLS   PUSH    B
111,372  376 000               10012X         CPI     0                                            /78.10.GC/
111,374  312 016 112           10013X         JZ      TBL2                                         /78.10.GC/
111,377  107                   10014X         MOV     B,A
112,000  176                   10015X TBL1    MOV     A,M             (A) = CHARACTER
112,001  043                   10016X         INX     H
112,002  270                   10017X         CMP     B
112,003  312 020 112           10018X         JZ      TBL3            IF MATCH
112,006  247                   10019X         ANA     A
112,007  043                   10020X         INX     H               SKIP PAST
112,010  302 000 112           10021X         JNZ     TBL1            IF NOT END OF TABLE
112,013  053                   10022X         DCX     H
112,014  053                   10023X         DCX     H
112,015  257                   10024X         XRA     A               SET TO ZERO FOR OLD USERS    /78.10.GC/
112,016  376 001               10025X TBL2    CPI     1               CLEAR ZERO                   /78.10.GC/
                               10026X
                               10027X *       DONE
                               10028X
112,020  301                   10029X TBL3    POP     B
112,021  311                   10030X         RET
112,022                        10031          XTEXT   TBRA

                               10033X **      $TBRA - BRANCH RELATIVE THOUGH TABLE.
                               10034X *
                               10035X *       $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                               10036X *       JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                               10037X *       ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                               10038X *
                               10039X *       CALL    $TBRA
                               10040X *       DB      LAB1-*          INDEX = 0 FOR LAB1
                               10041X *       DB      LAB2-*          INDEX = 1 FOR LAB2
                               10042X *       DB      LABN-*          INDEX = N-1 FOR LABN
                               10043X *
                               10044X *       ENTRY   (A) = INDEX
                               10045X *               (RET) = TABLE FWA
```

```
                        10046X *        EXIT    TO COMPUTED ADDRESS
                        10047X *        USES    F,H,L
                        10048X
                        10049X
    031.076             10050X $TBRA    EQU     31076A          IN H17 ROM
    112.022             10051           XTEXT   TJMP


                        10053X **       $TJMP - TABLE JUMP.
                        10054X *
                        10055X *        USAGE
                        10056X *
                        10057X *        CALL      $TJMP             (A) = INDEX
                        10058X *        DW        ADDR1
                        10059X *        .           .
                        10060X *        .           .
                        10061X *        .           .
                        10062X *        DW        ADDRN
                        10063X *
                        10064X *        ENTRY     (A) = INDEX
                        10065X *        EXIT      TO PROCESSOR
                        10066X *                  (A) = INDEX*2
                        10067X *        USES      NONE.
                        10068X
                        10069X
    031,061             10070X $TJMP    EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        10071X
    031,062             10072X $TJMP.   EQU     31062A          IN H17 ROM
    112.022             10073           XTEXT   TYPCH


                        10075X **       $TYPCH - TYPE SINGLE CHARACTER.
                        10076X *
                        10077X *        ENTRY   (RET) = CHARACTER
                        10078X *        EXIT    TO (RET)+1
                        10079X *                (A) = CHARACTER TYPED
                        10080X
                        10081X
    112.022  343        10082X $TYPCH   XTHL                     (HL) = RETURN ADDRESS
    112,023  176        10083X          MOV     A,M             (A) = CHARACTER
    112.024  043        10084X          INX     H
    112,025  343        10085X          XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        10086X
                        10087X **       $TYPC. - TYPE SINGLE CHARACTER.
                        10088X *
                        10089X *        ENTRY   (A) = CHARACTER
                        10090X *        EXIT    TO (RET)
                        10091X
    112.026  377 002    10092X $TYPC.   DB      SYSCALL,.SCOUT
    112,030  311        10093X          RET
    112.031             10094           XTEXT   TYPTX
```

```
                        10096X **       $TYPTX - TYPE TEXT.
                        10097X *
                        10098X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        10099X *
                        10100X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        10101X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        10102X *
                        10103X *       ENTRY   (RET) = TEXT
                        10104X *       EXIT    TO (RET+LENGTH)
                        10105X *       USES    A,F
                        10106X
                        10107X
    031.136             10108X $TYPTX  EQU     31136A          IN H17 ROM
                        10109X
    031.144             10110X $TYPTX. EQU     31144A          IN H17 ROM
    112.031             10111          XTEXT   GNL


                        10113X **       $GNL - GUARANTEE NEW LINE.
                        10114X *
                        10115X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                        10116X *       IF THE CURSOR IS NOT AT COLUMN 1.
                        10117X *
                        10118X *       ENTRY   NONE
                        10119X *       EXIT    NONE
                        10120X *       USES    ALL
                        10121X
                        10122X
    112.031  076 002    10123X $GNL    MVI     A,I.CUSOR
    112.033  001 000 000 10124X        LXI     B,0
    112.036  377 006    10125X        DB      SYSCALL+.CONSL          READ CURSOR
    112.040  075        10126X        DCR     A
    112.041  310        10127X        RZ                      AT COLUMN 1
    112.042  303 312 111 10128X       JMP     $CRLF           NEW LINE
    112.045             10129          XTEXT   CHL


                        10131X **       $CHL - COMPLEMENT (HL).
                        10132X *
                        10133X *       (HL) = -(HL)            TWO'S COMPLEMENT
                        10134X *
                        10135X *       ENTRY   NONE
                        10136X *       EXIT    NONE
                        10137X *       USES    A,F,H,L
                        10138X
                        10139X
    030.224             10140X $CHL    EQU     30224A          IN H17 ROM
    112.045             10141          XTEXT   COMP
```

```
                        10143X **       $COMP - COMPARE TWO CHARACTER STRINGS.
                        10144X *
                        10145X *       $COMP COMPARES TWO BYTE STRINGS.
                        10146X *
                        10147X *       ENTRY   (C) = COMPARE COUNT
                        10148X *               (DE) = FWA OF STRING #1
                        10149X *               (HL) = FWA OF STRING #2
                        10150X *       EXIT    'Z' CLEAR, IS MIS-MATCH
                        10151X *                (C) = LENGTH REMAINING
                        10152X *               (DE) = ADDRESS OF MISMATCH IN STRING#1
                        10153X *               (HL) = ADDRESS OF MISMATCH IN STRING #2
                        10154X *               'C' SET, HAVE MATCH
                        10155X *                (C) = 0
                        10156X *               (DE) = (DE) + (OC)
                        10157X *               (HL) = (HL) + (OC)
                        10158X *       USES    A,F,C,D,E,H,L
                        10159X
                        10160X
    030.060             10161X $COMP   EQU     30060A          IN H17 ROM
    112.045             10162          XTEXT   MCU


                        10164X **       MCU - MAP LOWER CASE TO UPPER CASE.
                        10165X *
                        10166X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                        10167X *       CASE.
                        10168X *
                        10169X *       ENTRY   (A) = CHARACTER
                        10170X *       EXIT    (A) = CHARACTER RESULT
                        10171X *       USES    A,F
                        10172X
                        10173X
    112.045  376 141   10174X $MCU    CPI     'a'
    112.047  330       10175X         RC                      NOT LOWER CASE
    112.050  376 173   10176X         CPI     'z'+1
    112.052  320       10177X         RNC                     NOT LOWER CASE
    112.053  326 040   10178X         SUI     'a'-'A'
    112.055  311       10179X         RET
    112.056             10180          XTEXT   MU10


                        10182X **       $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                        10183X *
                        10184X *       (HL) = (DE)*10
                        10185X *
                        10186X *       ENTRY   (DE) = MULTIPLIER
                        10187X *       EXIT    'C' CLEAR IF OK
                        10188X *                (HL) = PRODUCT
                        10189X *                'C' SET IF ERROR
                        10190X *       USES    D,E,H,L,F
                        10191X
                        10192X
```

```
030.324             10193X $MU10   EQU     30324A          IN H17 ROM
112.056             10194          XTEXT   SAVALL


                    10196X **      $RSTALL - RESTORE ALL REGISTERS.
                    10197X *
                    10198X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                    10199X *       RETURNS TO THE PREVIOUS CALLER.
                    10200X *
                    10201X *       ENTRY   (SP) = PSW
                    10202X *               (SP+2) = BC
                    10203X *               (SP+4) = DE
                    10204X *               (SP+6) = HL
                    10205X *               (SP+8) = RET
                    10206X *       EXIT    TO *RET*, REGISTERS RESTORED
                    10207X *       USES    ALL
                    10208X
                    10209X
031.047             10210X $RSTALL EQU     31047A          IN H17 ROM


                    10212X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                    10213X *
                    10214X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                    10215X *
                    10216X *       ENTRY   NONE
                    10217X *       EXIT    (SP) = PSW
                    10218X *               (SP+2) = BC
                    10219X *               (SP+4) = DE
                    10220X *               (SP+6) = HL
                    10221X *       USES    H,L
                    10222X
                    10223X
031.054             10224X $SAVALL EQU     31054A          IN H17 ROM
112.056             10225          XTEXT   INDL


                    10227X **      $INDL - INDEXED LOAD.
                    10228X *
                    10229X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                    10230X *
                    10231X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                    10232X *
                    10233X *       (DE) = ( (HL) + DSPLACEMENT )
                    10234X *
                    10235X *       ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                    10236X *               (HL) = TABLE ADDRESS
                    10237X *       EXIT    TO (RET+2)
                    10238X *       USES    A,F,D,E
                    10239X
                    10240X
030.234             10241X $INDL   EQU     30234A          IN H17 ROM
```

```
    112.056             10242          XTEXT   MTDOC

                        10244X ***     MANAGED TABLES.
                        10245X *
                        10246X *       THE FOLLOWING STRUCTURES ARE MANAGED TABLES.
                        10247X *
                        10248X *       SEVERAL TABLES OF DATA ARE 'MANAGED' BY A SUBROUTINE
                        10249X *       PACKAGE SO THAT THEIR SIZES MAY VARY INDEFINITELY.
                        10250X *
                        10251X *       EACH TABLE HAS A CERTAIN AMOUNT OF FREE SPACE LOCATED AFTER
                        10252X *       IT. WHEN A TABLE NEEDS TO BE ENLARGED, $ATS (ALLOCATE
                        10253X *       TABLE SPACE) PERFORMS THE ALLOCATION. IF SUFFICIENT FREE S PCE
                        10254X *       FOLLOWS THE TABLE, IT IS SIMPLY ALLOCATED.
                        10255X *
                        10256X *       IF THE FREE SPACE FOLLOWING THE TABLE IS INSUFFICIENT, ALL
                        10257X *       TABLES ARE MOVED, REDUCING THE FREE SPACE BEHIND EACH ONE, IN
                        10258X *       ORDER TO CONCENTRATE SUFFICIENT FREE SPACE BEHIND THE ONE
                        10259X *       NEEDING IT. THUS, WHEN TABLE OVERFLOW OCCURS, ALL TABLES HAVE
                        10260X *       OVERFLOWED, IN THAT THERE IS NO MORE FREE SPACE AVAILABLE
                        10261X *       BEHIND ANY OF THEM.
                        10262X *
                        10263X *       STORAGE USED:
                        10264X *
                        10265X *       THE MANAGED TABLE PACKAGE USES MEMORY STARTING AT SYMBOL 'MTAREA
                        10266X *       EXTENDING TO THE VALUE IN (MEML). (MEML) MAY BE INCREASED DURING
                        10267X *       EXECUTION, BUT IT SHOULD NOT BE DECREASED.
                        10268X *
                        10269X *       FREE SPACE ALLOCATION:
                        10270X *
                        10271X *       WHEN THE TABLES MUST BE MOVED, $ATS DIVIDES UP THE MEMORY FREE
                        10272X *       SPACE AMONG THE TABLES. HOWEVER, THIS SPLITTING IS NOT NECESSARI
                        10273X *       EVEN. EACH TABLE CONTAINS A ONE BYTE ALLOCATION FACTOR,
                        10274X *       INDICATING HOW MANY 1/16THS SHARES IT WILL RECEIVE. THESE
                        10275X *       NUMBERS MUST ALL ADD UP TO 16 (THUS, THE NEX NUMBER OF TABLES
                        10276X *       IS 16, SINCE NO ALLOCATION FACTOR MAY BE 0).
                        10277X *
                        10278X *       TABLE USAGE:
                        10279X *
                        10280X *       NO TABLE ITEM (EXCEPT ITEMS IN THE 1ST TABLE) MAY BE REFERENCED
                        10281X *       BY ADDRESS, SINCE THE ADDRESS MAY BE CHANGED (VIA TABLE MOVES)
                        10282X *       AT ANY TIME. INSTEAD, THE ITEMS SHOULD BE REFERENCED BY
                        10283X *       TABLE INDEX, THAT IS, THEIR SEQUENTIAL POSITION WITHIN THE
                        10284X *       TABLE.
```

