```
4  ***     BASIC - *WINTEK* BASIC INTERPRETER.
5  *
6  *          J. G. L.,  09/76, FOR *WINTEK* CORPORATION.
7  *
8  *       H. W. S       ; 12/77, FOR HEATH CO.
9  *
10 *          J. G. L.,  1/78, FOR HEATH COMPANY
11 *
12 *             G. C.,     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/
24 *               110.X6.00                                  /80.06.GC/

26 ***     COPYRIGHT 09/1976, *WINTEK* CORPORATION, LAFAYETTE, IND.
27 *
28 *       COPYRIGHT 12/1977, 05/1979 HEATH CO.
29 *
30 *               HEATH CO.
31 *               BENTON HARBOR, MI
32 *               49022
33 *
```

```
                        37  ****    ASSEMBLY CONSTANTS.
                        38
000.005                 39  CHANMAX EQU     5               MAXIMUM CHANNEL # = 5


                        41  **      RUN MODE FLAGS.
                        42  *
                        43  *       THESE ARE SET IN *RUNMODE*.
                        44
000.000                 45  RM.IMM  EQU     0               IMMEDIATE MODE
000.001                 46  RM.STE  EQU     01Q             STEP MODE
000.004                 47  RM.CON  EQU     04Q             CONTINUOUS MODE
000.200                 48  RM.HLT  EQU     200Q            HALT EXECUTION


                        50  **      MACHINE INSTRUCTIONS.
                        51  *
                        52
000.200                 53  MI.ADDB EQU     200Q            ADD B
000.303                 54  MI.JMP  EQU     303Q            JMP
000.077                 55  MI.CMC  EQU     077Q            CMC
000.072                 56  MI.LDA  EQU     072Q            LDA
000.076                 57  MI.MVIA EQU     076Q            MVI A
000.323                 58  MI.OUT  EQU     323Q            OUT
000.220                 59  MI.SUBB EQU     220Q            SUB B
000.000                 60  MI.NOP  EQU     0               NOP
000.311                 61  MI.RET  EQU     311Q            RET
000.333                 62  MI.IN   EQU     333Q            IN
000.041                 63  MI.LXIH EQU     041Q            LXI  H
000.021                 64  MI.LXID EQU     021Q            LXI  D
000.001                 65  MI.LXIB EQU     001Q            LXI  B


                        67  **      THE CT. SYMBOLS DEFINE INDEXED OF TOKENS.
                        68  *
                        69
                        70
                        71  **      CHARACTER TYPES.
                        72
000.000                 73          ORG     0
                        74
000.000                 75  CT.FIN  DS      1               00 OR ;
000.001                 76  CT.ALP  DS      1               ALPHABETIC
000.002                 77  CT.NUM  DS      1               NUMERIC
000.003                 78  CT.SEP  DS      1               UNSPECIFIED SEPERATOR
                        79
                        80  *       THE FOLLOWING ARE NOT COMPRESSED IN THE TEXT INTO THESE TOKENS,
                        81  *       BUT THE VARIOUS SCANNER ROUTINES RETURN THESE VALUES.
                        82
000.004                 83          ERRMI   10Q-*
000.004                 84          DS      10Q-*
000.010                 85          DS      1               PLACE HOLDER TO POSITION CT.EQ
000.000                 86          ERRNZ   *-011Q          REQUIRED FOR COMPARE PROCESSING
```

```
000.011                87  CT.EQ   DS      1               = 1
000.012                88  CT.GT   DS      1               > 2
000.013                89  CT.GE   DS      1               >= 3
000.014                90  CT.LT   DS      1               < 4
000.015                91  CT.LE   DS      1               <= 5
000.016                92  CT.NE   DS      1               <> 6
                       93
000.017                94  CT.PAL  DS      1               (
000.020                95  CT.PAR  DS      1               )
000.021                96  CT.PL   DS      1               +
000.022                97  CT.MI   DS      1               -
000.023                98  CT.MU   DS      1               *
000.024                99  CT.DI   DS      1               /
000.025               100  CT.EX   DS      1               ^
000.026               101  CT.CMA  DS      1               ,
000.027               102  CT.SEM  DS      1               ;
000.030               103  CT.QUO  DS      1               "
000.031               104  CT.PS   DS      1               #
                      105
                      106
                      107  **      BASIC VERBS AND KEYWORDS.
                      108
                      109
000.200               110          ORG     200Q
000.200               111  CT.BLD  DS      1               BUILD   (MUST BE FIRST)
000.201               112  CT.BYE  DS      1               BYE
000.202               113  CT.CNT  DS      1               CONTINUE
000.203               114  CT.DEL  DS      1               DELETE
000.204               115  CT.LIS  DS      1               LIST
000.205               116  CT.REP  DS      1               REPLACE
000.206               117  CT.RUN  DS      1               RUN
000.207               118  CT.SAV  DS      1               SAVE
000.210               119  CT.SCR  DS      1               SCRATCH
000.211               120  CT.STE  DS      1               STEP
                      121
000.212               122  CT.RUA  EQU     *               FOLLOWING COMMANDS 'RUN USAGE ALLOWED'
                      123
000.212               124  CT.SYE  DS      1               SYNTAX ERROR
000.213               125  CT.CHA  DS      1               CHAIN
000.214               126  CT.CLR  DS      1               CLEAR
000.215               127  CT.CLO  DS      1               CLOSE
000.216               128  CT.CTL  DS      1               CNTRL
000.217               129  CT.DIM  DS      1               DIM
000.220               130  CT.FN   DS      1               FN
000.221               131  CT.FOR  DS      1               FOR
000.222               132  CT.FRE  DS      1               FREE
000.223               133  CT.FRZ  DS      1               FREEZE
000.224               134  CT.GOS  DS      1               GOSUB
000.225               135  CT.GOT  DS      1               GOTO
000.226               136  CT.IF   DS      1               IF
000.227               137  CT.LET  DS      1               LET
000.230               138  CT.LCK  DS      1               LOCK
000.231               139  CT.NXT  DS      1               NEXT
000.232               140  CT.OLD  DS      1               OLD
000.233               141  CT.ON   DS      1               ON
000.234               142  CT.OPE  DS      1               OPEN
```

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

```
                        199
000.310                 200  CT.AND  DS      1              AND
000.311                 201  CT.AS   DS      1              AS
000.312                 202  CT.FIL  DS      1              FILE
000.313                 203  CT.WRI  DS      1              WRITE
000.314                 204  CT.NOT  DS      1              NOT
000.315                 205  CT.OR   DS      1              OR
000.316                 206  CT.THN  DS      1              THEN
000.317                 207  CT.TO   DS      1              TO
                        208
                        209  *       FUNCTION DEFINITIONS
                        210
000.320                 211  CT.FCN  EQU     *              ALL FUNCTIONS FOLLOW
                        212
000.320                 213  CT.ABS  DS      1              ABS(
000.321                 214  CT.ATN  DS      1              ATN(
000.322                 215  CT.CHR  DS      1              CHR$(
000.323                 216  CT.CIN  DS      1              CIN(
000.324                 217  CT.COS  DS      1              COS(
000.325                 218  CT.EXP  DS      1              EXP(
000.326                 219  CT.INT  DS      1              INT(
000.327                 220  CT.LNO  DS      1              LNO(
000.330                 221  CT.LOG  DS      1              LOG(
000.331                 222  CT.MAX  DS      1              MAX(
000.332                 223  CT.MIN  DS      1              MIN(
000.333                 224  CT.PAD  DS      1              PAD(
000.334                 225  CT.PEK  DS      1              PEEK(
000.335                 226  CT.PIN  DS      1              PIN(
000.336                 227  CT.POS  DS      1              POS(
000.337                 228  CT.RND  DS      1              RND(
000.340                 229  CT.SEG  DS      1              SEG(
000.341                 230  CT.SGN  DS      1              SGN(
000.342                 231  CT.SIN  DS      1              SIN(
000.343                 232  CT.SPC  DS      1              SPC(
000.344                 233  CT.SQR  DS      1              SQR(
000.345                 234  CT.STR  DS      1              STR$(
000.346                 235  CT.TAB  DS      1              TAB(
000.347                 236  CT.TAN  DS      1              TAN(
                        237
                        238  *       THE FOLLOWING FUNCTIONS REQUIRE STRING ARGUMENTS.
                        239
000.350                 240  CT.SRA  EQU     *              REQUIRE STRING ARGUMENTS
000.350                 241  CT.ASC  DS      1              ASC(
000.351                 242  CT.LEF  DS      1              LEFT$(
000.352                 243  CT.LEN  DS      1              LEN(
000.353                 244  CT.MAT  DS      1              MATCH$(
000.354                 245  CT.MID  DS      1              MID$(
000.355                 246  CT.RIG  DS      1              RIGHT$(
000.356                 247  CT.VAL  DS      1              VAL(
000.357                 248  CT.FNM  DS      0              MAX FUNCTION VALUE
```

```
000.357                 250         XTEXT   MTR
```

```
                        253X **     MTR - PAM/8 EQUIVALENCES.
                        254X *
                        255X *      THIS DECK CONTAINS SYMBOLIC DEFINITIONS USED TO
                        256X *      MAKE USE OF THE PAM/8 CODE AND CONTROL BYTES.


                        258X **     IO PORTS
                        259X
    000.360             260X IP.PAD EQU     360Q            PAD INPUT PORT
    000.360             261X OP.CTL EQU     360Q            CONTROL OUTPUT PORT
    000.360             262X OP.DIG EQU     360Q            DIGIT SELECT OUTPUT PORT
    000.361             263X OP.SEG EQU     361Q            SEGMENT SELECT OUTPUT PORT
    000.362             264X IP.CON EQU     362Q            H-88/H-89/HA-8-8 Configuration  /80.07.sc/
    000.362             265X OP2.CTL EQU    362Q            H-88/H-89/HA-8-8 Control Port   /80.07.sc/


                        267X **     FRONT PANEL CONTROL BITS.                               /80.07.sc/
                        268X *
                        269X *      CB.* set in OP.CTL
                        270X *      CB2.* set in OP2.CTL
                        271X *
                        272X
    000.020             273X CB.SSI EQU     00010000B       SINGLE STEP INTERRUPT
    000.040             274X CB.MTL EQU     00100000B       MONITOR LIGHT
    000.100             275X CB.CLI EQU     01000000B       CLOCK INTERRUPT ENABLE
    000.200             276X CB.SPK EQU     10000000B       SPEAKER ENABLE
                        277X
    000.001             278X CB2.SSI EQU    00000001B       Single Step Interrupt
    000.002             279X CB2.CLI EQU    00000010B       Clock Interrupt Enable
    000.040             280X CB2.ORG EQU    00100000B       ORG 0 Select
    000.100             281X CB2.SID EQU    01000000B       Side 1 Select


                        283X **     Secondary Control Bits
                        284X


                        286X **     MONITOR MODE FLAGS.
                        287X
    000.000             288X DM.MR  EQU     0               MEMORY READ
    000.001             289X DM.MW  EQU     1               MEMORY WRITE
    000.002             290X DM.RR  EQU     2               REGISTER READ
    000.003             291X DM.RW  EQU     3               REGISTER WRITE
```

```
                        293X **     USER OPTION BITS.
                        294X *
                        295X *      THESE BITS ARE SET IN CELL .MFLAG.
                        296X
    000.200             297X UO.HLT EQU     10000000B       DISABLE HALT PROCESSING
    000.100             298X UO.NFR EQU     CB.CLI          NO REFRESH OF FRONT PANEL
    000.002             299X UO.DDU EQU     00000010B       DISABLE DISPLAY UPDATE
    000.001             300X UO.CLK EQU     00000001B       ALLOW PRIVATE INTERRUPT PROCESSING


                        302X **     MONITOR IDENTIFICATION FLAGS
                        303X *
                        304X *      THESE BYTES IDENTIFY THE ROM MONITOR.
                        305X *      THEY ARE THE VARIOUS VALUES OF LOCATION .IDENT
                        306X
    000.021             307X M.PAM8 EQU     021Q            'LXI' INSTRUCTION AT 000.000 IN PAM-8
    000.303             308X M.FOX  EQU     303Q            'JMP' INSTRUCTION AT 000,000 IN FOX ROM


                        310X **     Configuration Flags                                 /80.07.sc/
                        311X *
                        312X *      These bits are read in IP.CON.
                        313X *
                        314X
    000.003             315X CN.174M EQU    00000011B       Port 174Q Device-Type Mask
    000.014             316X CN.170M EQU    00001100B       Port 170Q Device-Type Mask
    000.020             317X CN.PRI EQU     00010000B       Primary/Secondary:  1=>Primary == 170Q
    000.040             318X CN.MEM EQU     00100000B       Memory Test/Normal Switch:  0=>Test; 1=>Normal
    000.100             319X CN.BAU EQU     01000000B       Baud Rate:  0=>9600; 1=>19,200
    000.200             320X CN.ABO EQU     10000000B       Auto-Boot:  1=>Auto-Boot
                        321X
    000.000             322X CND.H17 EQU    00B             H-17 Disk,              Valid only in CN.174M
    000.000             323X CND.NDI EQU    00B             No Device Installed,    Valid only in CN.170M
    000.001             324X CND.H47 EQU    01B             H-47 Disk


                        326X **     ROUTINE ENTRY POINTS.
                        327X *
                        328X
    000.000             329X .IDENT EQU     0000A           IDENTIFICATION LOCATION
    000.053             330X .DLY   EQU     0053A           DELAY
    001.267             331X .LOAD  EQU     1267A           TAPE LOAD
    001.374             332X .DUMP  EQU     1374A           TAPE DUMP
    002.136             333X .ALARM EQU     2136A           ALARM ROUTINE
    002.140             334X .HORN  EQU     2140A           HORN
    002.172             335X .CTC   EQU     2172A           CHECK TAPE CHECKSUM
    002.205             336X .TPERR EQU     2205A           TAPE ERROR ROUTINE
    002.264             337X .PCHL  EQU     2264A           PCHL INSTRUCTION
    002.265             338X .SRS   EQU     2265A           SCAN RECORD START
    002.325             339X .RNP   EQU     2325A           READ NEXT PAIR
    002.331             340X .RNB   EQU     2331A           READ NEXT BYTE
```

```
002.347                 341X .CRC    EQU     2347A          CRC-16 CALCULATOR
003.017                 342X .WNP    EQU     3017A          WRITE NEXT PAIR
003.024                 343X .WNB    EQU     3024A          WRITE NEXT BYTE
003.122                 344X .DOD    EQU     3122A          DECODE FOR OCTAL DISPLAY
003.260                 345X .RCK    EQU     3260A          READ CONSOLE KEYSET
003.356                 346X .DODA   EQU     3356A          SEGMENT CODE TABLE


                        348X **      RAM CELLS USED BY H8MTR.
                        349X *
                        350X
040.000                 351X .START  EQU     40000A         START DUMP ADDRESS
040.002                 352X .IOWRK  EQU     40002A         IN OR OUT INSTRUCTION
040.005                 353X .REGI   EQU     40005A         DISPLAYED REGISTER INDEX
040.006                 354X .DSPROT EQU     40006A         PERIOD FLAG BYTE
040.007                 355X .DSPMOD EQU     40007A         DISPLAY MODE
040.010                 356X .MFLAG  EQU     40010A         USER OPTION BYTE
040.011                 357X .CTLFLG EQU     40011A         PANEL CONTROL BYTE
040.013                 358X .ALEDS  EQU     40013A         ABUSS LEDS
040.021                 359X .DLEDS  EQU     40021A         DBUSS LEDS
040.024                 360X .ABUSS  EQU     40024A         ABUSS REGISTER
040.027                 361X .CRCSUM EQU     40027A         CRCSUM WORD
040.031                 362X .TPERRX EQU     40031A         TAPE ERROR EXIT VECTOR
040.033                 363X .TICCNT EQU     40033A         CLOCK TICK COUNTER
040.035                 364X .REGPTR EQU     40035A         REGISTER POINTER
040.037                 365X .UIVEC  EQU     40037A         USER INTERRUPT VECTORS
040.064                 366X .NMIRET EQU     40064A         H88/H89 NMI Return Address      /80.07.sc/
040.066                 367X .CTL2FL EQU     40066A         OP2.CTL Control Byte            /80.07.sc/
000.357                 368          XTEXT   ASCII


                        370X **      ASCII CHARACTER EQUIVALENCES.
                        371X
000.015                 372X CR      EQU     13             CARRIAGE RETURN
000.012                 373X LF      EQU     10             LINE FEED
000.200                 374X NULL    EQU     200Q           PAD CHARACTER
000.000                 375X NUL2    EQU     0
000.007                 376X BELL    EQU     7              BELL CHARACTER
000.177                 377X RUBOUT  EQU     177Q
000.010                 378X BKSP    EQU     10Q            CTL-H
000.026                 379X C.SYN   EQU     26Q            SYNC
000.002                 380X C.STX   EQU     2              STX
000.047                 381X QUOTE   EQU     47Q
000.011                 382X TAB     EQU     11Q
000.033                 383X ESC     EQU     33Q
000.012                 384X NL      EQU     12Q            NEW LINE (HDOS SYSTEMS)
000.212                 385X ENL     EQU     NL+200Q        NL + END-OF-LINE-FLAG
000.014                 386X FF      EQU     14Q            FORM FEED
000.001                 387X CTLA    EQU     01Q            CTL-A
000.002                 388X CTLB    EQU     02Q            CTL-B
000.003                 389X CTLC    EQU     03Q            CTL-C
000.004                 390X CTLD    EQU     04Q            CTL-D
000.017                 391X CTLO    EQU     17Q            CTL-O
000.020                 392X CTLP    EQU     20Q            CTL-P
000.021                 393X CTLQ    EQU     21Q            CTL-Q
```

```
000.023                394X CTLS   EQU     23Q             CTL-S
000.032                395X CTLZ   EQU     32Q             CTL-Z
000.357                396         XTEXT   BECDEF

                       398X **     BASIC ERROR CODE DEFINITIONS.
                       399X
                       400X
000.200                401X        ORG     128             USE 128 AND ABOVE
000.200                402X BEC.CC DS      1               CONTROL-C HIT
000.201                403X BEC.CB DS      1               CONTROL-B HIT
000.202                404X BEC.DE DS      1               DATA EXHAUSTED
000.203                405X BEC.DO DS      1               /0
000.204                406X BEC.IN DS      1               ILLEGAL NUMBER
000.205                407X BEC.IU DS      1               ILLEGAL USAGE
000.206                408X BEC.LK DS      1               DATA LOCK ENGAGED
000.207                409X BEC.NV DS      1               NEXT VARIABLE MISSING
000.210                410X BEC.OV DS      1               NUMERIC OVERFLOW
000.211                411X BEC.RE DS      1               RETURN ERROR
000.212                412X BEC.SL DS      1               STRING LENGTH
000.213                413X BEC.SN DS      1               STATEMENT NUMBER
000.214                414X BEC.SY DS      1               SYNTAX ERROR
000.215                415X BEC.TC DS      1               TYPE CONFLICT
000.216                416X BEC.TO DS      1               TABLE OVERFLOW
000.217                417X BEC.SR DS      1               SUBSCRIPT RANGE
000.220                418X BEC.SC DS      1               SUBSCRIPT COUNT
000.221                419X BEC.ND DS      1               NOT DIMENSIONED
000.222                420X BEC.IC DS      1               ILLEGAL CHARACTER
000.223                421X BEC.UD DS      1               UNDEFINED FUNCTION
000.224                422X BEC.EN DS      1               END
000.225                423X BEC.ST DS      1               STOP
000.226                424X BEC.FAE DS     1               FILE ALREADY EXISTS
000.227                425X BEC.ILF DS     1               ILLEGAL FILE NAME
000.230                426X BEC.AC DS      1               ILLEGAL ARGUMENT COUNT
000.231                427X BEC.FNO DS     1               FILE NOT OPEN
000.232                428X BEC.LTL DS     1               LINE TOO LONG
000.233                429X BEC.CIU DS     1               CHANNEL IN USE
000.234                430         XTEXT   ECDEF

                       432X **     ERROR CODE DEFINITIONS.
                       433X
000.000                434X        ORG     0
000.000                435X        DS      1               NO ERROR #0
000.001                436X EC.EOF DS      1               END OF FILE
000.002                437X EC.EOM DS      1               END OF MEDIA
000.003                438X EC.ILC DS      1               ILLEGAL SYSCALL CODE
000.004                439X EC.CNA DS      1               CHANNEL NOT AVAILABLE
000.005                440X EC.DNS DS      1               DEVICE NOT SUITABLE
000.006                441X EC.IDN DS      1               ILLEGAL DEVICE NAME
000.007                442X EC.IFN DS      1               ILLEGAL FILE NAME
000.010                443X EC.NRD DS      1               NO ROOM FOR DEVICE DRIVER
000.011                444X EC.FNO DS      1               CHANNEL NOT OPEN
```

```
000.012                 445X EC.ILR  DS      1               ILLEGAL REQUEST
000.013                 446X EC.FUC  DS      1               FILE USAGE CONFLICT
000.014                 447X EC.FNF  DS      1               FILE NAME NOT FOUND
000.015                 448X EC.UND  DS      1               UNKNOWN DEVICE
000.016                 449X EC.ICN  DS      1               ILLEGAL CHANNEL NUMBER
000.017                 450X EC.DIF  DS      1               DIRECTORY FULL
000.020                 451X EC.IFC  DS      1               ILLEGAL FILE CONTENTS
000.021                 452X EC.NEM  DS      1               NOT ENOUGH MEMORY
000.022                 453X EC.RF   DS      1               READ FAILURE
000.023                 454X EC.WF   DS      1               WRITE FAILURE
000.024                 455X EC.WPV  DS      1               WRITE PROTECTION VIOLATION
000.025                 456X EC.WP   DS      1               DISK WRITE PROTECTED
000.026                 457X EC.FAP  DS      1               FILE ALREADY PRESENT
000.027                 458X EC.DDA  DS      1               DEVICE DRIVER ABORT
000.030                 459X EC.FL   DS      1               FILE LOCKED
000.031                 460X EC.FAO  DS      1               FILE ALREADY OPEN
000.032                 461X EC.IS   DS      1               ILLEGAL SWITCH
000.033                 462X EC.UUN  DS      1               UNKNOWN UNIT NUMBER
000.034                 463X EC.FNR  DS      1               FILE NAME REQUIRED
000.035                 464X EC.DIW  DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 465X EC.UNA  DS      1               UNIT NOT AVAILABLE
000.037                 466X EC.ILV  DS      1               ILLEGAL VALUE
000.040                 467X EC.ILO  DS      1               ILLEGAL OPTION
000.041                 468X EC.VPM  DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 469X EC.NVM  DS      1               NO VOLUME PRESENTLY MOUNTED
000.043                 470X EC.FOD  DS      1               FILE OPEN ON DEVICE
000.044                 471X EC.NPM  DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 472X EC.DNI  DS      1               DISK NOT INITIALIZED
000.046                 473X EC.DNR  DS      1               DISK IS NOT READABLE
000.047                 474X EC.DSC  DS      1               DISK STRUCTURE IS CORRUPT
000.050                 475X EC.NCV  DS      1               NOT CORRECT VERSION OF HDOS
000.051                 476X EC.NOS  DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 477X EC.IOI  DS      1               ILLEGAL OVERLAY INDEX
000.053                 478X EC.OTL  DS      1               OVERLAY TO LARGE
000.054                 479          XTEXT   FBDEF

                        481X **      FILE BLOCK DEFINITIONS.
                        482X
000.000                 483X         ORG     0
000.000                 484X FB.CHA  DS      1               CHANNEL NUMBER
000.001                 485X FB.FLG  DS      1               FLAGS
000.002                 486X FB.FWA  DS      2               BUFFER FWA
000.004                 487X FB.PTR  DS      2               BUFFER POINTER
000.006                 488X FB.LIM  DS      2               LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 489X FB.LWA  DS      2               LWA OF BUFFER
000.012                 490X FB.NAM  DS      4+8+4+1         NAME OF FILE
000.021                 491X FB.NAML EQU     *-FB.NAM
000.033                 492X FBENL   EQU     *               ENTRY LENGTH
000.033                 493          XTEXT   DIRDEF
```

```
                        495X **     DIRECTORY ENTRY FORMAT.
                        496X
000.000                 497X        ORG     0
                        498X
                        499X
000.377                 500X DF.EMP EQU     377Q            FLAGS ENTRY EMPTY
000.376                 501X DF.CLR EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        502X
000.000                 503X DIR.NAM DS     8               NAME
000.010                 504X DIR.EXT DS     3               EXTENSION
000.013                 505X DIR.PRO DS     1               PROJECT
000.014                 506X DIR.VER DS     1               VERSION
000.015                 507X DIRIDL EQU     *               FILE IDENTIFICATION LENGTH
                        508X
000.015                 509X DIR.CLU DS     1               CLUSTER FACTOR
000.016                 510X DIR.FLG DS     1               FLAGS
000.017                 511X        DS      1               RESERVED
000.020                 512X DIR.FGN DS     1               FIRST GROUP NUMBER
000.021                 513X DIR.LGN DS     1               LAST GROUP NUMBER
000.022                 514X DIR.LSI DS     1               LAST SECTOR INDEX (IN LAST GROUP)
000.023                 515X DIR.CRD DS     2               CREATION DATE
000.025                 516X DIR.ALD DS     2               LAST ALTERATION DATE
                        517X
000.027                 518X DIRELEN EQU    *               DIRECTORY ENTRY LENGTH
000.027                 519         XTEXT   IOCDEF


                        521X **     I/O CHANNEL DEFINITIONS.
                        522X
000.000                 523X        ORG     0
                        524X
000.000                 525X IOC.LNK DS     2               ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 526X IOC.DDA DS     2               THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        527X
000.004                 528X IOC.FLG DS     1               FILE TYPE FLAGS
000.001                 529X FT.DD  EQU     00000001B       =1 IF DIRECTORY DEVICE
000.002                 530X FT.OR  EQU     00000010B       =1 IF OPEN FOR READ
000.004                 531X FT.OW  EQU     00000100B       =1 IF OPEN FOR WRITE
000.010                 532X FT.OU  EQU     00001000B       =1 IF OPEN FOR UPDATE
000.020                 533X FT.OC  EQU     00010000B       =1 IF OPEN FOR CHARACTER MODE   /80.02.GC/
000.003                 534X IOC.SQL EQU    *-IOC.DDA       LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        535X
000.005                 536X IOC.GRT DS     2               ADDRESS OF GROUP RESERVATION TABLE
000.007                 537X IOC.SPG DS     1               SECTORS PER GROUP, THIS DEVICE
000.010                 538X IOC.CGN DS     1               CURRENT GROUP NUMBER
000.011                 539X IOC.CSI DS     1               CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 540X IOC.LGN DS     1               LAST GROUP NUMBER
000.013                 541X IOC.LSI DS     1               LAST SECTOR INDEX (IN LAST GROUP)
000.010                 542X IOC.DRL EQU    *-IOC.FLG       LENGTH OF INFO NORMALLY COPIED BACK TO
                        543X *                              THE CHANNEL TABLE
000.014                 544X IOC.DTA DS     2               DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 545X IOC.DES DS     2               SECTOR NUMBER OF DIRECTORY ENTRY
000.020                 546X IOC.DEV DS     2               DEVICE CODE
000.022                 547X IOC.UNI DS     1               UNIT NUMBER (0-9)
000.021                 548X IOC.DIL EQU    *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
```

```
                        549X
000.023                 550X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                        551X
000.052                 552X IOCELEN EQU     *               IOC ENTRY LENGTH
                        553X
000.001                 554X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)
000.052                 555          XTEXT   HOSDEF


                        557X **      HOSDEF - DEFINE HOS PARAMETER.
                        558X *
                        559X
                        560X
000.040                 561X VERS    EQU     2*16+0          VERSION 2.0
                        562X
000.377                 563X SYSCALL EQU     377Q            SYSCALL INSTRUCTION
                        564X
                        565X
000.000                 566X         ORG     0
                        567X
                        568X *       RESIDENT FUNCTIONS
                        569X
000.000                 570X .EXIT   DS      1               EXIT (MUST BE FIRST)
000.001                 571X .SCIN   DS      1               SCIN
000.002                 572X .SCOUT  DS      1               SCOUT
000.003                 573X .PRINT  DS      1               PRINT
000.004                 574X .READ   DS      1               READ
000.005                 575X .WRITE  DS      1               WRITE
000.006                 576X .CONSL  DS      1               SET/CLEAR CONSOLE OPTIONS
000.007                 577X .CLRCO  DS      1               CLEAR CONSOLE BUFFER
000.010                 578X .LOADO  DS      1               LOAD AN OVERLAY
000.011                 579X .VERS   DS      1               RETURN HDOS VERSION NUMBER
000.012                 580X .SYSRES DS      1               PRECEDING FUNCTIONS ARE RESIDENT
                        581X
                        582X
                        583X *       *HDOSOVL0.SYS*  FUNCTIONS
                        584X
000.040                 585X         ORG     40A
                        586X
000.040                 587X .LINK   DS      1               LINK  (MUST BE FIRST)
000.041                 588X .CTLC   DS      1               CTL-C
000.042                 589X .OPENR  DS      1               OPENR
000.043                 590X .OPENW  DS      1               OPENW
000.044                 591X .OPENU  DS      1               OPENU
000.045                 592X .OPENC  DS      1               OPENC
000.046                 593X .CLOSE  DS      1               CLOSE
000.047                 594X .POSIT  DS      1               POSITION
000.050                 595X .DELET  DS      1               DELETE
000.051                 596X .RENAM  DS      1               RENAME
000.052                 597X .SETTP  DS      1               SETTOP
000.053                 598X .DECODE DS      1               NAME DECODE
000.054                 599X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                 600X .CLEAR  DS      1               CLEAR CHAN
000.056                 601X .CLEARA DS      1               CLEAR ALL CHANS
000.057                 602X .ERROR  DS      1               LOOKUP ERROR
```

```
000.060                603X .CHFLG  DS      1               CHANGE FLAGS
000.061                604X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000.062                605X .LOADD  DS      1               LOAD DEVICE DRIVER
000.063                606X .OPEN   DS      1               Parametrized Open
                       607X
                       608X
                       609X *       *HDOSOVL1.SYS*  FUNCTIONS
                       610X
000.200                611X         ORG     200Q
                       612X
000.200                613X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201                614X .DMOUN  DS      1               DISMOUNT
000.202                615X .MONMS  DS      1               MOUNT/NO MESSAGE
000.203                616X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000.204                617X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                618X .CLEAN  DS      1               Clean device
000.206                619X .DAD    DS      1               Dismount All Disks              /80.08.sc/
000.207                620          XTEXT   OVLDEF



                       622X **      OVERLAY TABLE ENTRYS.
                       623X
000.000                624X         ORG     0
                       625X
000.000                626X OVL.COD DS      2               FIRST SECTOR OF OVERLAY CODE
000.002                627X OVL.SIZ DS      2               OVERLAY SIZE
000.004                628X OVL.ENT DS      2               OVERLAY ENTRY POINT
000.006                629X OVL.FLB DS      1               OVERLAY FLAG BYTE
000.007                630X         DS      1               DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010                631X OVL.ENS EQU     *               OVERLAY ENTRY SIZE
                       632X
                       633X *       OVERLAY INDICES
                       634X
000.000                635X         ORG     0
                       636X
000.000                637X OVL0    DS      1
000.001                638X OVL1    DS      1
000.002                639          XTEXT   HOSEQU



                       641X **      HDOS SYSTEM EQUIVALENCES.
                       642X *
                       643X
024.000                644X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
025.000                645X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
026.000                646X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                       647X
030.000                648X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                       649X
040.100                650X         ORG     40100A          FREE SPACE FROM PAM-8
                       651X
040.100                652X         DS      8               JUMP TO SYSTEM EXIT
```

```
040.110 653X D.CON   DS    16          DISK CONSTANTS
040.130 654X SYDD    EQU   *           SYSTEM DISK ENTRY POINT
040.130 655X D.VEC   DS    24*3        SYSTEM ROM ENTRY VECTORS
040.240 656X D.RAM   DS    31          SYSTEM ROM WORK AREA
040.277 657X S.VAL   DS    36          SYSTEM VALUES
040.343 658X S.INT   DS    115         SYSTEM INTERNAL WORK AREAS
041.126 659X         DS    16
041.146 660X S.SOVR  DS    2           STACK OVERFLOW WARNING
041.150 661X         DS    42200A-*    SYSTEM STACK
001.032 662X STACKL  EQU   *-S.SOVR    STACK SIZE
        663X
042.200 664X STACK   EQU   *           LWA+1 SYSTEM STACK
042.200 665X USERFWA EQU   *           USER FWA
042.200 666          XTEXT ESVAL


        668X **      S.VAL - SYSTEM VALUE DEFINTIONS.