```
                        10286X **       TABLE INDEX.
                        10287X *
                        10288X *       THE FOLLOWING INDEXES ARE USED TO KEEP TRACK OF TABLES. ALL
                        10289X *       TABLE INDIXES MUST APPEAR CONTIGUOUSLY IN MEMORY.
                        10290X *
                        10291X *
                        10292X * MTABIND EQU *
                        10293X *       [INDEX FOR TABLE 1]
                        10294X *         .
                        10295X *         .
                        10296X *       [INDEX FOR TABLE N]
                        10297X *       DB      0               DUMY ALLOCATION BYTE
                        10298X * MEML  DW      0               MEMORY LIMIT
                        10299X *MTABL  EQU     *-MTABIND/5     NUMBER OF TABLES.
                        10300X *
                        10301X *       INDEX FORMAT:
                        10302X *
                        10303X *       DB      FACT            ALLOCATION FACTOR (NUMBER OF 1/16THS)
                        10304X *       DW      FWA             TABLE FWA
                        10305X *       DW      LEN             TABLE LENGTH
                        10306X
    000.000             10307X MT.AFC  EQU     0               ALLOCATION FACTOR
    000.001             10308X MT.FWA  EQU     1               FWA INDEX
    000.003             10309X MT.LEN  EQU     3               LENGTH FIELD
    112.056             10310  MTABIND EQU     *               FWA OF 1ST TABLE HEADER
```

```
                        10314  **      TXTTAB - USER SOURCE TEXT TABLE.
                        10315  *
                        10316  *       FORMAT:
                        10317  *
                        10318  *       DW      LINE            LINE NUMBER
                        10319  *       DB      'TEXT'          LINE TEXT
                        10320  *       DB      0               END OF LINE
                        10321  *
                        10322  *       LINE NUMBER 65535 (377377A) IS ALWAYS PRESENT IN THE TABLE,
                        10323  *       BUT MAY NOT BE ALTERED OR DISPLAYED.
                        10324
     112.056  001      10325  TXTTAB  DB      1               ALLOCATION COUNT
     112.057  346 114  10326          DW      MTAREA          FWA
     112.061  003 000  10327          DW      3               LENGTH



                        10329  **      SYMTAB - SYMBOL TABLE.
                        10330  *
                        10331  *       SYMTAB CONTAINS THE USER SYMBOL TABLE.
                        10332  *       AN ENTRY IS PRESENT FOR EACH
                        10333  *
                        10334  *       1) SCALAR NUMERIC VARIABLE
                        10335  *       2) SCALAR NUMERIC FUNCTION
                        10336  *       3) SCALAR STRING VARIABLE
                        10337  *       4) SCALAR STRING FUNCTION
                        10338  *       5) NUMERIC VECTOR
                        10339  *       6) STRING VECTOR
                        10340  *
                        10341  *       ENTRY   FORMAT:
                        10342  *
                        10343  *       THE ENTRY FORMAT DEPENDS UPON THE SYMBOL TYPE.
                        10344  *       ALL SCALAR ENTRYS ARE 6 BYTES LONG WITH VECTORS BEING LONGER.(SEE BELOW)
                        10345  *       THE FIRST TWO BYTES OF ALL ENTRYS ARE ALWAYS FORMATTED:
                        10346  *
                        10347  *       DB      'C'             1ST CHARACTER OF VARIABLE NAME
                        10348  *       DB      N+F             N * 2ND CHARACTER INDEX
                        10349  *                                (0=NONE, 0001B='0',...,1010B='9')
                        10350  *                               F=00000000 SCALAR NUMERIC VARIABLE
                        10351  *                                =00000001 SCALAR STRING VARIABLE
                        10352  *                                =00000010 NUMERIC VECTOR
                        10353  *                                =00000011 STRING VECTOR
                        10354  *
                        10355  *       THE REMAINING  BYTES ARE FORMATTED:
                        10356  *
                        10357  *       1) SCALAR NUMERIC VARIABLE:
                        10358  *
                        10359  *       DW      V1              4 BYTE FLOATING POINT VALUE
                        10360  *       DW      V2
                        10361  *
                        10362  *       2) SCALAR NUMERIC FUNCTION
                        10363  *
                        10364  *       DB      201*    DB      2010            FUNCTION FLAG
                        10365  *       DW      ADDR            TEXT ADDRESS OF FUNCTION LINE
                        10366  *       DB      0               UNUSED
```

```
                        10367  *
                        10368  *       3) SCALAR STRING VARIABLE
                        10369  *
                        10370  *       DB      LEN,0           LENGTH
                        10371  *       DW      STRNAM          STRING NAME ((LABEL) SEE SYMTAB)
                        10372  *
                        10373  *       4) SCALAR STRING FUNCTION
                        10374  *
                        10375  *       DB      201Q            FUNCTION FLAG
                        10376  *       DW      ADDR            TEXT ADDRESS OF FUNCTION LINE
                        10377  *       DB      0               UNUSED
                        10378  *
                        10379  *       5) NUMERIC VECTOR
                        10380  *
                        10381  *       DB      DIM,0           NUMBER OF DIMENSIONS
                        10382  *       DW      SIZE            SIZE OF ARRAY(# OF BYTES FROM L1 TO NEXT ENTRY)
                        10383  * L1    DW      DIM 1           DIMENSION 1
                        10384  *       .       .
                        10385  *       .       .
                        10386  *       .       .
                        10387  *       DW      DIM N           DIMENSION N
                        10388  *       DW      V1              4 BYTE FLOATING POINT VALUE
                        10389  *       DW      V2
                        10390  *       .       .
                        10391  *       .       .
                        10392  *       .       .
                        10393  *       DW      V M-1
                        10394  *       DW      V M
                        10395  *
                        10396  *       6) STRING VECTOR
                        10397  *
                        10398  *       DB      DIM,0           NUMBER OF DIMENSIONS
                        10399  *       DW      SIZE            SIZE OF ARRAY
                        10400  *       DW      DIM 1           DIMENSION 1
                        10401  *       .       .
                        10402  *       .       .
                        10403  *       .       .
                        10404  *       DW      DIM N           DIMENSION N
                        10405  *       DW      LABEL 1         STRING LABEL
                        10406  *       DW      LEN 1           LENGTH OF STRING
                        10407  *       .       .
                        10408  *       .       .
                        10409  *       .       .
                        10410  *       DW      LABEL M
                        10411  *       DW      LEN M
                        10412
                        10413
    112.963  001        10414  SYMTAB  DB      1               ALLOCATION FACTOR
    112.064  351 114    10415          DW      NTAREA+3
    112.066  000 000    10416          DW      0
```

```
                        10418  **      FORTAB - FOR/NEXT LOOP TABLE.
                        10419  *
                        10420  *       FORTAB IS USED TO KEEP TRACK OF THE INDEX VARIABLE FOR
                        10421  *       'FOR/NEXT' LOOPS.
                        10422  *
                        10423  *       ENTRY   FORMAT:
                        10424  *
                        10425  *       DB      'C',N+F         SYMBOL TABLE KEY (SEE SYMTAB)   /80,01,GC/
                        10426  *       DW      INC,INC         INCREMENT VALUE
                        10427  *       DW      TRM,TRM         TERMINATEION VALUE
                        10428  *       DW      LOOPADR         ADDRESS FOR 'FOR' LOOP
                        10429
                        10430
112,070  001            10431  FORTAB  DB      1               ALLOCATION FACTOR
112,071  351 114        10432          DW      MTAREA+3
112,073  000 000        10433          DW      0               LENGTH


                        10435  **      GOSTAB - GOSUB TABLE.
                        10436  *
                        10437  *       GOSTAB CONTAINS THE RETURN ADDRESSES (AND LINE NUMBERS)
                        10438  *       FOR GOSUB CONSTRUCTS.
                        10439  *
                        10440  *       ENTRY   FORMAT:
                        10441  *
                        10442  *       DW      ADDR            RETURN TEXT ADDRESS
                        10443  *       DW      STATNO          RETURN LINE NUMBER
                        10444
                        10445
112,075  001            10446  GOSTAB  DB      1               ALLOCATION FACTOR
112,076  351 114        10447          DW      MTAREA+3
112,100  000 000        10448          DW      0


                        10450  **      WRKTAB - WORKING STORAGE TABLE.
                        10451  *
                        10452  *       WRKTAB IS USED BY THE EXPRESSION EVAULATOR TO STORE
                        10453  *       (ON A STACK) WORKING VALUES.
                        10454  *
                        10455  *       EACH ENTRY CONSISTS OF 5 BYTES, USUALLY A DESCRIPTOR BYTE
                        10456  *       AND 4 VALUE BYTES.
                        10457
                        10458
112,102  001            10459  WRKTAB  DB      1               ALLOCATION INDEX
112,103  351 114        10460          DW      MTAREA+3
112,105  000 000        10461          DW      0
```

```
                        10463  **      STRTAB - PERMANENT STRING TABLE.
                        10464  *
                        10465  *       STRTAB HOLDS PERMANENT STRING VARIABLES USED IN BASIC.
                        10466  *
                        10467  *       EACH STRING IS INDEXED BY AN ENTRY IN SYNTAB OR VECTAB.
                        10468  *
                        10469  *       ENTRY   FORMAT:
                        10470  *
                        10471  *       EACH STRING APPEARS CONTIGUOUSLY IN MEMORY, NO TRAILING
                        10472  *       CHARACTER IS USED SINCE LENGTHS ARE KNOWN IN THE POINTER.
                        10473  *       EXAMPLE:
                        10474  *
                        10475  *       DS      2               STRING LABEL(2NN NNN FOR PERM. STRING; 3NN NNN
                        10476  *                               FOR A TEMPORARY STRING)
                        10477  *       DS      N               ASCII STRING ( N=1 TO 256 )
                        10478  *       .       .
                        10479  *       .       .
                        10480  *       .       .
                        10481  *       DS      2               NTH LABEL
                        10482  *       DS      N
                        10483
                        10484
112.107  002           10485  STRTAB  DB      2               ALLOCATION INDEX
112.110  351 114       10486          DW      MTAREA+3
112.112  000 000       10487  STRLEN  DW      0


                        10489  **      TSTTAB - TEMPORARY STRING TABLE
                        10490  *
                        10491  *       TSTTAB HOLDS ALL TEMPORARY STRING VARIABLES USED IN BASIC.
                        10492  *
                        10493  *       THE FORMAT USED IS SIMILAR TO THAT OF STRTAB.
                        10494
                        10495
112.114  001           10496  TSTTAB  DB      1
112.115  351 114       10497          DW      MTAREA+3
112.117  000 000       10498          DW      0