        669X *
        670X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
        671X *
        672X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
        673X
        674X
040.277 675X         ORG   S.VAL
        676X
040.277 677X S.DATE  DS    9           SYSTEM DATE (IN ASCII)
040.310 678X S.DATC  DS    2           CODED DATE
040.312 679X S.TIME  DS    4           TIME FROM MIDNIGHT (IN TICS)
040.316 680X S.HIMEM DS    2           HARDWARE HIGH MEMORY ADRESS+1
        681X
040.320 682X S.SYSM  DS    2           FWA RESIDENT SYSTEM
        683X
040.322 684X S.USRM  DS    2           LWA USER MEMORY
        685X
040.324 686X S.OMAX  DS    2           MAX OVERLAY SIZE FOR SYSTEM
        687X
        688X
        689X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
        690X
000.200 691X CSL.ECH EQU   10000000B     SUPPRESS ECHO
000.004 692X CSL.RAW EQU   00000100B     Raw Mode I/O                    /80.09.sc/
000.002 693X CSL.WRP EQU   00000010B     WRAP LINES AT WIDTH
000.001 694X CSL.CHR EQU   00000001B     OPERATE IN CHARACTER MODE
        695X
000.000 696X I.CSLMD EQU   0             S.CSLMD IS FIRST BYTE
040.326 697X S.CSLMD DS    1           CONSOLE MODE
        698X
000.200 699X CTP.BKS EQU   10000000B     TERMINAL PROCESSES BACKSPACES
000.100 700X CTP.FF  EQU   01000000B     Terminal Processes Form-Feed    /80.09.sc/
000.040 701X CTP.MLI EQU   00100000B     MAP LOWER CASE TO UPPER ON INPUT
000.020 702X CTP.MLO EQU   00010000B     MAP LOWER CASE TO UPPER ON OUTPUT
000.010 703X CTP.2SB EQU   00001000B     TERMINAL NEEDS TWO STOP BITS
000.002 704X CTP.BKM EQU   00000010B     MAP BKSP (UPON INPUT) TO RUBOUT
000.001 705X CTP.TAB EQU   00000001B     TERMINAL SUPPORTS TAB CHARACTERS
```

```
                    706X
000.001             707X I.CONTY EQU     1               S.CONTY IS 2ND BYTE
000.000             708X         ERRNZ   *-S.CSLMD-I.CONTY
040.327             709X S.CONTY DS      1               CONSOLE TYPE FLAGS
000.002             710X I.CUSOR EQU     2               S.CUSOR IS 3RD BYTE
000.000             711X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330             712X S.CUSOR DS      1               CURRENT CURSOR POSITION
000.003             713X I.CONWI EQU     3               S.CONWI IS 4TH BYTE
000.000             714X         ERRNZ   *-S.CSLMD-I.CONWI
040.331             715X S.CONWI DS      1               CONSOLE WIDTH
                    716X
000.001             717X CO.FLG  EQU     00000001B       CTL-O FLAG
000.200             718X CS.FLG  EQU     10000000B       CTL-S FLAG
                    719X
000.004             720X I.CONFL EQU     4               S.CONFL IS 5TH BYTE
000.000             721X         ERRNZ   *-S.CSLMD-I.CONFL
040.332             722X S.CONFL DS      1               CONSOLE FLAGS
                    723X
040.333             724X S.CAADR DS      2               ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335             725X S.CCTAB DS      6               ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343             726          XTEXT   ESINT


                    728X **      S.INT - SYSTEM INTERNAL WORKAREA DEFINITIONS.
                    729X *
                    730X *       THESE CELLS ARE REFERENCED BY OVERLAYS AND MAIN CODE, AND
                    731X *       MUST THEREFORE RESIDE IN FIXED LOW MEMORY.
                    732X
                    733X
040.343             734X         ORG     S.INT
                    735X
                    736X **      CONSOLE STATUS FLAGS
                    737X
040.343             738X S.CDB   DS      1               CONSOLE DESCRIPTOR BYTE
000.000             739X CDB.H85 EQU     00000000B
000.001             740X CDB.H84 EQU     00000001B       =0 IF H8-5, =1 IF H8-4
040.344             741X S.BAUD  DS      2               [0-14]  H8-4 BAUD RATE, =0 IF H8-5
                    742X *                               [15]    =1 IF BAUD RATE => 2 STOP BITS
                    743X
                    744X **      TABLE ADDRESS WORDS
                    745X
040.346             746X S.DLINK DS      2               ADDRESS OF DATA IN HDOS CODE
040.350             747X S.OFWA  DS      2               FWA  OVERLAY  TABLE
040.352             748X S.CFWA  DS      2               FWA  CHANNEL  TABLE
040.354             749X S.DFWA  DS      2               FWA  DEVICE   TABLE
040.356             750X S.RFWA  DS      2               FWA  RESIDENT HDOS CODE
                    751X
                    752X **      DEVICE DRIVER DELAYED LOAD FLAGS
                    753X
040.360             754X S.DDLDA DS      2               DRIVER LOAD ADDRESS (HIGH BYTE=0 IF NO LOAD PENDING)
040.362             755X S.DDLEN DS      2               CODE LENGTH IN BYTES
040.364             756X S.DDGRP DS      1               GROUP NUMBER FOR DRIVER
040.365             757X         DS      1               HOLD PLACE
                    758X *S.DDSEC        DS      2               SECTOR NUMBER FOR DRIVER ( * OBSOLETE ! * )
```

```
040.366                 759X S.DDDTA DS      2              DEVICE'S ADDRESS IN DEVLST +DEV.RES
040.370                 760X S.DDOPC DS      1              OPEN OPCODE PENDEDING
                        761X
                        762X **      OVERLAY MANAGEMENT FLAGS
                        763X
000.001                 764X OVL.IN  EQU     00000001B      IN MEMORY
000.002                 765X OVL.RES EQU     00000010B      PERMINANTLY RESIDENT
000.014                 766X OVL.NUM EQU     00001100B      OVERLAY NUMBER MASK
000.200                 767X OVL.UCS EQU     10000000B      USER CODE SWAPPED FOR OVERLAY
                        768X
040.371                 769X S.OVLFL DS      1              OVERLAY FLAG
040.372                 770X S.UCSF  DS      2              FWA SWAPPED USER CODE
040.374                 771X S.UCSL  DS      2              LENGTH SWAPPED USER CODE
040.376                 772X S.OVLS  DS      2              SIZE OF OVERLAY CODE
041.000                 773X S.OVLE  DS      2              ENTRY POINT OF OVERLAY CODE
                        774X
041.002                 775X S.SSN   DS      2              SWAP AREA SECTOR NUMBER
041.004                 776X S.OSN   DS      2              OVERLAY SECTOR NUMBER
                        777X
                        778X *       SYSCALL PROCESSING WORK AREAS
                        779X
041.006                 780X S.CACC  DS      1              (ACC) UPON SYSCALL
041.007                 781X S.CODE  DS      1              SYSCALL INDEX IN PROGRESS
                        782X
                        783X *       JUMPS TO ROUTINES IN RESIDENT HDOS CODE
                        784X
041.010                 785X S.JUMPS DS      0              START OF DUMP VECTORS
041.010                 786X S.SDD   DS      3              JUMP TO STAND-IN DEVICE DRIVER
041.013                 787X S.FASER DS      3              JUMP TO FATSERR (FATAL SYSTEM ERROR)
041.016                 788X S.DIREA DS      3              JUMP TO DIREAD (DISK FILE READ)
041.021                 789X S.FCI   DS      3              JUMP TO FCI (FETCH CHANNEL INFO)
041.024                 790X S.SCI   DS      3              JUMP TO SCI (STORE CHANNEL INFO)
041.027                 791X S.GUP   DS      3              JUMP TO GUP (GET UNIT POINTER)
                        792X
041.032                 793X S.MOUNT DS      1              <>0 IF THE SYSTEM DISK IS MOUNTED
041.033                 794X S.DCS   DS      1              DEFAULT CLUSTER SIZE-1
                        795X
041.034                 796X S.BOOTF DS      1              BOOT FLAGS
000.001                 797X BOOT.P  EQU     00000001B      EXECUTE PROLOGUE UPON BOOTUP
                        798X
                        799X *       STACK VALUE SAVED FOR OVERLAY SYSCALLS
                        800X
041.035                 801X S.OVSTK DS      2              VALUE OF SP UPON SYSCALLS USING OVERLAY
                        802X
041.037                 803X         DS      1              RESERVED


                        805X **      ACTIVE I/O AREA.
                        806X *
                        807X *       THE AIO.XXX AREA CONTAINS INFORMATION ABOUT THE I/O OPERATION
                        808X *       CURRENTLY BEING PERFORMED. THE INFORMATION IS OBTAINED FROM
                        809X *       THE CHANNEL TABLE, AND WILL BE RESTORED THERE WHEN DONE.
                        810X *
                        811X *       NORMALLY, THE AIO.XXX INFORMATION WOULD BE OBTAINED DIRECTLY
```

```
                        812X *      FROM VARIOUS SYSTEM TABLES VIA POINTER REGISTERS. SINCE THE
                        813X *      8080 HAS NO GOOD INDEXED ADDRESSING, THE DATA IS MANUALLY
                        814X *      COPIED INTO THE AIO.XXX CELLS BEFORE PROCESSING, AND
                        815X *      BACKDATED AFTER PROCESSING.
                        816X
041.040                 817X AIO.VEC DS     3           JUMP INSTRUCTION
041.041                 818X AIO.DDA EQU    *-2         DEVICE DRIVER ADDRESS
041.043                 819X AIO.FLG DS     1           FLAG BYTE
041.044                 820X AIO.GRT DS     2           ADDRESS OF GROUP RESERV TABLE
041.046                 821X AIO.SPG DS     1           SECTORS PER GROUP
041.047                 822X AIO.CGN DS     1           CURRENT GROUP NUMBER
041.050                 823X AIO.CSI DS     1           CURRENT SECTOR INDEX
041.051                 824X AIO.LGN DS     1           LAST GROUP NUMBER
041.052                 825X AIO.LSI DS     1           LAST SECTOR INDEX
041.053                 826X AIO.DTA DS     2           DEVICE TABLE ADDRESS
041.055                 827X AIO.DES DS     2           DIRECTORY SECTOR
041.057                 828X AIO.DEV DS     2           DEVICE CODE
041.061                 829X AIO.UNI DS     1           UNIT NUMBER (0-9)
                        830X
041.062                 831X AIO.DIR DS     DIRELEN     DIRECTORY ENTRY
                        832X
041.111                 833X AIO.CNT DS     1           SECTOR COUNT
041.112                 834X AIO.EOM DS     1           END OF MEDIA FLAG
041.113                 835X AIO.EOF DS     1           END OF FILE FLAG
041.114                 836X AIO.TFP DS     2           TEMP FILE POINTERS
041.116                 837X AIO.CHA DS     2           ADDRESS OF CHANNEL BLOCK (IOC.DDA)


041.120                 839X S.BDA   DS     1           Boot Device Address (Setup by ROM) /80.09.sc/
041.121                 840X S.SCR   DS     2           SYSTEM SCRATCH AREA ADDRESS
041.123                 841          XTEXT  MTRDEF


                        843X **     HDOS MONITOR PRIVATE RAM AREA DEFINITIONS.
                        844X
000.000                 845X         ORG    0
000.000                 846X M.SYSM  DS     1           SYSCALL ITTERATION COUNT
000.001                 847X M.SALO  DS     1           STAND-ALONE FLAG
000.002                 848X M.CSLC  DS     1           LINES IN CONSOLE BUFFER
000.003                 849X M.CPRE  DS     1           CONSOLE PREVIOUS CHARACTER
000.004                 850X M.CRUB  DS     1           CONSOLE RUBOUT FLAG
000.005                 851X M.CINT  DS     1           CONSOLE INTERRUPT FLAG
000.006                 852X M.CIN   DS     2           CONSOLE CB IN POINTER
000.010                 853X M.COUT  DS     2           CONSOLE CB OUT POINTER
000.012                 854X M.CFWA  DS     2           CONSOLE CB FWA POINTER
000.014                 855X M.CLWA  DS     2           CONSOLE CB LWA POINTER
000.016                 856X M.CDLY  DS     1           CONSOLE PAD CHARACTER COUNT
000.017                 857X M.CDCA  DS     2           ADDRESS OF CHARACTER BEING PADDED
000.021                 858X M.SUNI  DS     1           System Unit Number              /80.05.sc/
000.022                 859X M.SYDD  DS     2           Address of Raw System Driver    /80.09.sc/
000.024                 860          XTEXT  FILDEF
```

```
                        862X **     FILDEF - FILE TYPE DEFINITIONS.
                        863X *
                        864X *      DB      377Q,FT.XXX
                        865X
                        866X
000.000                 867X FT.ABS EQU     0           ABSOLUTE BINARY
000.001                 868X FT.PIC EQU     1           POSITION INDEPENDANT CODE
000.002                 869X FT.REL EQU     2           RELOCATABLE CODE
000.003                 870X FT.BAC EQU     3           COMPILED BASIC CODE
000.024                 871         XTEXT   ABSDEF


                        873X **     ABS FORMAT EQUIVALENCES.
                        874X
000.000                 875X        ORG     0
                        876X
000.000                 877X ABS.ID DS      1           377Q = BINARY FILE FLAG
000.001                 878X        DS      1           FILE TYPE (FT.ABS)
000.002                 879X ABS.LDA DS     2           LOAD ADDRESS
000.004                 880X ABS.LEN DS     2           LENGTH OF ENTIRE RECORD
000.006                 881X ABS.ENT DS     2           ENTRY POINT
                        882X
000.010                 883X ABS.COD DS     0           CODE STARTS HERE
```

```
                                886
042.170                         887             ORG     USERFWA-ABS.COD
042.170  377 000                888             DB      377Q,FT.ABS             ABS FILE
042.172  200 042                889             DW      USERFWA         LOAD ADDRESS
042.174  267 050                890             DW      LOADL-USERFWA   SIZE
042.176  252 112                891             DW      PRS             ENTRY
                                892
                                894 **          LOW-MEMORY CELLS USED BY BASIC
                                895
042.200                         896             DS      2               ACCX TYPE
042.202                         897 ACCX        DS      4
042.206                         898             DS      2               ACCY TYPE
042.210                         899 ACCY        DS      4
                                900
                                901 **          SPECIAL LEXICAL VARIABLE AREA.
                                902 *
                                903 *           VARIABLES ARE STORED HERE SO THAT BASIC CAN QUICKLY TELL THAT
                                904 *           THEY DONT RESIDE IN THE SYMBOL TABLE BY SIMPLY CHECKING THE
                                905 *           BANK ADDRESS.
                                906
042.214  000                    907             DB      0               TYPE OF LEXC
042.215  000 000 000            908 LEXC        DB      0,0,0,0         SPECIAL LEXICAL UNDEFINED VARIABLE VALUE
                                909
042.221  000                    910             DB      0               TYPE OF LEXB
042.222  000 000 000            911 LEXB        DB      0,0,0,0         SPECIAL LEXICAL CELL FOR NUMERIC LITTERALS
                                912
042.226                         913 LEXLIM      EQU     *               ALL SYMTAB VARIABLES OCCUR IN HIGHER MEM
                                915 **          FBLIST - FILE BLOCK LIST.                               /80.02.GC/
                                916 *
                                917 *           FBLIST CONTAINS TTHE FILE BLOCK FOR ALL POSSIBLE USER
                                918 *           CHANNELS, IN ORDER #1 TO #N.
                                919 *
                                920 *           THE FIRST ENTRY IN FBLIST IS NOT A USER ACCESSABLE FILE, BUT IS
                                921 *           THE SYSTEM'S INTERNAL WORK FILE.
                                922 *           THE 2ND ENTRY IS CHANNEL #2, THE 3RD CHANNEL #3, ETC.
                                923 *
                                924 *           NOTE:   These tables were moved to the front to avoid problems
                                925 *                   when overlays are loaded, etc.                  /80.02.sc/
                                926
042.226  013 115                927 FBUFAD      DW      MTAREA+3        CURRENT CONTENTS OF FILTAB+MT.FWA
                                928
042.230                         929 FBLIST      DS      0
                                930
042.230  000 000                931 FBSCR       DB      0,0             CHANNEL AND STATUS
042.232  000 000 000            932             DW      0,0,0,0+512     USE  *HDOS*  SCRATCH RAM (PRS INITIALIZES IT)
042.242                         933             DS      FB.NAML
                                934
                                935 *           CHANNEL #1
                                936
042.263  001 000                937             DB      1,0             CHANNEL AND STATUS
```

```
042.265  013 115 013   938          DW      MTAREA+3,MTAREA+3,MTAREA+3,MTAREA+3+256
042.275                939          DS      FB.NAML
                       940
                       941  *       CHANNEL #2
                       942
042.316  002 000       943          DB      2,0             CHANNEL AND STATUS
042.320  013 116 013   944          DW      MTAREA+3+256,MTAREA+3+256,MTAREA+3+256,MTAREA+3+256+256
042.330                945          DS      FB.NAML
                       946
                       947  *       CHANNEL #3
                       948
042.351  003 000       949          DB      3,0             CHANNEL AND STATUS
042.353  013 117 013   950          DW      MTAREA+3+512,MTAREA+3+512,MTAREA+3+512,MTAREA+3+512+256
042.363                951          DS      FB.NAML
                       952
                       953  *       CHANNEL #4
                       954
043.004  004 000       955          DB      4,0             CHANNEL AND STATUS
043.006  013 120 013   956          DW      MTAREA+3+768,MTAREA+3+768,MTAREA+3+768,MTAREA+3+768+256
043.016                957          DS      FB.NAML
                       958
                       959  *       CHANNEL #5
                       960
043.037  005 000       961          DB      5,0             CHANNEL AND STATUS
043.041  013 121 013   962          DW      MTAREA+3+1024,MTAREA+3+1024,MTAREA+3+1024,MTAREA+3+1024+256
043.051                963          DS      FB.NAML


043.072  123 131 060   965  DEFALTP DB      'SYOBAS'        PROGRAM FILE DEFAULTS           /80.02.GC/
043.100  123 131 060   966  DEFALTD DB      'SYODAT'        DATA    FILE DEFAULTS           /80.02.GC/
```

```
                                968  ***     BASIC - MAIN EXEC LOOP.
                                969  *
                                970
043.106                         971  START   EQU     *
043.106  041 370 100            972          LXI     H,CBINT
043.111  076 002                973          MVI     A,CTLB
043.113  377 041                974          DB      SYSCALL,.CTLC   SETUP CTL-B HANDLER
043.115  041 363 100            975          LXI     H,CCINT
043.120  076 003                976          MVI     A,CTLC
043.122  377 041                977          DB      SYSCALL,.CTLC   SETUP CTL-C HANDLER
                                978
                                979  **      ACCEPT COMMAND OR TEXT.
                                980
043.124                         981  RESTART EQU     *               RESTART ADDRESS
                                982
                                983  *       AM IN COMMAND MODE. RESTORE SYSTEM TO COMMAND MODE.
                                984
043.124  061 200 042            985          LXI     SP,STACK        RESTORE STACK POINTER
043.127  041 124 043            986          LXI     H,RESTART
043.132  345                    987          PUSH    H               SET *RETURN ADDRESS*
043.133  257                    988          XRA     A
000.000                         989          ERRNZ   MI,NOP
043.134  062 111 076            990          STA     PNTC            CLEAR TOKEN PIPELINE
043.137  062 204 112            991          STA     CTLFLAG         CLEAR CTL-C AND CTL-B FLAGS
043.142  062 326 040            992          STA     S.CSLMD
000.000                         993          ERRNZ   RM.IMM
043.145  062 343 114            994          STA     RUNMOD          SET IMMEDIATE MODE
043.150  315 115 074            995          CALL    FOC             FILE OPEN CLEANUP
043.153  076 001                996          MVI     A,1
043.155  062 315 112            997          STA     COLCNTS+0       SET COLUMN NUMBER FOR CONSOLE (*PRINT* CMD)
043.160  315 016 112            998          CALL    $CCO            CLEAR CTL-O
043.163  315 073 112            999          CALL    $GNL            GUARANTEE NEW LINE
043.166  315 136 031           1000          CALL    $TYPTX
043.171  252                   1001          DB      '*'+200Q        PROMPT
043.172  315 364 065           1002          CALL    ICL             INPUT COMMAND LINE
043.175  322 203 043           1003          JNC     BAS1            NO CTL-C HIT
                               1004
                               1005  *       CTL-C HIT. CLEAR CONSOLE AND RESTART
                               1006
043.200  377 007               1007          DB      SYSCALL,.CLRCO
043.202  311                   1008          RET     RESTART         START AGAIN
                               1009
043.203  302 152 070           1010  BAS1    JNZ     ERR.SY          SYNTAX ERROR IN STATEMENT       /80.01.GC/
043.206  001 327 112           1011          LXI     B,LINE
043.211  315 230 072           1012          CALL    CNC             CLASSIFY NEXT CHARACTER         /80.01.GC/
043.214  247                   1013          ANA     A               SEE IF KEYWORD
043.215  372 233 043           1014          JM      BAS3            IS KEYWORD
043.220  376 002               1015          CPI     CT.NUM                                          /80.01.GC/
043.222  302 240 043           1016          JNZ     BAS2            IS NOT A NUMBER                 /80.01.GC/
                               1017
                               1018  *       HAVE STATEMENT WITH NUMBER.
                               1019
043.225  315 021 045           1020          CALL    CLR1            CLEAR REFERENCES TO TEXT
043.230  303 270 070           1021          JMP     MTL             INSERT TEXT LINE
                               1022
                               1023  *       IS KEYWORD. SEE IF ALLOWED IMMEDIATE USAGE
```

```
                                1024
043.233  376 250               1025  BAS3    CPI     CT.IUA          IMMEDIATE USAGE ALLOWED?
043.235  322 125 070            1026          JNC     ERR.IU          ILLEGAL USAGE
                                1027
043.240  257                    1028  BAS2    XRA     A
000.000                         1029          ERRNZ   RM.IMM          SET IMMEDIATE MODE
                                1030  *       JMP     EXEC            EXECUTE IN IMMEDIATE MODE
```

```
                                1033  **      EXEC - EXECUTE BASIC STATEMENTS.
                                1034  *
                                1035  *       EXEC CAUSES ONE OR MORE BASIC STATEMENTS TO BE EXECUTED.
                                1036  *
                                1037  *       ENTRY   (CURNUM) = CURNET LINE NUMBER
                                1038  *               (CURADR) = CURRENT LINE ADDRESS
                                1039  *               (A) = RUN MODE CONTROL
                                1040  *               (BC) = TEXT START ADDRESS
                                1041  *                                      (STEP, IMMEDIATE, CONTINUOUS)
                                1042  *       EXIT    WHEN MODE CONTROL IS CLEARED, OR AT END OF LINE
                                1043  *               FOR STEP AND IMMEDIATE MODES.
                                1044  *       USES    ALL
                                1045
                                1046
        043.241                 1047  EXEC    EQU     *
        043.241  062 343 114    1048          STA     RUNMOD          SET RUN MODE
                                1049
                                1050  *       PERFORM THE NEXT COMMAND.
                                1051
        043.244  041 124 043    1052  EXEC1   LXI     H,RESTART       SET ABNORMAL EXIT ADDRESS
        043.247  042 077 075    1053          SHLD    ILMA
        043.252  257            1054          XRA     A
        043.253  062 202 112    1055          STA     IOCHAN          SET OUTPUT TO CONSOLE
        043.256  315 072 076    1056          CALL    PNT             PREVIEW NEXT TOKEN
        000.000                 1057          ERRNZ   CT.FIN
        043.261  247            1058          ANA     A
        043.262  302 372 043    1059          JNE     EXEC3
        043.265  315 056 071    1060          CALL    ANT             CLEAR 'PNT' PIPELINE
                                1061
                                1062  *       END OF STATEMENT.
                                1063
        043.270                 1064  EXEC2   EQU     *
        043.270  315 201 044    1065          CALL    EXEC7           SAVE CURRENT TEXT ADDRESS
                                1066
                                1067  *       CHECK FOR CONTROL CHARACTERS.
                                1068
        043.273  041 204 112    1069          LXI     H,CTLFLAG
        043.276  176            1070          MOV     A,M
        043.277  037            1071          RAR
        043.300  332 106 070    1072          JC      ERR,CC          CONTROL-C HIT
        043.303  037            1073          RAR
        043.304  334 215 044    1074          CC      EXEC8           USER INTERRUPT
                                1075
                                1076  *       CHECK FOR HALT
                                1077
        043.307  072 343 114    1078          LDA     RUNMOD
        043.312  147            1079          MOV     H,A
        043.313  247            1080          ANA     A
        000.000                 1081          ERRNZ   RM.HLT-200Q
        043.314  370            1082          RM                      AM TO HALT
                                1083
                                1084  *       SETUP CORRECT DISPLAY MODE FOR FPLEDS.
                                1085
        043.315  076 000        1086          MVI     A,0             (A) = MODE INDEX
        043.316                 1087  FPMODE  EQU     *-1
        043.317  021 244 044    1088          LXI     D,EXEC4
```

```
043.322  203           1089          ADD     E
043.323  137           1090          MOV     E,A
043.324  032           1091          LDAX    D               (A) = FLAG VALUE
043.325  062 010 040   1092          STA     .MFLAG          SET TYPE OF DISPLAY
                       1093
                       1094  *       CHECK TO SEE IF ANOTHER STATEMENT ON THIS LINE
                       1095
043.330  012           1096          LDAX    B
043.331  003           1097          INX     B
043.332  247           1098          ANA     A
043.333  302 244 043   1099          JNZ     EXEC1           DO NEXT STATEMENT
043.336  174           1100          MOV     A,H             (A) = RUNMODE
043.337  247           1101          ANA     A
000.000                1102          ERRNZ   RM.HLT-200Q     REMOVE HALT FLAG
043.340  310           1103          RZ                      IMMEDIATE MODE
                       1104
                       1105  *       ADVANCE TO NEXT PROGRAM LINE.
                       1106
043.341  012           1107          LDAX    B
043.342  003           1108          INX     B
043.343  157           1109          MOV     L,A             SET LINE NUMBER
043.344  012           1110          LDAX    B
043.345  003           1111          INX     B
043.346  147           1112          MOV     H,A
043.347  042 175 112   1113          SHLD    CURNUM
043.352  245           1114          ANA     L               (A) = PRODUCT OF LINE NUMBER BYTES
043.353  074           1115          INR     A
043.354  076 253       1116          MVI     A,CT.END
043.356  312 040 044   1117          JZ      EXEC6           END OF TEXT - GENERATE 'END'
043.361  315 201 044   1118          CALL    EXEC7
043.364  376 001       1119          CPI     RM.STE
043.366  310           1120          RE                      DONE STEPPING
043.367  303 244 043   1121          JMP     EXEC1           PROCESS NEXT STATEMENT
                       1122
                       1123  *       PROCESS LINE.
                       1124
043.372  315 007 044   1125  EXEC3   CALL    EXEC4
                       1126
                       1127  *       RETURN FROM STATEMENT PROCESSOR. MUST HAVE END OF STATEMENT.
                       1128
043.375  315 305 077   1129  EXEC3.5 CALL    RNT
044.000  000           1130          DB      CT.FIN          REQUIRE CT.FIN
044.001  315 357 073   1131          CALL    DTS             DELETE TEMP STRINGS
044.004  303 270 043   1132          JMP     EXEC2
                       1133
                       1134
044.007  376 200       1135  EXEC4   CPI     CT.BLD
044.011  332 125 070   1136          JC      ERR.IU          ILLEGAL USAGE
                       1137
044.014  376 256       1138          CPI     CT.CMD
044.016  322 374 050   1139          JNC     LET             MUST BE 'LET', IS NOT COMMAND
                       1140
044.021  376 212       1141          CPI     CT.RUA
044.023  322 035 044   1142          JNC     EXEC5           RUN USAGE ALLOWED
                       1143
044.026  072 343 114   1144          LDA     RUNMOD
```

```
000.000                 1145            ERRNZ   RM.IMM
044.031  247           1146            ANA     A
044.032  302 125 070    1147            JNE     ERR.IU          ILLEGAL USAGE FOR IMMEDIATE MODE
                        1148
044.035  315 056 071    1149    EXEC5   CALL    ANT             ACCEPT NEXT TOKEN
044.040  326 200        1150    EXEC6   SUI     200Q            REMOVE BIAS
044.042  315 061 031    1151            CALL    $TJMP           ENTER PROCESSOR
                        1152
044.045  247 044        1153            DW      BUILD
044.047  337 044        1154            DW      BYE
044.051  163 045        1155            DW      CONT            CONTINUE
044.053  162 046        1156            DW      DELETE
044.055  020 051        1157            DW      LIST
044.057  233 053        1158            DW      REPLACE
044.061  155 045        1159            DW      RUN
044.063  302 053        1160            DW      SAVE
044.065  351 044        1161            DW      SCRATCH
044.067  356 053        1162            DW      STEP
                        1163
044.071  152 070        1164            DW      ERR.SY          LEXICAL SYNTAX ERROR FOUND
044.073  205 045        1165            DW      CHAIN
044.075  363 044        1166            DW      CLEAR
044.077  260 045        1167            DW      CLOSE
044.101  320 045        1168            DW      CNTRL           CNTRL
044.103  236 046        1169            DW      DIM             DIMENSION
044.105  152 070        1170            DW      ERR.SY          FN
044.107  060 047        1171            DW      FOR
044.111  213 047        1172            DW      FREE
044.113  336 047        1173            DW      FREEZE
044.115  026 050        1174            DW      GOSUB
044.117  031 050        1175            DW      GOTO
044.121  051 050        1176            DW      IF
044.123  374 050        1177            DW      LET
044.125  175 051        1178            DW      LOCK
044.127  203 051        1179            DW      NEXT
044.131  332 051        1180            DW      OLD
044.133  355 051        1181            DW      ON
044.135  036 052        1182            DW      OPEN
044.137  220 052        1183            DW      OUT
044.141  251 052        1184            DW      PAUSE
044.143  336 052        1185            DW      POKE
044.145  343 052        1186            DW      PRINT
044.147  171 053        1187            DW      READ
044.151  121 050        1188            DW      IF2             REM
044.153  053 045        1189            DW      RESTORE
044.155  242 053        1190            DW      RETURN
044.157  041 054        1191            DW      UNFREZ          UNFREEZE
044.161  176 051        1192            DW      UNLOCK
044.163  065 054        1193            DW      UNSAVE
                        1194
044.165  137 050        1195            DW      LINPUT
044.167  342 077        1196            DW      SES             DATA
044.171  133 046        1197            DW      DEF
044.173  044 047        1198            DW      END
044.175  150 050        1199            DW      INPUT
044.177  030 054        1200            DW      STOP
```

```
                                1202  *       END OF EXEC SEQUENCE. SAVE TEXT POINTER.
                                1203
044.201  072 343 114           1204  EXEC7   LDA     RUNMOD
044.204  346 177               1205          ANI     377Q-RM.HLT
000.000                        1206          ERRNZ   RM.IMM
044.206  310                   1207          RZ                      AM IN IMMEDIATE MODE
044.207  140                   1208          MOV     H,B             (HL) = TEXT ADDRESS
044.210  151                   1209          MOV     L,C
044.211  042 177 112           1210          SHLD    CURADR
044.214  311                   1211          RET


                                1213  **      CTL-B (USER INTERRUPT) HIT
                                1214
044.215  021 000 000           1215  EXEC8   LXI     D,0             (DE) = INTERRUPT EXIT ADDRESS
044.216                        1216  ACTLB   EQU     *-2
044.220  172                   1217          MOV     A,D
044.221  263                   1218          ORA     E
044.222  312 111 070           1219          JZ      ERR.CB          NO USER PROCESSING
                                1220
                                1221  *       USER PROGRAM PROCESSING SPECIFIED.
                                1222
044.225  176                   1223  EXEC9   MOV     A,M
044.226  346 375               1224          ANI     377Q-CFCTLB
044.230  167                   1225          MOV     M,A             CLEAR FLAG
044.231  341                   1226  EXEC10  POP     H               DISCARD 'RETURN ADDRESS'
044.232  353                   1227          XCHG                    (HL) = TEXT ADDRESS
044.233  315 143 100           1228          CALL    SRA             SAVE TEXT RETURN ADDRESS
044.236  315 042 050           1229          CALL    GOTO2           PROCESS AS GOTO
044.241  303 375 043           1230          JMP     EXEC3.5         EXIT FROM GOSUB
                                1231
                                1232  **      TABLE OF .MFLAG VALUES FOR DISPLAY CONTROL.
                                1233
044.244  301                   1234  EXECA   DB      UO.NFR+UO.HLT+UO.CLK    NO DISPLAY
044.245  203                   1235          DB      UO.DDU+UO.HLT+UO.CLK    DISABLE UPDATE
044.246  201                   1236          DB      UO.HLT+UO.CLK           LEAVE ON AND UPDATING
                                1237
000.044                        1238  .       SET     */256
000.000                        1239          ERRNZ   EXECA/256-.     ASSUME IN SAME BANK
                                1240


                                1243  **      BUILD - PROCESS BUILD COMMAND.
                                1244  *
                                1245  *       BUILD N,M
                                1246  *
                                1247  *       STARTING AT LINE N, INCREMENT BY M
                                1248
                                1249
044.247                        1250  BUILD   EQU     *
044.247  315 313 075           1251          CALL    LFC             CHECK FOR DATA LOCK
044.252  315 235 052           1252          CALL    OUT1            (DE) = INC, (HL) = VAL
```

```
044.255  325          1253  BLD1    PUSH    D               SAVE INC
044.256  345          1254          PUSH    H               SAVE NUMBER
044.257  353          1255          XCHG
044.260  315 206 072  1256          CALL    CLN             CHECK FOR LEGAL NUMBER
044.263  315 206 100  1257          CALL    TDI             TYPE LINE NUMBER
044.266  315 364 065  1258          CALL    ICL             ACCEPT NEW LINE
044.271  332 124 043  1259          JC      RESTART         CTL-C HIT
044.274  302 320 044  1260          JNZ     BLD2            ERROR IN LINE
044.277  041 327 112  1261          LXI     H,LINE
044.302  321          1262          POP     D
044.303  325          1263          PUSH    D               (DE) = NUMBER
044.304  315 304 070  1264          CALL    MTLO            INSERT TEXT LINE
044.307  341          1265          POP     H               (HL) = NUMBER
044.310  321          1266          POP     D               (DE) = INC
044.311  031          1267          DAD     D
044.312  332 122 070  1268          JC      ERR.IN          OVERFLOW
044.315  303 255 044  1269          JMP     BLD1
                      1270
                      1271  *       ERROR IN LINE
                      1272
044.320  315 136 031  1273  BLD2    CALL    $TYPTX
044.323  207          1274          DB      BELL+200Q
044.324  076 214      1275          MVI     A,BEC.SY
044.326  046 012      1276          MVI     H,NL
044.330  377 057      1277          DB      SYSCALL,.ERROR  SHOW ERROR
044.332  341          1278          POP     H
044.333  321          1279          POP     D
044.334  303 255 044  1280          JMP     BLD1            RE-TRY LINE ENTRY


                      1282  ***     BYE - RETURN TO HDOS.
                      1283  *
                      1284  *       BYE
                      1285
                      1286
044.337               1287  BYE     EQU     *
044.337  315 313 075  1288          CALL    LFC             CHECK FOR DATA LOCK
044.342  315 146 071  1289          CALL    AYS             ARE YOU SURE?
044.345  300          1290          RNE                     NOT SURE
044.346  257          1291          XRA     A
044.347  377 000      1292          DB      SYSCALL,.EXIT   EXIT


                      1294  **      SCRAT - SCRATCH SYSTEM.
                      1295  *
                      1296  *       DESTROY TEXT, CLEAR VARIABLES.
                      1297
                      1298
044.351               1299  SCRATCH EQU     *
044.351  315 313 075  1300          CALL    LFC             CHECK FOR DATA LOCK
044.354  315 146 071  1301          CALL    AYS             ARE YOU SURE?
044.357  300          1302          RNE                     NOT SURE
```

```
044.360  315 320 077  1303  SCR.    CALL    SCRA            INSERT DUMMY LAST LINE INTO TEXT TABLE
                      1304  *       JMP     CLEAR


                      1306  **      CLEAR - MASTER CLEAR.
                      1307  *
                      1308  *       CLEAR RESETS ALL CONTROL STRUCTURES:
                      1309  *
                      1310  *       1) GOSUB STACK
                      1311  *       2) 'FOR' STACK
                      1312  *       3) NEXT STATEMENT INDEX
                      1313  *       4) CLEAR VARIABLE LIST
                      1314  *       5) DATA POINTER
                      1315
                      1316
044.363               1317  CLEAR   EQU     *
044.363  315 313 075  1318          CALL    LFC             CHECK FOR DATA LOCK
044.366  315 056 071  1319          CALL    ANT
000.000               1320          ERRNZ   CT.FIN
044.371  247          1321          ANA     A
044.372  302 062 045  1322          JNZ     CLR2            HAVE VARIABLE
044.375  315 357 073  1323  CLEAR.  CALL    DTS
045.000  041 000 000  1324          LXI     H,0
045.003  042 154 112  1325          SHLD    STRTAB+MT.LEN
045.006  042 130 112  1326          SHLD    SYMTAB+MT.LEN
045.011  056 200      1327          MVI     L,200Q
045.013  042 205 112  1328          SHLD    STRVI           CLEAR STRING INDEX
045.016  315 171 072  1329          CALL    CLF             CLEAR FILE STRUCTURES
                      1330
                      1331
                      1332  *       ENTRY POINT FOR ROUTINES TO CLEAR REFERENCES TO TXTTAB.
                      1333
045.021  041 000 000  1334  CLR1    LXI     H,0             ENTRY TO JUST CLEAR TXTTAB REFERENCES
045.024  042 135 112  1335          SHLD    FORTAB+MT.LEN
045.027  042 142 112  1336          SHLD    GOSTAB+MT.LEN
045.032  042 147 112  1337          SHLD    WRKTAB+MT.LEN
045.035  042 216 044  1338          SHLD    ACTLB
045.040  056 300      1339          MVI     L,300Q
045.042  042 366 073  1340          SHLD    DTSA            CLEAR TEMP ONDEX
045.045  041 007 115  1341          LXI     H,MTAREA-1
045.050  042 177 112  1342          SHLD    CURADR          CLEAR ADDREAA


                      1344  **      RESTORE - RESTORE DATA POINTER
                      1345  *
                      1346  *       RESTORE
                      1347
                      1348
045.053  041 007 115  1349  RESTORE LXI     H,MTAREA-1
045.056  042 345 114  1350          SHLD    DATPTR
045.061  311          1351          RET
                      1352
```

```
                                1353  *       CLEAR VARIABLE
                                1354
045.062  376 300                1355  CLR2    CPI     CT.VARL
045.064  332 152 070            1356          JC      ERR,SY          NOT VARIABLE
045.067  376 306                1357          CPI     CT.SSF+1
045.071  322 152 070            1358          JNC     ERR.SY          NOT VARIABLE
045.074  147                    1359          MOV     H,A             SAVE (A) IN H
045.075  076 042                1360          MVI     A,LEXLIM/256
045.077  272                    1361          CMP     D
045.100  320                    1362          RNC                     IS NOT IN SYMBOL TABLE
045.101  174                    1363          MOV     A,H             (A) = VARIABLE TYPE
045.102  041 006 000            1364          LXI     H,6             (HL) = SIZE TO CLEAR
045.105  346 002                1365          ANI     CF.VEC
045.107  312 132 045            1366          JZ      CLR3            NOT VECTOR
045.112  032                    1367          LDAX    D
045.113  247                    1368          ANA     A
045.114  372 132 045            1369          JM      CLR3            IS FUNCTION
045.117  325                    1370          PUSH    D               SAVE ADDR OF AREA+2
045.120  345                    1371          PUSH    H               SAVE #6
045.121  023                    1372          INX     D
045.122  023                    1373          INX     D
045.123  353                    1374          XCHG
045.124  136                    1375          MOV     E,M
045.125  043                    1376          INX     H
045.126  126                    1377          MOV     D,M             (DE) = SIZE OF ARRAY
045.127  341                    1378          POP     H               (HL) = 6
045.130  031                    1379          DAD     D               (HL) = TOTAL SIZE
045.131  321                    1380          POP     D               (DE) = VARIABLE AREA+2
045.132  033                    1381  CLR3    DCX     D
045.133  033                    1382          DCX     D               (DE) = VARIABLE FWA
045.134  345                    1383          PUSH    H               SAVE COUNT TO REMOVE
045.135  052 126 112            1384          LHLD    SYMTAB+MT.FWA
045.140  173                    1385          MOV     A,E             COMPUTE INDEX INTO SYMTAB
045.141  225                    1386          SUB     L
045.142  157                    1387          MOV     L,A
045.143  172                    1388          MOV     A,D
045.144  234                    1389          SBB     H
045.145  147                    1390          MOV     H,A             (HL) = INDEX
045.146  321                    1391          POP     D               (DE) = DELETE COUNT
045.147  315 203 104            1392          CALL    $DBT            DELETE FROM SYMTAB
045.152  126 112                1393          DW      SYMTAB+1
045.154  311                    1394          RET                     DONE


                                1396  **      RUN - BEGIN EXECUTION.
                                1397  *
                                1398  *       RUN IS THE SAME AS
                                1399  *
                                1400  *       CLEAR: CONTINUE
                                1401
                                1402
045.155  315 313 075            1403  RUN     CALL    LFC             CHECK FOR DATA LOCK
045.160  315 375 044            1404          CALL    CLEAR.
```

```
                                1406  **      CONT - RESUME EXECUTION.
                                1407  *
                                1408
                                1409
045.163  076 004               1410  CONT    MVI     A,RM.CON        (A) = NEW RUN MODE
045.165  052 177 112           1411  CONT1   LHLD    CURADR
045.170  104                   1412          MOV     B,H
045.171  115                   1413          MOV     C,L             (BC) = CURRENT TEXT ADDRESS
045.172  315 241 043           1414          CALL    EXEC            EXECUTE WITH REQUESTED MODE
045.175  001 007 115           1415          LXI     B,ZERO          POINT TO ZERO BYTE
000.000                        1416          ERRNZ   RM.IMM
045.200  257                   1417          XRA     A
045.201  062 343 114           1418          STA     RUNMOD          RESTORE IMMEDIATE MODE
045.204  311                   1419          RET


                                1421  ***     CHAIN - CHAIN TO NEW PROGRAM.
                                1422  *
                                1423  *       CHAIN <STRING> [<LINE NUMBER>]
                                1424  *
                                1425  *       LEAVE DATA, VARIABLES, AND CHANNELS INTACT
                                1426
                                1427
045.205                         1428  CHAIN   EQU     *
045.205  341                   1429          POP     H               ** KLUDGE ** TO CLEAN STACK FOR RECURSIVE CALL TO *CONT*
045.206  341                   1430          POP     H
045.207  315 053 072           1431          CALL    CFN             COPY FILE NAME
045.212  315 072 076           1432          CALL    PNT             SEE IF LINE # FOLLOWS
045.215  247                   1433          ANA     A
000.000                        1434          ERRNZ   CT.FIN
045.216  312 231 045           1435          JZ      CHAIN1          NO LINE NUMBER
045.221  315 223 072           1436          CALL    CMA             GOBBLE COMMA
045.224  315 033 074           1437          CALL    ELN             EVAL LINE NUMBER
045.227  366 001                1438          ORI     1               CLEAR 'Z'
045.231  325                   1439  CHAIN1  PUSH    D               SAVE LINE NUMBER (GARBAGE IF NO NUMBER)
045.232  365                   1440          PUSH    PSW             'Z' SET IF NO LINE NUMBER
045.233  315 206 077           1441          CALL    RNP             READ NEW PROGRAM
045.236  361                   1442          POP     PSW             'Z' SET IF NO NUMBER
045.237  321                   1443          POP     D               (DE) = NUMBER
045.240  312 163 045           1444          JZ      CONT            JUST CONTINUE
                                1445
                                1446  *       HAD LINE NUMBER, NOW FIND IT
                                1447
045.243  315 242 074           1448          CALL    FLN             FIND LINE BY NUMBER
045.246  332 147 070           1449          JC      ERR.SN
045.251  053                   1450          DCX     H               POINT TO TERMINATOR OF PREVIOUS LINE
045.252  042 177 112           1451          SHLD    CURADR
045.255  303 163 045           1452          JMP     CONT            PROCESS AS CONTINUE
```

```
                                1454  ***     CLOSE - CLOSE FILE.
                                1455  *
                                1456  *       CLOSE   #I [,#J,...,#N]
                                1457  *
                                1458  *       CLOSE FILES #I THROUGH #N
                                1459  *
                                1460  *       NO ERROR MESSAGE IF FILE ALREADY CLOSED.
                                1461
                                1462
045.260                         1463  CLOSE   EQU     *
045.260  315 273 073            1464          CALL    DCN.            DECODE CHANNEL NUMBER
045.263  305                    1465          PUSH    B               SAVE TEXT POINTER
045.264  072 202 112            1466          LDA     IOCHAN
045.267  075                    1467          DCR     A
045.270  315 005 072            1468          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
045.273  332 304 045            1469          JC      CLOSE1          CHANNEL DOSENT EXIST
045.276  315 335 102            1470          CALL    $FCLO           CLOSE IT
045.301  315 326 073            1471          CALL    DNF             DELETE NON-OPEN FILE BLOCKS
045.304  301                    1472  CLOSE1  POP     B
045.305  315 072 076            1473          CALL    PNT             CHECK NEXT TOKEN
000.000                         1474          ERRNZ   CT.FIN
045.310  247                    1475          ANA     A
045.311  310                    1476          RZ                      DONE WITH STATEMENT
045.312  315 223 072            1477          CALL    CMA             REQUIRE COMMA
045.315  303 260 045            1478          JMP     CLOSE           CRACK ANOTHER


                                1480  **      CNTRL - CONTROL COMMAND.
                                1481  *
                                1482  *       CNTRL I,J
                                1483  *
                                1484  *       I=0     SET CTL-B PROCESSOR LINE
                                1485  *               J=N             LINE NUMBER
                                1486  *
                                1487  *       I=1     SET PRINTING MODE
                                1488  *               J=N             SET SCIENTIFIC THRESHOLD
                                1489  *
                                1490  *       I=2     SET DISPLAY MODE
                                1491  *               J=0  DISPLAYS OFF
                                1492  *               J=1  DISPLAYS REFRESHED, NOT UPDATED
                                1493  *               J=2  DISPLAYS REFRESHED AND UPDATED
                                1494  *
                                1495  *       I=3     SET TAB SIZE
                                1496  *               J=NN            WIDTH OF TAB FIELD
                                1497  *
                                1498  *       I=4     SET OVERLAY FLAG
                                1499  *               J=0  USE MAXIMUM AMOUNT OF MEMORY
                                1500  *               J=1  ALLOW OVERLAY TO REMAIN RESIDENT
                                1501
                                1502
045.320                         1503  CNTRL   EQU     *
045.320  315 235 052            1504          CALL    OUT1            (L) = I, (E) = J
045.323  175                    1505          MOV     A,L
045.324  376 005                1506          CPI     CNTLMX
```

```
045.326  322 122 070  1507          JNC     ERR.IN          TOO BIG A NUMBER
045.331  315 076 031  1508          CALL    $TBRA
045.334  005          1509  CNTLA   DB      CNTL1-*
045.335  016          1510          DB      CNTL2-*
045.336  033          1511          DB      CNTL3-*
045.337  044          1512          DB      CNTL4-*
045.340  101          1513          DB      CNTL5-*
000.005               1514  CNTLMX  EQU     *-CNTLA         MAX NUMBER OF FUNCTIONS - 1
                      1515
                      1516
                      1517  *       SET CTL-B PROCESSOR.
                      1518
045.341  315 242 074  1519  CNTL1   CALL    FLN             FIND LINE BY NUMBER
045.344  332 147 070  1520          JC      ERR.SN          NOT FOUND
045.347  042 216 044  1521          SHLD    ACTLB           SET ADDRESS
045.352  311          1522          RET
                      1523
                      1524  *       SET SCIENTIFIC THRESHOLD
                      1525
045.353  173          1526  CNTL2   MOV     A,E             SET THRESHOLD
045.354  074          1527          INR     A
045.355  376 010       1528          CPI     7+1                                     /78.10.GC/
045.357  322 122 070  1529          JNC     ERR.IN          LIM SIZE DUE TO ACC. OF FLT. PT./78.10.GC/
045.362  062 022 111  1530          STA     FTAC
045.365  062 032 111  1531          STA     FTAD                                    /78.10.GC/
045.370  311          1532          RET
                      1533
                      1534  *       SET DISPLAY MODE.
                      1535
045.371  173          1536  CNTL3   MOV     A,E
045.372  376 003       1537          CPI     3
045.374  322 122 070  1538          JNC     ERR.IN          IF ILLEGAL VALUE
045.377  062 316 043  1539          STA     FPMODE          SET DISPLAY MODE
046.002  311          1540          RET
                      1541
                      1542  *       SET TAB SIZE
                      1543
046.003  173          1544  CNTL4   MOV     A,E
046.004  247          1545          ANA     A
046.005  312 122 070  1546          JZ      ERR.IN          BAD VALUE
046.010  062 062 053  1547          STA     PRIC
                      1548
046.013  257          1549          XRA     A                                       /80.01.GC/
046.014  041 331 040  1550          LXI     H,S.CONWI                               /80.01.GC/
046.017  203          1551  CNTL43  ADD     E                                       /80.01.GC/
046.020  332 033 046  1552          JC      CNTL46                                  /80.01.GC/
046.023  276          1553          CMP     M                                       /80.01.GC/
046.024  332 017 046  1554          JC      CNTL43          NOT >= CONSOLE WIDTH    /80.01.GC/
046.027  312 033 046  1555          JZ      CNTL46          IS AN INTEGRAL MULTIPLE /80.01.GC/
046.032  223          1556          SUB     E                                       /80.01.GC/
046.033  223          1557  CNTL46  SUB     E                                       /80.01.GC/
046.034  074          1558          INR     A               ADJUST AT THE LIMIT POINTS /80.01.GC/
                      1559
046.035  062 050 053  1560          STA     PRIB            SET TAB-FIELD WRAP WIDTH
046.040  311          1561          RET
                      1562
```

```
                        1563  *       SET OVERLAY LOAD OPTIONS
                        1564
046.041  172            1565  CNTL5   MOV     A,D
046.042  247            1566          ANA     A                                       /78.10.GC/
046.043  302 122 070    1567          JNZ     ERR.IN          BAD VALUE               /78.10.GC/
046.046  263            1568          ORA     E
046.047  376 002        1569          CPI     1+1                                     /78.10.GC/
046.051  322 122 070    1570          JNC     ERR.IN                                  /78.10.GC/
046.054  062 203 112    1571          STA     OVLMAN          SET OVERLAY MANAGE FLAGS
046.057  247            1572          ANA     A                                       /78.10.GC/
046.060  312 115 074    1573          JZ      FOC             OPEN TABLES             /78.10.GC/
                        1574
                        1575  *       GET THE NEW OVERLAY MEMORY                      /80.01.GC/
                        1576
046.063  315 054 031    1577          CALL    $SAVALL                                 /80.01.GC/
046.066  315 230 074    1578          CALL    FOP.            SQUEEZE TABLES          /80.01.GC/
046.071  345            1579          PUSH    H               SAVE LWA                /80.01.GC/
                        1580
046.072  052 350 040    1581          LHLD    S.OFWA                                  /80.01.GC/
000.000                 1582          ERRNZ   OVL0                                    /80.01.GC/
046.075  021 006 000    1583          LXI     D,OVL.FLB                               /80.01.GC/
046.100  031            1584          DAD     D               HL = ADDR. OF FLAG BYTE /80.01.GC/
046.101  176            1585          MOV     A,M             A  = FLAG BYTE          /80.01.GC/
046.102  346 001        1586          ANI     OVL.IN                                  /80.01.GC/
046.104  052 320 040    1587          LHLD    S.SYSM                                  /80.01.GC/
046.107  021 360 377    1588          LXI     D,-16                                   /80.01.GC/
046.112  031            1589          DAD     D               LEAVE SOME SLOP         /80.01.GC/
046.113  302 126 046    1590          JNZ     CNTL52          ALREADY IN MEMORY       /80.01.GC/
                        1591
                        1592  *       LEAVE ROOM FOR THE OVERLAY                      /80.01.GC/
                        1593
046.116  353            1594          XCHG                                            /80.01.GC/
046.117  052 324 040    1595          LHLD    S.OMAX                                  /80.01.GC/
046.122  315 224 030    1596          CALL    $CHL            HL = -HL                /80.01.GC/
046.125  031            1597          DAD     D                                       /80.01.GC/
                        1598
046.126  321            1599  CNTL52  POP     D                                       /80.01.GC/
046.127  353            1600          XCHG                    DE = PROSPECTUS, HL = LIMIT /80.01.GC/
046.130  303 152 074    1601          JMP     FOC1.3                                  /80.01.GC/


                        1603  **      DEF - DEFINE FUNCTION.
                        1604  *
                        1605  *       1 LINE FUNCTIONS:
                        1606  *
                        1607  *       DEF FN X(P1,...,PN) = EXPR
                        1608
                        1609
046.133                 1610  DEF     EQU     *
046.133  315 305 077    1611          CALL    RNT
046.136  220            1612          DB      CT.FN           REQUIRE 'FN'
046.137  315 263 075    1613          CALL    IVT             INSERT VECTOR IN TABLE
046.142  032            1614          LDAX    D
046.143  075            1615          DCR     A
```

```
046.144  362 152 070  1616          JP      ERR.SY          IS DIMENSIONED
                      1617
                      1618  *       IS SINGLE LINE DEFINITION.
                      1619
046.147  076 201      1620          MVI     A,201Q
046.151  022          1621          STAX    D
046.152  023          1622          INX     D
046.153  353          1623          XCHG
046.154  161          1624          MOV     M,C
046.155  043          1625          INX     H
046.156  160          1626          MOV     M,B             SET FUNCTION ADDRESS
046.157  303 342 077  1627          JMP     SES             SKIP TO STATEMENT END AND EXIT


                      1629  **      DELETE - DELETE LINES.
                      1630  *
                      1631  *       DELETE NNN;MMM
                      1632
                      1633
046.162               1634  DELETE  EQU     *
046.162  315 313 075  1635          CALL    LFC             CHECK FOR DATA LOCK
046.165  315 036 057  1636          CALL    EVALI           (DE) = 1ST LINE NUMBER
046.170  315 223 072  1637          CALL    CMA             REQUIRE ','
046.173  315 242 074  1638          CALL    FLN             FIND LINE BY NUMBER
046.176  345          1639          PUSH    H               SAVE ADDRESS
046.177  315 036 057  1640          CALL    EVALI
046.202  023          1641          INX     D
046.203  315 242 074  1642          CALL    FLN             FIND LAST
046.206  353          1643          XCHG
046.207  341          1644          POP     H               (HL) = FWA, (DE) = LWA
046.210  175          1645          MOV     A,L
046.211  223          1646          SUB     E
046.212  137          1647          MOV     E,A
046.213  174          1648          MOV     A,H
046.214  232          1649          SBB     D
046.215  127          1650          MOV     D,A             (DE) = BYTE COUNT TO DELETE
046.216  322 152 070  1651          JNC     ERR.SY          FIRST > LAST
046.221  325          1652          PUSH    D               SAVE COUNT
046.222  021 370 262  1653          LXI     D,-NTAREA
046.225  031          1654          DAD     D               (HL) = TABLE INDEX OF 1ST LINE TO DELETE
046.226  321          1655          POP     D               (DE) = COUNT
046.227  067          1656          STC                     NUMBER IS NEG.  SET 17TH BIT OF NUMBER
046.230  315 213 104  1657          CALL    $IBT            REMOVE BYTES
046.233  121 112      1658          DW      TXTTAB+1
046.235  311          1659          RET
```

```
                                1661  **      DIM - PROCESS DIMENSION DECLARATION.
                                1662  *
                                1663  *       DIM ITEM1(X1,...,XN),...,ITEMN(X1,...,XP)
                                1664
                                1665
046.236                         1666  DIM     EQU     *
046.236  052 130 112            1667          LHLD    SYMTAB+MT.LEN
046.241  042 034 047            1668          SHLD    DIMA            SET BEFORE SYMTAB LEN
046.244  041 033 047            1669          LXI     H,DIM5
046.247  042 077 075            1670          SHLD    ILMA            SET ABORT PROCESSOR
046.252  315 263 075            1671          CALL    IVT             INSERT VECTOR IN SYMBOL TABLE
                                1672
046.255  315 000 073            1673          CALL    CSI             (DE) = INDEX INTO SYMTAB
046.260  325                    1674          PUSH    D               SAVE INDEX INTO SYMTAB
                                1675
                                1676  *       DECODE AND STORE DIMENSION BOUNDS IN VECTAB.
                                1677
046.261  041 001 000            1678          LXI     H,1             (HL) = ARRAY SIZE ACCUMULATOR
046.264  134                    1679          MOV     E,H             (E) = 0 = DIMENSION COUNT
046.265  034                    1680  DIM2    INR     E               INCREMENT DIMENSION COUNT
046.266  325                    1681          PUSH    D
046.267  315 036 057            1682          CALL    EVALI           EVALUATE NUMERIC EXPRESSION
046.272  023                    1683          INX     D               (DE) = BOUND+1
046.273  325                    1684          PUSH    D               SAVE BOUND
046.274  305                    1685          PUSH    B               SAVE (BC)
046.275  104                    1686          MOV     B,H             (BC) = CURRENT ARRAY SIZE
046.276  115                    1687          MOV     C,L
046.277  315 337 030            1688          CALL    $MU66           (HL) = NEW ARRAY SIZE
046.302  302 160 070            1689          JNZ     ERR.TO          OVERFLOW
046.305  301                    1690          POP     B
046.306  343                    1691          XTHL                    PUSH SIZE UNDER DIMENSION BOUND
046.307  345                    1692          PUSH    H
046.310  041 002 000            1693          LXI     H,2
046.313  021 126 112            1694          LXI     D,SYMTAB+1
046.316  315 026 071            1695          CALL    AMB             ALLOCATE 2 BYTES TO STORE BOUND
046.321  321                    1696          POP     D               (DE) = DIMENSION BOUND
046.322  163                    1697          MOV     M,E
046.323  043                    1698          INX     H
046.324  162                    1699          MOV     M,D             STORE IN TABLE
046.325  315 056 071            1700          CALL    ANT             ACCEPT NEXT TOKEN
046.330  341                    1701          POP     H               (HL) = ARRAY SIZE
046.331  321                    1702          POP     D               (E) = DIMENSION COUNT
046.332  376 026                1703          CPI     CT.CMA
046.334  312 265 046            1704          JE      DIM2            GET ANOTHER
046.337  376 020                1705          CPI     CT.PAR
046.341  302 152 070            1706          JNE     ERR.SY          REQUIRE )
                                1707
                                1708  *       READ ALL BOUNDS. SET SUBSCRIPT COUNT IN SYMTAB.
                                1709
046.344  173                    1710          MOV     A,E             (A) = SUBSCRIPT COUNT
046.345  321                    1711          POP     D               (DE) = INDEX INTO SYMBOL
                                1712
046.346  315 366 072            1713          CALL    CSA             (DE) = ABSOLUTE ADDRESS IN SYMTAB
                                1714
046.351  325                    1715          PUSH    D
046.352  022                    1716          STAX    D               SET DIMENSION COUNT
```

```
046.353  051            1717          DAD     H               (HL) = 2*(HL)
046.354  332 122 070    1718          JC      ERR.IN          TOO LARGE
046.357  051            1719          DAD     H               (HL) = 4*HL
046.360  332 122 070    1720          JC      ERR.IN          TOO LARGE
                        1721
                        1722  *       INSERT LENGTH OF AREA IN HEADER, (HL) = STORAGE NEEDED
                        1723
046.363  207            1724          ADD     A               (A) = NUMBER OF DIMENSIONS *2
046.364  353            1725          XCHG
046.365  046 000        1726          MVI     H,0
046.367  157            1727          MOV     L,A             (HL) = LENGTH OF BOUNDS
046.370  031            1728          DAD     D               (HL) = TOTAL LENGTH
046.371  353            1729          XCHG
046.372  343            1730          XTHL                    (HL) = ADDRESS OF HEADER;((SP)) = STORAGE NEEDED
046.373  043            1731          INX     H
046.374  043            1732          INX     H
                        1733
046.375  163            1734          MOV     M,E
046.376  043            1735          INX     H
046.377  162            1736          MOV     M,D             SET TOTAL LENGTH
047.000  341            1737          POP     H               (HL) = LENGTH OF VALUE STORE AREA
047.001  021 126 112    1738          LXI     D,SYMTAB+1
047.004  345            1739          PUSH    H               SAVE COUNT
047.005  315 026 071    1740          CALL    AMB             ALLOCATE MEMORY
047.010  321            1741          POP     D               (DE) = COUNT
                        1742
                        1743  *       ZERO NEWLY CREATED VALUES.
                        1744
047.011  066 000        1745  DIM3    MVI     M,0
047.013  033            1746          DCX     D               ZERO ENTRYS
047.014  043            1747          INX     H
047.015  172            1748          MOV     A,D
047.016  263            1749          ORA     E
047.017  302 011 047    1750          JNZ     DIM3
                        1751
                        1752  *       DONE WITH DECLARATION. SEE IF ANOTHER FOLLOWS.
                        1753
047.022  315 056 071    1754          CALL    ANT             GET NEXT TOKEN
047.025  376 026        1755          CPI     CT.CMA
047.027  300            1756          RNE                     NOT COMMA
047.030  303 236 046    1757          JMP     DIM             PROCESS ANOTHER
                        1758
                        1759  *       ERROR OCCURED. PUT SYMBOL TABLE BACK.
                        1760
047.033  041 000 000    1761  DIM5    LXI     H,0
047.034                 1762  DIMA    EQU     *-2             PREVIOUS LENGTH
047.036  042 130 112    1763          SHLD    SYMTAB+MT.LEN
047.041  303 124 043    1764          JMP     RESTART         EXIT: RESTART RESTORES ABORT ADDRESS
```

```
                                1766  **      END - END PROGRAM.
                                1767  *
                                1768
                                1769
047.044  041 007 115           1770  END     LXI     H,MTAREA-1
047.047  042 177 112           1771          SHLD    CURADR          SET EXECUTION ADDRESS TO TOP
047.052  076 224               1772          MVI     A,BEC.EN
047.054  365                   1773          PUSH    PSW             SAVE CODE
047.055  303 063 075           1774          JMP     ILM             ISSUE LINE MESSAGE



                                1776  **      FOR - PERFORM 'FOR' LOOP.
                                1777  *
                                1778  *       FOR VAR = VAL1 TO VAL2 [STEP VAL3]
                                1779  *
                                1780  *       KEPT ON 'FOR' STACK:
                                1781  *
                                1782  *       1) INDEX VARIABLE ADDRESS (2BYTES)
                                1783  *       2) STEP VALUE (4 BYTES)
                                1784  *       3) FINAL VALUE (4 BYTES)
                                1785  *       4) LOOP ADDRESS (2 BYTES)
                                1786  *
                                1787  *       IF THE 'FOR' VARIABLE IS ALREADY PRESENT IN THE 'FOR' STACK,
                                1788  *       REMOVE IT AND THEN ADD IT TO THE END.
                                1789
                                1790
047.060                         1791  FOR     EQU     *
047.060  315 362 077           1792          CALL    SFS             SEARCH 'FOR' STACK
047.063  315 000 073           1793          CALL    CSI             CONVERT TO INDEX                /80.01.GC/
047.066  325                   1794          PUSH    D
047.067  302 104 047           1795          JNZ     FOR1            NONE PRE-EXISTING
047.072  053                   1796          DCX     H
047.073  053                   1797          DCX     H
047.074  021 014 000           1798          LXI     D,12
047.077  315 203 104           1799          CALL    $DBT            REMOVE FROM TABLE
047.102  133 112               1800          DW      FORTAB+1
                                1801
                                1802  *       ALLOCATE SPACE FOR ENTRY.
                                1803
047.104                         1804  FOR1    EQU     *
047.104  041 014 000           1805          LXI     H,12
047.107  021 133 112           1806          LXI     D,FORTAB+1
047.112  315 026 071           1807          CALL    AMB             ALLOCATE 12 BYTES
047.115  321                   1808          POP     D               (DE) = FOR INDEX
047.116  315 366 072           1809          CALL    CSA             CONVERT BACK TO ABS. AFTER DEL  /80.01.GC/
                                1810
                                1811  *       STORE THE KEY ENTRY
                                1812
047.121  033                   1813          DCX     D                               /80.01.GC/
047.122  033                   1814          DCX     D                               /80.01.GC/
047.123  032                   1815          LDAX    D                               /80.01.GC/
047.124  167                   1816          MOV     M,A                             /80.01.GC/
047.125  023                   1817          INX     D                               /80.01.GC/
047.126  043                   1818          INX     H                               /80.01.GC/
```

```
047.127  032          1819          LDAX    D                                       /80.01.GC/
047.130  167          1820          MOV     M,A                                     /80.01.GC/
047.131  023          1821          INX     D                                       /80.01.GC/
047.132  043          1822          INX     H                                       /80.01.GC/
047.133  315 000 073  1823          CALL    CSI             CONVERT IT TO AN INDEX  /80.01.GC/
                      1824
047.136  076 300      1825          MVI     A,CT.SNV
047.140  315 377 050  1826          CALL    LET.            ASSIGN VALUE
047.143  315 305 077  1827          CALL    RNT
047.146  317          1828          DB      CT.TO           REQUIRE *TO*
047.147  315 022 057  1829          CALL    EVALN
047.152  043          1830          INX     H               GO PAST 'STEP' VALUE
047.153  043          1831          INX     H
047.154  043          1832          INX     H
047.155  043          1833          INX     H
047.156  315 051 076  1834          CALL    MOV4            STORE LIMIT
047.161  021 370 377  1835          LXI     D,-8
047.164  031          1836          DAD     D               (HL) = ADDRESS FOR STEP
047.165  315 056 071  1837          CALL    ANT             ACCEPT NEXT TOKEN
047.170  021 211 112  1838          LXI     D,FP1.0
047.173  376 211      1839          CPI     CT.STE
047.175  314 022 057  1840          CE      EVALN           EVALUATE STEP VALUE
047.200  315 051 076  1841          CALL    MOV4            STORE STEP
047.203  043          1842          INX     H               SKIP 'LIMIT'
047.204  043          1843          INX     H
047.205  043          1844          INX     H
047.206  043          1845          INX     H
047.207  161          1846          MOV     M,C
047.210  043          1847          INX     H
047.211  160          1848          MOV     M,B             STORE STATEMENT RETURN ADDRESS
047.212  311          1849          RET


                      1851  **      FREE - TYPE FREE SPACE.
                      1852  *
                      1853  *       FREE
                      1854
                      1855
047.213               1856  FREE    EQU     *
047.213  305          1857          PUSH    B               SAVE (BC)
047.214  041 123 112  1858          LXI     H,MTABIND+MT.LEN
047.217  345          1859          PUSH    H               SAVE TABLE INDEX ON STACK
047.220  006 021      1860          MVI     B,MTABL*2+1     (B) = NUMBER OF TABLES * 2 +1
047.222  041 272 047  1861          LXI     H,FREEA         (HL) = HEADER MESSAGES
                      1862
047.225  377 003      1863  FREE1   DB      SYSCALL+.PRINT  PRINT HEADER
047.227  315 136 031  1864          CALL    $TYPTX
047.232  040 075 240  1865          DB      ' =',' '+200Q
047.235  343          1866          XTHL                    (HL) = ADDRESS OF INDEX
047.236  136          1867          MOV     E,M
047.237  043          1868          INX     H
047.240  126          1869          MOV     D,M
047.241  043          1870          INX     H
047.242  043          1871          INX     H
```

```
047.243  043          1872          INX     H
047.244  043          1873          INX     H
047.245  343          1874          XTHL
047.246  005          1875          DCR     B
047.247  304 264 047  1876          CNZ     TDI.            TYPE VALUE IF NOT LAST ONE
047.252  005          1877          DCR     B
047.253  362 225 047  1878          JP      FREE1           MORE TO GO
047.256  341          1879          POP     H               DISCARD TABLE ADDRESS
047.257  315 127 072  1880          CALL    $CFS            COMPUTE FREE SPACE
047.262  353          1881          XCHG
047.263  301          1882          POP     B               RESTORE (BC)
                      1883
                      1884  **      TDI. - TYPE DECIMAL INTEGER FOLLOWED BY $CRLF
                      1885
047.264  315 206 100  1886  TDI.    CALL    TDI
047.267  303 354 111  1887          JMP     $CRLF
                      1888
047.272               1889  FREEA   EQU     *               TABLE OF TABLE NAMES
047.272  124 145 170  1890          DB      'Tex','t'+200Q
047.276  123 171 155  1891          DB      'Sym','b'+200Q
047.302  106 157 162  1892          DB      'For','l'+200Q
047.306  107 163 165  1893          DB      'Gsu','b'+200Q
047.312  127 157 162  1894          DB      'Wor','k'+200Q
047.316  123 164 162  1895          DB      'Str','n'+200Q
047.322  124 123 164  1896          DB      'TSt','r'+200Q
047.326  106 151 154  1897          DB      'Fil','e'+200Q
047.332  106 162 145  1898          DB      'Fre','e'+200Q


                      1900  ***     FREEZE - FREEZE PROGRAM AND BASIC.