                        10500  **      FILE TABLE.
                        10501  *
                        10502  *       CONTAINS BUFFER FOR EACH OPEN FILE.
                        10503
112.121  000           10504  FILTAB  DB      0               ALLOCATION INDEX
112.122  351 114       10505          DW      MTAREA+3        FWA
112.124  000 000       10506          DW      0               LWA
```

```
                                10508  **      DUMY LAST TABLE.
                                10509  *
                                10510  *       FORMATTED LIKE REGULAR TABLE, BUT CONTAINS
                                10511  *       MOVE COUNT, AND MEMORY LIMIT VALUES.
                                10512
000.010                         10513  MTABL   EQU     *-MTABIND/5     NUMBER OF TABLES
112.126  000                    10514          DB      0               STORAGE MOVES (IN ALLOCATING INDEX CELL)
112.127  351 114                10515  MEML    DW      MTAREA+3        MEMORY LIMIT ADDRESS (IN FWA CELL)
112.131                         10516          DS      2               TABLE LENGTH CELL NOT USED
                                10517
000.005                         10518  MTABLEN EQU     5               LENGTH OF EACH TABLE HEADER


                                10520  **      POINTERS TO CURRENT INFORMATION ABOUT RUN.
                                10521
112.133  000 000                10522  CURNUM  DW      0               CURRENT LINE NUMBER
112.135  000 000                10523  CURADR  DW      0               CURRENT LINE ADDRESS
112.137  000                    10524  LCKFLG  DB      0               DATA LOCK FLAG
                                10525
                                10526  **      CURRENT I.O CHANNEL.
                                10527  *
                                10528  *       =0      SYSTEM CONSOLE
                                10529  *       =1      INTERNAL FILE
                                10530  *       =1+N    BASIC CHANNEL N (N=1 TO X, IF N=0 THEN IOCHAN=0)
                                10531
112.140  000                    10532  IOCHAN  DB      0
                                10533
112.141  000                    10534  OVLMAN  DB      0               <>0 IF TO LOCK OVERLAY
                                10535
                                10536
112.142  000                    10537  CTLFLAG DB      0               CTL CHARACTERS FLAG BYTE
000.001                         10538  CFCTLC  EQU     0010            CTL-C HIT
000.002                         10539  CFCTLB  EQU     0020            CTL-B HIT


                                10541  **      STRING INDEXES.
                                10542  *
                                10543
112.143  200 000                10544  STRVJ   DW      000200A
112.145  300 000                10545  STRTI   DW      000300A


                                10547  **      FLOATING POINT VALUES.
                                10548  *
                                10549
112.147  000 000 100            10550  FP1.0   DB      0,0,100Q,201Q
112.153  000 000 120            10551  FP10.   DB      0,0,120Q,204Q
112.157  146 146 146            10552  FP0.1   DB      146Q,146Q,146Q,175Q
031.320                         10553  FP0.0   EQU     $ZEROS
112.163  022 170 233            10554  NPI.2   DB      022Q,170Q,233Q,201Q     -PI/2
112.167  022 170 233            10555  NPI2    DB      022Q,170Q,233Q,203Q     -PI*2
112.173  022 170 233            10556  NPI     DB      022Q,170Q,233Q,202Q     -PI
112.177  022 170 233            10557  NPI.4   DB      022Q,170Q,233Q,200Q     -PI/4
```

```
112.203  356 207 144 10558  PI.4    DB      356Q,207Q,144Q,200Q      PI/4
                     10559
112.207  040        10560  SPACE   DB      ' '                      SPACE CHARACTER
```

```
                        10563  **      PRS - PRESET BASIC,
                        10564  *
                        10565  *       PRS PERFORMS PRESET INITIALIZATION,
                        10566  *
                        10567
      112,210           10568  PRS     EQU     *
                        10569
                        10570  *       CHECK THE MDOS VERSION
                        10571
      112,210  377 011  10572          DB      SYSCALL,.VERS
      112,212  332 014 113 10573       JC      PRSERR1         NO SYSTEM CALL
      112,215  376 026  10574          CPI     VERS
      112,217  302 014 113 10575       JNZ     PRSERR1         NOT THE CORRECT VERSION
                        10576
                        10577  *       REQUEST MINIMAL MEMORY
                        10578
      112,222  041 346 114 10579       LXI     H,MTAREA
      112,225  377 052     10580       DB      SYSCALL,.SETTP
      112,227  332 016 113 10581       JC      PRSERR          NOT EVEN ENOUGH MEMORY TO START
                        10582
                        10583  *       SET UP THE INTERNAL WORK FILE BLOCK
                        10584
      112,232  052 120 041 10585       LHLD    S.SCR           HL = ADDRESS OF  *MDOS*  SCRATCH BUFFER
      112,235  042 232 042 10586       SHLD    FBSCR+2+0
      112,240  042 234 042 10587       SHLD    FBSCR+2+2
      112,243  042 236 042 10588       SHLD    FBSCR+2+4
      112,246  021 000 002 10589       LXI     D,512
      112,251  031         10590       DAD     D
      112,252  042 240 042 10591       SHLD    FBSCR+2+6
                        10592
                        10593  *       PROCEED WITH INITIALIZATION
                        10594
      112,255  315 357 073 10595       CALL    DTS             DELETE TEMP, STRINGS
      112,260  072 033 040 10596       LDA     .TICCNT         INITIALIZE RANDOM NUMBER SEED
      112,263  147         10597       MOV     H,A
      112,264  157         10598       MOV     L,A
      112,265  042 101 061 10599       SHLD    RNDA            INITIALIZE SEED
      112,270  021 016 000 10600       LXI     D,14                                    /80.01.GC/
      112,273  315 003 046 10601       CALL    CNTL4           SET TAB-FIELD WIDTH TO 14       /80.01.GC/
      112,276  041 370 100 10602       LXI     H,CBINT
      112,301  076 002     10603       MVI     A,CTLB
      112,303  377 041     10604       DB      SYSCALL,.CTLC   SETUP CTL-B HANDLER
      112,305  041 363 100 10605       LXI     H,CCINT
      112,310  076 003     10606       MVI     A,CTLC
      112,312  377 041     10607       DB      SYSCALL,.CTLC   SETUP CTL-C HANDLER
      112,314  315 136 031 10608       CALL    $TYPTX
      112,317  012 012 105 10609  PRSA DB      NL,NL,'Extended Renton Harbor BASIC #110.05.00',ENL
      112,371  315 115 074 10610       CALL    FDC             SET TABLES TO MAXIMUM AREA
      112,374  315 360 044 10611       CALL    SCR.            SCRATCH TEXT
                        10612
                        10613  ********************************************************************************
                        10614  ********************************************************************************
                        10615  **                                                                            **
                        10616  **      Note:   Be very careful about the following initializations           **
                        10617  **              Be sure that the instructions do not destroy thems-           **
                        10618  **              selves,                                                       **
```

```
                                10619  **
                                10620  **      Note:   If you don't understand the following error messages   **
                                10621  **              neither do I, Just love it and leave it.               **
                                10622  **                                                                     **
                                10623  *************************************************************************
                                10624  *************************************************************************
                                10625
 112.377  257                   10626          XRA     A
                                10627
 113.000  062 345 114           10628          STA     ZERO            CLEAR LINE-1
 114.345                        10629  .       SET     ZERO
 000.000                        10630          IF      *-1/.           *-1 < .
 001.331                        10631          ERRMI   .-PRSB           . < PRSB
                                10632          ENDIF
                                10633
 113.003  062 272 113           10634          STA     LINE+LINEL+6    INSURE 0 AT END OF LINE
 113.272                        10635  .       SET     LINE+LINEL+6
 000.000                        10636          IF      *-1/.           *-1 < .
 000.256                        10637          ERRMI   .-PRSB           . < PRSB
                                10638          ENDIF
                                10639
 113.006  062 300 114           10640          STA     LINE2+LINEL+6   INSURE 0 AT END OF LINE
 114.300                        10641  .       SET     LINE2+LINEL+6
 000.000                        10642          IF      *-1/.           *-1 < .
 001.264                        10643          ERRMI   .-PRSB           . < PRSB
                                10644          ENDIF
                                10645
 113.011  303 124 043           10646          JMP     RESTART         START PROGRAM
 113.014                        10647  PRSB    EQU     *
                                10648
 113.014  076 050               10649  PRSERR1 MVI     A,EC.NCV        NOT THE CORRECT VERSION OF  *HDOS*
                                10650
 113.016  046 012               10651  PRSERR  MVI     H,NL
 113.020  377 057               10652          DB      SYSCALL+.ERROR  OUTPUT THE ERROR
 113.022  257                   10653          XRA     A
 113.023  377 000               10654          DB      SYSCALL+.EXIT   QUIT BEFORE PROBLEMS ARISE
                                10655
 113.025                        10656  PRSLIM  EQU     *               LWA OF PRS CODE
                                10657
 113.025                        10658  LOADL   EQU     *               LOAD LWA
                                10659
                                10660
                                10661  **      OVERLAID BUFFER AREA
                                10662
 112.253                        10663          ORG     PRSLIM-106
                                10664
                                10665  **      COLUMN COUNTERS.
                                10666  *
                                10667  *       SINCE SEVERAL CHANNELS MAY BE PRINTED ON, INTERMINGLED,
                                10668  *       A SEPERATE COLUMN COUNTER IS KEPT FOR EACH.
                                10669  *       THIS TABLE IS INDEXED BY THE CONTENTS OF IOCHAN
                                10670
 112.253                        10671  COLCNTS DS      CHANMAX+1+2     ONE FOR EACH CHANNEL, +2 FOR TTY AND INTERNAL
                                10672
                                10673
 112.263                        10674          DS      2               USED BY ITL (WHEN CALLED BY BUILD)
```

```
                    10675  LINE    DS      0       LINE BUFFER                     /80.01.GC/
112.265             10676          DS      255                                     /80.01.GC/
112.265             10677  LINEL   EQU     *-LINE  LINE LENGTH
000.377             10678          ERRMI   *-PRSLIM FOLLOWING CELLS CHANGED BY PRS CODE
000.237             10679          DS      6       ROOM FOR EXPANDED LINE NUMBER   /78.10.GC/
113.264             10680          DS      1       ALWAYS 0 TO GUARANTEE END OF LINE
113.272             10681
                    10682  LINE2   DS      LINEL   WORK AREA
113.273             10683          DS      6       ROOM FOR EXPANDED LINE NUMBER   /78.10.GC/
114.272             10684          DS      1       ALWAYS ZERO TO GUARANTEE END OF LINE
114.300             10685
                    10686  FNRMA   EQU     LINE    FNRM WORK AREA
112.265             10687
                    10688  RUNMOD  DS      1
114.301             10689  STATE   DS      1
114.302             10690  DATPTR  DS      2
114.303


                    10692  **      PATCH AREA.
                    10693
                    10694  PATCH   DS      32
114.305


                    10696  **      BEGINNING OF MANAGED TABLE ADDRESS.
                    10697  *
                    10698
                    10699  ZERO    DS      1       DUMY END OF FIRST LINE -1
114.345             10700  MTAREA  EQU     *       BEGINNING OF MANAGED TABLES AREA
114.346             10701
                    10702          DS      100     AUX. PATCH AREA
114.346             10703
                    10704          END
115.112
ASSEMBLY COMPLETE
10704 STATEMENTS
    0 ERRORS DETECTED
18656 BYTES FREE
```