                      1901  *
                      1902  *       FREEZE <STRING>
                      1903  *
                      1904  *       FREEZE THE BASIC PROGRAM, BASIC, AND ALL MEMORY ONTO
                      1905  *       FILE 'STRING'
                      1906
                      1907
047.336               1908  FREEZE  EQU     *
047.336  315 041 072  1909          CALL    CFN.            COPY FILE NAME, DO FILE OPEN PRESET
047.341  257          1910          XRA     A
047.342  315 005 072  1911          CALL    CFA             PRESET FOR I/O OPERATION
047.345  021 057 054  1912          LXI     D,UNFREZA       (DE) = DEFAULTS
047.350  315 030 101  1913          CALL    $FOPEW          OPEN FOR WRITE
047.353  345          1914          PUSH    H               SAVE FB ADDRESS
047.354  052 171 112  1915          LHLD    MEML
047.357  021 200 335  1916          LXI     D,-USERFWA
047.362  031          1917          DAD     D               (HL) = LENGTH
047.363  042 022 050  1918          SHLD    FREZEB
047.366  021 016 050  1919          LXI     D,FREZEA        (DE) = HEADER ADDRESS
047.371  343          1920          XTHL                    (HL) = FB ADDRESS, ((SP)) = LEN
047.372  001 010 000  1921          LXI     B,FREZEAL
047.375  315 047 102  1922          CALL    $FWRIB          WRITE HEADER
050.000  301          1923          POP     B               (BC) = LEN OF PROGRAM
050.001  021 200 042  1924          LXI     D,USERFWA
```

```
050.004  315 047 102  1925          CALL    $FWRIB          WRITE IT
050.007  315 335 102  1926          CALL    $FCLO           CLOSE FILE
050.012  001 007 115  1927          LXI     B,ZERO          NO MORE TEXT LINE
050.015  311          1928          RET                     LET HIM KEEP RUNNING
                      1929
050.016  377 000      1930  FREZEA  DB      377Q,FT.ABS     ABS HEADER FOR IMAGE
050.020  200 042      1931          DW      USERFWA
050.022  000 000      1932  FREZEB  DW      0               LENGTH
050.024  106 043      1933          DW      START           ENTRY ADDRESS
000.010               1934  FREZEAL EQU     *-FREZEA        LENGTH OF HEADER


                      1936  **      GOSUB - CALL SUBROUTINE.
                      1937  *
                      1938  *       GOSUB EXIR
                      1939
                      1940
050.026               1941  GOSUB   EQU     *
050.026  315 143 100  1942          CALL    SRA             STACK RETURN ADDRESS
                      1943  *       JMP     GOTO            PROCESS AS GOTO


                      1945  **      GOTO - GO TO STATEMENT.
                      1946  *
                      1947  *       GOTO EXPR
                      1948
                      1949
050.031               1950  GOTO    EQU     *
050.031  315 033 074  1951          CALL    ELN             EVAL LINE NUMBER
050.034  315 242 074  1952  GOTO1   CALL    FLN             FIND LINE BY NUMBER
050.037  332 147 070  1953          JC      ERR.SN          CANT FIND IT
                      1954
050.042               1955  GOTO2   EQU     *
050.042  053          1956          DCX     H               (HL) = PREVIOUS LINE TERMINATOR
050.043  104          1957          MOV     B,H
050.044  115          1958          MOV     C,L
050.045  042 177 112  1959          SHLD    CURADR          SAVE CURRENT TEXT ADDRESS
050.050  311          1960          RET                     LET EXEC 'FIND' NEW LINE


                      1962  **      IF - PROCESS IF STATEMENT.
                      1963  *
                      1964  *       IF EXPR THEN <STATEMENT>
                      1965  *       IF EXPR THEN <STATEMENT NUMBER>
                      1966
                      1967
050.051               1968  IF      EQU     *
050.051  315 036 057  1969          CALL    EVALI           EVALUATE EXPRESSION
                      1970
                      1971  *       WILL EXECUTE. REQUIRE 'THEN'
```

```
                                1972
050.054  315 056 071           1973          CALL    ANT             GET NEXT TOKEN
050.057  376 225               1974          CPI     CT.GOT
050.061  312 127 050           1975          JE      IF3             IS IF <EXPR> GOTO <EXPR>
050.064  376 316               1976          CPI     CT.THN
050.066  302 152 070           1977          JNE     ERR.SY          NOT *THEN*
050.071  173                   1978          MOV     A,E             (A) = TEST CODE
050.072  037                   1979          RAR
050.073  322 121 050           1980          JNC     IF2             FALSE - WILL SKIP
050.076  315 126 100           1981          CALL    SOB             SKIP OVER BLANKS
050.101  012                   1982          LDAX    B
050.102  376 060               1983          CPI     '0'
050.104  332 114 050           1984          JC      IF0             NOT DIGIT - MUST BE STATEMENT
050.107  376 072               1985          CPI     '9'+1
050.111  332 031 050           1986          JC      GOTO            IS DIGIT - MUST BE GOTO
050.114  341                   1987  IF0     POP     H
050.115  303 244 043           1988          JMP     EXEC1           PROCESS AS STATEMENT
                                1989
                                1990  *       SKIP REST OF LINE.
                                1991
050.120  003                   1992  IF1     INX     B
050.121  012                   1993  IF2     LDAX    B
050.122  247                   1994          ANA     A
050.123  302 120 050           1995          JNZ     IF1             SKIP STATEMENT
050.126  311                   1996          RET                     DONE
                                1997
                                1998  *       IF <EXPR> GOTO <EXPR>
                                1999
050.127  173                   2000  IF3     MOV     A,E
050.130  037                   2001          RAR
050.131  322 121 050           2002          JNC     IF2             IF TO SKIP
050.134  303 031 050           2003          JMP     GOTO            PROCESS GOTO


                                2005  **      LINE INPUT - INPUT ONE LINE FROM CONSOLE.
                                2006  *
                                2007  *       SAME AS *INPUT*, EXCEPT THAT THE FIRST VARIABLE MUST
                                2008  *       BE A STRING VARIABLE, AND THE FIRST LINE IS
                                2009  *       TAKEN AS THE VALUE.
                                2010
                                2011
050.137                         2012  LINPUT  EQU     *
050.137  315 305 077           2013          CALL    RNT
050.142  254                   2014          DB      CT.INP          REQUIRE *INPUT*
050.143  076 001               2015          MVI     A,1
050.145  303 151 050           2016          JMP     INP1            PROCESS AS INPUT
```

```
                        2018  **      INPUT - INPUT FROM CONSOLE.
                        2019  *
                        2020  *       INPUT "PROMPT";V1,...,VN
                        2021
                        2022
050.150  257           2023  INPUT   XRA     A
050.151  062 370 050    2024  INP1    STA     INPUTA          SAVE FLAG FOR LINE INPUT
050.154  315 253 073    2025          CALL    DCN             DECODE CHANNEL NUMBER
050.157  305            2026          PUSH    B               SAVE (BC)(
050.160  315 016 112    2027          CALL    $CCO            CLEAR CTL-O
050.163  301            2028          POP     B
050.164  315 072 076    2029          CALL    PNT             PEEK AT NEXT TOKE
050.167  041 371 050    2030          LXI     H,INPUTB        ASSUME '?' PROMPT
050.172  376 027        2031          CPI     CT.SEM
050.174  302 205 050    2032          JNE     INP2            MAY HAVE PROMPT
050.177  315 056 071    2033          CALL    ANT             NO PROMPT, GOBBLE ;
050.202  303 233 050    2034          JMP     INP4            PROVIDE DEFALT PROMPT
                        2035
050.205  376 301        2036  INP2    CPI     CT.SSV          SCALAR STRING VALUE
050.207  302 233 050    2037          JNE     INP4            NO PROMPT
                        2038
                        2039  *       HAVE PROMPT
                        2040
050.212  072 202 112    2041          LDA     IOCHAN
050.215  247            2042          ANA     A
050.216  314 200 100    2043          CZ      TCS             TYPE CHARACTER STRING IFF CONSOLE INPUT
050.221  315 056 071    2044          CALL    ANT             ACCEPT ALREADY PROCESSED STRING
050.224  315 305 077    2045          CALL    RNT
050.227  027            2046          DB      CT.SEM          REQUIRE ;
050.230  041 373 050    2047          LXI     H,INPUTC        SUPPRESS OUR PROMPT
                        2048
                        2049  *       READY TO INPUT VALUES
                        2050
050.233  072 202 112    2051  INP4    LDA     IOCHAN
050.236  247            2052          ANA     A               SEE IF OUTPUT TO CONSOLE
050.237  302 244 050    2053          JNZ     INP4.5          DISK I/O, NO PROMPT
050.242  377 003        2054          DB      SYSCALL,.PRINT  PRINT PROMPT
050.244                 2055  INP4.5  EQU     *                                       /80.01.GC/
                        2056
                        2057  *       MAKE SURE WE HAVE VARIABLES
                        2058
050.244  315 072 076    2059          CALL    PNT                                     /80.01.GC/
050.247  376 300        2060          CPI     CT.VARL         LOWEST VARIABLE         /80.01.GC/
050.251  332 152 070    2061          JC      ERR.SY          < LOWEST VARIABLE       /80.01.GC/
050.254  376 310        2062          CPI     CT.VARH+1                               /80.01.GC/
050.256  322 152 070    2063          JNC     ERR.SY          > HIGHEST VARIABLE      /80.01.GC/
                        2064
050.261  041 330 112    2065          LXI     H,LINE+1                                /80.01.GC/
050.264  315 141 077    2066          CALL    RLF             READ LINE FROM FILE
050.267  332 106 070    2067          JC      ERR.CC          IF CTL-C HIT
050.272  072 370 050    2068          LDA     INPUTA
050.275  247            2069          ANA     A
050.276  041 330 112    2070          LXI     H,LINE+1        ASSUME START LINE AT FIRST CHARACTER
050.301  312 356 050    2071          JZ      INP6            IS REGULAR INPUT
                        2072
                        2073  *       IS LINE INPUT, ENCLOSE LINE IN QUOTES
```

```
                              2074
050.304  315 072 076         2075  INP5    CALL    PNT             CHECK INPUT VARIABLE
050.307  346 375             2076          ANI     377Q-CF.VEC     ALLOW VECTORS, TOO
050.311  376 301             2077          CPI     CT,SSV
050.313  302 152 070         2078          JNE     ERR.SY          MUST BE SCALAR STRING VALUE
050.316  345                 2079          PUSH    H               SAVE DATA POINTER
050.317  315 136 075         2080          CALL    IST             INSERT SYMBOL IN TABLE
050.322  341                 2081          POP     H               (HL) = DATA POINTER
050.323  325                 2082          PUSH    D               SAVE TARGET VARIABLE INDEX
050.324  305                 2083          PUSH    B               SAVE TEXT POINTER
050.325  104                 2084          MOV     B,H
050.326  115                 2085          MOV     C,L             (BC) = INPUT TEXT ADDRESS
050.327  315 012 055         2086          CALL    LEX11.5         BUILD INTO STRING
050.332  001 005 000         2087          LXI     B,5
050.335  033                 2088          DCX     D
050.336  041 201 042         2089          LXI     H,ACCX-1
050.341  315 252 030         2090          CALL    $MOVE           MOVE TEMP DESCRIPTOR INTO ACCX
050.344  301                 2091          POP     B
050.345  321                 2092          POP     D
050.346  315 366 072         2093          CALL    CSA             CONVERT INDEX TO ABSOLUTE
050.351  076 301             2094          MVI     A,CT.SSV        IS STRING ASSIGNMENT
050.353  303 202 071         2095          JMP     AVV             ASSIGN VALUE TO VARIABLE, EXIT 'INPUT' PROCESSING
                              2096
                              2097  *       ASSIGN VALUES
                              2098
050.356  315 135 076         2099  INP4    CALL    PVI             PERFORM VALUE INPUT
050.361  310                 2100          RE                      DONE
050.362  041 371 050         2101          LXI     H,INPUTB        USE '?' PROMPT
050.365  303 233 050         2102          JMP     INP4            INPUT MORE
                              2103
050.370  000                 2104  INPUTA  DB      0               <>0 IF LINE INPUT
050.371  077 240             2105  INPUTB  DB      '?',' '+200Q    DEFAULT PROMPT
050.373  200                 2106  INPUTC  DB      200Q            NULL PROMPT


                              2108  **      LET - ASSIGN VALUE.
                              2109  *
                              2110  *       LET VAL = EXPR
                              2111
                              2112
050.374  315 136 075         2113  LET     CALL    IST             PREPARE VALUE FOR ASSIGNMENT
                              2114
050.377  365                 2115  LET.    PUSH    PSW             SAVE TYPE
                              2116
051.000  325                 2117          PUSH    D               SAVE INDEX
051.001  315 305 077         2118          CALL    RNT
051.004  011                 2119          DB      CT.EQ           REQUIRE =
051.005  315 244 055         2120          CALL    EVAL            (ACCX) = VALUE
051.010  321                 2121          POP     D               (DE) = VALUE INDEX
051.011  315 366 072         2122          CALL    CSA             (DE) = ABSOLUTE ADDRESS INTO SYMTAB
051.014  361                 2123          POP     PSW             (A) = TYPE
051.015  303 202 071         2124          JMP     AVV             ASSIGN VALUE TO VARIABLE
```

```
                        2126  **      LIST - PROCESS LIST COMMAND.
                        2127  *
                        2128  *       LIST                    LIST ALL
                        2129  *       LIST NNN                LIST NNN
                        2130  *       LIST NNN,MMM            LIST NNN TO MMM
                        2131  *
                        2132  *       LIST [#CHAN][;NNN[,MMM]] ETC.                                   /78.10.GC/
                        2133
051.020                 2134  LIST    EQU     *
                        2135
051.020  315 253 073    2136          CALL    DCN             DECODE CHANNEL NUMBER                   /78.10.GC/
                        2137
                        2138  *       DECODE RANGE.
                        2139
051.023  021 000 000    2140  LIST.   LXI     D,0
051.026  325            2141          PUSH    D               SET DEFAULT NN
051.027  033            2142          DCX     D
051.030  033            2143          DCX     D               (DE) = 377376A
051.031  315 072 076    2144          CALL    PNT             PEEK AT NEXT TOKEN
000.000                 2145          ERRNZ   CT.FIN
051.034  247            2146          ANA     A
051.035  312 065 051    2147          JZ      LIST1           IS LIST 0;377376A
051.040  315 036 057    2148          CALL    EVALI           (DE) = NNN
051.043  341            2149          POP     H               DISARED DEFAULT FIRST
051.044  325            2150          PUSH    D               SET 1ST = LAST = NNN
051.045  315 072 076    2151          CALL    PNT             PEEK AT NEXT TOKEN
000.000                 2152          ERRNZ   CT.FIN
051.050  247            2153          ANA     A
051.051  312 065 051    2154          JZ      LIST1           IS NNN
051.054  315 223 072    2155          CALL    CMA             REQUIRE ','
051.057  315 036 057    2156          CALL    EVALI           IS NNN,MMM
051.062  315 242 074    2157          CALL    FLN             CHECK VALIDITY OF LAST LINE NUMBER  /78.10.GC/
                        2158
                        2159  *       LIST TEXT
                        2160
051.065  341            2161  LIST1   POP     H               (HL) = START
051.066  325            2162          PUSH    D               SAVE END
051.067  353            2163          XCHG                    (DE) = 1ST, ((SP)) = LAST
051.070  315 242 074    2164          CALL    FLN             FIND LINE BY NUMBER
                        2165
                        2166  *       SEE IF OFF THE END
                        2167
051.073  116            2168  LIST2   MOV     C,M
051.074  043            2169          INX     H
051.075  106            2170          MOV     B,M             (BC) = LINE NUMBER OF NEXT LINE
051.076  043            2171          INX     H
051.077  343            2172          XTHL                    (HL) = LIMIT
051.100  175            2173          MOV     A,L
051.101  221            2174          SUB     C               COMPARE TO CURRENT
051.102  174            2175          MOV     A,H
051.103  230            2176          SBB     B
051.104  332 170 051    2177          JC      LIST6           ALL DONE
051.107  343            2178          XTHL                    RESTORE LIMIT
051.110  345            2179          PUSH    H               SAVE LINE ADDRESS
051.111  041 335 113    2180          LXI     H,LINE2
051.114  076 005        2181          MVI     A,5
```

```
051.116  315 157 031  2182          CALL    $UDD            UNPACK DECIMAL DIGITS
051.121  066 040      2183          MVI     M,' '           ADD BLANK
051.123  043          2184          INX     H
051.124  353          2185          XCHG                    (DE) = LINE ADDRESS
051.125  341          2186          POP     H               (HL) = PROGRAM TEXT ADDRESS
051.126  176          2187  LIST3   MOV     A,M             (A) = NEXT CHARACTER
051.127  043          2188          INX     H
051.130  247          2189          ANA     A
051.131  374 374 073  2190          CM      EKA             EXPAND KEYWORD TO ASCII
051.134  022          2191          STAX    D               STORE IN LISTING LINE
051.135  023          2192          INX     D
051.136  247          2193          ANA     A
051.137  302 126 051  2194          JNZ     LIST3           MORE TO GO
                      2195
                      2196  *       SEE IF TO WRITE TO FILE, OR TO CONSOLE
                      2197
051.142  345          2198          PUSH    H               SAVE PROGRAM TEXT ADDRESS
051.143  353          2199          XCHG                    (HL) = LINE NEXT ADDRESS
051.144  053          2200          DCX     H               BACKUP OVER END OF LINE
051.145  066 012      2201          MVI     M,NL
051.147  043          2202          INX     H
051.150  066 000      2203          MVI     M,0             ADD END OF LINE
051.152  315 242 100  2204          CALL    WLF             WRITE LINE TO FILE
051.155  341          2205          POP     H               (HL) = TEXT FWA
051.156  072 204 112  2206          LDA     CTLFLAG
000.000               2207          ERRNZ   CFCTLC-1
051.161  037          2208          RAR
051.162  332 106 070  2209          JC      ERR.CC          CTL-C STRUCK
051.165  303 073 051  2210          JMP     LIST2           DO NEXT
                      2211
                      2212  *       ALL DONE.
                      2213
051.170  341          2214  LIST6   POP     H
051.171  001 007 115  2215          LXI     B,ZERO          END OF COMMAND LINE
051.174  311          2216          RET


                      2218  **      LOCK - LOCK OUT DATA CHANGE
                      2219  *
                      2220  *       LOCK PREVENTS ANY DATA OR LINES OF TEXT TO BE
                      2221  *       CHANGED
                      2222  *
                      2223
                      2224  **      UNLOCK - ENABLE DATA CHANGE
                      2225  *
                      2226  *       UNLOCK CLEARS THE LOCK FLAG ENABLING
                      2227  *       DATA CHANGES
                      2228  *
                      2229  *       UNLOCK
                      2230  *       LOCK
                      2231
                      2232
                      2233  ************************************************************************
                      2234  *                                                                      *
```

```
                                2235  *       LOCK USES THE XRA  A  OPCODE (257) AS THE VALUE TO PUT IN      *
                                2236  *       LCKFLG THROUGH THE USE OF THE MVI INSTRUCTION IMPLEMENTED      *
                                2237  *       TO USE THE XRA  A  OPCODE AS THE SECOND BYTE OF THE MVI        *
                                2238  *       INSTRUCTION.                                                   *
                                2239  *                                                                      *
                                2240  ************************************************************************
                                2241
                                2242
                                2243
051.175  076                   2244  LOCK    DB      MI.MVIA         MVI OPCODE
                                2245
051.176  257                   2246  UNLOCK  XRA     A
051.177  062 201 112           2247          STA     LCKFLG          STORE EITHER 0 OR 257 (FROM XRA OPCODE)
051.202  311                   2248          RET                     EXIT


                                2250  **      NEXT - PROCESS NEXT.
                                2251  *
                                2252  *       NEXT VAR
                                2253  *
                                2254  *       PERFORM LOOPING FOR REQUESTED VARIABLE. IF NOT THE MOST
                                2255  *       RECENT, DISCARD 'FORTAB' ENTRYS UNTIL IS FOUND.
                                2256
                                2257
051.203                         2258  NEXT    EQU     *
051.203  315 352 077           2259          CALL    SFS.            SEARCH 'FOR' STACK
051.206  302 133 070           2260          JNZ     ERR.NV          NEXT MISSING VARIABLE
051.211  345                   2261          PUSH    H               SAVE FORTAB INDEX
                                2262  **      CALL    CSA             (DE) = ABS. ADDR. OF VARIABLE   /80.01.GC/
051.212  315 210 073           2263          CALL    CVX             COPY VALUE TO ACCX
051.215  315 000 073           2264          CALL    CSI             (DE) = INDEX INTO SYMTAB OF VARIBLE
051.220  341                   2265          POP     H               (HL) = FORTAB INDEX
051.221  325                   2266          PUSH    D               SAVE INDEX ADDRESS
051.222  353                   2267          XCHG
051.223  041 012 000           2268          LXI     H,12-2
051.226  031                   2269          DAD     D               (HL) = NEW TABLE LENGTH
051.227  042 135 112           2270          SHLD    FORTAB+MT.LEN   DISCARD ANY MORE INNER ENTRYS
051.232  052 133 112           2271          LHLD    FORTAB+MT.FWA
051.235  031                   2272          DAD     D               (HL) = TABLE ADDRESS
051.236  353                   2273          XCHG
051.237  315 352 104           2274          CALL    FPADD           ADD STEP TO INDEX
051.242  353                   2275          XCHG                    (HL) = ADDRESS OF STEP VALUE
051.243  321                   2276          POP     D               (DE) = ADDRESS OF INDEX
051.244  315 366 072           2277          CALL    CSA             (DE) = ABS. ADDR. OF VARIBLE
051.247  315 237 073           2278          CALL    CXV             COPY ACCX TO VALUE
                                2279
                                2280  *       COMPARE RESULT TO LIMIT.
                                2281  *
                                2282  *       IF INC >= 0, VAL-LIM>0 => TERMINATE
                                2283  *       IF INC < 0, LIM-VAL>0 => TERMINATE
                                2284
051.252  043                   2285          INX     H
051.253  043                   2286          INX     H
051.254  176                   2287          MOV     A,M             (A) = SIGN OF INCREMENT
```

```
051.255  043            2288          INX     H
051.256  043            2289          INX     H
051.257  345            2290          PUSH    H               SAVE ADDRESS OF LIMIT
051.260  247            2291          ANA     A
051.261  353            2292          XCHG                    (DE) = ADDRESS OF LIMIT
051.262  372 271 051     2293          JM      NXT1            IS < 0
                        2294
                        2295  *       COMPUTE VALUE-LIMIT
                        2296
051.265  315 210 073     2297          CALL    CVX             (ACCX) = LIMIT
051.270  353            2298          XCHG                    (DE) = ADDRESS OF VALUE
                        2299
                        2300  *       COMPUTE LIMIT-VALUE
                        2301
051.271  315 166 105     2302  NXT1    CALL    FPSUB           COMPARE
051.274  341            2303          POP     H               (HL) = ADDRESS OF LIMIT
051.275  072 204 042     2304          LDA     ACCX+2
051.300  247            2305          ANA     A
051.301  312 307 051     2306          JZ      NXT1.5          IS MATCH
051.304  362 317 051     2307          JP      NXT2            ALL DONE
                        2308
                        2309  *       LOOP TO AFTER 'FOR' STATEMENT
                        2310
051.307  043            2311  NXT1.5  INX     H
051.310  043            2312          INX     H
051.311  043            2313          INX     H
051.312  043            2314          INX     H
051.313  116            2315          MOV     C,M
051.314  043            2316          INX     H
051.315  106            2317          MOV     B,M
051.316  311            2318          RET
                        2319
                        2320  *       DONE. COLLAPSE 'FOR' OUT OF TABLE.
                        2321
051.317  052 135 112     2322  NXT2    LHLD    FORTAB+MT.LEN
051.322  021 364 377     2323          LXI     D,-12
051.325  031            2324          DAD     D
051.326  042 135 112     2325          SHLD    FORTAB+MT.LEN
051.331  311            2326          RET
```

```
                        2328  ***     OLD - GET NEW PROGRAM.
                        2329  *
                        2330  *       OLD     <STRING>
                        2331  *
                        2332  *       OLD CLEARS ALL THE TABLE, THEN LOADS A PROGRAM.
                        2333
                        2334
051.332                 2335  OLD     EQU     *
051.332  315 053 072     2336          CALL    CFN             COPY FILE NAME
051.335  041 203 112     2337          LXI     H,OVLMAN
051.340  176            2338          MOV     A,M             (A) = CURRENT VALUE
051.341  064            2339          INR     M               MAKE NON-ZERO
051.342  345            2340          PUSH    H
```

```
051.343  365           2341          PUSH    PSW
051.344  315 360 044   2342          CALL    SCR.            CLEAR PROGRAM
051.347  361           2343          POP     PSW
051.350  341           2344          POP     H
051.351  167           2345          MOV     M,A             RESTORE FLAG
051.352  303 206 077   2346          JMP     RNP             READ NEW PROGRAM AND RETURN


                       2348  **      ON - PROCESS 'ON' STATEMENT.
                       2349  *
                       2350  *       ON EXPR GOTO EXP1,...,EXPN
                       2351  *       ON EXPR GOSUB EXP1,...,EXPN
                       2352  *
                       2353  *       IF EXPR < 0, FLAG ERROR
                       2354  *       IF EXPR = 1,...,N TAKE EXP1,...,EXPN
                       2355  *       IF EXPR>N TAKE EXPN
                       2356  *       IF EXPR=0 TAKE EXPN
                       2357
                       2358
051.355  315 036 057   2359  ON      CALL    EVALI           EVALUATE INTEGER
051.360  353           2360          XCHG                    (HL) = INDEX
051.361  315 056 071   2361          CALL    ANT             ACCEPT NEXT TOKEN
051.364  376 225       2362          CPI     CT.GOT
051.366  312 001 052   2363          JE      ON1             GOTO
051.371  376 224       2364          CPI     CT.GOS
051.373  302 152 070   2365          JNE     ERR.SY          NOT GOSUB
051.376  315 143 100   2366          CALL    SRA             SET RETURN ADDRESS
052.001                2367  ON1     EQU     *
                       2368
                       2369  *       SKIP DOWN LIST UNTIL INDEX FOUND.
                       2370
052.001  315 033 074   2371  ON2     CALL    ELN             EVALUATE LINE NUMBER
052.004  315 056 071   2372          CALL    ANT             GET DELIMITER
052.007  062 035 052   2373          STA     ONA             SAVE FOR LATER EXAM
052.012  053           2374          DCX     H
052.013  174           2375          MOV     A,H
052.014  265           2376          ORA     L
052.015  312 034 050   2377          JZ      GOTO1           HAVE PROPER LABEL
052.020  072 035 052   2378          LDA     ONA
052.023  247           2379          ANA     A
000.000                2380          ERRNZ   CT.FIN
052.024  310           2381          RZ                      END OF LINE
052.025  376 026       2382          CPI     CT.CMA
052.027  302 152 070   2383          JNE     ERR.SY
052.032  303 001 052   2384          JMP     ON2
                       2385
052.035  000           2386  ONA     DB      0               TEMP AREA
```

```
                        2388  ***     OPEN - OPEN FILE
                        2389  *
                        2390  *       OPEN <STRING> FOR <VERB> AS FILE #N
                        2391  *
                        2392  *       <STRING> = STRING CONTAINING FILE NAME
                        2393  *
                        2394  *       <VERB> = READ           OPEN FILE FOR READ ACCESS
                        2395  *       <VERB> = WRITE          OPEN FILE FOR WRITE ACCESS
                        2396
                        2397
052.036                 2398  OPEN    EQU     *
052.036 315 053 072     2399          CALL    CFN             CRACK FILE NAME
052.041 315 305 077     2400          CALL    RNT
052.044 221             2401          DB      CT.FOR          REQUIRE FOR
052.045 315 056 071     2402          CALL    ANT
052.050 376 241         2403          CPI     CT.REA
052.052 312 062 052     2404          JE      OPEN1           VERB IS OK
052.055 376 313         2405          CPI     CT.WRI
052.057 302 152 070     2406          JNE     ERR.SY          NOT A GOOD VERB
052.062 365             2407  OPEN1   PUSH    PSW             SAVE VERB TOKEN
052.063 315 305 077     2408          CALL    RNT
052.066 311             2409          DB      CT.AS
052.067 315 305 077     2410          CALL    RNT
052.072 312             2411          DB      CT.FIL          FILE
052.073 315 273 073     2412          CALL    DCN,            DECODE CHANNEL NUMBER, NO COMMA
052.076 315 302 075     2413          CALL    LCC             LOCATE CHANNEL COLUMN NUMBER
052.101 066 001         2414          MVI     M,1             SET AT FRONT OF LINE
052.103 361             2415          POP     PSW             (A) = FUNCTION KEYWORD
052.104 305             2416          PUSH    B               SAVE TEXT POINTER
052.105 365             2417          PUSH    PSW             SAVE FUNCTION KEYWORD
052.106 072 202 112     2418          LDA     IOCHAN
052.111 075             2419          DCR     A               (A) = CHANNEL NUMBER
                        2420
                        2421  *       FIND THE FILE BLOCK, CREATE IT IF NECESSARY
                        2422
052.112 365             2423  OPEN2   PUSH    PSW             SAVE CHANNEL/BLOCK NUMBER
052.113 315 005 072     2424          CALL    CFA             COMPUTE FILEBLOCK ADDRESS
052.116 322 144 052     2425          JNC     OPEN3           GOTIT
                        2426
052.121 315 374 071     2427          CALL    CEF             CREATE EMPTY FILE BLOCK
052.124 072 167 112     2428          LDA     FILTAB+MT.LEN+1 A  = BUFFER JUST ADDED          /80.01.GC/
052.127 315 005 072     2429          CALL    CFA             HL = FILE-BLOCK ADDRESS         /80.01.GC/
052.132 332 160 070     2430          JC      ERR.TO          SHOULD NOT HAPPEN ! (NOT FOUND) /80.01.GC/
052.135 043             2431          INX     H                                               /80.01.GC/
000.000                 2432          ERRNZ   FB.FLG-1                                        /80.01.GC/
052.136 066 000         2433          MVI     M,0             ZERO THE FLAG                   /80.01.GC/
052.140 361             2434          POP     PSW
052.141 303 112 052     2435          JMP     OPEN2           SEE IF WE'VE CREATED ENOUGH
                        2436
                        2437  *       GOT THE FILE BLOCK,
                        2438  *       (HL) = FB FWA (ABS)
                        2439
052.144 043             2440  OPEN3   INX     H               (HL) = #FB.FLG
000.000                 2441          ERRNZ   FB.FLG-1
052.145 176             2442          MOV     A,M                                             /80.01.GC/
052.146 247             2443          ANA     A                                               /80.01.GC/
```

```
052.147  302 221 070  2444           JNZ     ERR.CIU         CHANNEL ALREADY IN USE          /80.01.6C/
                      2445
052.152  001 011 000  2446           LXI     B,FB.NAM-FB.FLG
052.155  011         2447           DAD     B               (HL) = ADDRESS FOR NAME IN FILE BLOCK
052.156  021 242 042  2448           LXI     D,FBLIST+FB.NAM (DE) = ADDRESS OF NAME IN SYSTEM FILE BLOCK
377.012               2449           ERRPL   FB.NAM-256      CODE ASSUMES 1 BYTE VALUE
377.021               2450           ERRPL   FB.NAML-256     CODE ASSUMES 1 BYTE VALUE
052.161  016 021      2451           MVI     C,FB.NAML       (BC) = #FB.NAML
052.163  315 252 030  2452           CALL    $MOVE           MOVE NAME TO PROPER BLOCK
052.166  315 217 074  2453           CALL    FOP             FILE OPEN PRESET
052.171  361         2454           POP     PSW             (A) = CHANNEL NUMBER
052.172  315 005 072  2455           CALL    CFA             COMPUTE FILE BLOCK ADDRESS
052.175  361         2456           POP     PSW             (A) = CT.REA OR CT.WRI
052.176  021 100 043  2457           LXI     D,DEFALTD       USE DATA DEFAULTS
052.201  315 210 052  2458           CALL    OPEN4           CALL OPEN ROUTINE
052.204  301         2459           POP     B               RESTORE TEXT POINTER
052.205  303 115 074  2460           JMP     FOC             FILE OPEN CLEANUP AND EXIT
                      2461
052.210  376 241      2462  OPEN4    CPI     CT.REA
052.212  312 021 101  2463           JE      $FOPER
052.215  303 030 101  2464           JMP     $FOPEW          OPEN FOR READ OR WRITE


                      2466  **       OUT - OUTPUT TO PORT.
                      2467  *
                      2468  *        OUT PORT,VALUE
                      2469
                      2470
052.220  315 235 052  2471  OUT      CALL    OUT1            EVALUATE PORT AND VALUE
052.223  145         2472           MOV     H,L             (H) = PORT
052.224  056 323      2473           MVI     L,MI.OUT
052.226  042 002 040  2474           SHLD    .IOWRK          SET VALUE
052.231  173         2475           MOV     A,E             (A) = VALUE
052.232  303 002 040  2476           JMP     .IOWRK          OUTPUT AND RETURN
                      2477
                      2478
                      2479  **       OUT1 - EVALUATE ADDRESS,VALUE
                      2480
052.235  315 036 057  2481  OUT1     CALL    EVAL1
052.240  325         2482           PUSH    D               SAVE ADDRESS
052.241  315 223 072  2483           CALL    CMA             REQUIRE ','
052.244  315 036 057  2484           CALL    EVAL1           (E) = VALUE
052.247  341         2485           POP     H               (HL) = ADDRESS
052.250  311         2486           RET
```

```
                                2488  **      PAUSE - PAUSE FOR TIME INTERVAL.