```
$ATP    071071  5327L   6229    8378
$ATS    103244  5276    5504    8213E   8489
$CCO    111354   956    1985    9928L
$CDEHL  030216  6196    7757    9846E
$CFS    072127  1838    5593L
$CHL    030224  1554    2484    2490    6183    10140E
$CLL    111273  5187    5204    7361    9756L
$COMP   030060  3733    10161E
$CRLF   111312  1845    9780L   10128
$CVD    111320  9670    9673    9701    9796L
$CVD.   111321  6132    6139    9797L
$DADA   030072  3667    6143    9944E
$DADA.  030101  3712    5351    6529    9954E
$DBT    104203  1350    1757    5897    8452E
$DU66   030106  2592    9881E
$FCLO   102335  1428    1884    2742    7091    7967L
$FCLO.  102344  7967    7971L
$FCLO1  103015  8010L   8013
$FCLO2  103025  8006    8015L
$FCLO3  103032  7979    7996    8026L
$FCLO4  103040  7977    8032L
$FERROR 070223  2727    2797    2814    5138E   7506    7510    7514    7614    7744    7791    7909    7969
$FFB    103122  7662    7759    8104L
$FOPE1  101101  7536    7548L
$FOPE2  101134  7572    7576L
$FOPEA  101142  7576    7580E
$FOPEB  101152  7570    7590L
$FOPER  101021  2421    7074    7504L
$FOPER. 101046  2724    7504    7517L
$FOPEU  101037  7512L
$FOPEU. 101054  7512    7521L
$FOPEW  101030  1871    2422    2732    7508L
$FOPEW. 101051  7508    7519L
$FREAL  101161  7032    7080    7611L
$FREAL. 101174  7611    7619E
$FREAD  101364  3504    7741L
$FREAD. 101377  7741    7749E
$FWBRK  102231  7906L
$FWBRK. 102240  7906    7911L
$FWBRK1 102254  7913    7918L
$FWRIB  102047  1880    1883    7398    7789L
$FWRIB. 102056  7789    7794E
$GNL    112031   957    10123L
$IBT    104213  1615    5221    5432    8477L
$INDL   030234  7983    7986    10241E
$MCU    112045  2895    4832    4845    4863    5373    5691    10174L
$MOVE   030252  2048    2410    2827    3026    5228    5246    5576    6842    8371    8432    8519    9978E
$MU10   030324  6142    10193E
$MU66   030337  1646    3082    5483    9990E
$MU86   031007  5539    6048    9896E
$RCHAR  103233  2474    8191L   8192
$REAL2  101205  7628L   7705
$REAL3  101226  7640L   7664
$REAL4  101254  7653    7656L
$REAL6  101272  7658    7673L
$REAL7  101304  7683L   7701
$REAL8  101352  7633    7636    7663    7719L
$REAL9  101360  7725L
```

```
$REA01  102007  7754L  7761
$REA02  102035  7758  7763L
$REA08  102043  7760  7772L
$REL7.3 101320  7688  7696L
$REL7.5 101343  7698  7709L
$RSTALL 031047  6201  9932  10210E
$SAVALL 031054  1535  6170  6223  9928  10224E
$SOB    111330  3978  4907  6739  9827L
$SOB1   111331  9828L  9831  9833
$TBLS   111371  5703  7574  10011L
$TBRA   031076  1466  10050E
$TJMP   031061  1109  3347  10070E
$TJMP.  031062  10072E
$TYPC.  112026  10092L
$TYPCC  103217  7327  7389  8172E  8181
$TYPCH  112022  10082L
$TYPTX  031136  958  1231  1822  2770  5155  5367  6414  7472  10108E  10608
$TYPTX. 031144  10110E
$UDD    031157  2140  9916E
$WCHAR  103241  7480  8195L
$WDR    031222  9740E
$WER    031241  9728E
$WRIB2  102062  7800L  7880
$WRIB3  102100  7810L  7846
$WRIB4  102126  7823  7826L
$WRIB6  102170  7828  7859L
$WRIB7  102202  7869L  7876
$WRIB8  102224  7803  7806  7850  7886L
$ZERO   031212  3751  9814E
$ZEROS  031320  9901E  10553
.  114300  11968  1197  10629S  10630  10631  10635S  10636  10637  10641S  10642  10643
.ABUSS  040024  329E
.ALARM  002136  302E
.ALEDS  040013  327E
.CHFLG  000060  569L
.CLEAR  000055  566L  6400
.CLEARA 000056  567L  5632
.CLOSE  000046  559L  8031
.CLRCD  000007  543L  965  5154
.CONSL  000006  542L  9931  10125
.CRC    002347  310E
.CRCSUM 040027  330E
.CTC    002172  304E
.CTLC   000041  554L  932  935  10604  10607
.CTLFLG 040011  326E
.DECODE 000053  564L
.DELET  000050  561L  2811
.DISMT  000061  570L
.DLEDS  040021  328E
.DLY    000053  299E
.DMNMS  000203  581L
.DMOUN  000201  579L
.DOD    003122  313E
.DODA   003356  315E  3920
.DSPMOD 040007  324E
.DSPROT 040006  323E
.DUMP   001374  301E
.ERROR  000057  568L  1235  6403  10652
```

```
.EXIT    000000   536L   1250   7573   7579   10654
.HORN    002140   303E
.IDENT   000000   298E
.IOWRK   040002   321E   2432   2434   3841   3842
.LINK    000040   553L   2796
.LOAD    001267   300E
.LOADD   000062   571L
.LOADO   000010   544L
.MFLAG   040010   325E   1050
.MONNS   000202   580L
.MOUNT   000200   578L
.NAME    000054   565L
.OPENC   000045   558L
.OPENR   000042   555L   7590
.OPENU   000044   557L   7592
.OPENW   000043   556L   7591
.PCHL    002264   306E
.POSIT   000047   560L
.PRINT   000003   539L   1821   2012
.RCK     003260   314E   3816
.READ    000004   540L   8123
.REGI    040005   322E
.REGPTR  040035   333E
.RENAM   000051   562L
.RESET   000204   582L
.RNB     002331   309E
.RNF     002325   308E
.SCIN    000001   537L   3488   6986   8191
.SCOUT   000002   538L   8178   8195   9781   10092
.SETTP   000052   563L   6199   10580
.SRS     002265   307E
.START   040000   320E
.SYSRES  000012   546L
.TICCNT  040033   332E   2480   2483   10596
.TPERR   002205   305E
.TPERRX  040031   331E
.UIVEC   040037   334E
.VERS    000011   545L   10572
.WNB     003024   312E
.WNP     003017   311E
.WRITE   000005   541L   7844   7952   8019
ABS      057055   3348   3428L
ABS.COD  000010   841L   845
ABS.ENT  000006   839L
ABS.ID   000000   835L
ABS.LDA  000002   837L
ABS.LEN  000004   838L
ACCX     042202   855L   2047   2262   3162   3269   3299   3336   3391   3428   3441   3456   3457
                  3465   3513   3559   3656   3658   3689   3749   3797   3830   3870   3877   3901   3934
                  3953   3954   3963   3971   4017   4065   4091   4145   4162   4208   4219   4313   4341
                  4367   4417   4450   4463   4491   4511   4536   4599   4603   4638   4718   4771   5399
                  5415   5444   5448   5744   5938   5953   5972   6342   6375   6801   6841   6870   6922
                  7412   8600   8616   8618   8619   8623   8660   8825   8860   8894   9002   9165   9166
                  9183   9267   9405   9408   9510   9526   9536   9544   9549   9578
ACCY     042210   857L   3600   3646   3662   3795   4095   4147   4156   4214   4328   4393   4458
                  4690   4732   4749   5745   5954   6887   6888   6933   7413   9417   9703   9708
ACILB    044216   1174E  1296   1479
ADD      104356   4392   4394   4462   4509   4556   4570   4625   4694   4769   8598E  8724   9709
```

```
ADD0     104367  8608L
ADD1     105022  8615    8629L
ADD2     105041  8641L   8653
ADD2.5   105057  8631    8651L
ADD2.7   105065  8654L   8656
ADD3     105074  8629    8637    8650    8660L
ADD4     105142  8685    8696L   8739
ADD5     105160  8610    8706L
AIO.CGN  041047  783L
AIO.CHA  041116  798L
AIO.CNT  041111  794L
AIO.CSI  041050  784L
AIO.DDA  041041  779E
AIO.DES  041055  788L
AIO.DEV  041057  789L
AIO.DIR  041062  792L
AIO.DTA  041053  787L
AIO.EOF  041113  796L
AIO.EOM  041112  795L
AIO.FLG  041043  780L
AIO.GRT  041044  781L
AIO.LGN  041051  785L
AIO.LSI  041052  786L
AIO.SPG  041046  782L
AIO.TEP  041114  797L
AIO.UNI  041061  790L
AIO.VEC  041040  778L
AMB      071026  1653    1698    1765    5264E   5821    6472    6946    7285
ANT      071056  1018    1107    1277    1658    1712    1795    1931    1991    2002    2319    2330    2360
                 2555    2576    2605    3051    3106    3114    3264    3313    3624    4236    4259    4278    5298L
                 6456    6508    6921    6962    7108    7145
ASC      057065  3375    3441L
ATAN1    065052  4608L
ATAN2    065137  4623    4627L
ATF      107323  2879    3979    6799    9349E
ATF0     107335  9355    9357L
ATF1     107340  9362L   9367    9371
ATF1.5   107365  9369    9375L
ATF10    110214  9475L   9482
ATF11    110230  9462    9487L
ATF2     110021  9395L   9403
ATF2.5   110035  9390    9404L
ATF3     110053  9363    9415L
ATF4     110067  9420L
ATF5     110102  9411    9429L
ATF6     110132  9439    9442    9445L
ATF7     110157  9457    9460L
ATF8     110160  9432    9461L
ATF9     110213  9467    9474L
ATFA     110217  9465    9470    9478E
ATN      065026  3349    4597E
ATP1     071116  5339L   5353
ATP2     071120  5340L   5349
ATS1     103321  8234    8242    8263L
ATS2     103376  8308L   8316
ATS3     104031  8333L   8377
ATS4     104057  8353L   8355
ATS5     104064  8352    8356L
```

```
ATSA    103356   8218    8285E
ATSB    104124   8326    8383L   8398
ATSC    104125   8384L   8397
AVV     071202   2053    2082    4277    5394L   6774
AVV1    071222   5410L
AYS     071146   1247    1259    5367L
BAS1    043203    961     968L
BAS2    043240    974     986L
BAS3    043233    972     983L
BEC.AC  000230    393L    5123
BEC.CB  000201    370L    5063
BEC.CC  000200    369L    5060
BEC.CIU 000233    396L    5135
BEC.DO  000203    372L    5069
BEC.DE  000202    371L    5066
BEC.EN  000224    389L    1730
BEC.FAE 000226    391L    5117
BEC.FNO 000231    394L    5126
BEC.IC  000222    387L    5114
BEC.ILF 000227    392L    5120
BEC.IN  000204    373L    5072
BEC.IU  000205    374L    5075
BEC.LK  000206    375L    5078
BEC.LTL 000232    395L
BEC.ND  000221    386L    5111
BEC.NV  000207    376L    5081
BEC.OV  000210    377L    5084
BEC.RE  000211    378L    5087
BEC.SC  000220    385L    5108
BEC.SL  000212    379L    5090
BEC.SN  000213    380L    5093
BEC.SR  000217    384L    5105
BEC.ST  000225    390L    2781
BEC.SY  000214    381L    1233    5096
BEC.TC  000215    382L    5099
BEC.TO  000216    383L    5102
BEC.UD  000223    388L    5132
BELL    000007    343E    1232    5041    5156    5368
BKSP    000010    345E
BLD1    044255   1211L    1227    1238
BLD2    044320   1218     1231L
BOOT.P  000001    758E
BUILD   044247   1111     1208E
BYE     044337   1112     1245E
C.STX   000002    347E
C.SYN   000026    346E
CAS     071334   5465L
CAS1    071346   5474E    5487
CB.CLI  000100    268E     283
CB.MTL  000040    267E
CB.SPK  000200    269E
CB.SSI  000020    266E
CBINT   100370    930     7469L   10602
CBINT1  100372   7460     7470L
CBT     103042   7624     7753    7796    7912    8044L
CBT1    103051   8049L    8058
CCINT   100363    933     7459L   10605
CDB.H84 000001    701E
```

```
CDB.M85 000000    700E
CEF     071374    2385    5502L
CF.FCN  000004    177E     187    189    193    194
CF.STR  000001    175E     188    189    191    193    194   2565   2918   3270   3339   3392   3690
                  5402    5564   5748   6440   6790
CF.VEC  000002    176E     190    191    193    194   1323   2034   2925   3053   3129   4237   5885
                  6468    6512   6602
CFA     072005    1426    1869   2382   2387   2413   3502   5518L  7026   7395
CFA1    072015    5531L
CFCTLB  000002    1182    7469   7476   10539E
CFCTLC  000001    2165    6983   7459   7477   10538E
CFN     072053    1389    2294   2357   2819   5555   5560L
CFN.    072041    1867    2682   2721   5555L
CFS1    072140    5597L   5608
CHAIN   045205    1123    1386E
CHAIN1  045231    1393    1397L
CHANMAX 000005      38E    5337   5518   6023   10671
CHR$    057103    3350    3453L
CIN     057140    3351    3476E
CIN0    057160    3489L   3505
CIN1    057170    3491    3493L
CIN2    057202    3484    3502L
CLEAR   044363    1124    1275E
CLEAR.  044375    1281L   1362
CLF     072171    1287    5629E
CLL1    111277    9760L   9764
CLN     072206    1214    5178   5648L
CLOSE   045260    1125    1421E  1436
CLOSE1  045304    1427    1430L
CLR1    045021     978    1292L  7070
CLR2    045062    1280    1313L
CLR3    045132    1324    1327   1339L
CMA     072223    1394    1435   1595   2113   2441   3583   3686   3694   3790   5667L  6001
CNC     072230     970    2853   5680L
CNC1    072265    5686    5693   5701L
CNCA    072302    5702    5712E
CNTL1   045341    1467    1477L
CNTL2   045353    1468    1484L
CNTL3   045371    1469    1494L
CNTL4   046003    1470    1502L  10601
CNTL43  046017    1509L   1512
CNTL46  046033    1510    1513   1515L
CNTL5   046041    1471    1523L
CNTL52  046126    1548    1557L
CNTLA   045334    1467L   1472
CNTLMX  000005    1464    1472E
CNTRL   045320    1126    1461E
CO.FLG  000001     678E    9930
COLCNTS 112253     955    3851   6527   6995   10671L
CONT    045163    1113    1368L  1402   1410
CONT1   045165    1369L   2764
COS     064125    3352    4488E
COS1    064175    4507    4511L
COS2    064233    4523L   4591   4627
COSA    064234    4506    4524E  4541   4552   4601
COT     072347    4059    4140   5744L  7048
CR      000015     339E
CS.FLG  000200     679E
```

CSA	072366	1671	1767	2051	2080	2235	4276	5412	5443	5763E	6443	6773	7175
CSE	073016	5810L	6450										
CSE.	073033	3023	3458	3659	3955	4154	5816L						
CSE..	073056	5812	5818	5829L									
CSE1	073045	5814	5821L										
CSE2	073065	5832	5834L										
CSI	073000	1631	1751	1781	2222	5410	5784E	6439					
CSL.CHR	000001	656E											
CSL.ECH	000200	654E											
CSL.WRP	000002	655E											
CT.ABS	000320	212L	4957										
CT.ALP	000001	75L	2865	5695									
CT.AND	000310	199L	3182	4958									
CT.AS	000311	200L	2367	4960									
CT.ASC	000350	240L	4959										
CT.ATN	000321	213L	4961										
CT.BLD	000200	110L	1093	4962									
CT.BYE	000201	111L	4963										
CT.CHA	000213	124L	4964										
CT.CHR	000322	214L	4965										
CT.CIN	000323	215L	4966										
CT.CLO	000215	126L	4968										
CT.CLR	000214	125L	4967										
CT.CMA	000026	100L	1661	1713	2340	2556	2627	3107	3625	4261	5668	5722	6966
CT.CMD	000256	163E	1096										
CT.CNT	000202	112L	4970										
CT.COS	000324	216L	4971										
CT.CTL	000216	127L	4969										
CT.DAT	000251	157L	2669	4900	4972								
CT.DEF	000252	158L	4973										
CT.DEL	000203	113L	4974										
CT.DI	000024	98L	3232	4186	5718								
CT.DIM	000217	128L	4975										
CT.END	000253	159L	1074	4976									
CT.EQ	000011	86L	2077	3196	4977	4289	5724						
CT.EX	000025	99L	3247	5719									
CT.EXP	000325	217L	4977										
CT.FCN	000320	210E	3329	3335	3591	4889	6106						
CT.FIL	000312	201L	2369	4978									
CT.FIN	000000	74L	1015	1088	1278	1392	1432	2103	2110	2338	2472	2546	2755
		2759	2854	4292	5683	5720	6734	6824	6964	7146	7167		
CT.FN	000220	129L	1570	3324	4979								
CT.FNM	000357	247L											
CT.FOR	000221	130L	2359	4980									
CT.FRE	000222	131L	4982										
CT.FRZ	000223	132L	4981										
CT.GE	000013	88L	2984										
CT.GOS	000224	133L	2322	4983									
CT.GOT	000225	134L	1932	2320	4984								
CT.GT	000012	87L	2863	2986	5725								
CT.IF	000226	135L	4985										
CT.INP	000254	160L	1972	4986									
CT.INT	000326	218L	4987										
CT.IUA	000250	154E	983										
CT.LCK	000230	137L	4993										
CT.LE	000015	90L	2967										
CT.LEF	000351	241L	3590	4989									
CT.LEN	000352	242L	4988										

CT.LET	000227	136L	4990										
CT.LIN	000250	156L	4991										
CT.LIS	000204	114L	4992										
CT.LNO	000327	219L	4994	6130									
CT.LOG	000330	220L	4995										
CT.LT	000014	89L	2861	2975	5723								
CT.MAT	000353	243L	4996										
CT.MAX	000331	221L	4997										
CT.MI	000022	96L	3215	3260	3273	5714							
CT.MID	000354	244L	3590	3591	3594	4998							
CT.MIN	000332	222L	4999										
CT.MU	000023	97L	3230	5717									
CT.NE	000016	91L	2973	3198									
CT.NOT	000314	203L	3262	5001									
CT.NUM	000002	76L	973	2867	5688	5721							
CT.NXT	000231	138L	5000										
CT.OLD	000232	139L	5002										
CT.ON	000233	140L	5003										
CT.OPE	000234	141L	5004										
CT.OR	000315	204L	3169	5005									
CT.OUT	000235	142L	5006										
CT.PAD	000333	223L	5007										
CT.PAL	000017	93L	3314	5715	5728								
CT.PAR	000020	94L	1663	2609	3115	3321	3622	3788	4263	4282	5716	5729	6156
		6737											
CT.PAU	000236	143L	5008										
CT.PEK	000334	224L	5009										
CT.PIN	000335	225L	5010										
CT.PL	000021	95L	3213	4137	5713								
CT.POK	000237	144L	5011										
CT.POS	000336	226L	5012										
CT.PRT	000240	145L	5013										
CT.PS	000031	103L	5730	5998	6010								
CT.QUO	000030	102L	2859	5727									
CT.REA	000241	146L	2361	2420	5014								
CT.REM	000242	147L	4898	5015									
CT.REF	000205	115L	5016										
CT.RES	000243	148L	5017										
CT.RET	000244	149L	5018										
CT.RIG	000355	245L	3594	5019									
CT.RND	000337	227L	5020										
CT.RUA	000212	121E	1099										
CT.RUN	000206	116L	5021										
CT.SAV	000207	117L	5022										
CT.SCR	000210	118L	5023										
CT.SEG	000340	228L	5024										
CT.SEM	000027	101L	1989	2004	2554	2629	5726						
CT.SEF	000003	77L	2869	5707									
CT.SGN	000341	229L	5025										
CT.SIN	000342	230L	5026										
CT.SNF	000304	187L											
CT.SNV	000300	186L	192	1783	3829	4018	6374	6800	7173				
CT.SPC	000343	231L	2552	2611	5027								
CT.SQR	000344	232L	5028										
CT.SRA	000350	239E	3335										
CT.SSF	000305	189L	1315										
CT.SSV	000301	188L	1994	2035	2052	3030	3962						
CT.STE	000211	119L	1797	5030									

CT.STP	000255	161L	5031						
CT.STR	000345	233L	5029						
CT.SYE	000212	123L	4927	5041					
CT.TAB	000346	234L	2550	5032					
CT.TAN	000347	235L	5033						
CT.THN	000316	205L	1934	5034					
CT.TO	000317	206L	1786	5035					
CT.UNF	000245	150L	5036						
CT.UNL	000246	151L	5037						
CT.UNS	000247	152L	5038						
CT.VAL	000356	246L	5039						
CT.VARH	000307	193E	2020	3293	6459	6703			
CT.VARL	000300	192E	1313	2018	3291	6457	6701		
CT.VNV	000302	190L							
CT.VSV	000303	191L							
CT.WRI	000313	202L	2363	5040					
CTB	103070	7726	7773	7888	7915	8070L			
CTB1	103101	8076L	8085						
CTLA	000001	354E							
CTLB	000002	355E	931	10603					
CTLC	000003	356E	934	10606					
CTLD	000004	357E							
CTLFLAG	112142	949	1027	2164	2506	6982	7482	10537L	
CTLO	000017	358E							
CTLP	000020	359E							
CTLQ	000021	360E							
CTLS	000023	361E							
CTLZ	000032	362E							
CTP.2SB	000010	664E							
CTP.BKM	000002	665E							
CTP.BKS	000200	661E							
CTP.NLI	000040	662E							
CTP.NLO	000020	663E							
CTP.TAB	000001	666E							
CUF	073104	5862E	7069						
CUF1	073107	5866L	5874						
CUF2	073126	5872	5878E						
CUF3	073163	5886	5904L						
CUF4	073172	5890	5912L						
CURADR	112135	1168	1300	1369	1409	1729	1917	10523L	
CURNUM	112133	1071	2710	2772	6416	7287	10522L		
CVX	073210	2221	2255	4168	4684	4748	5936L		
CXV	073237	2236	2884	5969L					
CXV.	073240	5405	5970L						
CXY	073223	4319	4388	4456	4679	4725	4746	4767	5951E
D.CON	040110	616L							
D.RAM	040240	619L							
D.VEC	040130	618L							
DATPTR	114303	1308	2651	2653	10690L				
DCN	073253	1983	2094	2542	5995L				
DCN.	073273	1422	2370	6000	6009L				
DCN..	073302	6019L							
DCN1	073322	6026	6028L						
DDN	111171	5177	9448	9669E					
DDN1	111202	9377	9673L	9692					
DDN2	111233	9381	9691L						
DDNERR	070122	9671	9683	9687	9741E				
DEF	046133	1155	1568E						