                                2489  *
                                2490  *       PAUSE <IEXP>
                                2491  *
                                2492  *       PAUSE FOR <IEXP>*2 MILLISECONDS. IF NO TIME IS
                                2493  *       SPECIFIED, PAUSE UNTIL A KEY IS STRUCK.
                                2494  *
                                2495  *       METHOD OF CALCULATION: (IF IEXP GIVEN)
                                2496  *
                                2497  *       AT EXAMINE TIME:
                                2498  *
                                2499  *       IF TARGET => TICCNT
                                2500  *                          THEN
                                2501  *                                  IF TAR. - TIC <> 0
                                2502  *                                  OR
                                2503  *                                  IF TAR. - TIC. < 377 000A  TIME UP
                                2504  *                                  ELSE WAIT
                                2505  *
                                2506  *       IF TARGET < TICCNT
                                2507  *                          THEN
                                2508  *                                  IF TIC. - TAR. < 000 377A , TIME UP
                                2509  *                                  ELSE WAIT
                                2510
                                2511
052.251                         2512  PAUSE   EQU     *
052,251  315 072 076           2513          CALL    PNT             CHECK NEXT TOKEN
000.000                         2514          ERRNZ   CT.FIN
052,254  247                   2515          ANA     A
052.255  312 233 103           2516          JZ      $RCHAR          NO PARAMETERS, JUST WAIT
                                2517
052.260  315 036 057           2518          CALL    EVALI           DECODE PAUSE INTERVAL
052,263  172                   2519          MOV     A,D             (A) = HIGH ORDER BYTE OF IEXP
052.264  074                   2520          INR     A
052,265  312 122 070           2521          JZ      ERR.IN          NUMBER TOO LARGE
052.270  052 033 040           2522          LHLD    .TICCNT
052,273  031                   2523          DAD     D               (HL) = TICCNT FINAL VALUE
052.274  353                   2524          XCHG
052,275  052 033 040           2525  PAUSE1  LHLD    .TICCNT         (HL) = TIC COUNTER
052.300  315 224 030           2526          CALL    $CHL            INVERT IT
052,303  031                   2527          DAD     D               TAR. - TIC.
052.304  332 320 052           2528          JC      PAUSE2          TAR. - TIC. => 0
                                2529
                                2530  *       TAR. < TIC.
                                2531
052.307  315 224 030           2532          CALL    $CHL            (HL) = TIC. - TAR.
052,312  174                   2533          MOV     A,H
052.313  247                   2534          ANA     A
052,314  310                   2535          RZ                      DONE
052.315  303 326 052           2536          JMP     PAUSE3          WAIT
                                2537
                                2538  *       TAR. => TIC.
                                2539
052.320  174                   2540  PAUSE2  MOV     A,H             CHECK FOR TAR. = TIC.
052,321  265                   2541          ORA     L
052.322  310                   2542          RZ                      DONE
                                2543
```

```
052.323  174           2544          MOV     A,H
052.324  074           2545          INR     A
052.325  310           2546          RZ                      DONE
                       2547
052.326  072 204 112   2548  PAUSE3  LDA     CTLFLAG
052.331  247           2549          ANA     A               SEE IF ANY CTL CHARACTERS HIT
052.332  300           2550          RNZ                     CONTROL CHARACTER HIT
052.333  303 275 052   2551          JMP     PAUSE1          CONTINUE WAITING


                       2553  **      POKE - WRITE VALUE INTO MEMORY.
                       2554  *
                       2555  *       POKE ADDR,VALUE
                       2556
                       2557
052.336                2558  POKE    EQU     *
052.336  315 235 052   2559          CALL    OUT1            READ ADDRESS AND VALUE
052.341  163           2560          MOV     M,E             SET VALUE
052.342  311           2561          RET


                       2563  ***     POSITION - SET FILE POSITION.
                       2564  *
                       2565  *       POSITION #N,IEXP
                       2566  *
                       2567  *       POSITION FILE #N AT BLOCK IEXP. FILE MUST BE OPEN FOR READ.
                       2568
                       2569


                       2571  **      PRINT - PROCESS PRINT STATEMENT.
                       2572  *
                       2573  *       PRINT VARLIST
                       2574  *
                       2575  *       IF VARIABLE SEPERATOR IS ',', TAB TO NEXT FIELD.
                       2576  *       IF SEPERATOR IS ';', DONT TAB.
                       2577  *       IF THE LAST TOKEN IN THE STATEMNET IS ',' OR ';', DONT
                       2578  *       CRLF AFTER LINE
                       2579
                       2580
052.343                2581  PRINT   EQU     *
052.343  257           2582          XRA     A
052.344  062 144 053   2583          STA     PRIA            CLEAR ';' OR ',' FLAG
052.347  315 253 073   2584          CALL    DCN             DECODE CHANNEL NUMBER
052.352  041 352 052   2585  PRI1    LXI     H,PRI1
052.355  345           2586          PUSH    H               SET 'RETURN ADDRESS'
052.356  315 072 076   2587          CALL    PNT             PREVIEW NEXT TOKEN
000.000                2588          ERRNZ   CT.FIN
052.361  247           2589          ANA     A
052.362  312 142 053   2590          JZ      PRI7            END OF STATEMENT
```

```
052.365  062 144 053  2591          STA     PRIA            SAVE TYPE
052.370  376 346      2592          CPI     CT.TAB
052.372  312 105 053  2593          JE      PRI6            TAB FUNCTION
052.375  376 343      2594          CPI     CT.SPC
052.377  312 105 053  2595          JE      PRI6            SPC FUNCTION
053.002  376 027      2596          CPI     CT.SEM
053.004  312 056 071  2597          JE      ANT             ACCEPT ; AND GO TO PRI1
053.007  376 026      2598          CPI     CT.CMA
053.011  312 040 053  2599          JE      PRI3            ,
                      2600
                      2601  *       MUST BE EXPRESSION.
                      2602
053.014  315 244 055  2603          CALL    EVAL            EVALUTE EXPRESSION
053.017  033          2604          DCX     D
053.020  032          2605          LDAX    D               (A) = TYPE
053.021  023          2606          INX     D
053.022  346 001      2607          ANI     CF.STR
053.024  302 200 100  2608          JNZ     TCS             IS STRING: TYPE CHARACTER STRING
                      2609
                      2610  *       HAVE NUMERIC VALUE.
                      2611
053.027  041 335 113  2612  PRI2    LXI     H,LINE2         USE SCRATCH AREA
053.032  315 301 110  2613          CALL    FTA             CONVERT FLOATING TO ASCII
053.035  303 251 100  2614          JMP     WLF.            WRITE LINE TO FILE AND RETURN TO PRI1
                      2615
                      2616  *       HAVE COMMA - SKIP TO NEXT FIELD
                      2617
053.040  315 056 071  2618  PRI3    CALL    ANT             ACCEPT ,
053.043  315 302 075  2619          CALL    LCC             LOCATE CHANNEL COLUMN COUNTER
053.046  176          2620          MOV     A,M             (A) = COLUMN COUNTER
053.047  376 072      2621          CPI     58
053.050               2622  PRIB    EQU     *-1             TAB LIMITS
053.051  322 225 100  2623          JNC     WEL             OVERFLOW - A NEW LINE
                      2624
                      2625  *       COMPUTE REQUIRED SPACES
                      2626
053.054  305          2627          PUSH    B                                       /80.01.GC/
                      2628
053.055  117          2629          MOV     C,A                                     /80.01.GC/
053.056  006 000      2630          MVI     B,0             BC = COLUMN COUNTER     /80.01.GC/
053.060  013          2631          DCX     B               ON RANGE [0,N]          /80.01.GC/
053.061  021 016 000  2632          LXI     D,14            DE = FIELD SIZE         /80.01.GC/
053.062               2633  PRIC    EQU     *-2                                     /80.01.GC/
053.064  315 106 030  2634          CALL    $DU66           DE = REMAINDER          /80.01.GC/
053.067  301          2635          POP     B                                       /80.01.GC/
053.070  072 062 053  2636          LDA     PRIC            A  = FIELD SIZE         /80.01.GC/
053.073  223          2637          SUB     E               A  = NUMBER OF SPACES REQUIRED  /80.01.GC/
                      2638
053.074  247          2639  PRI5    ANA     A
053.075  310          2640          RZ                      NO MORE SPACES          /80.01.GC/
053.076  315 156 053  2641          CALL    PRI8            OUTPUT A SPACE          /80.01.GC/
053.101  075          2642          DCR     A
053.102  303 074 053  2643          JMP     PRI5
                      2644
                      2645  *       HAVE TAB OR SPC FUNCTION
                      2646
```

```
053.105  315 056 071  2647  PRI6    CALL    ANT             ACCEPT TAB OR SPC
053.110  365         2648          PUSH    PSW             SAVE FUNCTION TYPE
053.111  315 044 057  2649          CALL    EVALI8          EVALUTE COUNT
053.114  315 305 077  2650          CALL    RNT
053.117  020         2651          DB      CT.PAR          REQUIRE ')'
053.120  361         2652          POP     PSW
053.121  376 343     2653          CPI     CT.SPC
053.123  173         2654          MOV     A,E             (A) = COUNT IF SPACE
053.124  312 074 053  2655          JE      PRI5            IS SPC                          /80.01.GC/
053.127  315 302 075  2656          CALL    LCC             LOCATE CHANNEL COLUMN COUNTER
053.132  176         2657          MOV     A,M             (A) = COLUMN
053.133  223         2658          SUB     E
053.134  057         2659          CMA
053.135  074         2660          INR     A                                               /78.10.GC/
053.136  362 074 053  2661          JP      PRI5            NOT PAST IT
053.141  311         2662          RET                     ALREADY PAST - DO NOTHING
                     2663
                     2664  *       HAVE END OF LINE
                     2665
053.142  341         2666  PRI7    POP     H               DISCARD 'RETURN' ADDRESS
053.143  076 000     2667          MVI     A,0
053.144              2668  PRIA    EQU     *-1
053.145  376 026     2669          CPI     CT.CMA          CHECK TEYP OF LAST TOKEN
053.147  310         2670          RE                      COMMA
053.150  376 027     2671          CPI     CT.SEM
053.152  310         2672          RE                      ;
053.153  303 225 100  2673          JMP     WEL             END LINE
                     2674
                     2675  *       OUTPUT A SPACE                                          /80.01.GC/
                     2676
053.156  365         2677  PRI8    PUSH    PSW                                             /80.01.GC/
053.157  041 251 112  2678          LXI     H,SPACE                                         /80.01.GC/
053.162  076 001     2679          MVI     A,1             COUNT = 1                       /80.01.GC/
053.164  315 251 100  2680          CALL    WLF.            WRITE CHARACTER TO THE FILE     /80.01.GC/
053.167  361         2681          POP     PSW
053.170  311         2682          RET


                     2684  **      READ - READ FROM DATA STATEMENT.
                     2685  *
                     2686  *       READ PERFORMS READS FROM DATA STATEMENTS.
                     2687  *
                     2688  *       THE 1ST DATA STATEMENT IS FOUND AND USED; THEN THE 2ND,
                     2689  *       ETC.
                     2690
                     2691
053.171              2692  READ    EQU     *
053.171  052 345 114  2693          LHLD    DATPTR          (HL) = DATA STATEMENT POINTER
053.174  315 135 076  2694  REA1    CALL    PVI             PERFORM VALUE INPUT
053.177  042 345 114  2695          SHLD    DATPTR          SAVE FOR NEXT TIME
053.202  310         2696          RE                      NO MORE DATA NEEDED
                     2697
                     2698  *       SCAN FOR NEXT DATA STATEMENT
                     2699
```

```
053.203  176           2700  REA2   MOV   A,M
053.204  043           2701         INX   H
053.205  247           2702         ANA   A
053.206  302 203 053   2703         JNZ   REA2          NOT AT END OF STATEMENT
053.211  176           2704         MOV   A,M
053.212  043           2705         INX   H
053.213  246           2706         ANA   M
053.214  043           2707         INX   H
053.215  074           2708         INR   A
053.216  312 114 070   2709         JZ    ERR.DE        DATA EXHAUSED AT LINE 377377A
053.221  176           2710         MOV   A,M
053.222  376 251       2711         CPI   CT.DAT
053.224  302 203 053   2712         JNE   REA2          NOT DATA
053.227  043           2713         INX   H
053.230  303 174 053   2714         JMP   REA1          READ NEW DATA STATEMENT


                       2716  **     REPLACE - SAVE PROGRAM OVERTOP ANY EXISTING PROGRAM.
                       2717  *
                       2718  *      REPLACE <STRING>
                       2719  *
                       2720  *      SAME AS SAVE, BUT DOESNT SQUAK IF ALREADY EXISTS.
                       2721
                       2722
053.233                2723  REPLACE EQU   *
053.233  315 041 072   2724         CALL  CFN.          COPY FILE NAME AND FILE OPEN PRESET
053.236  305           2725         PUSH  B             SAVE (BC)
053.237  303 324 053   2726         JMP   SAVE1         SAVE IT


                       2728  **     RETURN - RETURN FROM GOSUB.
                       2729  *
                       2730
                       2731
053.242  052 140 112   2732  RETURN LHLD  GOSTAB+MT.FWA
053.245  353           2733         XCHG                (DE) = FWA
053.246  052 142 112   2734         LHLD  GOSTAB+MT.LEN
053.251  174           2735         MOV   A,H
053.252  265           2736         ORA   L
053.253  312 141 070   2737         JZ    ERR.RE        RETURN ERROR
053.256  053           2738         DCX   H
053.257  053           2739         DCX   H
053.260  053           2740         DCX   H
053.261  053           2741         DCX   H
053.262  042 142 112   2742         SHLD  GOSTAB+MT.LEN SET REDUCED SIZE
053.265  031           2743         DAD   D             (HL) = ABS ADDRESS OF ENTRY
053.266  116           2744         MOV   C,M
053.267  043           2745         INX   H
053.270  106           2746         MOV   B,M           (BC) = RETURN ADDRESS
053.271  043           2747         INX   H
053.272  176           2748         MOV   A,M
053.273  043           2749         INX   H
```

```
053.274  146           2750          MOV     H,M
053.275  157           2751          MOV     L,A
053.276  042 175 112   2752          SHLD    CURNUM          (HL) = OLD LINE NUMBER
053.301  311           2753          RET


                       2755  ***     SAVE - SAVE PROGRAM ON DISK.
                       2756  *
                       2757  *       SAVE <FNAME>
                       2758  *
                       2759  *       WILL COMPLAIN IF FILE ALREADY EXISTS.
                       2760
                       2761
053.302                2762  SAVE    EQU     *
053.302  315 041 072   2763          CALL    CFN.            COPY FILE NAME AND FILE OPEN PRESET
053.305  305           2764          PUSH    B               SAVE (BC)
053.306  021 072 043   2765          LXI     D,DEFALTP       PROGRAM DEFAULT
053.311  315 046 101   2766          CALL    $FOPER.         OPEN FILE
053.314  322 177 070   2767          JNC     ERR.FAE         FILE ALREADY EXISTS
053.317  376 014       2768          CPI     EC.FNF
053.321  302 223 070   2769          JNE     $FERROR         NON-EXPECTED ERROR
                       2770
                       2771  *       ENTERED HERE FROM 'REPLACE' TO SAVE FILE REGARDLESS
                       2772
053.324  021 072 043   2773  SAVE1   LXI     D,DEFALTP       (DE) = DEFAULTS FOR SAVE
053.327  315 030 101   2774          CALL    $FOPEW          OPEN FOR WRITE, THEN
053.332  301           2775          POP     B               RESTORE TEXT POINTER
                       2776
                       2777  *       FILE IS OPEN. LIST PROGRAM TO IT
                       2778
053.333  076 001       2779          MVI     A,1
053.335  062 202 112   2780          STA     IOCHAN          SET I/O CHANNEL
053.340  345           2781          PUSH    H               SAVE ADDRESS OF BUFFER
053.341  315 023 051   2782          CALL    LIST.           LIST TO FILE                    /78.10.GC/
053.344  341           2783          POP     H
053.345  315 335 102   2784          CALL    $FCLO           CLOSE IT
053.350  001 007 115   2785          LXI     B,ZERO
053.353  303 115 074   2786          JMP     FOC             FILE OPEN CLEANUP, AND EXIT


                       2788  **      STEP - PERFORM SINGLE STEPPING.
                       2789  *
                       2790  *       STEP I
                       2791  *
                       2792
                       2793
053.356  315 072 076   2794  STEP    CALL    PNT             PREVIEW NEXT TOKEN
053.361  127           2795          MOV     D,A
053.362  137           2796          MOV     E,A
000.000                2797          ERRNZ   CT.FIN
053.363  247           2798          ANA     A
053.364  304 036 057   2799          CNZ     EVALI           EVALUATE COUNT
```

```
053.367  315 305 077  2800          CALL    RNT             FLUSH TOKEN PIPELINE
053.372  000          2801          DB      CT.FIN
053.373  033          2802          DCX     D
                      2803
053.374  325          2804  STEP1   PUSH    D               SAVE COUNT
053.375  076 001      2805          MVI     A,RM.STE
053.377  315 165 045  2806          CALL    CONT1           STEP 1
054.002  321          2807          POP     D
054.003  033          2808          DCX     D
054.004  172          2809          MOV     A,D
054.005  247          2810          ANA     A
054.006  362 374 053  2811          JP      STEP1           MORE TO GO
054.011  315 136 031  2812          CALL    $TYPTX
054.014  116 145 170  2813          DB      'Next',='@200Q
054.021  052 175 112  2814          LHLD    CURNUM
054.024  353          2815          XCHG
054.025  303 264 047  2816          JMP     TDI.            TYPE AS DECIMAL INTEGER


                      2818  **      STOP - STOP EXECUTION.
                      2819  *
                      2820
                      2821
054.030  315 201 044  2822  STOP    CALL    EXEC7           STORE BC
054.033  076 225      2823          MVI     A,BEC.ST
054.035  365          2824          PUSH    PSW
054.036  303 063 075  2825          JMP     ILM             ISSUE LINE MESSAGE


                      2827  ***     UNFREEZE - UNFREEZE FROZEN PROGRAM.
                      2828  *
                      2829  *       UNFREEZE <FNAME>
                      2830  *
                      2831  *       SAME AS 'RUN SY0:FNAME,BAF'
                      2832
                      2833
054.041               2834  UNFREZ  EQU     *
054.041  315 313 075  2835          CALL    LFC             CHECK FOR DATA LOCK
054.044  315 103 054  2836          CALL    UNSAVE1         PRESET
054.047  021 057 054  2837          LXI     D,UNFREZA
054.052  377 040      2838          DB      SYSCALL,.LINK   LINK IT
054.054  303 223 070  2839          JMP     $FERROR         GOT PROBLEMS
                      2840
054.057  123 131 060  2841  UNFREZA DB      'SY0BAF'        DEFAULT BLOCK
```

```
                                2843  ***     UNSAVE - DELETE PROGRAM.
                                2844  *
                                2845  *       UNSAVE <FNAME>
                                2846
                                2847
                                2848
          054.065               2849  UNSAVE  EQU     *
          054.065  315 103 054  2850          CALL    UNSAVE1         PRESET
          054.070  021 072 043  2851          LXI     D,DEFALTP       PROGRAM DEFAULTS
          054.073  305          2852          PUSH    B
          054.074  377 050      2853          DB      SYSCALL,.DELET  DELETE IT
          054.076  301          2854          POP     B               (BC) = TEXT POINTER
          054.077  320          2855          RNC                     NO ERROR
          054.100  303 223 070  2856          JMP     $FERROR         FLAG ERROR
                                2857
                                2858
                                2859  **      GET READY FOR OPERATION
                                2860
          054.103  315 053 072  2861  UNSAVE1 CALL    CFN             CRACK FILE NAME
          054.106  021 012 000  2862          LXI     D,FB.NAM
          054.111  031          2863          DAD     D
          054.112  353          2864          XCHG
          054.113  041 335 113  2865          LXI     H,LINE2
          054.116  305          2866          PUSH    B
          054.117  001 021 000  2867          LXI     B,FB.NAML
          054.122  345          2868          PUSH    H               SAVE $FOPWRK
          054.123  315 252 030  2869          CALL    $MOVE           MOVE IN NAME
          054.126  341          2870          POP     H
          054.127  301          2871          POP     B
          054.130  311          2872          RET
```

```
                                2876  **      LEXCAL - PERFORM LEXICAL ANALYSIS.
                                2877  *
                                2878  *       LEXCAL PARSES THE NEXT TOKEN FROM THE SOURCE LINE.
                                2879  *
                                2880  *       IF THE VARIABLE HAS NOT BEEN DEFINED, A SPECIAL ADDRESS,
                                2881  *       *LEXC* IS RETURNED. THIS ADDRESS CONTAINS A 0 OR A NULL STRING,
                                2882  *       DEPENDING UPON THE VARIABLE TYPE.
                                2883  *
                                2884  *       ENTRY   (BC) = SOURCE TEXT POINTER
                                2885  *       EXIT    (A) = TYPE (CT. CODE)
                                2886  *               (DE) = SYMTAB ENTRY ADDRESS+2 (IF SYMBOL)
                                2887  *               'C' SET IF VARIABLE AND NOT DEFINED
                                2888  *                (DE) = LEXC
                                2889  *                LEXA = VARIABLE NAM
                                2890  *       USES    A,F,B,C,D,E
                                2891
                                2892
 054.131                        2893  LEXCAL  EQU     *
 054.131  315 126 100           2894          CALL    SOB             SKIP OVER BLANKS
 054.134  315 230 072           2895          CALL    CNC             CLASSIFY NEXT CHARACTER
 000.000                        2896          ERRNZ   CT.FIN
 054.137  247                  2897          ANA     A
 054.140  310                  2898          RZ                      IS CT.FIN
 054.141  003                  2899          INX     B               ACCEPT CHARACTER
 054.142  370                  2900          RM                      IS KEYWORD
 054.143  376 030              2901          CPI     CT.QUO
 054.145  312 015 055          2902          JE      LEX12           HAVE STRING
 054.150  376 014              2903          CPI     CT.LT
 054.152  314 354 054          2904          CE      LEX10           IS <
 054.155  376 012              2905          CPI     CT.GT
 054.157  314 377 054          2906          CE      LEX11           HAVE >
 054.162  376 001              2907          CPI     CT.ALP
 054.164  312 231 054          2908          JE      LEX1            IS ALPHABETIC
 054.167  376 002              2909          CPI     CT.NUM
 054.171  312 202 054          2910          JE      LEX0            IS NUMERIC VALUE
 054.174  376 003              2911          CPI     CT.SEP
 054.176  300                  2912          RNE                     IS SOME KNOWN CHARACTER
 054.177  303 174 070          2913          JMP     ERR.IC          ILLEGAL CHARACTER
                                2914
                                2915  *       IS NUMERIC VALUE. FLOAT IT.
                                2916
 054.202  013                  2917  LEX0    DCX     B               (BC) = ADDRESS OF FIRST DIGIT
 054.203  353                  2918          XCHG                    SAVE (HL) IN (DE)
 054.204  140                  2919          MOV     H,B
 054.205  151                  2920          MOV     L,C
 054.206  315 323 107          2921          CALL    ATF             ASCII TO FLOATING
 054.211  104                  2922          MOV     B,H
 054.212  115                  2923          MOV     C,L
 054.213  353                  2924          XCHG                    RESTORE (HL)
 054.214  021 222 042          2925          LXI     D,LEXB
 054.217  315 237 073          2926          CALL    CXV             COPY NUMBER INTO 'LEXB' HOLD AREA
 054.222  076 300              2927          MVI     A,300Q          SET TYPE
 054.224  247                  2928          ANA     A               CLEAR CARRY
 054.225  062 221 042          2929          STA     LEXB-1          SET TYPE
 054.230  311                  2930          RET
                                2931
```

```
                        2932  *       IS ALPHABETIC. MUST BE VARIABLE.
                        2933
054.231                 2934  LEX1    EQU     *
054.231  013           2935          DCX     B               POINT TO 1ST CHAR OF NAME
054.232  012           2936          LDAX    B               (A) = 1ST CHARACTER OF NAME
054.233  315 107 112   2937          CALL    $MCU            MAP CHARACTER TO UPPER CASE
054.236  127           2938          MOV     D,A
054.237  036 000       2939          MVI     E,0             (DE) = CARIABLE NAME
054.241  003           2940          INX     B
054.242  012           2941          LDAX    B
054.243  326 060       2942          SUI     '0'
054.245  332 264 054   2943          JC      LEX2            NOT NUMBER
054.250  376 012       2944          CPI     9+1
054.252  322 264 054   2945          JNC     LEX2            NOT NUMBER
054.255  074           2946          INR     A               DIFFERENTIATE BETWEEN X AND X0
054.256  007           2947          RLC
054.257  007           2948          RLC
054.260  007           2949          RLC
054.261  007           2950          RLC
054.262  137           2951          MOV     E,A             (E) = NUMBER INDEX
054.263  003           2952          INX     B               ADVANCE PAST NUMBER
                        2953
                        2954  *       HAVE VARIABLE NAME IN (DE), CHECK FOR $ AND (
                        2955
054.264  012           2956  LEX2    LDAX    B
054.265  376 044       2957          CPI     '$'
054.267  302 274 054   2958          JNE     LEX3            NOT $
054.272  003           2959          INX     B
000.000                 2960          ERRNZ   CF.STR-1
054.273  034           2961          INR     E               SET CF.STR
054.274  315 126 100   2962  LEX3    CALL    SOB             SKIP OVER BLANKS
054.277  012           2963          LDAX    B
054.300  376 050       2964          CPI     '('
054.302  302 312 054   2965          JNE     LEX3.5
054.305  003           2966          INX     B
054.306  076 002       2967          MVI     A,CF.VEC        SET VECTOR TYPE
054.310  263           2968          ORA     E               SET VECTOR TYPE
054.311  137           2969          MOV     E,A
                        2970
                        2971  *       (DE) = VARIABLE NAME AND TYPE. FIND IN SYMTAB
                        2972
054.312  345           2973  LEX3.5  PUSH    H                                       /80.01.GC/
054.313  325           2974          PUSH    D                                       /80.01.GC/
054.314  315 323 075   2975          CALL    LVS             DE = SYMTAB ADDRESS     /80.01.GC/
054.317  341           2976          POP     H               HL = SAVED TYPE         /80.01.GC/
054.320  023           2977          INX     D                                       /80.01.GC/
054.321  302 334 054   2978          JNZ     LEX7                                    /80.01.GC/
                        2979
                        2980  *       HAVE FOUND MATCH.
                        2981
054.324  032           2982          LDAX    D                                       /80.01.GC/
054.325  023           2983          INX     D                                       /80.01.GC/
054.326  346 007       2984          ANI     7               (A) = TYPE CODE
054.330  366 300       2985          ORI     300Q
054.332  341           2986          POP     H               RESTORE (HL)
054.333  311           2987          RET
```

```
                        2988
                        2989  *       ITEM NOT IN TABLE.
                        2990  *
                        2991  *       RETURN NULL OR ZERO VALUE.
                        2992
054.334  042 236 075    2993  LEX7    SHLD    LEXA            SAVE
054.337  175            2994          MOV     A,L             (A) = FLAG FIELD OF NAME
054.340  021 214 042    2995          LXI     D,LEXC-1        (DE) = RESULT POINTER-1
054.343  346 007        2996          ANI     7               STRIP FLAGS
054.345  366 300        2997          ORI     300Q            SET VARIABLE TYPE
054.347  022            2998          STAX    D               SET TYPE
054.350  023            2999          INX     D               SET RESULT POINTER
054.351  341            3000          POP     H               RESTORE (HL)
054.352  067            3001          STC                     FLAG UNDEFINED
054.353  311            3002          RET
                        3003
                        3004  *       HAVE <, SEE IF <= OR <>
                        3005
054.354  012            3006  LEX10   LDAX    B
054.355  003            3007          INX     B               ASSUME IS <= OR <>
054.356  376 075        3008          CPI     '='
054.360  076 015        3009          MVI     A,CT.LE         ASSUME <=
054.362  310            3010          RE
054.363  013            3011          DCX     B               GET TESTING CHARACTER
054.364  012            3012          LDAX    B
054.365  003            3013          INX     B               RESTORE (BC)
054.366  376 076        3014          CPI     '>'
054.370  076 016        3015          MVI     A,CT.NE         ASSUME <>
054.372  310            3016          RE                      IS <>
054.373  076 014        3017          MVI     A,CT.LT         IS JUST <
054.375  013            3018          DCX     B
054.376  311            3019          RET
                        3020
                        3021  *       HAVE >, SEE IF >=
                        3022
054.377  012            3023  LEX11   LDAX    B
055.000  003            3024          INX     B               ASSUME IS >=
055.001  376 075        3025          CPI     '='
055.003  076 013        3026          MVI     A,CT.GE
055.005  310            3027          RE                      IS <=
055.006  076 012        3028          MVI     A,CT.GT         IS >
055.010  013            3029          DCX     B
055.011  311            3030          RET
                        3031
                        3032
                        3033  **      LEX12 - PUT TEXT STRING INTO STRINGTABLE AS TEMP STRING.
                        3034  *
                        3035  *       ENTRY   (BC) = ADDRESS OF 1ST CHARACTER
                        3036  *       EXIT    LEXB = STRING HEADER
                        3037  *               (DE) = #LEXB
                        3038  *       USES    A,F,B,C,D,E
                        3039
055.012  076 000        3040  LEX11.5 MVI     A,0
                        3041
055.014  021            3042          DB      MI.LXID         USE 'LXI,D' TO GOBBLE NEXT MVI
055.015  076 042        3043  LEX12   MVI     A,'"'
```

```
                                3044
055.017  062 033 055  3045            STA     LEXD            SET END OF STRING MATCH CHARACTER
055.022  345          3046            PUSH    H               SAVE (HL)
055.023  305          3047            PUSH    B               SAVE TEXT POINTER
055.024  041 377 377  3048            LXI     H,-1            (HL) = CHARACTER COUNTER
055.027  012          3049  LEX13     LDAX    B
055.030  003          3050            INX     B
055.031  043          3051            INX     H
055.032  376 042      3052            CPI     '"'
055.033               3053  LEXD      EQU     *-1             END OF STRING MATCH
055.034  312 044 055  3054            JE      LEX14           GOT END QUOTE
055.037  247          3055            ANA     A
055.040  302 027 055  3056            JNZ     LEX13           NOT AT END OF LINE
055.043  053          3057            DCX     H               RAN OFF END OF LINE, DONT COUNT 00 BYTE
                                3058
055.044  174          3059  LEX14     MOV     A,H
055.045  247          3060            ANA     A
055.046  302 144 070  3061            JNZ     ERR.SL          STRING LENGTH ERROR
055.051  345          3062            PUSH    H               SAVE LENGTH IN (L)
055.052  042 222 042  3063            SHLD    LEXB            SET IN DESCRIPTOR
055.055  021 222 042  3064            LXI     D,LEXB
055.060  315 033 073  3065            CALL    CSE.            CREATE TEMP STRING TAB ENTRY
055.063  301          3066            POP     B               (BC) = COUNT
055.064  321          3067            POP     D               (DE) = FROM ADDRESS
055.065  315 252 030  3068            CALL    $MOVE           COPY STRING INTO TEMP
055.070  102          3069            MOV     B,D
055.071  113          3070            MOV     C,E
055.072  003          3071            INX     B
055.073  076 301      3072            MVI     A,CT.SSV        SCALAR STRING VALUE
055.075  021 221 042  3073            LXI     D,LEXB-1
055.100  022          3074            STAX    D               SET TYPE
055.101  023          3075            INX     D               (DE) = DESCRIPTOR POINTER
055.102  341          3076            POP     H               RESTORE (HL)
055.103  311          3077            RET
```

```
                                3081  **      VARIAB - DECODE VARIABLE.
                                3082  *
                                3083  *       VARIAB IS CALLED TO EVALUATE A VARIABLE SPECIFICATION.
                                3084  *       VARIAB RESOLVES SUBSCRIPTS.
                                3085  *
                                3086  *       ENTRY   (BC) = TEXT POINTER
                                3087  *       EXIT    (BC) UPDATED
                                3088  *               (DE) = VARIABLE POINTER
                                3089  *       USES    A,F,B,C,D,E
                                3090
                                3091
055.104                         3092  VARIAB  EQU     *
055.104  315 056 071           3093          CALL    ANT             ACCEPT NEXT TOKEN
055.107  365                  3094  VARIAB. PUSH    PSW             SAVE TYPE
055.110  346 002               3095          ANI     CF.VEC
055.112  302 117 055           3096          JNZ     VAR2            IS VECTOR
055.115  361                  3097          POP     PSW
055.116  311                  3098          RET                     IS SIMPLE VARIABLE
                                3099
                                3100  *       HAVE SUBSCRIPT.
                                3101
055.117  345                  3102  VAR2    PUSH    H
055.120  032                  3103          LDAX    D               (A) = DIMENSION COUNT
055.121  247                  3104          ANA     A
055.122  372 152 070           3105          JM      ERR.SY          IS FUNCTION
055.125  312 171 070           3106          JZ      ERR.ND          NOT DECLARED
                                3107
                                3108  *       EVALUATE SUBSCRIPT.