```
DEFALTD 043100    924L    2415
DEFALTF 043072    923L    2723    2731    2809    7073
DELETE  046162   1114     1592E
DF.CLR  000376    468E
DF.EMP  000377    467E
DFD     111237   9420     9701L   9714
DFD1    111267   9424     9713L
DIM     046236   1127     1624E   1715
DIM2    046265   1638L    1662
DIM3    047011   1703L    1708
DIM5    047033   1627     1719L
DIMA    047034   1626     1720E
DIR.ALD 000025    483L
DIR.CLU 000015    476L
DIR.CRD 000023    482L
DIR.EXT 000010    471L
DIR.FGN 000020    479L
DIR.FLG 000016    477L
DIR.LGN 000021    480L
DIR.LSI 000022    481L
DIR.NAM 000000    470L
DIR.PRO 000013    472L
DIR.VER 000014    473L
DIRELEN 000027    485E     516     792
DIRIDL  000015    474E
DIV     106264   4460     4574    4658    4728    4750    9031E   9469
DIV0    106305   9034     9040L
DIV1    106355   9061L    9109
DIV2    106365   9066L    9100
DIV3    107020   9078     9088L
DIVA    106367   9046     9068E
DIVB    106373   9048     9072E
DIVC    106377   9050     9076E   9121
DM.MR   000000    273E
DM.MW   000001    274E
DM.RR   000002    275E
DM.RW   000003    276E
DNF     073326   1429     6044L   6056
DTS     073357   1089     1281    6068L   10595
DTSA    073366   1298     6071E
EC.CNA  000004    406L
EC.DDA  000027    425L
EC.DIF  000017    417L
EC.DIW  000035    431L
EC.DNI  000045    439L
EC.DNR  000046    440L
EC.DNS  000005    407L
EC.DSC  000047    441L
EC.EOF  000001    403L    5129    7613    7743    8132
EC.EOM  000002    404L
EC.FAD  000031    427L    7543
EC.FAP  000026    424L
EC.FL   000030    426L
EC.FNF  000014    414L    2726
EC.FND  000011    411L    7631
EC.FNR  000034    430L
EC.FOD  000043    437L
EC.FUC  000013    413L
```

```
EC.ICN  000016  416L
EC.IDN  000006  408L
EC.IFC  000020  418L
EC.IFN  000007  409L
EC.ILC  000003  405L
EC.ILO  000040  434L
EC.ILR  000012  412L
EC.ILV  000037  433L
EC.IOI  000052  444L
EC.IS   000032  428L
EC.NCV  000050  442L  10649
EC.NEM  000021  419L
EC.NOS  000051  443L
EC.NPM  000044  438L
EC.NRD  000010  410L
EC.NVM  000042  436L
EC.OTL  000053  445L
EC.RF   000022  420L
EC.UNA  000036  432L
EC.UND  000015  415L
EC.UUN  000033  429L
EC.VPM  000041  435L
EC.WF   000023  421L
EC.WP   000025  423L
EC.WPV  000024  422L
EKA     073374  2148  6086L
EKA1    074001  6089L  6092
EKA2    074011  6098L  6103
ELN     074033  1395  1909  2329  6125L
ELN1    074055  6138L  6148
ELN2    074102  6131  6152L
END     047044  1156  1728L
ENL     000212  352E  10609
EOFFLG  103216  7622  7751  8104  8131  8158L
ERR.AC  070205  4281  5123L
ERR.CB  070111  1177  5063L
ERR.CC  070106  1030  2025  2167  5060L  5371
ERR.CIU 070221  2402  5135L
ERR.DO  070117  5069L  6904  9037
ERR.DE  070114  2667  5066L
ERR.EQE 070213  5129L  7033
ERR.FAE 070177  2725  5117L
ERR.FNO 070210  3503  5126L  7027  7396
ERR.IC  070174  2871  5114L
ERR.ILF 070202  5120L  5548  5521
ERR.IN  070122  1226  1465  1487  1496  1504  1525  1528  1676  1678  2479  3419  3481
                3619  3710  4370  4420  5072L  5484  5519  5650  5655  6021  6024  6147  6349
                6352  9453  9741
ERR.IU  070125  984  1094  1105  5075L  6511
ERR.LK  070130  5078L  6543
ERR.ND  070171  3064  5111L  6469
ERR.NV  070133  2218  5081L
ERR.OV  070136  4325  4344  4371  5084L  8692  8768  8848  8976  9040  9042  9310
ERR.RE  070141  2695  5087L
ERR.SC  070166  3108  3116  5108L
ERR.SL  070144  3019  4149  5090L
ERR.SN  070147  1407  1478  1911  5093L  6255
ERR.SR  070163  3094  5105L
```

```
ERR.SY  070152  968  1122  1128  1314  1316  1574  1609  1664  1935  2019  2021  2036
                2323  2341  2364  3063  3330  3367  3370  3626  4238  4264  5096L  5565  6133
                6458  6460  6968  7114  7174
ERR.TC  070155  3271  3340  3393  3691  4019  5099L  5403  5749  7050
ERR.TO  070160  1647  2388  5102L  6207  8269  8278  8320
ERR.UD  070216  4241  5132L
ERROR1  070244  5146  5151E
ESC     000033  350E
EVAL    055244  2078  2561  3158E  3316  3332  3390  3688  4271  4290  5560
EVALI   057036  1594  1598  1640  1927  2106  2114  2317  2439  2442  2476  2757  3087
                3404L  3415  6011  6153
EVALI8  057044  2607  3415L  3585  3629  3696
EVALN   057022  1787  1798  3390L  3404  3792
EXEC    043241  1005E  1372
EXEC1   043244  1010L  1057  1079  1946
EXEC10  044231  1184L
EXEC2   043270  1022E  1090
EXEC3   043372  1017  1083L
EXEC3.5 043375  1087L  1188
EXEC4   044007  1083  1093L
EXEC5   044035  1100  1107L
EXEC6   044040  1075  1108L
EXEC7   044201  1023  1076  1162L  2780
EXEC8   044215  1032  1173L
EXEC9   044225  1181L
EXECA   044244  1046  1192L  1197
EXP     063075  3353  4227  4311E
EXP1    063133  4323  4326L
EXPA    063221  4317  4350L
FB.CHA  000000  451L  7527  7530  8017  8028  8144
FB.FLB  000001  452L  2390  2399  2404  5579  5634  6049  7530  7547  7585  7972  7984
                7987  8146
FB.FWA  000002  453L  5338  7547  7553  7987  8148
FB.LIM  000006  455L  7558  7563  8152
FB.LWA  000010  456L  8154
FB.NAM  000012  457L  458  2404  2406  2407  2820  5573  7563
FB.NAML 000021  458E  891  897  903  909  915  921  2408  2409  2825  5570
FB.PTR  000004  454L  7553  7558  7984  8150
FBENL   000033  459E  5338  5350  5538  6047
FBLIST  042230  887L  2406  5338  5540  5557  5573  5579  5580  5634  6049  7072  7078
                7090
FBSCR   042230  889L  10586  10587  10588  10591
FBUFAD  042226  885L  5327  5330
FF      000014  353E
FILTAB  112121  2386  5329  5502  5531  5631  6044  6054  10504L
FLN     074242  1406  1477  1596  1600  1910  2115  2122  5199  6245L
FLN1    074260  6252  6257L  6272
FLN1.5  074300  6264  6267L
FLN2    074301  6268L  6271
FLN3    074312  6263  6266  6276L
FNRMA   112265  10686E
FOC     074115  953  1531  2418  2744  6170L  7075  10610
FOC1    074146  6180  6185L
FOC1.3  074152  1559  6187L
FOC1.5  074162  6193L  6225
FOC2    074205  6197  6201L
FOC3    074210  6191  6205L
FOF     074217  2411  5556  6223L  6398  7071
```

FOP.	074230	1536	6224	6227L									
FOR	047060	1129	1749E										
FOR1	047104	1753	1762E										
FORTAB	112070	1293	1758	1764	2228	2229	2280	2283	7178	7181	10431L		
FP0.0	031320	10553E											
FP0.1	112157	10552L											
FP1.0	112147	1796	4218	4747	10550L								
FP10.	112153	9466	9525	9605	9706	10551L							
FPADD	104352	2232	3519	4141	4486	8586L	9547						
FPDIV	106260	4193	9022L	9533	9538								
FPMODE	043316	1045E	1497										
FPMUL	105323	4191	4226	8837L	9529	9606							
FPNEG	105302	3278	3430	3496	3517	3549	3940	8809L	9488	9515			
FPNRM	105202	3909	8737L	9603									
FPSUB	105166	2260	3796	4064	4177	8719L							
FPTST	105316	8825L											
FRC	061262	3459	3962L										
FREE	047213	1130	1814E										
FREE1	047225	1821L	1836										
FREEA	047272	1819	1847E										
FREEZE	047336	1131	1866E										
FREZEA	050016	1877	1888L	1892									
FREZEAL	000010	1879	1892E										
FREZEB	050022	1876	1890L										
FSE	074315	3442	3663	3700	3704	3972	4092	4096	4157	4163	5413	5446	5449
		5566	6290L	7310									
FSE0	074341	6303	6307L										
FSE1	074344	6305	6308	6312L	6325								
FSE2	074363	6317	6321L	6324									
FSE3	074374	6314	6320	6329L									
FT.ABS	000000	825E	846	1888									
FT.BAC	000003	828E											
FT.DD	000001	496E											
FT.DR	000002	497E	7517	7521	7590	7592	7630						
FT.DU	000010	499E											
FT.DW	000004	498E	7519	7521	7591	7592	7802	7978					
FT.PIC	000001	826E											
FT.REL	000002	827E											
FTA	110237	2571	3948	7326	9506E								
FTA1	110261	9512	9517L										
FTA10	111112	9626L	9628										
FTA11	111125	9632L	9634										
FTA12	111135	9612	9639L	9642									
FTA13	111152	9635	9644	9646L									
FTA2	110267	9523	9523L	9551									
FTA2.5	110332	9535	9539L										
FTA2.7	110356	9519	9555E										
FTA3	110367	9560	9562L										
FTA4	110372	9568L	9607										
FTA4.5	111001	9569	9572L										
FTA5	111005	9577L											
FTA6	111023	9585L	9588										
FTA7	111040	9596L	9598										
FTA7.5	111045	9583	9599L										
FTA8	111055	9604E											
FTA8.5	111066	9573	9611L										
FTA9	111107	9620	9624L										
FTAA	111165	9546	9658L										

```
FTAC     110360   1488     9558E
FTAD     110370   1489     9563E
FWBRK2   102273   7932L    7939
FWBRK3   102307   7934     7941L
GOSTAB   112075   1294     2690    2692    2700    7284    10446L
GOSUB    050026   1132     1899E
GOTO     050031   1133     1908E   1944    1961
GOTO1    050034   1910L    2335
GOTO2    050042   1187     1913E
HLCPDE   111346   5868     9863L
I.CONFL  000004    681E     682    9929
I.CONTY  000001    668E     669
I.CONWI  000003    674E     675
I.CSLMD  000000    658E
I.CUSOR  000002    671E     672    10123
IBT1     104270   8519L    8571
IBT2     104301   8487     8527L
IBTA     104244   8477     8500E   8522    8552
ICL      065364    960     1216    4808E
ICL.     065373   4817L    7083
ICL1     066000   4825L    4836    4838    4847    4849    4922
ICL1.5   066001   4826L    4903    4909
ICL10    066234   4831     4948L
ICL2     066056   4854L    4879
ICL3     066062   4858L    4868
ICL4     066104   4873L    4876
ICL5     066122   4860     4886L
ICL5.5   066142   4894     4896L
ICL6     066163   4890     4907L
ICL7     066176   4834     4914L   4921
ICL8     066216   4880     4910    4919    4927L
ICL9     066223   4899     4901    4935L   4940
IF       050051   1134     1926E
IF0      050114   1942     1945L
IF1      050120   1950L    1953
IF2      050121   1146     1938    1951L   1960
IF3      050127   1933     1958L
IFIX     075002   3405     3453    3477    3824    3838    3850    3919    6343L   6859
IFIX.    074377   4020     4320    6342L   6861
IFIX1    075026   6355L    6357
IFLT     075040   3494     3765    3828    3936    3938    3939    4027    4327    4390    6369L   7324
ILM      075063   1732     2783    5160    6398L
ILMA     075077   1011     1628    6406E
INP1     050151   1974     1982L
INP2     050205   1990     1994L
INP4     050233   1992     1995    2009L   2060
INP4.5   050244   2011     2013E
INP5     050304   2033L
INP6     050356   2029     2057L
INPUT    050150   1157     1981L
INPUTA   050370   1982     2026    2062L
INPUTB   050371   1988     2059    2063L
INPUTC   050373   2005     2064L
INT      057216   3354     3512E   4729
INT1     057245   3529L    3533
INT2     057260   3530     3535L
INTA     057302   3518     3552L
IOC.CGN  000010    504L
```

IOC.CSI	000011	505L											
IOC.DDA	000002	493L	500	514									
IOC.DES	000016	511L											
IOC.DEV	000020	512L											
IOC.DIL	000021	514E											
IOC.DIR	000023	516L											
IOC.DRL	000010	508E											
IOC.DTA	000014	510L											
IOC.FLG	000004	495L	508										
IOC.GRT	000005	502L											
IOC.LGN	000012	506L											
IOC.LNK	000000	492L											
IOC.LSI	000013	507L											
IOC.SPG	000007	503L											
IOC.SRL	000003	500E											
IOC.UNI	000022	513L											
IOCGTD	000001	520E											
IOCELEN	000052	518E											
IOCHAN	112140	1013	1424	1999	2009	2376	2738	5996	6028	6528	7018	7381	10532L
IP.PAD	000360	259E											
IST	075136	2038	2071	4255	6437L	6747	7172						
IST0	075172	6437	6455E										
IST00	075170	6441	6452L										
IST1	075216	6464L	6513										
IST2	075257	6463	6493L										
IVT	075263	1571	1629	6508L									
KEYTAB	066240	4853	4956E	6086									
L1	106024	8875L	8885										
L2	106035	8879	8882L										
L4	106077	8902	8910L										
L4,5	106120	8896	8917	8919	8921L								
L5	106122	8924L	8949										
L6	106144	8928	8940L										
L7	106174	8950	8952	8954	8958L								
LDC	075302	2371	2577	2614	6527L	7340	7376						
LDKFLG	112137	2205	6541	10524L									
LDD	107250	6346	8599	9171	9275L								
LDX	107245	8738	8810	9057	9209	9267L							
LEFT$	057314	3376	3579E										
LEFT$1	060032	3593	3641E										
LEN	057306	3377	3559L										
LET	050374	1097	1135	2071L									
LET.	050377	1784	2073L										
LEV1	055255	3160	3168L										
LEV11	055260	3169L	3177										
LEV2	055304	3168	3172	3181L									
LEV21	055307	3182L	3190										
LEV3	055333	3181	3185	3191E									
LEV4	055333	3195L											
LEV41	055336	3196L	3208										
LEV5	055367	3195	3202	3212L									
LEV51	055372	3213L	3225										
LEV52	056002	3214	3217L										
LEV6	056025	3212	3219	3229L									
LEV61	056030	3230L	3242										
LEV62	056040	3231	3234L										
LEV7	056063	3229	3236	3246L									
LEV71	056066	3247L	3255										

```
LEV8      056112  3246    3250    3259L
LEV81     056127  3261    3264L
LEV82     056163  3274    3284L
LEV9      056170  3263    3266    3290L
LEV90     056307  3337    3339L
LEV92     056234  3292    3294    3313L
LEV93     056256  3315    3324L
LEV94     056247  3320L   3342
LEV95     056220  3304L   3309
LEX0      054202  2868    2875L
LEX1      054231  2866    2892E
LEX10     054354  2862    2964L
LEX11     054377  2864    2981L
LEX11.5   055012  2044    2998L
LEX12     055015  2860    3001L   6837
LEX13     055027  3007L   3014
LEX14     055044  3012    3017L
LEX2      054264  2901    2903    2914L
LEX3      054274  2916    2920L
LEX3.5    054312  2923    2931L
LEX7      054334  2936    2951L
LEXA      075236  2951    6474E
LEXB      042222  869L    2883    2887    3021    3022    3031
LEXC      042215  866L    2953
LEXCAL    054131  2851E   6709
LEXD      055033  3003    3011E
LEXLIM    042226  871E    1318    6461    6509
LF        000012  340E
LFC       075313  1209    1246    1258    1276    1361    1593    2793    5175    6541L
LINE      112265  969     1219    2023    2028    4809    5176    5369    7077    10634   10635   10675L  10677
                  10686
LINE2     113273  2138    2570    2823    3947    3974    3977    6815    6832    7325    7360    10640   10641
                  10682L
LINEL     000377  7030    7076    10634   10635   10640   10641   10677E  10682
LINPUT    050137  1153    1970E
LIST      051020  1115    2092E
LIST.     051023  2098L   2740
LIST1     051065  2105    2112    2119L
LIST2     051073  2126L   2168
LIST3     051126  2145L   2152
LIST6     051170  2135    2172L
LNO       057064  3355    3434L
LOADL     113025  848     10658E
LOCK      051175  1136    2202L
LOG       063225  3356    4224    4365E
LOGA      063350  4391    4400L
LOGB      063354  4395    4401L
LSH       107101  8788    9098    9127    9138L
LVS       075323  2933    6561E   7241
LVS1      075335  6570L   6630    6639
LVS2      075363  6578    6580    6593L
LVS3      076027  6603    6607    6634L
LVS4      076040  6572    6643L
M.CDCA    000017  817L
M.CDLY    000016  816L
M.CFWA    000012  814L
M.CIN     000006  812L
M.CINT    000005  811L
```

```
M.CLWA  000014    815L
M.COUT  000010    813L
M.CPRE  000003    809L
M.CRUB  000004    810L
M.CSLC  000002    808L
M.FOX   000303    293E
M.FAMB  000021    292E
M.SALO  000001    807L
M.SYSM  000000    806L
MATC2.3 060257    3715    3747L
MATC2.5 060260    3719    3748L
MATCH   060111    3378    3683E
MATCH1  060225    3726L   3745
MATCH2  060247    3728    3741L
MATCH3  060273    3737    3757L
MATCHA  060307    3684    3703    3767L
MATCHB  060276    3707    3760E
MATCHC  060313    3692    3699    3768L
MAX     060317    3357    3780L
MAX1    060325    3787L   3804
MAX2    061002    3789    3808L
MEML    112127    1873    6186    6193    8271    8327    10515L
MI.ADDB 000200      52E   8845
MI.CMC  000077      54E   9032
MI.IN   000333      61E   3840
MI.JMP  000303      53E
MI.LDA  000072      55E
MI.LXIB 000001      64E   5061    5064    5067    5070    5073    5076    5079    5082    5085    5088    5091
                    5094  5097    5100    5103    5106    5109    5112    5115    5118    5121    5124    5127    5130
                    5133
MI.LXID 000021      63E   3000
MI.LXIH 000041      62E
MI.MVIA 000076      56E   2202    3780
MI.NOP  000000      59E    947    5300
MI.OUT  000323      57E   2431
MI.RET  000311      60E   6715
MI.SUBB 000220      58E   9032
MID$    057314    3379    3581E
MID$1   057365    3592    3615E
MID$2   060032    3609    3623    3642E
MID$3   060043    3649    3651L
MID$4   060050    3652    3654L
MIN     060320    3358    3781L
MOV     071015    3958    3975    4160    4166    5243L   5452
MOV4    076051    1792    1799    3685    3693    4220    5939    5973    6668L
MOV5    076045    5955    6663E   6910    6948
MT.AFC  000000   10307E
MT.FWA  000001    1342    2229    2690    5197    5224    5329    5422    5436    5767    5788    5880    6304
                  6307    6568    6906    7181   10308E
MT.LEN  000003    1283    1284    1293    1294    1295    1625    1721    1816    2228    2280    2283    2386
                  2692    2700    5141    5531    5593    5631    5866    6044    6054    6069    6565    6900    6903
                  7127    7178   10309E
MTABIND 112056    1816    5141    5593    8384   10310E   10513
MTABL   000010    1818    5596    8383   10513E
MTABLEN 000005   10518E
MTAREA  114346     885     896     896     896     896     902     902     902     902     908     908     908
                   908     914     914     914     914     920     920     920     920    1299    1307    1611    1728
                  5595    6246    7129    7130   10326   10415   10432   10447   10460   10486   10497   10505   10515
```

10579 10700E

MTD	104127	6227	8263	8397L									
MTD1	104151	8417L	8437										
MTL	070270	979	5174E	7085									
MTL0	070304	1222	5182L										
MTL1	070307	5184L	5186										
MTL1.5	070325	5189	5191L										
MTL2	070361	5201	5209L										
MUL	105327	4318	4396	4512	4586	4621	4639	4691	4731	8844E	9464	9477	9707
MUL2.5	106047	8874	8886	8891L									
MUL3	106061	8898L	8916										
MUL3.5	106216	8963	8975L	9126									
MUL4	106222	8980L	9128										
MUL5	106240	8847	8991L	9038	9415								
MULA	106034	8852	8881E										
MULB	106051	8865	8893E										
MULC	106072	8856	8905E										
MULD	106076	8853	8909E										
MULE	106121	8862	8922E										
MULF	106133	8859	8931E										
MULG	106137	8857	8935E										
MULH	106143	8854	8939E										
NEG	105305	4316	4493	4510	4526	4542	4557	4571	4602	4626	8722	8810L	
NEXT	051203	1137	2216E										
NL	000012	351E	352	1234	2159	5156	6988	7347	7697	9780	10609	10609	10651
NPI	112173	4543	10556L										
NPI.2	112163	4485	4502	4508	4548	4555	4624	10554L					
NPI.4	112177	4561	4569	10557L									
NPI2	112167	4497	4726	10555L									
NRM	105213	6372	8697	8706	8748E	9305							
NRM.	105205	8738L											
NRM0	105221	8701	8756L	8790									
NRM1	105227	8763L	8784										
NRM2	105242	8752	8772L										
NRM3	105257	8781	8783L										
NRM4	105263	8788L	8795										
NUL2	000000	342E											
NULL	000200	341E											
NXT1	051271	2251	2260L										
NXT1.5	051307	2264	2269L										
NXT2	051317	2265	2280L										
OLD	051332	1138	2293E										
ON	051355	1139	2317L										
ON1	052001	2321	2325E										
ON2	052001	2329L	2342										
ONA	052035	2331	2336	2344L									
OP.CTL	000360	260E											
OP.DIG	000360	261E											
OP.SEG	000361	262E											
OPEN	052036	1140	2356E										
OPEN1	052062	2362	2365L										
OPEN2	052112	2381L	2393										
OPEN3	052144	2383	2398L										
OPEN4	052210	2416	2420L										
OUT	052220	1141	2429L										
OUT1	052235	1210	1462	2429	2439L	2517							
OVL.COD	000000	589L											
OVL.ENS	000010	594E											