                                3109
055.130  353                  3110          XCHG                    (HL) = SUBSCRIPT LIST-2
055.131  043                  3111          INX     H
055.132  043                  3112          INX     H
055.133  021 000 000           3113          LXI     D,0             (DE) = INDEX
                                3114
055.136  365                  3115  VAR4    PUSH    PSW
055.137  043                  3116          INX     H
055.140  043                  3117          INX     H               POINT TO NEXT SUBSCRIPT LIMIT
055.141  345                  3118          PUSH    H               SAVE VECTAB POINTER
055.142  305                  3119          PUSH    B               SAVE (BC)
055.143  116                  3120          MOV     C,M
055.144  043                  3121          INX     H
055.145  106                  3122          MOV     B,M             (BC) = LIMIT
055.146  305                  3123          PUSH    B               SAVE LIMIT
055.147  315 337 030           3124          CALL    $MU66           (HL) = NEW INDEX
055.152  321                  3125          POP     D               (DE) = NEW LIMIT
055.153  301                  3126          POP     B               (BC) = TEXT POINTER
055.154  345                  3127          PUSH    H               SAVE INDEX
055.155  325                  3128          PUSH    D               SAVE LIM
055.156  315 036 057           3129          CALL    EVALI           EVALUATE SUBSCRIPT
055.161  341                  3130          POP     H               (HL) = LIM
055.162  053                  3131          DCX     H
055.163  175                  3132          MOV     A,L
055.164  223                  3133          SUB     E
055.165  174                  3134          MOV     A,H
055.166  232                  3135          SBB     D
055.167  332 163 070           3136          JC      ERR.SR          SUBSCRIPT RANGE
```

```
055.172  341          3137          POP     H               (HL) = INDEX
055.173  031          3138          DAD     D               (HL) = NEW INDEX
055.174  353          3139          XCHG
055.175  341          3140          POP     H               (HL) = SYMTAB POINTER
055.176  361          3141          POP     PSW
055.177  075          3142          DCR     A
055.200  312 220 055  3143          JZ      VAR5            NO MORE SUBSCRIPTS
                      3144
                      3145  *       EXPECT ,
                      3146
055.203  365          3147          PUSH    PSW
055.204  315 056 071  3148          CALL    ANT             ACCEPT NEXT TOKEN
055.207  376 026      3149          CPI     CT.CMA
055.211  302 166 070  3150          JNE     ERR.SC          NOT ENOUGH SUBSCRIPTS
055.214  361          3151          POP     PSW             (A) = REMAINING SUBSCRIPT COUNT
055.215  303 136 055  3152          JMP     VAR4            READ NEXT
                      3153
                      3154  *       EXPECT )
                      3155
055.220  315 056 071  3156  VAR5    CALL    ANT             ACCEPT NEXT TOKEN
055.223  376 020      3157          CPI     CT.PAR
055.225  302 166 070  3158          JNE     ERR.SC          TOO MANY SUBSCRIPTS
                      3159
                      3160  *       SUBSCRIPT EVALUATED, (DE) = INDEX
                      3161
055.230  043          3162          INX     H
055.231  043          3163          INX     H
055.232  353          3164          XCHG
055.233  051          3165          DAD     H
055.234  051          3166          DAD     H
055.235  031          3167          DAD     D
055.236  353          3168          XCHG                    (DE) = ADDRESS OF ENTRY IN SYMTAB
055.237  341          3169          POP     H
055.240  361          3170          POP     PSW
055.241  346 375      3171          ANI     377Q-CF.VEC     CLEAR VECTOR TYPE
055.243  311          3172          RET
```

```
                                3175  **      EVAL - EVALUATES AN EXPRESSION.
                                3176  *
                                3177  *       EVAL EVALUATES EXPRESSIONS MADE UP OF OPERATORS AND
                                3178  *       SYMBOLS.
                                3179  *
                                3180  *       VALID OPERATORS ARE (IN ORDER OF PRECIDENCE)
                                3181  *
                                3182  *       - NOT   (UNARY MINUS, NOT)
                                3183  *       ^                       (EXPONENTIATION)
                                3184  *       *,/
                                3185  *       +,-
                                3186  *       < <= = <> >= >
                                3187  *       AND
                                3188  *       OR
                                3189  *
                                3190  *       EVAL PROCESSES EXPRESSIONS UNTIL AN INAPPROPRIATE TOKEN IS
                                3191  *       ENCOUNTERED.
                                3192  *
                                3193  *
                                3194  *       ENTRY   (BC) = TEXT POINTER
                                3195  *       EXIT    (BC) UPDATED
                                3196  *               (DE) = VALUE POINTER
                                3197  *       USES    A,F,B,C,D,E
                                3198
                                3199
          055.244               3200  EVAL    EQU     *
          055.244  345         3201          PUSH    H               SAVE (HL)
          055.245  315 255 055 3202          CALL    LEV1
          055.250  341         3203          POP     H               RESTORE (HL)
          055.251  021 202 042 3204          LXI     D,ACCX          (DE) = RESULT ADDRESS
          055.254  311         3205          RET


                                3207
                                3208  **      LEV1 - OR
                                3209
          055.255  315 304 055 3210  LEV1    CALL    LEV2
          055.260  376 315     3211  LEV11   CPI     CT.OR
          055.262  300         3212          RNE                     NOT 'OR'
          055.263  315 030 077 3213          CALL    PSHX.           ACCEPT '-' AND SAVE ACCX
          055.266  315 304 055 3214          CALL    LEV2
          055.271  315 365 076 3215          CALL    POPY
          055.274  365         3216          PUSH    PSW             SAVE TYPE
          055.275  315 323 061 3217          CALL    P.OR            PREFORM 'OR'
          055.300  361         3218          POP     PSW
          055.301  303 260 055 3219          JMP     LEV11
                                3220
                                3221  *       LEV2 - AND
                                3222
          055.304  315 333 055 3223  LEV2    CALL    LEV3
          055.307  376 310     3224  LEV21   CPI     CT.AND
          055.311  300         3225          RNE
          055.312  315 030 077 3226          CALL    PSHX.           ACCEPT 'AND' AND SAVE ACCX
          055.315  315 333 055 3227          CALL    LEV3
          055.320  315 365 076 3228          CALL    POPY
```

```
055.323  365              3229          PUSH    PSW
055.324  315 336 061      3230          CALL    P.AND           PERFORM AND
055.327  361              3231          POP     PSW
055.330  303 307 055      3232          JMP     LEV21
055.333                   3233  LEV3    EQU     *               NOT USED
                          3234
                          3235  *       LEV4 - COMPARE OPERATORS.
                          3236
055.333  315 367 055      3237  LEV4    CALL    LEV5
055.336  376 011          3238  LEV41   CPI     CT.EQ
055.340  330              3239          RC                      NOT COMPARE
055.341  376 017          3240          CPI     CT.NE+1
055.343  320              3241          RNC                     NOT COMPARE
055.344  365              3242          PUSH    PSW             SAVE TYPE
055.345  315 030 077      3243          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
055.350  315 367 055      3244          CALL    LEV5
055.353  315 365 076      3245          CALL    POPY
055.356  341              3246          POP     H               (H) = COMPARE TYPE
055.357  365              3247          PUSH    PSW             SAVE NEXT TYPE
055.360  315 375 061      3248          CALL    P.CMP           DO BOOLEAN
055.363  361              3249          POP     PSW
055.364  303 336 055      3250          JMP     LEV41
                          3251
                          3252  *       LEV5 - +, -
                          3253
055.367  315 025 056      3254  LEV5    CALL    LEV6
055.372  376 021          3255  LEV51   CPI     CT.PL
055.374  312 002 056      3256          JE      LEV52           IS +
055.377  376 022          3257          CPI     CT.MI
056.001  300              3258          RNE                     NOT + OR -
056.002  365              3259  LEV52   PUSH    PSW             SAVE TYPE
056.003  315 030 077      3260          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
056.006  315 025 056      3261          CALL    LEV6
056.011  315 365 076      3262          CALL    POPY
056.014  341              3263          POP     H               (H) = TYPE
056.015  365              3264          PUSH    PSW
056.016  315 134 062      3265          CALL    P.ADD
056.021  361              3266          POP     PSW
056.022  303 372 055      3267          JMP     LEV51
                          3268
                          3269  *       LEV6 - * /
                          3270
056.025  315 063 056      3271  LEV6    CALL    LEV7
056.030  376 023          3272  LEV61   CPI     CT.MU
056.032  312 040 056      3273          JE      LEV62           IS *
056.035  376 024          3274          CPI     CT.DI
056.037  300              3275          RNE                     NOT * /
056.040  365              3276  LEV62   PUSH    PSW
056.041  315 030 077      3277          CALL    PSHX.           ACCEPT OPERATOR AND SAVE ACCX
056.044  315 063 056      3278          CALL    LEV7
056.047  315 365 076      3279          CALL    POPY
056.052  341              3280          POP     H               (HL) = TYPE
056.053  365              3281          PUSH    PSW
056.054  315 247 062      3282          CALL    P.MUL
056.057  361              3283          POP     PSW
056.060  303 030 056      3284          JMP     LEV61
```

```
                                3285
                                3286  *       LEV7 - ^
                                3287
056.063  315 112 056            3288  LEV7    CALL    LEV8
056.066  376 025                3289  LEV71   CPI     CT.EX
056.070  300                    3290          RNE                     NOT EXPONENTIAL
056.071  315 030 077            3291          CALL    PSHX.           ACCEPT ^ AND SAVE ACCX
056.074  315 112 056            3292          CALL    LEV8
056.077  315 365 076            3293          CALL    POPY
056.102  365                    3294          PUSH    PSW
056.103  315 270 062            3295          CALL    F.EXP
056.106  361                    3296          POP     PSW
056.107  303 066 056            3297          JMP     LEV71
                                3298
                                3299  *       LEV8 - UNARY  - NOT
                                3300
056.112  315 072 076            3301  LEV8    CALL    PNT             PEEK AT NEXT TOKEN
056.115  376 022                3302          CPI     CT.MI
056.117  312 127 056            3303          JE      LEV81           IS MINUS
056.122  376 314                3304          CPI     CT.NOT
056.124  302 170 056            3305          JNE     LEV9            NOT - OR NOT
056.127  315 056 071            3306  LEV81   CALL    ANT             READ '-' OR 'NOT'
056.132  365                    3307          PUSH    PSW             SAVE TYPE
056.133  315 170 056            3308          CALL    LEV9            PROCESS OPERAND
056.136  341                    3309          POP     H               (H) = TYPE
056.137  365                    3310          PUSH    PSW             SAVE NEXT TOKEN CODE
056.140  072 201 042            3311          LDA     ACCX-1
056.143  346 001                3312          ANI     CF.STR
056.145  302 155 070            3313          JNZ     ERR.TC          MUST BE NUMERIC
056.150  174                    3314          MOV     A,H
056.151  376 022                3315          CPI     CT.MI
056.153  302 163 056            3316          JNE     LEV82           IS NOT
                                3317
                                3318  *       IS -
                                3319
056.156  315 302 105            3320          CALL    FPNEG
056.161  361                    3321          POP     PSW             (A) = CODE FOR NEXT TOKEN
056.162  311                    3322          RET
                                3323
                                3324  *       IS NOT
                                3325
056.163  315 351 061            3326  LEV82   CALL    F.NOT
056.166  361                    3327          POP     PSW
056.167  311                    3328          RET
                                3329
                                3330  *       LEV9 - TOKEN
                                3331
056.170  315 072 076            3332  LEV9    CALL    PNT             PREVIEW NEXT TOKEN
056.173  376 300                3333          CPI     CT.VARL
056.175  332 234 056            3334          JC      LEV92           NOT VARIABLE
056.200  376 310                3335          CPI     CT.VARH+1
056.202  322 234 056            3336          JNC     LEV92           NOT VARIABLE
                                3337
                                3338  *       IS VARIABLE.
                                3339
056.205  315 104 055            3340          CALL    VARIAB          DECODE
```

```
056.210  041 201 042  3341          LXI     H,ACCX-1
056.213  167          3342          MOV     M,A
056.214  043          3343          INX     H
056.215  305          3344          PUSH    B               SAVE (BC)
056.216  006 004      3345          MVI     B,4             (B) = LOOP COUNT
056.220  032          3346  LEV95   LDAX    D
056.221  167          3347          MOV     M,A             COPY VALUE INTO ACCX
056.222  023          3348          INX     D
056.223  043          3349          INX     H
056.224  005          3350          DCR     B
056.225  302 220 056  3351          JNZ     LEV95
056.230  301          3352          POP     B               RESTORE (BC)
056.231  303 072 076  3353          JMP     PNT             PREVIEW NEXT TOKEN AND EXIT
                      3354
056.234  315 056 071  3355  LEV92   CALL    ANT             ACCEPT TOKEN
056.237  376 017      3356          CPI     CT.PAL
056.241  302 256 056  3357          JNE     LEV93           NOT (
056.244  315 244 055  3358          CALL    EVAL            IS PARENTHESISED EXPRESSION
                      3359
                      3360  *       FUNCTION COMPLETE, REQUIRE ')'
                      3361
056.247  315 305 077  3362  LEV94   CALL    RNT
056.252  020          3363          DB      CT.PAR          REQUIRE ')'
056.253  303 072 076  3364          JMP     PNT             READ NEXT TOKEN AND EXIT
                      3365
056.256  376 220      3366  LEV93   CPI     CT.FN
056.260  312 340 062  3367          JE      TXTFN           TEXT FUNCTION
                      3368
                      3369  *       IS NOT SIMPLE STRING OR VALUE, MUST BE FUNCTION
                      3370
056.263  326 320      3371          SUI     CT.FCN
056.265  332 152 070  3372          JC      ERR.SY          NOT FUNCTION
056.270  365          3373          PUSH    PSW
056.271  315 244 055  3374          CALL    EVAL            EVALUATE INNARDS
056.274  361          3375          POP     PSW
056.275  365          3376          PUSH    PSW             (A) = FUNCTION INDEX
056.276  376 030      3377          CPI     CT.SRA-CT.FCN
056.300  072 201 042  3378          LDA     ACCX-1          (A) = PARAMETER TYPE
056.303  332 307 056  3379          JC      LEV90           REQUIRE NUMERIC ARGUMENT
056.306  057          3380          CMA
056.307  346 001      3381  LEV90   ANI     CF.STR
056.311  302 155 070  3382          JNZ     ERR.TC          TYPE CONFLICT
056.314  361          3383          POP     PSW             (A) = FUNCTION CODE
056.315  041 247 056  3384          LXI     H,LEV94
056.320  345          3385          PUSH    H               SAVE 'LEV94' AS RETURN
                      3386
                      3387  *       IS SYSTEM FUNCTION
                      3388
056.321  315 061 031  3389          CALL    $TJMP           ENTER PROCESSOR
056.324  055 057      3390          DW      ABS
056.326  026 065      3391          DW      ATN
056.330  103 057      3392          DW      CHR$
056.332  140 057      3393          DW      CIN
056.334  125 064      3394          DW      COS
056.336  075 063      3395          DW      EXP
056.340  216 057      3396          DW      INT
```

```
056.342  064 057      3397          DW      LNO
056.344  225 063      3398          DW      LOG
056.346  317 060      3399          DW      MAX
056.350  320 060      3400          DW      MIN
056.352  006 061      3401          DW      PAD
056.354  014 061      3402          DW      PEEK
056.356  034 061      3403          DW      PIN
056.360  053 061      3404          DW      POS
056.362  074 061      3405          DW      RND
056.364  170 061      3406          DW      SEG
056.366  205 061      3407          DW      SGN
056.370  117 064      3408          DW      SIN
056.372  152 070      3409          DW      ERR.SY          SPC
056.374  360 063      3410          DW      SQR
056.376  231 061      3411          DW      STR$
057.000  152 070      3412          DW      ERR.SY          TAB
057.002  243 064      3413          DW      TAN
                      3414
                      3415  *       THESE FUNCTIONS REQUIRE STRING ARGUMENTS.
                      3416
057.004  065 057      3417          DW      ASC
057.006  314 057      3418          DW      LEFT$
057.010  306 057      3419          DW      LEN
057.012  111 060      3420          DW      MATCH
057.014  314 057      3421          DW      MID$
057.016  314 057      3422          DW      RIGHT$
057.020  270 061      3423          DW      VAL


                      3425  **      EVALN - EVALUATE NUMERIC EXPRESSION.
                      3426  *
                      3427  *       ENTRY   SAME AS EVAL.
                      3428  *       EXIT    SAME AS EVAL
                      3429  *       USES    A,F,B,C,D,E
                      3430
                      3431
057.022  315 244 055  3432  EVALN   CALL    EVAL
057.025  072 201 042  3433          LDA     ACCX-1
057.030  346 001      3434          ANI     CF.STR
057.032  302 155 070  3435          JNZ     ERR.TC          TYPE CONFLICT
057.035  311          3436          RET                     OK


                      3438  **      EVALI - EVALUATE INTEGER EXPRESSION.
                      3439  *
                      3440  *       ENTRY   SAME AS EVAL
                      3441  *       EXIT    (DE) = INTEGER VALUE
                      3442  *               (BC) UPDATED
                      3443  *       USES    A,F,B,C,D,E
                      3444
                      3445
057.036  315 022 057  3446  EVALI   CALL    EVALN
```

```
057.041  303 002 075  3447          JMP     IFIX            FIX IT


                      3449  **      EVALI8 - EVALUATE 8 BIT INTEGER EXPRESSION.
                      3450  *
                      3451  *       ENTRY   SAME AS EVAL
                      3452  *       EXIT    (BC) UPDATED
                      3453  *               (E) = VALUE
                      3454  *       USES    A,F,B,C,D,E
                      3455
                      3456
057.044  315 036 057  3457  EVALI8  CALL    EVALI
057.047  172          3458          MOV     A,D
057.050  247          3459          ANA     A
057.051  310          3460          RZ                      OK
057.052  303 122 070  3461          JMP     ERR.IN          TOO LARGE
```

```
                        3465  **      ABS - ABSOLUTE VALUE.
                        3466  *
                        3467  *       Y=ABS(X)
                        3468
                        3469
057.055  072 204 042   3470  ABS     LDA     ACCX+2
057.060  247           3471          ANA     A
057.061  374 302 105   3472          CM      FPNEG
                        3473
                        3474  *       IDENTIFY FUNCTION
                        3475
057.064  311           3476  LNO     RET


                        3478  **      ASC - DECODE ASCII VALUE
                        3479  *
                        3480  *       X=ASC("C")
                        3481
                        3482
057.065  021 202 042   3483  ASC     LXI     D,ACCX
057.070  315 315 074   3484          CALL    FSE             FIND STRING TABLE ENTRY
057.073  247           3485          ANA     A
057.074  312 020 061   3486          JZ      PEEK1           NULL STRING YEILDS 0
057.077  176           3487          MOV     A,M             GIVE VALUE
057.100  303 020 061   3488          JMP     PEEK1


                        3490  **      CHR$ - CONVERT VALUE INTO ASCII CHARACTER.
                        3491  *
                        3492  *       C$=CHR$(X)
                        3493
                        3494
057.103  315 002 075   3495  CHR$    CALL    IFIX            MAKE INTEGER
057.106  325           3496          PUSH    D               SAVE VALUE
057.107  041 001 000   3497          LXI     H,1
057.112  042 202 042   3498          SHLD    ACCX            SET LENGTH
057.115  021 202 042   3499          LXI     D,ACCX
057.120  315 033 073   3500          CALL    CSE.            CREATE TEMP STRINGTAB ENTRY
057.123  315 262 061   3501          CALL    FRC             SET FUNCTION RETURNS CHARACTER
057.126  321           3502          POP     D               (DE) = VALUE
057.127  173           3503          MOV     A,E
057.130  346 177       3504          ANI     177Q            CLEAR BIT
057.132  167           3505          MOV     M,A             STORE
057.133  300           3506          RNZ                     IF NOT NULL
057.134  062 202 042   3507          STA     ACCX            NULL STRING IF 00
057.137  311           3508          RET
```

```
                                3510  **      CIN - CHARACTER INPUT FUNCTION.
                                3511  *
                                3512  *       I=CIN(CHAN)
                                3513  *
                                3514  *       INPUT SINGLE CHARACTER, NO MAPPING OR PARITY ADJUSTMENT.
                                3515  *       I=-1 IF NO CHARACTER AVAILABLE
                                3516
                                3517
057.140                         3518  CIN     EQU     *
057.140  315 002 075           3519          CALL    IFIX            GET CHANNEL NUMBER
057.143  305                   3520          PUSH    B               SAVE TEXT POINTER
057.144  172                   3521          MOV     A,D
057.145  247                   3522          ANA     A
057.146  302 122 070           3523          JNZ     ERR.IN          TOO LARGE A NUMBER
057.151  173                   3524          MOV     A,E             (A) = CHANNEL NUMBER
057.152  247                   3525          ANA     A
057.153  302 202 057           3526          JNZ     CIN2            FROM FILE
                                3527
                                3528  *       IS INPUT FROM CONSOLE
                                3529
057.156  377 001               3530          DB      SYSCALL,.SCIN   READ CHARACTER
057.160  137                   3531  CIN0    MOV     E,A
057.161  026 000               3532          MVI     D,0
057.163  322 170 057           3533          JNC     CIN1            GOT CHARACTER
057.166  036 001               3534          MVI     E,1
057.170  365                   3535  CIN1    PUSH    PSW
057.171  315 040 075           3536          CALL    IFLT            FLOAT IT
057.174  361                   3537          POP     PSW
057.175  334 302 105           3538          CC      FPNEG           NEGATE IT, IF NO CHARACTER
057.200  301                   3539          POP     B               RESTORE TEXT POINTER
057.201  311                   3540          RET                     EXIT
                                3541
                                3542  *       READ CHARACTER FROM FILE
                                3543
057.202  315 005 072           3544  CIN2    CALL    CFA             COMPUTE FILE ADDRESS
057.205  332 210 070           3545          JC      ERR.FNO         FILE NOT OPEN
057.210  315 364 101           3546          CALL    $FREAD          READ CHARACTER
057.213  303 160 057           3547          JMP     CIN0            PROCESS VALUE ('C' SET IF EOF)


                                3549  **      INT - TRUNCATE TO NEAREST INTEGER.
                                3550  *
                                3551  *       Y=INT(X)
                                3552
                                3553
057.216                         3554  INT     EQU     *
057.216  041 204 042           3555          LXI     H,ACCX+2
057.221  176                   3556          MOV     A,M             (A) = SIGN
057.222  247                   3557          ANA     A
057.223  365                   3558          PUSH    PSW             SAVE TEST RESULTS
057.224  374 302 105           3559          CM      FPNEG           MAKE POSITIVE
057.227  021 302 057           3560          LXI     D,INTA
057.232  315 352 104           3561          CALL    FPADD           ROUND UP
057.235  043                   3562          INX     H               (HL) = $ACCX+3
```

```
057.236  305          3563          PUSH    B               SAVE (BC)
057.237  106          3564          MOV     B,M             (B) = EXPONENT
057.240  004          3565          INR     B
057.241  021 000 000  3566          LXI     D,0
057.244  112          3567          MOV     C,D             (C,D,E) = MASK
                      3568
                      3569  *       SHIFT IN ONE BITS TO CORRESPOND TO NON-FRACTIONAL BITS.
                      3570
057.245  005          3571  INT1    DCR     B
057.246  362 260 057  3572          JP      INT2            NO MORE
057.251  067          3573          STC
057.252  315 233 107  3574          CALL    SRS..           SHIFT (BCD) RIGHT THROUGH CARRY
057.255  303 245 057  3575          JMP     INT1
                      3576
057.260  053          3577  INT2    DCX     H
057.261  176          3578          MOV     A,M             MASK OFF VALUE
057.262  241          3579          ANA     C
057.263  167          3580          MOV     M,A
057.264  053          3581          DCX     H
057.265  176          3582          MOV     A,M
057.266  242          3583          ANA     D
057.267  167          3584          MOV     M,A
057.270  053          3585          DCX     H
057.271  176          3586          MOV     A,M
057.272  243          3587          ANA     E
057.273  167          3588          MOV     M,A
057.274  301          3589          POP     B               RESTORE (BC)
057.275  361          3590          POP     PSW             (A) = ORIGINAL SIGN TEST RESULTS
057.276  374 302 105  3591          CM      FPNEG           RE-INVERT IF NECESSARY
057.301  311          3592          RET
                      3593
057.302  000 000 101  3594  INTA    DB      000Q,000Q,101Q,157Q


                      3596  **      LEN - LENGTH OF STRING.
                      3597  *
                      3598  *       X=LEN(S$)
                      3599
                      3600
057.306  072 202 042  3601  LEN     LDA     ACCX            (A) = LENGTH
057.311  303 020 061  3602          JMP     PEEK1           FLOAT INTO ACCX


                      3604  **      LEFT$ - GET LEFTMOST CHARACTERS.
                      3605  *
                      3606  *       Y$=LEFT$(X$,CNT)
                      3607
                      3608
                      3609  **      RIGHT$ - GET RIGHTMOST CHARACTERS.
                      3610  *
                      3611  *       Y$=RIGHT$(X$,CNT)
                      3612
```

```
                                3613
                                3614
                                3615
                                3616  **      MID$ - GET SEGMENT OF CHARACTER STRING.
                                3617  *
                                3618  *       Y$=MID$(X$,POS,LEN)
                                3619
                                3620
057.314                         3621  LEFT$   EQU     *
057.314                         3622  RIGHT$  EQU     *
057.314                         3623  MID$    EQU     *
057.314  365                   3624          PUSH    PSW             SAVE TYPE CODE
057.315  315 223 072            3625          CALL    CMA             REQUIRE ';'
057.320  315 033 077            3626          CALL    PSHX            SAVE X$ POINTER
057.323  315 044 057            3627          CALL    EVAL18          EVALUATE 8 BIT RESULT
057.326  123                   3628          MOV     D,E             (D) = LEN
057.327  036 001               3629          MVI     E,1             (E) = POS
057.331  315 370 076            3630          CALL    POPY.           (Y) = X$ POINTER
057.334  361                   3631          POP     PSW
000.003                         3632          ERRMI CT.MID-CT.LEF
057.335  376 070               3633          CPI     CT.MID-CT.FCN*2
057.337  312 365 057            3634          JE      MID$1           IS MID$
057.342  332 032 060            3635          JC      LEFT$1          IS LEFT$
377.377                         3636          ERRPL CT.MID-CT.RIG
                                3637
                                3638  *       IS RIGHT$
                                3639  *
                                3640  *       GENERATE MID$(X$,LENX$-MIN(LENX$,CNT),MIN(LENX$,CNT)
                                3641
057.345  072 210 042            3642          LDA     ACCY
057.350  137                   3643          MOV     E,A             (E) = LENX$
057.351  272                   3644          CMP     D
057.352  322 356 057            3645          JNC     RIGHT$1
057.355  127                   3646          MOV     D,A             (D) = MIN(LENX$,CNT)
057.356  173                   3647  RIGHT$1 MOV     A,E             (A) = LENX$
057.357  222                   3648          SUB     D               (A) = LENX$-MIN(LENX$,CNT)
057.360  137                   3649          MOV     E,A             (E) = LENX$-MIN(LENX$,CNT)
057.361  034                   3650          INR     E
057.362  303 032 060            3651          JMP     MID$2           MOVE
                                3652
                                3653  *       IS MID$
                                3654  *
                                3655  *       EVALUATE CNT
                                3656
057.365                         3657  MID$1   EQU     *
057.365  132                   3658          MOV     E,D             (E) = POS
057.366  172                   3659          MOV     A,D
057.367  247                   3660          ANA     A
057.370  312 122 070            3661          JZ      ERR.IN          0 ILLEGAL
057.373  026 377               3662          MVI     D,255           ASSUME NULL (LEN=255)
057.375  315 072 076            3663          CALL    PNT             PREVIEW NEXT TOKEN
060.000  376 020               3664          CPI     CT.PAR
060.002  312 032 060            3665          JE      MID$2           IS NULL
060.005  315 056 071            3666          CALL    ANT             ACCEPT ,
060.010  376 026               3667          CPI     CT.CMA
060.012  302 152 070            3668          JNE     ERR.SY
```

```
060.015  325          3669          PUSH    D               SAVE CURRENT POS
060.016  315 041 077  3670          CALL    PSHY            SAVE STRING
060.021  315 044 057  3671          CALL    EVAL1B          EVALUATE LEN
060.024  315 370 076  3672          CALL    POPY.           (ACCY) = X$ POINTER
060.027  353          3673          XCHG
060.030  321          3674          POP     D
060.031  125          3675          MOV     D,L             SET NEW LEN
                      3676
                      3677  *       (ACCX) = X$ POINTER
                      3678  *       (D) = LEN
                      3679  *       (E) = POS
                      3680  *
                      3681  *       COMPUTE Y$ = MID$(X$,POS,LEN)
                      3682
060.032               3683  LEFT$1  EQU     *
060.032               3684  MID$2   EQU     *
                      3685
                      3686  *       COMPUTE LEN' = MIN(LEN,MAX(LENX$-POS+1,0))
                      3687
060.032  072 210 042  3688          LDA     ACCY            (A) = LENX$
060.035  223          3689          SUB     E
060.036  074          3690          INR     A               (A) = LENX$-POS+1
060.037  322 043 060  3691          JNC     MID$3           IS >= 0
060.042  257          3692          XRA     A               USE 0
060.043  272          3693  MID$3   CMP     D
060.044  322 050 060  3694          JNC     MID$4           (D) = MIN VALUE
060.047  127          3695          MOV     D,A             (D) = MIN VALUE
060.050  046 000      3696  MID$4   MVI     H,0
060.052  152          3697          MOV     L,D             (HL) = LEN
060.053  042 202 042  3698          SHLD    ACCX            SET IN BLOCK
060.056  325          3699          PUSH    D               SAVE LENGTH
060.057  021 202 042  3700          LXI     D,ACCX
060.062  315 033 073  3701          CALL    CSE.            CREATE TEMP STRINGTAB ENTRY
060.065  343          3702          XTHL                    SAVE ADDRESS
060.066  345          3703          PUSH    H               SAVE LEN, POS
060.067  021 210 042  3704          LXI     D,ACCY
060.072  315 315 074  3705          CALL    FSE             FIND STRING ENTRY
060.075  321          3706          POP     D               (E) = POS
060.076  173          3707          MOV     A,E
060.077  075          3708          DCR     A
060.100  315 072 030  3709          CALL    $DADA           (HL) = FROM ADDRESS
060.103  172          3710          MOV     A,D             (A) = LEN
060.104  353          3711          XCHG
060.105  341          3712          POP     H               (HL) = TO ADDRESS
060.106  303 257 061  3713          JMP     STR$1           MOVE, EXIT WITH TYPE = CT.SSV


                      3715  ***     MATCH - FIND SUBSTRING IN STRING.
                      3716  *
                      3717  *       I=MATCH$(S1$,S2$,J)
                      3718  *
                      3719  *       SCAN S1$ FOR OCCURANCE OF S2$, STARTING AT CHARACTER J
                      3720  *
                      3721  *       I=0 IF NOT FOUND
```

```
                        3722  *       I = CHARACTER NUMBER OF START OF MATCH IF FOUND
                        3723
                        3724
060.111                 3725  MATCH   EQU     *
060.111  041 307 060    3726          LXI     H,MATCHA
060.114  315 051 076    3727          CALL    MOV4            SAVE S1 DESCRIPTOR
060.117  315 223 072    3728          CALL    CMA             GOBBLE CMA
                        3729
060.122  315 244 055    3730          CALL    EVAL                                    /80.01.GC/
060.125  072 201 042    3731          LDA     ACCX-1                                  /80.01.GC/
060.130  346 001        3732          ANI     CF.STR                                  /80.01.GC/
060.132  312 155 070    3733          JZ      ERR.TC          REQUIRE A STRING        /80.01.GC/
060.135  041 313 060    3734          LXI     H,MATCHC
060.140  315 051 076    3735          CALL    MOV4            SAVE S2 DESCRIPTION
060.143  315 223 072    3736          CALL    CMA             GOBBLE ','
                        3737
060.146  315 044 057    3738          CALL    EVALI8          EVALUATE INDEX
060.151  305            3739          PUSH    B               SAVE TEXT POINTER
060.152  103            3740          MOV     B,E             (B) = J
060.153  021 313 060    3741          LXI     D,MATCHC
060.156  315 315 074    3742          CALL    FSE             FIND S2
060.161  345            3743          PUSH    H               SAVE S2 ADDRESS
060.162  365            3744          PUSH    PSW             SAVE S2 COUNT
060.163  021 307 060    3745          LXI     D,MATCHA
060.166  315 315 074    3746          CALL    FSE             FIND S1
060.171  365            3747          PUSH    PSW             SAVE S1 LENGTH
060.172  175            3748          MOV     A,L
060.173  062 276 060    3749          STA     MATCHB          SAVE ADDRESS (IN PAGE) OF START
060.176  170            3750          MOV     A,B
060.177  247            3751          ANA     A
060.200  312 122 070    3752          JZ      ERR.IN          ILLEGAL NUMBER
060.203  075            3753          DCR     A
060.204  315 101 030    3754          CALL    $DADA.          (HL) = START OF SEARCH AREA
060.207  361            3755          POP     PSW
060.210  220            3756          SUB     B               (A) = LEN(S1)-J
060.211  332 257 060    3757          JC      MATC2.3         NOT ANYWHERE
060.214  074            3758          INR     A
060.215  301            3759          POP     B               (B) = S2 LENGTH
060.216  220            3760          SUB     B               (A) = # OF TRYS -1
060.217  332 260 060    3761          JC      MATC2.5         NONE
060.222  321            3762          POP     D               (DE) = S2 ADDRESS
060.223  074            3763          INR     A
060.224  365            3764          PUSH    PSW             SAVE TRY COUNT
                        3765
                        3766  *       SEE IF MATCH
                        3767
060.225  032            3768  MATCH1  LDAX    D
060.226  276            3769          CMP     M
060.227  302 247 060    3770          JNE     MATCH2          NOT THIS ONE
060.232  325            3771          PUSH    D
060.233  345            3772          PUSH    H
060.234  305            3773          PUSH    B               SAVE ALL REGS
060.235  110            3774          MOV     C,B             (C) = S2 LENGTH = COMPARE COUNT
060.236  315 060 030    3775          CALL    $COMP           COMPARE THE REST
060.241  301            3776          POP     B
060.242  341            3777          POP     H
```

```
060.243  321           3778           POP     D
060.244  312 273 060   3779           JE      MATCH3          GOT IT
                       3780
                       3781  *        NO MATCH AT THIS ONE
                       3782
060.247  043           3783  MATCH2   INX     H
060.250  361           3784           POP     PSW
060.251  075           3785           DCR     A
060.252  365           3786           PUSH    PSW             COUNT IT
060.253  302 225 060   3787           JNZ     MATCH1          MORE TO TRY
060.256  365           3788           PUSH    PSW             SET STACK PROPERLY
060.257  361           3789  MATC2.3  POP     PSW
060.260  361           3790  MATC2.5  POP     PSW
060.261  041 202 042   3791           LXI     H,ACCX
060.264  006 004       3792           MVI     B,4
060.266  315 212 031   3793           CALL    $ZERO           RETURN ZERO FOR ANSWER
060.271  301           3794           POP     B               RESTORE (BC)
060.272  311           3795           RET
                       3796
                       3797  *        GOT ONE
                       3798
060.273  361           3799  MATCH3   POP     PSW
060.274  175           3800           MOV     A,L             (A) = FWA OF STRING
060.275  326 000       3801           SUI     0               SUBTRACT START ADDRESS
060.276                3802  MATCHB   EQU     *-1             INDEX OF START OF S1
060.277  137           3803           MOV     E,A
060.300  026 000       3804           MVI     D,0
060.302  023           3805           INX     D               BIAS INTO 1 TO 256
060.303  301           3806           POP     B               RESTORE TEXT POINTER
060.304  303 040 075   3807           JMP     IFLT            FLOAT RESULT, AND EXIT
                       3808
060.307                3809  MATCHA   DS      4               HOLD AREA FOR STRING DESCRIPTOR
060.313                3810  MATCHC   DS      4               HOLD AREA FOR S2 DESCRIPTOR


                       3812  **       MAX - COMPUTE 'MAXIMUM' FUNCTION.
                       3813  *
                       3814  *        Y=MAX(X1,...,XN)
                       3815
                       3816
                       3817  **       MIN - COMPUTE 'MINIMUM' FUNCTION.
                       3818  *
                       3819  *        Y=MIN(X1,...,XN)
                       3820
                       3821
060.317  076           3822  MAX      DB      MI.MVIA
060.320  257           3823  MIN      XRA     A               (A) = MI.MVI IF MAX, 0 IF MIN
060.321  365           3824           PUSH    PSW             SAVE CODE
060.322  315 317 100   3825           CALL    XCY             (ACCY) = CURRENT CANDIDATE
                       3826
                       3827  *        (ACCY) = CURRENT CANDIDATE
                       3828
060.325  315 072 076   3829  MAX1     CALL    PNT             PEEK AT NEXT TOKEN
060.330  376 020       3830           CPI     CT.PAR
```

```
060.332  312 002 061  3831          JE      MAX2            IS )
060.335  315 223 072  3832          CALL    CMA             REQUIRE ','
060.340  315 041 077  3833          CALL    PSHY            SAVE CURRENT BEST
060.343  315 022 057  3834          CALL    EVALN           EVALUATE NUMBER
060.346  315 365 076  3835          CALL    POPY            RESTORE CURRENT BEST
060.351  315 033 077  3836          CALL    PSHX            SAVE LATEST
060.354  021 210 042  3837          LXI     D,ACCY
060.357  315 166 105  3838          CALL    FPSUB           COMPUTE (CANDIDATE-LATEST)
060.362  072 204 042  3839          LDA     ACCX+2
060.365  127          3840          MOV     D,A
060.366  315 357 076  3841          CALL    POPX            RESTORE LATEST TRY
060.371  361          3842          POP     PSW
060.372  365          3843          PUSH    PSW             (A) = MIN/MAX FLAG
060.373  252          3844          XRA     D               (A) = CODE
060.374  364 317 100  3845          CP      XCY             LATEST IS SUPERIOR
060.377  303 325 060  3846          JMP     MAX1
                      3847
                      3848  *       AT END OF LIST.
                      3849
061.002  361          3850  MAX2    POP     PSW
061.003  303 317 100  3851          JMP     XCY             (ACCX) = BEST FOUND


                      3853  **      PAD - READ KEYPAD.
                      3854  *
                      3855  *       Y=PAD(0)
                      3856
                      3857
061.006  315 260 003  3858  PAD     CALL    .RCK            ;READ VALUE
061.011  303 020 061  3859          JMP     PEEK1           RETURN VALUE


                      3861  **      PEEK - PEEK AT MEMORY.
                      3862  *
                      3863  *       X=PEEK(ADDR)
                      3864
                      3865
061.014  315 002 075  3866  PEEK    CALL    IFIX            EVAL TO 16 BITS
061.017  032          3867          LDAX    D
061.020  137          3868  PEEK1   MOV     E,A
061.021  026 000      3869          MVI     D,0             (DE) = VALUE
061.023  315 040 075  3870  PEEK1.5 CALL    IFLT            FLOAT INTO ACCX
061.026  076 300      3871  PEEK2   MVI     A,CT.SNV        SCALAR NUMERIC VALUE
061.030  062 201 042  3872          STA     ACCX-1
061.033  311          3873          RET
```

```
                        3875  **      PIN - PORT INPUT.
                        3876  *
                        3877  *       Y=PIN(PORT)
                        3878
                        3879
061.034  315 002 075   3880  PIN     CALL    IFIX
061.037  143           3881          MOV     H,E
061.040  056 333       3882          MVI     L,MI.IN
061.042  042 002 040   3883          SHLD    .IOWRK
061.045  315 002 040   3884          CALL    .IOWRK          INPUT
061.050  303 020 061   3885          JMP     PEEK1           FLOAT AND EXIT


                        3887  **      POS - RETURN PRINT HEAD POSITION.
                        3888  *
                        3889  *       X=POS(PORT)
                        3890
                        3891
061.053  315 002 075   3892  POS     CALL    IFIX
061.056  041 315 112   3893          LXI     H,COLCNTS
061.061  173           3894          MOV     A,E
061.062  247           3895          ANA     A
061.063  312 067 061   3896          JZ      POS1            IS CHANNEL 0
061.066  023           3897          INX     D               (DE) = INDEX INTO COLCNTS
061.067  031           3898  POS1    DAD     D
061.070  176           3899          MOV     A,M             (A) = POSITION
061.071  303 020 061   3900          JMP     PEEK1           FLOAT


                        3902  **      RND - COMPUTE TAUSWORTH 15 BIT RANDOM NUMBER.
                        3903  *
                        3904  *       X=RND(Y)
                        3905  *
                        3906  *       Y = 0, GIVE LAST RANDOM NUMBER
                        3907  *       Y > 0, GIVE NEXT RANDOM NUMBER
                        3908  *       Y < 0, SEED WITH Y
                        3909
                        3910
061.074                 3911  RND     EQU     *
061.074  072 204 042   3912          LDA     ACCX+2          EXAMINE SIGN
061.077  247           3913          ANA     A
061.100  041 114 107   3914          LXI     H,'GL'          (HL) = SEED
061.101                 3915  RNDA    EQU     *-2             SEED
061.103  026 017       3916          MVI     D,15            (D) = BIT COUNT
061.105  312 150 061   3917          JZ      RND2            JUST RETURN SEED
061.110  362 116 061   3918          JP      RND1            GENERATE NEW NUMBER
061.113  052 203 042   3919          LHLD    ACCX+1          USE NEW SEED
061.116  174           3920  RND1    MOV     A,H             SHIFT RIGHT ONE
061.117  247           3921          ANA     A
061.120  037           3922          RAR
061.121  147           3923          MOV     H,A
061.122  175           3924          MOV     A,L
```

```
061.123  037            3925           RAR
061.124  157            3926           MOV     L,A
061.125  027            3927           RAL                     'C' = 1
061.126  027            3928           RAL
061.127  027            3929           RAL
061.130  027            3930           RAL                     'C' = 10Q
061.131  255            3931           XRA     L               XOR WITH VALUE
061.132  027            3932           RAL
061.133  027            3933           RAL
061.134  027            3934           RAL
061.135  346 100        3935           ANI     100Q
061.137  264            3936           ORA     H               INSERT IN LEFT
061.140  147            3937           MOV     H,A
061.141  025            3938           DCR     D
061.142  302 116 061    3939           JNZ     RND1            MORE TO GO
061.145  042 101 061    3940           SHLD    RNDA            SAVE SEED
                        3941
061.150  353            3942  RND2     XCHG                    (DE) = VALUE
061.151  041 202 042    3943           LXI     H,ACCX
061.154  066 000        3944           MVI     M,0             ZERO LOW 8
061.156  043            3945           INX     H
061.157  163            3946           MOV     M,E
061.160  043            3947           INX     H
061.161  162            3948           MOV     M,D
061.162  043            3949           INX     H
061.163  066 200        3950           MVI     M,200Q          EXPONENT
061.165  303 202 105    3951           JMP     FPNRM           NORMALIZE AND EXIT


                        3953  **       SEG - DECODE SEGMENT VALUE.
                        3954  *
                        3955  *        Y=SEG(NUM)
                        3956  *
                        3957  *        DECODE VALUES 0-9.
                        3958  *        NUM = 10 GIVES BLANK.
                        3959
                        3960
061.170  315 002 075    3961  SEG      CALL    IFIX
061.173  041 356 003    3962           LXI     H,.DODA
061.176  031            3963           DAD     D
061.177  176            3964           MOV     A,M
061.200  366 200        3965           ORI     200Q            CLEAR DECIMAL
061.202  303 020 061    3966           JMP     PEEK1           DECODE VALUE


                        3968  **       SGN - RETURN SIGN OF NUMBER.
                        3969  *
                        3970  *        Y=SGN(X)
                        3971  *
                        3972  *        ; = -1 IF X<0, =0 IF X=0, =1 IF X > 0
                        3973
                        3974
```

```
061.205  021 000 000  3975  SGN     LXI     D,0
061.210  072 204 042  3976          LDA     ACCX+2
061.213  247          3977          ANA     A
061.214  312 040 075  3978          JZ      IFLT            IS 0
061.217  023          3979          INX     D
061.220  362 040 075  3980          JP      IFLT            IS POSITIVE
061.223  315 040 075  3981          CALL    IFLT
061.226  303 302 105  3982          JMP     FPNEG           MAKE -1


                      3984  **      STR$ - CONVERT FLOATING TO ASCII.
                      3985  *
                      3986  *       Y$=STR$(X)
                      3987
                      3988
061.231  041 335 113  3989  STR$    LXI     H,LINE2
061.234  315 301 110  3990          CALL    FTA             FLOATING TO ASCII
061.237  345          3991          PUSH    H               SAVE 'FROM'
061.240  157          3992          MOV     L,A
061.241  046 000      3993          MVI     H,0
061.243  365          3994          PUSH    PSW             SAVE LENTRH
061.244  042 202 042  3995          SHLD    ACCX            SET LENGTH
061.247  021 202 042  3996          LXI     D,ACCX
061.252  315 033 073  3997          CALL    CSE.            CREATE TEMP ENTRY
061.255  361          3998          POP     PSW             (A) = COUNT
061.256  321          3999          POP     D
061.257  315 015 071  4000  STR$1   CALL    MOV             MOVE IT
                      4001
                      4002  **      FRC - FUNCTION RETURNS CHARACTER VALUE.
                      4003
061.262  076 301      4004  FRC     MVI     A,CT.SSV        SCALAR STRING VALUE
061.264  062 201 042  4005          STA     ACCX-1
061.267  311          4006          RET


                      4008  **      VAL - CONVERT ASCII TO FLOATING POINT.
                      4009  *
                      4010  *       Y=VAL(X$)
                      4011
                      4012
061.270  021 202 042  4013  VAL     LXI     D,ACCX
061.273  315 315 074  4014          CALL    FSE             FIND STRINGTAB ENTRY
061.276  353          4015          XCHG
061.277  041 335 113  4016          LXI     H,LINE2
061.302  315 015 071  4017          CALL    MOV             MOVE TO *LINE*
061.305  066 000      4018          MVI     M,0             MAKE SURE TERMINATES
061.307  041 335 113  4019          LXI     H,LINE2
061.312  315 372 111  4020          CALL    $SOB            SKIP OVER BLANKS
061.315  315 323 107  4021          CALL    ATF             ASCII TO FLOATING
061.320  303 026 061  4022          JMP     PEEK2
```

```
                                4024  **      P.OR - PROCESS BOOLEAN 'OR'.
                                4025  *
                                4026  *       ENTRY   (ACCX) = 1ST VALUE
                                4027  *               (ACCY) = 1ND VALUE
                                4028  *       EXIT    (ACCX) = 1ST 'OR' 2ND
                                4029
                                4030
061.323  315 345 076            4031  P.OR    CALL    PBO             PROCESS BOOLEAN OPERATOR
061.326  172                    4032          MOV     A,D
061.327  264                    4033          ORA     H
061.330  127                    4034          MOV     D,A
061.331  173                    4035          MOV     A,E
061.332  265                    4036          ORA     L
061.333  303 371 061            4037          JMP     P.NOT1



                                4039  **      P.AND - PROCESS BOOLEAN AND.
                                4040  *
                                4041  *       ENTRY   NONE
                                4042  *       EXIT    (ACCX) = IFLT(IFIX(ACCX).AND.IFIX(ACCY))
                                4043
                                4044
061.336  315 345 076            4045  P.AND   CALL    PBO             PREPARE BOOLEAN
061.341  172                    4046          MOV     A,D
061.342  244                    4047          ANA     H
061.343  127                    4048          MOV     D,A
061.344  173                    4049          MOV     A,E
061.345  245                    4050          ANA     L
061.346  303 371 061            4051          JMP     P.NOT1



                                4053  **      P.NOT - PROCESS BOOLEAN 'NOT'.
                                4054  *
                                4055  *       ENTRY   NONE
                                4056  *       EXIT    (ACCX) = IFLT(.NOT.IFIX(ACCX))
                                4057
                                4058
061.351  072 201 042            4059  P.NOT   LDA     ACCX-1          (A) = TYPE OF ARGUMENT
061.354  376 300                4060          CPI     CT.SNV
061.356  302 155 070            4061          JNE     ERR.TC          WRONG TYPE
061.361  315 377 074            4062          CALL    IFIX.           (DE) = IFIX(ACCX)
061.364  172                    4063          MOV     A,D
061.365  057                    4064          CMA
061.366  127                    4065          MOV     D,A
061.367  173                    4066          MOV     A,E
061.370  057                    4067          CMA
061.371  137                    4068  P.NOT1  MOV     E,A             (DE) = RESULT
061.372  303 040 075            4069          JMP     IFLT            FLOAT AND EXIT
```

```
                                4071  **      P.CMP - PROCESS COMPARES.
                                4072  *
                                4073  *       Y= X1 'OP' X2
                                4074  *
                                4075  *       OP = < <= = <> >= >
                                4076  *
                                4077  *       THE TWO OPERANDS ARE COMPARED, AND THE RESULT GENERATES
                                4078  *       A BIT PATTERN:
                                4079  *
                                4080  *       001     EQUAL
                                4081  *       010     POSITIVE
                                4082  *       100     NEGATIVE
                                4083  *
                                4084  *       THIS PATTERN IS THEN MASKED WITH THE PATTERNS GENERATED BY THE
                                4085  *       OPERATORS:
                                4086  *
                                4087  *       =  001
                                4088  *       >  010
                                4089  *       >= 011
                                4090  *       <  100
                                4091  *       <= 101
                                4092  *       <> 110
                                4093  *
                                4094  *       ENTRY   (ACCX) = X2
                                4095  *               (ACCY) = X1
                                4096  *               (L) = OPERATOR 'CT.' CODE
                                4097  *       USES    A,F,D,E,H,L
                                4098
                                4099
061.375  345                   4100  P.CMP   PUSH    H               SAVE (H)
061.376  315 347 072           4101          CALL    COT             CHECK OPERAND TYPE
062.001  302 047 062           4102          JNZ     P.CMP2          IS STRING
                                4103
                                4104  *       IS NUMERIC COMPARE.
                                4105
062.004  315 166 105           4106          CALL    FPSUB           (ACCX) = X1-X2
062.007  072 204 042           4107          LDA     ACCX+2          (A) = SIGN OF RESULT
062.012  247                   4108          ANA     A               SET FLAGS
062.013  341                   4109  P.CMP1  POP     H
062.014  067                   4110          STC
062.015  077                   4111          CMC                     CLEAR CARRY
062.016  076 001               4112          MVI     A,1             ASSUME IS ZERO
062.020  312 030 062           4113          JZ      P.CMP13         IS ZERO
062.023  027                   4114          RAL
062.024  362 030 062           4115          JP      P.CMP13         IS POSITIVE
062.027  027                   4116          RAL                     IS NEGATIVE
062.030  157                   4117  P.CMP13 MOV     L,A             (L) = RESULTS OF TEST
062.031  174                   4118          MOV     A,H             (A) = TYPE
062.032  326 010               4119          SUI     CT.EQ-1         (A) = CODE FOR DCONDIIONS
062.034  245                   4120          ANA     L
062.035  021 000 000           4121          LXI     D,0             ASSUME FALSE
062.040  312 023 061           4122          JZ      PEEK1,5         IS FALSE
062.043  033                   4123          DCX     D
062.044  303 023 061           4124          JMP     PEEK1,5         IS TRUE
```

```
                          4126  **      STRING COMPARES.
                          4127  *
                          4128  *       COMPARE CHARACTER FOR CHARACTER. IF A STRING RUNS OUT, ITS
                          4129  *       NEXT CHARACTER IS CONSIDERED TO BE '00'
                          4130
                          4131
062.047  305             4132  P.CMP2  PUSH    B               SAVE (BC)
062.050  021 202 042     4133          LXI     D,ACCX
062.053  315 315 074     4134          CALL    FSE             FIND STRING ENTRY
062.056  107             4135          MOV     B,A             (B) = LEN(X2)
062.057  345             4136          PUSH    H               SAVE ADDRESS OF X2
062.060  021 210 042     4137          LXI     D,ACCY
062.063  315 315 074     4138          CALL    FSE             FIND ENTRY
062.066  117             4139          MOV     C,A             (C) = LEN(X1)
062.067  321             4140          POP     D               (DE) = ADR(X2), (HL) = ADR(X1)
062.070  353             4141          XCHG
062.071  004             4142          INR     B               (B) = LEN(X2)+1
062.072  014             4143          INR     C               (C) = LEN(X1)+1
                          4144
                          4145  *       COMPARE STRINGS.
                          4146
062.073  005             4147  P.CMP3  DCR     B
062.074  312 121 062     4148          JZ      P.CMP5          OUT OF X2
062.077  015             4149          DCR     C
062.100  312 115 062     4150          JZ      P.CMP4          OUT OF X1, BUT NOT X2
062.103  032             4151          LDAX    D
062.104  226             4152          SUB     M
062.105  302 130 062     4153          JNZ     P.CMP6          HAVE RESULT
062.110  023             4154          INX     D
062.111  043             4155          INX     H
062.112  303 073 062     4156          JMP     P.CMP3          TOO EARLY TO TELL
                          4157
                          4158  *       RAN OUT OF X1, BUT NOT X2. RESULT IS X1 < X2
                          4159
062.115  015             4160  P.CMP4  DCR     C               SET 'M' FLAG
062.116  303 130 062     4161          JMP     P.CMP6
                          4162
                          4163  *       RAN OUT OF X2, DONT KNOW ABOUT X1
                          4164
062.121  015             4165  P.CMP5  DCR     C
062.122  312 130 062     4166          JZ      P.CMP6          OUT OF BOTH
062.125  076 001         4167          MVI     A,1
062.127  247             4168          ANA     A               X2 > X1
                          4169
                          4170  *       HAVE COMPUARE RESULT IN PSW
                          4171
062.130  301             4172  P.CMP6  POP     B               RESTORE (BC)
062.131  303 013 062     4173          JMP     P.CMP1
```

```
                        4175  **      P.ADD - PROCESS ADD AND SUBTRACT.
                        4176  *
                        4177
                        4178
062.134  076 021       4179  P.ADD   MVI     A,CT.PL
062.136  274             4180          CMP     H
062.137  302 241 062     4181          JNE     P.SUB           IS -
062.142  315 347 072     4182          CALL    COT             CHECK OPERAND TYPE
062.145  312 352 104     4183          JZ      FPADD           IS NUMERIC ADD
                        4184
                        4185  *       IS STRING CONCATINATE.
                        4186
062.150  072 202 042     4187          LDA     ACCX
062.153  157             4188          MOV     L,A             (L) = LEN(X2)
062.154  072 210 042     4189          LDA     ACCY
062.157  205             4190          ADD     L               (A) = RESULTANT LENGTH
062.160  332 144 070     4191          JC      ERR.SL          STRING LENGTH ERROR
062.163  157             4192          MOV     L,A             (HL) = LEN
062.164  046 000         4193          MVI     H,0
062.166  042 235 062     4194          SHLD    P.ADDA          SAVE INDEX IN BUILD BLOCK AREA
062.171  021 235 062     4195          LXI     D,P.ADDA
062.174  315 033 073     4196          CALL    CSE.            CREATE TEMP STRINGTAB ENTRY
062.177  345             4197          PUSH    H               SAVE 'TO'
062.200  021 210 042     4198          LXI     D,ACCY
062.203  315 315 074     4199          CALL    FSE             FIND ENTRY
062.206  353             4200          XCHG                    (DE) = FROM
062.207  341             4201          POP     H               (HL) = TO
062.210  315 015 071     4202          CALL    MOV             COPY 1ST
062.213  345             4203          PUSH    H               SAVE TO
062.214  021 202 042     4204          LXI     D,ACCX
062.217  315 315 074     4205          CALL    FSE
062.222  353             4206          XCHG                    (DE) = FROM
062.223  341             4207          POP     H               (HL) = TO
062.224  315 015 071     4208          CALL    MOV
062.227  021 235 062     4209          LXI     D,P.ADDA
062.232  303 210 073     4210          JMP     CVX             COPY BLOCK TO ACCX
                        4211
062.235                 4212  P.ADDA  DS      4



                        4214  *
                        4215  *       (ACCX) = (ACCY-ACCX)
                        4216
                        4217
062.241  315 177 077     4218  P.SUB   CALL    RNO             REQUIRE NUMERIC OPERANDS
062.244  303 166 105     4219          JMP     FPSUB
```

```
                          4221  **      M.MUL - PROCESS MULTIPLICATION AND DIVISION.
                          4222  *
                          4223  *       (ACCX) = (ACCX)*(ACCY)
                          4224  *
                          4225  *       ENTRY   (DE) = #ACCY
                          4226
                          4227
062.247  076 024         4228  P.MUL   MVI     A,CT.DI
062.251  274             4229          CMP     H               SEE IF /
062.252  365             4230          PUSH    PSW             SAVE RESULT
062.253  315 177 077     4231          CALL    RND             REQUIRE NUMERIC OPERANDS
062.256  361             4232          POP     PSW
062.257  302 323 105     4233          JNE     FPMUL           IS *
062.262  315 317 100     4234          CALL    XCY             INVERT
062.265  303 260 106     4235          JMP     FPDIV           DIVIDE


                          4237  **      P.EXP - EXPONENTIATION.
                          4238  *
                          4239  *       (ACCX) = (VAL)^(POWR)
                          4240  *
                          4241  *       IF (ACCY)>0, COMPUTE RSLT=EXP(X*LOG(Y))
                          4242  *
                          4243  *       ENTRY   (ACCY) = VAL
                          4244  *               (ACCX) = POWER
                          4245  *       EXIT    (ACCX) = RESULT
                          4246
                          4247
062.270                   4248  P.EXP   EQU     *
062.270  315 317 100     4249          CALL    XCY             (ACCX) = VAL
062.273  072 205 042     4250          LDA     ACCX+3          CHECK IF VAL IS 0
062.276  247             4251          ANA     A
062.277  302 321 062     4252          JNZ     P.EXP1          VAL NON = ZERO
                          4253
                          4254  *       CHECK FOR 0^0
                          4255
062.302  072 213 042     4256          LDA     ACCY+3          CHECK IF POWER IS 0
062.305  247             4257          ANA     A
062.306  300             4258          RNZ                     EXIT; POWER NON = ZERO; VAL = 0
                          4259
062.307  021 211 112     4260          LXI     D,FP1.0         POWER = ZERO; RETURN RESULT OF 1
062.312  041 202 042     4261          LXI     H,ACCX
062.315  315 051 076     4262          CALL    MOV4            (ACCX) = 1
062.320  311             4263          RET                     EXIT
                          4264
062.321  315 041 077     4265  P.EXP1  CALL    PSHY            SAVE EXPONENT
062.324  315 225 063     4266          CALL    LOG             (ACCX) = LOG(Y)
062.327  315 365 076     4267          CALL    POPY
062.332  315 323 105     4268          CALL    FPMUL
062.335  303 075 063     4269          JMP     EXP             (ACCX) = EXP(X*LOG(Y))
```

```
                        4273  **      TXTFN - PERFORM TEXT DEFINED FUNCTIONS.
                        4274  *
                        4275
                        4276
062.340                 4277  TXTFN   EQU     *
062.340  315 056 071    4278          CALL    ANT             ACCEPT NEXT TOKEN
062.343  346 002        4279          ANI     CF.VEC
062.345  312 152 070    4280          JZ      ERR.SY          NOT VECTOR TYPE
062.350  032            4281          LDAX    D
062.351  247            4282          ANA     A
062.352  362 216 070    4283          JP      ERR.UD          NOT DECLARED AS FUNCTION
062.355  023            4284          INX     D
062.356  353            4285          XCHG
062.357  136            4286          MOV     E,M
062.360  043            4287          INX     H
062.361  126            4288          MOV     D,M             (DE) = ADDRESS OF FUNCTION DEFINITION
062.362  353            4289          XCHG
062.363  305            4290  TXTF1   PUSH    B
062.364  343            4291          XTHL
062.365  301            4292          POP     B               (BC) = ADDRESS OF PARAMETER LIST
                        4293
                        4294  *       ASSIGN VALUES TO PARAMETER LIST.
                        4295
062.366  345            4296  TXTF2   PUSH    H               SAVE (HL)
062.367  315 136 075    4297          CALL    IST             INSERT SYMBOL IN TABLE
062.372  341            4298          POP     H
062.373  325            4299          PUSH    D               SAVE INDEX
062.374  365            4300          PUSH    PSW
062.375  315 056 071    4301          CALL    ANT             EXAMINE NEXT TOKEN
063.000  365            4302          PUSH    PSW             SAVE FOR LATER
063.001  376 026        4303          CPI     CT.CMA
063.003  312 013 063    4304          JE      TXTF3           IS ,
063.006  376 020        4305          CPI     CT.PAR
063.010  302 152 070    4306          JNE     ERR.SY          BAD SYNTAX
                        4307
                        4308  *       SWAP (BC) (HL) TO DECODE VALUE FOR VARIABLE
                        4309
063.013  305            4310  TXTF3   PUSH    B
063.014  343            4311          XTHL
063.015  301            4312          POP     B
063.016  315 244 055    4313          CALL    EVAL            EVALUATE PARAMETER VALUE
063.021  361            4314          POP     PSW             (A) = NEXT CHARACTER FROM *ANT*
063.022  062 041 063    4315          STA     TXTFNA          SAVE FOR COMPARISON
063.025  361            4316          POP     PSW             RESTORE TYPE
063.026  321            4317          POP     D               RESTORE PARM ADDRESS
063.027  315 366 072    4318          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL
063.032  315 202 071    4319          CALL    AVV             ASSIGN VALUE TO VARIABLE
063.035  315 056 071    4320          CALL    ANT             CHECK SEPERATOR
063.040  376 000        4321          CPI     0               MUST BE SAME AS FUNCTION LIST
063.041                 4322  TXTFNA  EQU     *-1
063.042  302 205 070    4323          JNE     ERR.AC          ARG COUNT ERROR
063.045  376 020        4324          CPI     CT.PAR
063.047  302 363 062    4325          JNE     TXTF1           MORE TO ASSIGN
063.052  305            4326          PUSH    B               EXCHANGE POINTERS
063.053  343            4327          XTHL
063.054  301            4328          POP     B
```

```
063.055  345           4329        PUSH    H           SAVE CALLER POINTER
063.056  315 305 077   4330        CALL    RNT
063.061  011           4331        DB      CT.EQ       REQUIRE =
063.062  315 244 055   4332        CALL    EVAL        EVALUATE FUNCTION
063.065  315 305 077   4333        CALL    RNT
063.070  000           4334        DB      CT.FIN      REQUIRE STATEMENT END
063.071  301           4335        POP     B
063.072  303 072 076   4336        JMP     PNT         EXIT WITH NEXT TOKEN PEEKED
```

```
                                4340  **      EXP - CALCULATE EXP(X).
                                4341  *
                                4342  *       Y=EXP(X)
                                4343  *
                                4344  *       VIA:
                                4345  *
                                4346  *       X1 = X * LN(2)^-1
                                4347  *
                                4348  *       Y = 2^(INT(X1)) * 2^(FRACT(X1))
                                4349  *
                                4350  *       FRACT(X) [0,1) = P5(X)
                                4351
                                4352
063.075                         4353  EXP     EQU     *
063.075  305                   4354          PUSH    B               SAVE TEXT POINTER
063.076  072 204 042           4355          LDA     ACCX+2          (A) = SIGN
063.101  247                   4356          ANA     A
063.102  365                   4357          PUSH    PSW             SAVE RESULTS
063.103  374 305 105           4358          CM      NEG             INSURE POSITIVE
063.106  041 221 063           4359          LXI     H,EXPA
063.111  315 327 105           4360          CALL    MUL             (ACCX) = X * LN(2)^-1
063.114  315 223 073           4361          CALL    CXY             SAVE IN ACCY
063.117  315 377 074           4362          CALL    IFIX.           (DE) = INT(X1)
063.122  172                   4363          MOV     A,D
063.123  247                   4364          ANA     A
063.124  312 133 063           4365          JZ      EXP1            EXPONENT NOT TOO BIG
063.127  074                   4366          INR     A
063.130  302 136 070           4367          JNZ     ERR,OV          EXPONENT TOO BIG
063.133  325                   4368  EXP1    PUSH    D               SAVE EXP
063.134  315 040 075           4369          CALL    IFLT            FLOAT INTO ACCX
063.137  041 210 042           4370          LXI     H,ACCY
063.142  315 172 105           4371          CALL    SUB             (ACCX) = FRACT(X1)
063.145  315 177 065           4372          CALL    POLY            EVALUATE P5(X)
063.150  006                   4373          DB      6
063.151  202 014 173           4374          DB      202Q,014Q,173Q,167Q     .001877576677
063.155  003 244 111           4375          DB      003Q,244Q,111Q,172Q     .008989340083
063.161  021 125 162           4376          DB      021Q,125Q,162Q,174Q     .05582631806
063.165  152 365 172           4377          DB      152Q,365Q,172Q,176Q     .2401536170
063.171  075 271 130           4378          DB      075Q,271Q,130Q,200Q     .6931530732
063.175  377 377 177           4379          DB      377Q,377Q,177Q,200Q     .9999999250
                                4380
063.201  321                   4381          POP     D               (DE) = EXP OF 2^(INT(X1))
063.202  173                   4382          MOV     A,E             (A) = EXPONENT ADJUSTMENT
063.203  041 205 042           4383          LXI     H,ACCX+3
063.206  206                   4384          ADD     M               ADJUST EXPONENT
063.207  167                   4385          MOV     M,A
063.210  332 136 070           4386          JC      ERR.OV          OVERFLOW
063.213  361                   4387          POP     PSW             (A) = RESULTS OF INITIAL SIGN TEST
063.214  374 312 065           4388          CM      RCX             EXP(X) = 1/EXP(-X)
063.217  301                   4389          POP     B               RESTORE BC
063.220  311                   4390          RET
                                4391
063.221  035 125 134           4392  EXPA    DB      035Q,125Q,134Q,201Q     1/LN(2)
```

```
                                4394  **      LOG - CALCULATE LOG BAS E
                                4395  *
                                4396  *       Y=LOG(X)
                                4397  *
                                4398  *       VIA:
                                4399  *
                                4400  *       LOGE(X) = LOGE(2)* LOG2(X)
                                4401  *
                                4402  *       LOG2(X) = EXPONENT(X) + LOG2(MANTISSA)
                                4403  *
                                4404  *       LOG2(N) [.5,1) = P3(X)/P2(X)
                                4405
                                4406
063.225                         4407  LOG     EQU     *
063.225  305                    4408          PUSH    B               SAVE TEXT POINTER
063.226  041 204 042            4409          LXI     H,ACCX+2
063.231  176                    4410          MOV     A,M             (A) = SIGN
063.232  247                    4411          ANA     A
063.233  372 122 070            4412          JM      ERR.IN          MUST BE > 0                     /80;01;6C/
063.236  312 136 070            4413          JZ      ERR.OV
063.241  043                    4414          INX     H
063.242  136                    4415          MOV     E,M
063.243  026 000                4416          MVI     D,0             (DE) = EXPONENT
063.245  325                    4417          PUSH    D               SAVE
063.246  066 200                4418          MVI     M,200Q          (ACCX) = MANTISSA
063.250  315 153 065            4419          CALL    POLYQ           COMPUTE P3(X)/P2(X)
063.253  004                    4420          DB      4
063.254  000 000 100            4421          DB      000Q,000Q,100Q,201Q       1.0
063.260  160 330 146            4422          DB      160Q,330Q,146Q,203Q       6.4278 42090
063.264  005 271 110            4423          DB      005Q,271Q,110Q,203Q       4.5451 70876
063.270  172 202 132            4424          DB      172Q,202Q,132Q,177Q       .35355 34252
063.274  004                    4425          DB      4
063.275  314 373 114            4426          DB      314Q,373Q,114Q,203Q       4.8114 74609
063.301  221 261 141            4427          DB      221Q,261Q,141Q,203Q       6.1058 51990
063.305  106 031 271            4428          DB      106Q,031Q,271Q,204Q      -8.8626 59939
063.311  054 100 276            4429          DB      054Q,100Q,276Q,202Q      -2.0546 66719
063.315  315 223 073            4430          CALL    CXY             (ACCY) = LOG2(MANTISSA)
063.320  321                    4431          POP     D               (DE) = EXPONENT
063.321  315 040 075            4432          CALL    IFLT
063.324  041 350 063            4433          LXI     H,LOGA
063.327  315 356 104            4434          CALL    ADD             REMOVE EXPONENT BIAS
063.332  041 210 042            4435          LXI     H,ACCY
063.335  315 356 104            4436          CALL    ADD             (ACCX) = EXPONENT+LOG2(MANTISSA)
063.340  041 354 063            4437          LXI     H,LOGB
063.343  315 327 105            4438          CALL    MUL             (ACCX) = LOGE(2)*LOG2(X)
063.346  301                    4439          POP     B
063.347  311                    4440          RET
                                4441
063.350  000 000 200            4442  LOGA    DB      000Q,000Q,200Q,207Q       -128.