```
OVL.ENT 000004      591L
OVL.FLB 000006      592L      1541
OVL.IN  000001      725E      1544
OVL.NUM 000014      727E
OVL.RES 000002      726E
OVL.SIZ 000002      590L
OVL.UCS 000200      728E
OVL0    000000      600L      1540
OVL1    000001      601L
OVLMAN  112141      1529      2295    6178    6206    10534L
P.ADD   062134      3223      4137L
P.ADDA  062235      4152      4153    4167    4170L
P.AND   061336      3188      4003L
P.CMP   061375      3206      4058L
P.CMP1  062013      4067L     4131
P.CMP13 062030      4071      4073    4075L
P.CMP2  062047      4060      4090L
P.CMP3  062073      4105L     4114
P.CMP4  062115      4108      4118L
P.CMP5  062121      4106      4123L
P.CMP6  062130      4111      4119    4124    4130L
P.EXP   062270      3253      4206E
P.EXP1  062321      4210      4223L
P.MUL   062247      3240      4186L
P.NOT   061351      3284      4017L
P.NOT1  061371      3995      4009    4026L
P.OR    061323      3175      3989L
P.SUB   062241      4139      4176L
PAD     061006      3359      3816L
PATCH   114305      10694L
PAUSE   052251      1142      2470E
PAUSE1  052275      2483L     2509
PAUSE2  052320      2486      2498L
PAUSE3  052326      2494      2506L
PBD     076345      3989      4003    6858L
PEEK    061014      3360      3824L
PEEK1   061020      3444      3446    3560    3817    3826L   3843    3858    3924
PEEK1.5 061023      3828L     4080    4082
PEEK2   061026      3829L     3980
PI.4    112203      4573      10558L
PIN     061034      3361      3838L
PLY     065204      4650      4669    4679L
PLY0    065207      4654      4680L
PLY1    065222      4688L     4702
PLY2    065240      4686      4696L
PMAA    107005      9056      9080L
PMAB    107011      9054      9083L
PMAC    107015      9052      9086L
PMD     107114      8846      9033    9161E
PMD1    107200      9195      9201L
PMD2    107204      9169      9209L
PMDA    107152      9167      9178E
PMDB    107164      9164      9186L
PNT     076072      1014      1390    1431    1987    2017    2033    2102    2109    2471    2545    2752    3259
                    3290      3311    3322    3621    3787    4294    5298    5997    6699L   6733    7166
PNT1    076130      5303      6716L
PNT2    076111      6702      6704    6707E
PNTA    076073      5302      6700E   6713
```

```
PNTR     076107    6706E     6711
PNTC     076111     948      6708L    6716
POKE     052336    1143      2516E
POLY     065177    4330      4441     4513     4668L
POLY2    065245    4701L
POLYQ    065153    4377      4649L
POP      076373    6871      6897L
POPX     076357    3799      4457     4470     6870L
POPY     076365    3173      3186     3203     3220     3237     3251     3793     4225     4584     4619     4656     6887L
POPY     076370    3588      3630     6888L
POS      061053    3362      3850L
POS1     061067    3854      3856L
PRI1     052352    2543      2543L
PRI2     053027    2570L
PRI3     053040    2557      2576L
PRI5     053074    2597L     2601     2613     2619
PRI6     053105    2551      2553     2605L
PRI7     053142    2548      2624L
PRI8     053156    2599      2635L
PRIA     053144    2541      2549     2626E
PRIB     053050    1518      2580E
PRIC     053062    1505      2591E    2594
PRINT    052343    1144      2539E
PRS      112210     849     10568E
PRSA     112317   10609L
PRSB     113014   10631     10637    10643    10647E
PRSERR   113016   10581     10651L
PRSERR1  113014   10573     10575    10649L
PRSLIM   113025   10656E    10663    10678
PSH      077044    6923      6943L
PSHX     077033    3584      3794     4416     4637     4652     6922L
PSHX.    077030    3171      3184     3201     3218     3235     3249     6921L
PSHY     077041    3628      3791     4223     6933L
PTS      065252    4490      4535     4718L
PUI      076135    2057      2652     6733L    6779
PUI1     076210    6759      6767L
PUI10    076325    6811      6833     6837L
PUI2     076230    6763      6775E
PUI3     076236    6742      6783L
PUI5     076240    6769      6791L
PUI6     076244    6797L
PUI7     076261    6792      6808L
PUI8     076274    6817L     6826
PUI9     076312    6823      6830L
QUOTE    000047     348E
RAR      065331    4498      4503     4544     4549     4562     4764L
RAR1     065332    4766L     4776
RAR2     065357    4773      4778L
RCE      077062    6750      6770     6962L
RCX      065312    4346      4590     4606     4746L
REA1     053174    2652L     2672
REA2     053203    2658L     2661     2670
READ     053171    1145      2650E
REPLACE  053233    1116      2681E
RESTART  043124     939E      944     1010     1217     1722     6405    10646
RESTORE  045053    1147      1307L
RETURN   053242    1148      2690L
RIGHT$   057314    3380      3580E
```

```
RIGHT$1 057356   3603    3605L
RIL     077075   4810    5370    6981L   7020
RIL1    077076   6982L   6987    6993
RIL2    077120   6989    6991L
RIL3    077137   6985    7005L
RLF     077141   2024    7018L
RM.CON  000004     46E   1368
RM.HLT  000200     47E   1039    1060    1163    6410    6412
RM.IMM  000000     44E    951     987    1103    1164    1374    6412
RM.STE  000001     45E   1077    2763
RND     061074   3363    3869E
RND1    061116   3876    3878L   3897
RND2    061150   3875    3900L
RNDA    061101   3873E   3898   10599
RNO     077177   4176    4189    6858   7048L
RNP     077206   1399    2304    7068L
RNP1    077236   7076L   7086
RNP2    077273   7082    7084    7090L
RNT     077305   1087    1569    1785   1971    2003    2076    2358    2366    2368    2608    2758    3320
                 4288    4291    5667   6009    6155    7108L
ROMBOOT 030000    611E
RUBOUT  000177    344E
RUN     045155   1117    1361L
RUNMOD  114301    952    1006    1036   1102    1162    1376    6408   10688L
RVU     107315   9123    9322L
S.BAUD  040344    702L
S.BOOTF 041034    757L
S.CAADR 040333    685L
S.CACC  041006    741L
S.CCTAB 040335    686L
S.CDB   040343    699L
S.CFWA  040352    709L
S.CODE  041007    742L
S.CONFL 040332    683L
S.CONTY 040327    670L
S.CONWI 040331    676L   1508
S.CSLMD 040326    659L    669     672     675     682     950
S.CUSOR 040330    673L
S.DATC  040310    641L
S.DATE  040277    640L
S.DCS   041033    755L
S.DDDTA 040366    720L
S.DDGRP 040364    717L
S.DDLDA 040360    715L
S.DDLEN 040362    716L
S.DDOPC 040370    721L
S.DFWA  040354    710L
S.DIREA 041016    749L
S.DLINK 040346    707L
S.FASER 041013    748L
S.FCI   041021    750L
S.GRT0  024000    607E
S.GRT1  025000    608E
S.GRT2  026000    609E
S.GUP   041027    752L
S.HIMEM 040316    643L
S.INT   040343    621L    695
S.JUMPS 041010    746L
```

```
S.MOUNT 041032    754L
S.OFWA  040350    708L    1539
S.OMAX  040324    649L    1553    6182
S.OSN   041004    737L
S.OVLE  041000    734L
S.OVLFL 040371    730L
S.OVLS  040376    733L
S.OVSTK 041035    762L
S.RFWA  040356    711L
S.SCI   041024    751L
S.SCR   041120    800L   10585
S.SDD   041010    747L
S.SOVR  041146    623L     625
S.SSN   041002    736L
S.SYSM  040320    645L    1545    6175
S.TIME  040312    642L
S.UCSF  040372    731L
S.UCSL  040374    732L
S.USRM  040322    647L    6195
S.VAL   040277    620L     638
SAVE    053302    1118    2720E
SAVE1   053324    2684    2731L
SCR.    044360    1261L   2300    10611
SCRA    077320    1261    5149    7068    7126L
SCRATCH 044351    1119    1257E
SEG     061170    3364    3919L
SERROR  070223    5137E   6200
SES     077342    1154    1585    7145L   7148    7282
SFS     077362    1750    7172L
SFS.    077352    2217    7166L
SFS0    077375    7170    7177L
SFS1    100025    7187    7198L   7221
SFS2    100060    7211    7215L
SFS3    100073    7200    7225L
SFS4    100074    7214    7229L
SFS5    100075    7230L   7253
SFS6    100120    7209    7250L
SGN     061205    3365    3933L
SIN     064117    3366    4484E
SOB     100126    1939    2852    2920    6125    7262L   7268
SOB1    100137    7264    7267L
SPACE   112207    2636   10560L
SPE     107215    8586    8719    8737    8809    8837    9022    9226L
SPEX    107225    9229    9235L
SQR     063360    3368    4414E
SQRT1   064011    4435L
SQRT2   064015    4431    4438L
SQRT3   064112    4421    4470L
SRA     100143    1186    1900    2324    7280L
SRS     107231    6355    8654    9246E
SRS.    107232    8690    9248L
SRS..   107233    3532    9249L   9311
STACK   042200     627E     943
STACKL  001032     625E
START   043106     929E    1891
STATE   114302   10689L
STEP    053356    1120    2752L
STEP1   053374    2762L    2769
```

```
STO     106250  9003L   9418
STOP    054030  1158    2780L
STR$    061231  3369    3947L
STR$1   061257  3671    3958L
STRLEN  112112  10487L
STRTAB  112107  1283    5422    5433    5436    5813    6307    10485L
STRTI   112145  5817    6072    10545L
STRVI   112143  1286    5811    10544L
STX     106245  6373    8702    8707    8812    8985    8987    9002L   9211
SUB     105172  4329    4733    8721L
SYDD    040130  617E
SYMTAB  112063  1284    1342    1351    1625    1652    1696    1721    5767    5788    5866    5880    5898
                6470    6565    6568    10414L
SYSCALL 000377  529E    932     935     965     1235    1250    1821    2012    2796    2811    3488    5154
                5632    6199    6400    6403    6986    7579    7844    7952    8019    8031    8123    8178    8191
                8195    9781    9931    10092   10125   10572   10580   10604   10607   10652   10654
T.CHA   103204  7843    7951    8047    8075    8122    8145L   8146    8148    8150    8152    8154    8156
T.FLG   103205  7629    7801    8147L
T.FWA   103206  7833    7941    8111    8149L
T.LIM   103212  7642    7754    8115    8135    8137    8153L
T.LWA   103214  7812    7836    7918    7944    8114    8155L
T.PTR   103210  7640    7703    7711    7756    7766    7810    7834    7878    7920    7942    8112    8151L
TAB     000011  349E    7265    9832
TAN     064243  3371    4534E
TAN1    064321  4551    4561L
TAN2    064347  4565    4573L
TANA    065016  4563    4588E
TBL1    112000  10015L  10021
TBL2    112016  10013   10025L
TBL3    112070  10018   10029L
TCS     100200  2001    2566    7309E
TCV     107260  8811    8982    9175    9210    9288E
TDI     100206  1215    1844    6418    7323L
TDI.    047264  1834    1844L   2774
TLEN    000012  8046    8156E
TSTTAB  112114  5819    6069    6304    10496L
TXTF1   062363  4248L   4283
TXTF2   062366  4254L
TXTF3   063013  4262    4268L
TXTFN   062340  3325    4235E
TXTFNA  063041  4273    4280E
TXTTAB  112056  1616    5197    5222    5224    7127    10325L
UNFREZ  054041  1149    2792E
UNFREZA 054057  1870    2795    2799L
UNLOCK  051176  1150    2204L
UNSAVE  054065  1151    2807E
UNSAVE1 054103  2794    2808    2819L
UO.CLK  000001  285E    1192    1193    1194
UO.RDU  000002  284E    1193
UO.HLT  000200  282E    1192    1193    1194
UO.NER  000100  283E    1192
USERFWA 042200  628E    845     847     848     1874    1882    1889
VAL     061270  3381    3971L
VAR2    055117  3054    3060L
VAR4    055136  3073L   3110
VAR5    055220  3101    3114L
VARIAB  055104  3050E   3298
VARTAB  055107  3052L   6493
```

VERS	000026	527E	10574							
WEL	100225	2581	2631	7340L						
WELA	100241	7343	7347L							
WLF	100242	2162	7360L							
WLF1	100251	2572	2638	7311	7345	7372L				
WLF2	100301	7383	7393L							
WRNTAB	112102	1295	6900	6903	6906	6945	10459L			
XCY	100317	3783	3803	3809	4192	4207	4778	7409L		
XCY1	100332	7415L	7423							
XPOLYQ	065142	4575	4608	4637L						
ZERO	114345	1373	1885	2173	2743	5375	7092	10628	10629	10699L
ZRO	100351	5441	7439L	7445						

13574 BYTES FREE