063.354  014 271 130            4443  LOGB    DB      014Q,271Q,130Q,200Q       LOGE(2)
```

```
                                4445  **      SQRT - SQUARE ROOT.
                                4446  *
                                4447  *       Y=SQRT(X)
                                4448  *
                                4449  *       VIA:
                                4450  *
                                4451  *       SQRT(X) = 2^B * SQRT(X*2^(-2*B))
                                4452  *
                                4453  *       SQRT(X1) [.25,1) = P2(X)/P2(X)
                                4454
                                4455
063.360                         4456  SQR     EQU     *
063.360  305                    4457          PUSH    B
063.361  315 033 077            4458          CALL    PSHX            SAVE X
063.364  041 204 042            4459          LXI     H,ACCX+2
063.367  176                    4460          MOV     A,M             (A) = SIGN
063.370  247                    4461          ANA     A
063.371  372 122 070            4462          JM      ERR.IN          MUST BE >= 0
063.374  312 112 064            4463          JZ      SQRT3           IS ZERO
063.377  043                    4464          INX     H
064.000  176                    4465          MOV     A,M             (A) = EXPONENT
                                4466
                                4467  *       EXPONENT >= 200Q. SCALE TO 177 OR 200
                                4468
064.001  326 177                4469          SUI     177Q
064.003  037                    4470          RAR                     (A) = B (SCALE FACTOR)
064.004  365                    4471          PUSH    PSW             SAVE FACTOR
064.005  077                    4472          CMC
064.006  303 015 064            4473          JMP     SQRT2
                                4474
                                4475  *       EXPONENT < 200Q. SCLAE TO 177 OR 200
                                4476
064.011  326 177                4477  SQRT1   SUI     177Q
064.013  037                    4478          RAR                     (A) = B (SCALE FACTOR)
064.014  365                    4479          PUSH    PSW             SAVE SCALE FACTOR
064.015  076 200                4480  SQRT2   MVI     A,200Q
064.017  336 000                4481          SBI     0               (A) = 200Q OR 177Q
064.021  167                    4482          MOV     M,A             (ACCX) = SCALED VALUE
064.022  315 177 065            4483          CALL    POLY            EVALUTE POLY
064.025  005                    4484          DB      5
064.026  053 017 255            4485          DB      053Q,017Q,255Q,177Q     -.32398 73450
064.032  327 005 104            4486          DB      327Q,005Q,104Q,201Q      1.062856525
064.036  153 213 241            4487          DB      153Q,213Q,241Q,201Q     -1.4758 65807
064.042  170 366 142            4488          DB      170Q,366Q,142Q,201Q      1.546293465
064.046  362 202 141            4489          DB      362Q,202Q,141Q,176Q      .19045 21794
                                4490
064.052  361                    4491          POP     PSW             (A) = EXPONENT ADJUST
064.053  041 205 042            4492          LXI     H,ACCX+3
064.056  206                    4493          ADD     M               ADJUST EXPONENT
064.057  167                    4494          MOV     M,A
                                4495
                                4496  *       APPLY HERON'S ITTERATION ONCE.
                                4497
064.060  315 223 073            4498          CALL    CXY             ACCY = GUESS
064.063  315 357 076            4499          CALL    POPX            ACCX = X
064.066  041 210 042            4500          LXI     H,ACCY
```

```
064.071  345              4501          PUSH    H
064.072  315 264 106      4502          CALL    DIV
064.075  341              4503          POP     H               (HL) = #ACCY
064.076  315 356 104      4504          CALL    ADD             (ACCX) = GUESS+X/GUESS
064.101  041 205 042      4505          LXI     H,ACCX+3
064.104  176              4506          MOV     A,M
064.105  326 001          4507          SUI     1               DIVIDE BY 2
064.107  167              4508          MOV     M,A
064.110  301              4509          POP     B               RESTORE (BC)
064.111  311              4510          RET
                          4511
064.112  315 357 076      4512  SQRT3   CALL    POPX            RESTORE STACK
064.115  301              4513          POP     B               RESTORE (BC)
064.116  311              4514          RET                     EXIT


                          4516  **      SINCOS - SIN AND COSIN.
                          4517  *
                          4518  *       Y=SIN(X)
                          4519  *       Y=COS(X)
                          4520  *
                          4521  *       REDUCE RANGE FROM 0 TO PI/2, APPROXIMATE WITH
                          4522  *
                          4523  *       COS(X) = P4(X)
                          4524
                          4525
064.117                   4526  SIN     EQU     *
064.117  021 225 112      4527          LXI     D,NPI.2
064.122  315 352 104      4528          CALL    FPADD           SIN(X) = COS(X-PI/2)
                          4529
064.125                   4530  COS     EQU     *
064.125  305              4531          PUSH    B
064.126  315 252 065      4532          CALL    PTS             PERFORM TRIG SCALEING
064.131  072 204 042      4533          LDA     ACCX+2
064.134  247              4534          ANA     A
064.135  374 305 105      4535          CM      NEG             COS(-X) = COS(X)
                          4536
                          4537  *       REDUCE RANGE TO 0<=X<=2*PI
                          4538
064.140  041 231 112      4539          LXI     H,NPI2          POINT TO -2PI
064.143  315 331 065      4540          CALL    RAR             REDUCE ARGUMENT RANGE
                          4541
                          4542  *       REDUCE RANGE TO 0<=X<=PI/2
                          4543
064.146  041 225 112      4544          LXI     H,NPI.2
064.151  315 331 065      4545          CALL    RAR
064.154  074              4546          INR     A
064.155  037              4547          RAR
064.156  062 234 064      4548          STA     COSA            (COSA) = ODD IF TO INVERT SIGN
064.161  332 175 064      4549          JC      COS1            IF < PI/2
064.164  041 225 112      4550          LXI     H,NPI.2         (X) = -(X-PI/2)
064.167  315 356 104      4551          CALL    ADD
064.172  315 305 105      4552          CALL    NEG
064.175  041 202 042      4553  COS1    LXI     H,ACCX
```

```
064.200  315 327 105  4554          CALL    MUL             (ACCX) = X**X
064.203  315 177 065  4555          CALL    POLY
064.206  005          4556          DB      5
064.207  130 035 141  4557          DB      130Q,035Q,141Q,161Q      .00002315393167
064.213  130 065 245  4558          DB      130Q,065Q,245Q,167Q     -.00138 53704 276
064.217  267 123 125  4559          DB      267Q,123Q,125Q,174Q      .04166358467
064.223  020 000 200  4560          DB      020Q,000Q,200Q,177Q     -.49999 90534
064.227  000 000 100  4561          DB      000Q,000Q,100Q,201Q      .9999999534
                      4562
                      4563  *       NEGATE SIGN OF RESULT, IF NECESSARY
                      4564
064.233  076 000      4565  COS2    MVI     A,0
064.234               4566  COSA    EQU     *-1             ODD IF TO TOGGLE SIGN
064.235  037          4567          RAR
064.236  334 305 105  4568          CC      NEG
064.241  301          4569          POP     B
064.242  311          4570          RET


                      4572  **      TAN - COMPUTE TANGENT FUNCTION.
                      4573  *
                      4574
                      4575
064.243               4576  TAN     EQU     *
064.243  315 252 065  4577          CALL    PTS             PERFORM TRIG SCALING
064.246  072 204 042  4578          LDA     ACCX+2
064.251  247          4579          ANA     A
064.252  310          4580          RZ                      TAN(0) = 0
064.253  305          4581          PUSH    B
064.254  007          4582          RLC
064.255  062 234 064  4583          STA     COSA            SET NEGATION FLAG
064.260  334 305 105  4584          CC      NEG             TAN(-X) = -TAN(X)
064.263  041 235 112  4585          LXI     H,NPI           REDUCE RANGE BY PI
064.266  315 331 065  4586          CALL    RAR             REDUCE ARGUMENT RANGE
                      4587
                      4588  *       REDUCE IT BY PI/2
                      4589
064.271  041 225 112  4590          LXI     H,NPI.2
064.274  315 331 065  4591          CALL    RAR
064.277  247          4592          ANA     A
064.300  312 321 064  4593          JZ      TAN1            WAS IN RANGE 0 - PI/2
064.303  041 234 064  4594          LXI     H,COSA
064.306  256          4595          XRA     M
064.307  167          4596          MOV     M,A             TAN(X) = -TAN(PI-X)
064.310  041 225 112  4597          LXI     H,NPI.2
064.313  315 356 104  4598          CALL    ADD
064.316  315 305 105  4599          CALL    NEG             ACCX = -(X-PI)
                      4600
                      4601  *       SCALE TO PI/4
                      4602
064.321  041 241 112  4603  TAN1    LXI     H,NPI.4
064.324  315 331 065  4604          CALL    RAR             REDUCE ARGUMENT RANGE
064.327  062 016 065  4605          STA     TANA            SAVE COUNT
064.332  247          4606          ANA     A
```

```
064.333  312 347 064  4607          JZ      TAN2
                      4608
                      4609  *       TAN(X) = 1/TAN(PI/2-X)
                      4610
064.336  041 241 112  4611          LXI     H,NPI.4
064.341  315 356 104  4612          CALL    ADD
064.344  315 305 105  4613          CALL    NEG             (ACCX) = -(X-PI/2)
                      4614
064.347  041 245 112  4615  TAN2    LXI     H,PI.4
064.352  315 264 106  4616          CALL    DIV             (ACCX) = X/(PI/4)
064.355  315 142 065  4617          CALL    XPOLYQ          COMPUTE P1(X^2)/P2(X^2)
064.360  003          4618          DB      3
064.361  000 000 100  4619          DB      000Q,000Q,100Q,201Q     1.
064.365  151 147 270  4620          DB      151Q,147Q,270Q,207Q     71.59606050
064.371  346 235 103  4621          DB      346Q,235Q,103Q,211Q     270.4672235
064.375  002          4622          DB      2
064.376  331 222 233  4623          DB      331Q,222Q,233Q,204Q     -12.55329742
065.002  124 066 152  4624          DB      124Q,066Q,152Q,210Q     212.42445758
                      4625
065.006  315 365 076  4626          CALL    POPY            (ACCY) = X
065.011  353          4627          XCHG
065.012  315 327 105  4628          CALL    MUL             X*P1/P2
065.015  076 000      4629          MVI     A,0
065.016               4630  TANA    EQU     *-1
065.017  037          4631          RAR
065.020  334 312 065  4632          CC      RCX             TAKE RECIPRICAL OF ACCX
065.023  303 233 064  4633          JMP     COS2            NEGATE RESULT, IF NECESSARY


                      4635  **      ATAN - ATAN(X)
                      4636  *
                      4637
                      4638
065.026               4639  ATN     EQU     *
065.026  305          4640          PUSH    B
065.027  072 204 042  4641          LDA     ACCX+2
065.032  007          4642          RLC
065.033  062 234 064  4643          STA     COSA            SET NEGATE FLAG
065.036  334 305 105  4644          CC      NEG             ATAN(-X) = -ATAN(X)
065.041  072 205 042  4645          LDA     ACCX+3
065.044  326 201      4646          SUI     201Q
065.046  365          4647          PUSH    PSW             SAVE RANGE
065.047  324 312 065  4648          CNC     RCX             IF VALUE > 1, TAKE RECIPROCAL
                      4649
065.052  315 142 065  4650  ATAN1   CALL    XPOLYQ          =X*P3(X^2)/P2(X^2)
065.055  003          4651          DB      3
065.056  000 000 100  4652          DB      000Q,000Q,100Q,201Q     1.
065.062  156 132 103  4653          DB      156Q,132Q,103Q,203Q     4.2095 84416
065.066  332 176 164  4654          DB      332Q,176Q,164Q,202Q     3.640485264
065.072  004          4655          DB      4
065.073  156 000 252  4656          DB      156Q,000Q,252Q,172Q     -.01049 78419 9
065.077  104 042 123  4657          DB      104Q,042Q,123Q,177Q     .32474 16032
065.103  013 340 137  4658          DB      013Q,340Q,137Q,202Q     2.996099356
065.107  332 176 164  4659          DB      332Q,176Q,164Q,202Q     3.640485163
```

```
                          4660
065.113  315 365 076    4661          CALL    POPY
065.116  353            4662          XCHG
065.117  315 327 105    4663          CALL    MUL             MULTIPLY RESULT BY X
065.122  361            4664          POP     PSW             RESTORE INVERT CODE
065.123  332 137 065    4665          JC      ATAN2           NOT INVERTED
065.126  041 225 112    4666          LXI     H,NPI.2         PI/2-ATAN(1/X)= ATAN(X)
065.131  315 356 104    4667          CALL    ADD
065.134  315 305 105    4668          CALL    NEG             ACCX = -(X-PI/2)
065.137  303 233 064    4669  ATAN2   JMP     COS2            NEGATE IF NECESSARY


                          4671  **      XPOLYQ - EVALUATE X*P(X^2)/Q(X^2)
                          4672  *
                          4673  *       ENTRY   (ACCX) = VALUE
                          4674  *               (RET) = QUOTIENT LIST, NUMERATOR FIRST
                          4675  *       EXIT    TO AFTER LIST
                          4676  *       USES    ALL
                          4677
                          4678
065.142  315 033 077    4679  XPOLYQ  CALL    PSHX            SAVE X
065.145  041 202 042    4680          LXI     H,ACCX
065.150  315 327 105    4681          CALL    MUL             X=X^2


                          4683  **      POLYQ - EVALUATE P(X)/Q(X)
                          4684  *
                          4685  *       ENTRY   (ACCX) = X
                          4686  *               (RET) = QUOTIENT LIST, NUMERATOR FIRST
                          4687  *       EXIT    TO AFTER LIST
                          4688  *       USES    ALL
                          4689
                          4690
065.153  341            4691  POLYQ   POP     H               (HL) = LIST ADDRESS
065.154  315 204 065    4692          CALL    PLY             COMPUTE DENOMINATOR
065.157  345            4693          PUSH    H               SAVE (HL)
065.160  315 033 077    4694          CALL    PSHX            SAVE QUOTIENT
065.163  341            4695          POP     H               RESTORE (HL)
065.164  315 207 065    4696          CALL    PLYO            COMPUTE NUMERATOR
065.167  345            4697          PUSH    H               SAVE RETURN ADDRESS
065.170  315 365 076    4698          CALL    POPY            (ACCY) = DENIMINATOR
065.173  353            4699          XCHG                    (HL) = #ACCY
065.174  303 264 106    4700          JMP     DIV             DIVIDE AND RETURN
```

```
                                4702  **      POLY - EVALUATE POLYNOMIAL.
                                4703  *
                                4704  *       ENTRY   ACCX = X
                                4705  *               (RET) = COEFFICIENT LIST
                                4706  *       EXIT    TO AFTER LIST
                                4707  *       USES    ALL
                                4708
                                4709
065.177  341                   4710  POLY    POP     H               (HL) = RETURN ADDRESS
065.200  315 204 065           4711          CALL    PLY             COMPUTE
065.203  351                   4712          PCHL


                                4714  **      PLY - COMPUTE POLYNOMIAL.
                                4715  *
                                4716  *       ACCX = PN(X)
                                4717  *
                                4718  *       COMPUTE A + X(B + X(C + X(D....)))
                                4719
                                4720
065.204  315 223 073           4721  PLY     CALL    CXY             (ACCY) = ACCX VALUE
065.207  176                   4722  PLY0    MOV     A,M             (A) = COUNT
065.210  365                   4723          PUSH    PSW
065.211  043                   4724          INX     H
065.212  353                   4725          XCHG                    (DE) = ADDRESS
065.213  315 210 073           4726          CALL    CVX             (ACCX) = D
065.216  353                   4727          XCHG                    (HL) = ADDRESS OF D
065.217  303 240 065           4728          JMP     PLY2
                                4729
065.222  365                   4730  PLY1    PUSH    PSW             SAVE COUNT
065.223  345                   4731          PUSH    H               SAVE ADDRESS
065.224  041 210 042           4732          LXI     H,ACCY
065.227  315 327 105           4733          CALL    MUL             COMPUTE X(...)
065.232  341                   4734          POP     H
065.233  345                   4735          PUSH    H
065.234  315 356 104           4736          CALL    ADD             COMPUTE A + X(...)
065.237  341                   4737          POP     H
065.240  361                   4738  PLY2    POP     PSW
065.241  043                   4739          INX     H
065.242  043                   4740          INX     H
065.243  043                   4741          INX     H
065.244  043                   4742          INX     H
065.245  075                   4743  POLY2   DCR     A
065.246  302 222 065           4744          JNZ     PLY1            IF MORE TO GO
065.251  311                   4745          RET                     DONE
```

```
                                4747  **      PTS - PERFORM TRIG SCALING.
                                4748  *
                                4749  *       PTS SCALES A VALUE INTO THE RANGE -2*PI <= X <= 2*PI
                                4750  *       ONLY IF -10*PI <= X <= 10*PI.
                                4751  *
                                4752  *       FOR VALUES WITHIN THIS RANGE, THE ADDITIVE SCALING OF THE
                                4753  *       FUNCTIONS THEMSELVES IS MORE EFFICIENT.
                                4754  *
                                4755  *       ENTRY   (ACCX) = X
                                4756  *       EXIT    (ACCX) = SCALED VALUE
                                4757  *       USES    A,F,D,E,H,L
                                4758
                                4759
065.252  072 205 042           4760  PTS     LDA     ACCX+3          (A) = EXPONENT
065.255  376 206               4761          CPI     206Q
065.257  330                   4762          RC                      DOSENT NEED IT
065.260  305                   4763          PUSH    B               SAVE (BC)
                                4764
                                4765  *       COMPUTE SCALED = X - INT(X/2*PI) * 2*PI
                                4766
065.261  315 223 073           4767          CALL    CXY             (ACCY) = X
065.264  041 231 112           4768          LXI     H,NPI2
065.267  345                   4769          PUSH    H               SAVE ADDRESS OF NPI2
065.270  315 264 106           4770          CALL    DIV
065.273  315 216 057           4771          CALL    INT             FIX
065.276  341                   4772          POP     H
065.277  315 327 105           4773          CALL    MUL
065.302  041 210 042           4774          LXI     H,ACCY
065.305  315 172 105           4775          CALL    SUB             TAKE DIFFERENCE
065.310  301                   4776          POP     B
065.311  311                   4777          RET


                                4779  **      RCX - TAKE RECIPROCAL OF (ACCX).
                                4780  *
                                4781  *       (ACCX) = 1/(ACCX)
                                4782  *
                                4783  *       ENTRY   NONE
                                4784  *       EXIT    NONE
                                4785  *       USES    ALL
                                4786
                                4787
065.312  315 223 073           4788  RCX     CALL    CXY             (ACCY) = X
065.315  021 211 112           4789          LXI     D,FP1.0
065.320  315 210 073           4790          CALL    CVX             COPY VALUE INTO ACCX
065.323  041 210 042           4791          LXI     H,ACCY
065.326  303 264 106           4792          JMP     DIV             ACCX = 1/(ACCX)
```

```
                                4794  **      RAR - REDUCE ARGUMENT RANGE.
                                4795  *
                                4796  *       RAR REDUCES THE ARGUMENT RANGE OF A VALUE BY REPEATED
                                4797  *       ADDITION WITH A NEGATIVE CONSTANT, UNTIL THE NUMBER IS
                                4798  *       SMALLER THAN ABS(CONSTANT)
                                4799  *
                                4800  *       ENTRY   (HL) = CONSTANT
                                4801  *       EXIT    (A) = ADDITION COUNT
                                4802  *               (HL) UNCHANGED
                                4803  *       USES    A,F,B,C,D,E
                                4804
                                4805
065.331  257                   4806  RAR     XRA     A
                                4807
065.332  365                   4808  RAR1    PUSH    PSW             SAVE COUNT
065.333  315 223 073           4809          CALL    CXY             SAVE VALUE IN ACCY
065.336  345                   4810          PUSH    H
065.337  315 356 104           4811          CALL    ADD             SUBTRACE
065.342  341                   4812          POP     H
065.343  072 204 042           4813          LDA     ACCX+2
065.346  247                   4814          ANA     A
065.347  372 357 065           4815          JM      RAR2            DONE
065.352  361                   4816          POP     PSW
065.353  074                   4817          INR     A
065.354  303 332 065           4818          JMP     RAR1
                                4819
065.357  315 317 100           4820  RAR2    CALL    XCY             COPY LAST VALUE INTO ACCX
065.362  361                   4821          POP     PSW
065.363  311                   4822          RET
```

```
                                4826  **      ICL - INPUT COMMAND LINE.
                                4827  *
                                4828  *       ICL INPUTS A COMMAND INTO *LINE*.
                                4829  *
                                4830  *       KEYWORDS ARE EXPANDED UNLESS
                                4831  *               1) THEY FOLLOW A 'REM' KEYWORD
                                4832  *               2) THEY APPEAR IN QUOTES
                                4833  *
                                4834  *       ICL MAKES (AND ENFORCES) CERTAIN ASSUMTTIONS ABOUT
                                4835  *       LEXICAL SYNTAX
                                4836  *               1) A PAIR OF ALPHA CHARACTERS MAY ONLY APPEAR IN A
                                4837  *                   KEYWORD, OR WITHIN A 'REM' OR QUOTED STRING.
                                4838  *               2) ALL KEYWORDS ARE UNIQUE WITHIN THE 1ST 3 CHARACTERS.
                                4839  *
                                4840  *       IF A CTL-C IS ENTERED, ICL EXITS WITH NO TEXT.
                                4841  *
                                4842  *       ENTRY   (A) = PROMPT CHARACTER
                                4843  *       EXIT    LINE READ
                                4844  *               'C' SET IF CTL-C ENTERED. NO TEXT.
                                4845  *               'C' CLEAR IF HAVE LINE
                                4846  *                'Z' SET IF NO ERROR IN LINE
                                4847  *       USES    ALL
                                4848
                                4849
065.364                         4850  ICL     EQU     *
065.364  041 330 112           4851          LXI     H,LINE+1
065.367  315 075 077           4852          CALL    RIL             READ INPUT LINE
065.372  330                   4853          RC                      CTL-C
                                4854
                                4855  **      ICL. - ENTRY FOR PRE-READ LINE
                                4856  *
                                4857  *               (HL) = LINE FWA (BUFFER ADDRESS  +1)
                                4858
065.373  053                   4859  ICL.    DCX     H               PRE-DECREMENT H
065.374  345                   4860          PUSH    H               SAVE BUFFER FWA
065.375  104                   4861          MOV     B,H
065.376  115                   4862          MOV     C,L             (BC) = TO, (HL) = FROM
065.377  013                   4863          DCX     B               PREDECREMENT (BC)
                                4864
                                4865  *       COPY ANOTHER CHARACTER
                                4866
066.000  003                   4867  ICL1    INX     B
066.001  043                   4868  ICL1.5  INX     H
066.002  176                   4869          MOV     A,M
066.003  002                   4870          STAX    B               COPY CHARACTER
066.004  127                   4871          MOV     D,A             (D) = 0 IFF END OF LINE
066.005  247                   4872          ANA     A
066.006  312 234 066           4873          JZ      ICL10           ALL DONE
066.011  315 107 112           4874          CALL    $MCU            MAP CHARACTER TO UPPER CASE
066.014  376 042               4875          CPI     '"'
066.016  312 176 066           4876          JE      ICL7            GOT QUOTES
066.021  376 101               4877          CPI     'A'
066.023  332 000 066           4878          JC      ICL1            NOT ALPHA
066.026  376 133               4879          CPI     'Z'+1
066.030  322 000 066           4880          JNC     ICL1            NOT ALPHA
                                4881
```

```
                          4882  *       HAVE AN ALPHA CHARACTER. SEE IF WE HAVE 2 IN A ROW
                          4883
066.033  043            4884          INX     H
066.034  176            4885          MOV     A,M
066.035  053            4886          DCX     H
066.036  315 107 112     4887          CALL    $MCU            MAP CHARACTER TO UPPER CASE
066.041  376 101         4888          CPI     'A'
066.043  332 000 066     4889          JC      ICL1            NOT TWO ALPHA
066.046  376 133         4890          CPI     'Z'+1
066.050  322 000 066     4891          JNC     ICL1            NOT TWO ALPHA
                          4892
                          4893  *       HAVE TWO ALPHA IN A ROW. MUST BE A KEYWORD. FIND IT IN LIST
                          4894
066.053  021 240 066     4895          LXI     D,KEYTAB
066.056  345            4896  ICL2    PUSH    H               SAVE *FROM* ADDRESS
066.057  032            4897          LDAX    D
066.060  002            4898          STAX    B               ASSUME IS THIS KEYWORD
066.061  023            4899          INX     D
066.062  032            4900  ICL3    LDAX    D               COMPARE LINE AGAINST TABLE ENTRY
066.063  247            4901          ANA     A
066.064  372 122 066     4902          JM      ICL5            GOT MATCH
066.067  353            4903          XCHG                    (DE) = LINE, (HL) = KEYTAB ADDR
066.070  032            4904          LDAX    D               (A) = CHARACTER
066.071  315 107 112     4905          CALL    $MCU            MAP TO UPPER
066.074  276            4906          CMP     M
066.075  353            4907          XCHG                    RESTORE (DE) AND (HL)
066.076  023            4908          INX     D
066.077  043            4909          INX     H
066.100  312 062 066     4910          JE      ICL3            STILL MATCHING
066.103  033            4911          DCX     D               PRE-DECREMENT KEYTAB POINTER
                          4912
                          4913  *       NOT THIS KEYWORD. SCAN TO NEXT ONE AND RETRY
                          4914
066.104  023            4915  ICL4    INX     D
066.105  032            4916          LDAX    D
066.106  247            4917          ANA     A
066.107  362 104 066     4918          JP      ICL4            NOT AT START OF NEXT
066.112  341            4919          POP     H               (HL) = FWA OF UNKNOWN KEYWORD
066.113  074            4920          INR     A               SEE IF AT END OF LIST
066.114  302 056 066     4921          JNZ     ICL2            NOT AT END OF LIST
066.117  303 216 066     4922          JMP     ICL8            INVALID KEYWORD
                          4923
                          4924  *       HAVE FOUND THE KEYWORD. SEE IF '(' FOLLOWS
                          4925  *
                          4926  *       (HL) POINTS JUST PAST THE KEYWORD ON THE LINE
                          4927
066.122  321            4928  ICL5    POP     D               DISCARD KEYWORD FWA
066.123  012            4929          LDAX    B               (A) = KEYWORD VALUE
066.124  003            4930          INX     B
066.125  376 320         4931          CPI     CT.FCN
066.127  322 163 066     4932          JNC     ICL6            IS FUNCTION
066.132  365            4933          PUSH    PSW             SAVE CODE
066.133  176            4934          MOV     A,M
066.134  376 040         4935          CPI     ' '
066.136  302 142 066     4936          JNE     ICL5.5          NO BLANK FOLLOWING
066.141  043            4937          INX     H
```

```
066.142  361          4938  ICL5.5  POP     PSW             (A) = KEYWORD CODE
066.143  026 000      4939          MVI     D,0             NO ERROR WHEN REACH END OF LINE
066.145  376 242      4940          CPI     CT.REM
066.147  312 223 066  4941          JE      ICL9            COPY REST OF LINE
066.152  376 251      4942          CPI     CT.DAT
066.154  312 223 066  4943          JE      ICL9            COPY REST OF LINE
066.157  053          4944          DCX     H               PRESET (HL) FOR INCREMENT
066.160  303 001 066  4945          JMP     ICL1.5
                      4946
                      4947  *       IS FUNCTION. REQUIRE '('
                      4948
066.163  315 372 111  4949  ICL6    CALL    $SOB            SKIP OVER BLANKS
066.166  376 050      4950          CPI     '('
066.170  312 001 066  4951          JE      ICL1.5          OK, GOBBLE '('
066.173  303 216 066  4952          JMP     ICL8            ERROR
                      4953
                      4954  *       GOT QUOTE. SCAN TO CLOSE QUOTE
                      4955
066.176  003          4956  ICL7    INX     B
066.177  043          4957          INX     H
066.200  176          4958          MOV     A,M
066.201  002          4959          STAX    B               STORE CHARACTER
066.202  247          4960          ANA     A
066.203  312 216 066  4961          JZ      ICL8            ERROR
066.206  376 042      4962          CPI     '"'
066.210  302 176 066  4963          JNE     ICL7            NOT CLOSE QUOTE
066.213  303 000 066  4964          JMP     ICL1            GOT CLOSE
                      4965
                      4966  *       ERROR IN LINE. FLAG IT, AND COPY THE REST VERBATIM
                      4967  *       (A) = 0
                      4968
066.216  076 212      4969  ICL8    MVI     A,CT.SYE
066.220  002          4970          STAX    B               SET ERROR
066.221  003          4971          INX     B
066.222  127          4972          MOV     D,A             (D) <> 0 INDICATING ERROR
                      4973
                      4974  *       COPY REST OF LINE VERBATIM
                      4975  *       (D) <> 0 IF ERROR
                      4976
066.223  176          4977  ICL9    MOV     A,M
066.224  002          4978          STAX    B
066.225  043          4979          INX     H
066.226  003          4980          INX     B
066.227  247          4981          ANA     A
066.230  302 223 066  4982          JNZ     ICL9            NOT DONE
066.233  013          4983          DCX     B
                      4984
                      4985  *       ALL DONE.
                      4986  *
                      4987  *       (BC) = LINE LWA
                      4988  *       (D) <> 0 IFF ERROR
                      4989
066.234  341          4990  ICL10   POP     H               (HL) = FWA
066.235  172          4991          MOV     A,D             (A) = ERROR FLAG
066.236  247          4992          ANA     A
066.237  311          4993          RET
```

```
                                4995  **      KEYTAB - KEYWORD TABLE.
                                4996  *
                                4997
066.240                         4998  KEYTAB  EQU     *
066.240  320 101 102           4999          DB      CT.ABS,'ABS'
066.244  310 101 116           5000          DB      CT.AND,'AND'
066.250  350 101 123           5001          DB      CT.ASC,'ASC'
066.254  311 101 123           5002          DB      CT.AS,'AS'
066.257  321 101 124           5003          DB      CT.ATN,'ATN'
066.263  200 102 125           5004          DB      CT.BLD,'BUILD'
066.271  201 102 131           5005          DB      CT.BYE,'BYE'
066.275  213 103 110           5006          DB      CT.CHA,'CHAIN'
066.303  322 103 110           5007          DB      CT.CHR,'CHR$'
066.310  323 103 111           5008          DB      CT.CIN,'CIN'
066.314  214 103 114           5009          DB      CT.CLR,'CLEAR'
066.322  215 103 114           5010          DB      CT.CLO,'CLOSE'
066.330  216 103 116           5011          DB      CT.CTL,'CNTRL'
066.336  202 103 117           5012          DB      CT.CNT,'CONTINUE'
066.347  324 103 117           5013          DB      CT.COS,'COS'
066.353  251 104 101           5014          DB      CT.DAT,'DATA'
066.360  252 104 105           5015          DB      CT.DEF,'DEF'
066.364  203 104 105           5016          DB      CT.DEL,'DELETE'
066.373  217 104 111           5017          DB      CT.DIM,'DIM'
066.377  253 105 116           5018          DB      CT.END,'END'
067.003  325 105 130           5019          DB      CT.EXP,'EXP'
067.007  312 106 111           5020          DB      CT.FIL,'FILE'
067.014  220 106 116           5021          DB      CT.FN,'FN'
067.017  221 106 117           5022          DB      CT.FOR,'FOR'
067.023  223 106 122           5023          DB      CT.FRZ,'FREEZE'         MUST APPEAR BEFORE 'FREE'
067.032  222 106 122           5024          DB      CT.FRE,'FREE'
067.037  224 107 117           5025          DB      CT.GOS,'GOSUB'
067.045  225 107 117           5026          DB      CT.GOT,'GOTO'
067.052  226 111 106           5027          DB      CT.IF,'IF'
067.055  254 111 116           5028          DB      CT.INP,'INPUT'
067.063  326 111 116           5029          DB      CT.INT,'INT'
067.067  352 114 105           5030          DB      CT.LEN,'LEN'
067.073  351 114 105           5031          DB      CT.LEF,'LEFT$'
067.101  227 114 105           5032          DB      CT.LET,'LET'
067.105  250 114 111           5033          DB      CT.LIN,'LINE'
067.112  204 114 111           5034          DB      CT.LIS,'LIST'
067.117  230 114 117           5035          DB      CT.LCK,'LOCK'
067.124  327 114 116           5036          DB      CT.LNO,'LNO'
067.130  330 114 117           5037          DB      CT.LOG,'LOG'
067.134  353 115 101           5038          DB      CT.MAT,'MATCH'
067.142  331 115 101           5039          DB      CT.MAX,'MAX'
067.146  354 115 111           5040          DB      CT.MID,'MID$'
067.153  332 115 111           5041          DB      CT.MIN,'MIN'
067.157  231 116 105           5042          DB      CT.NXT,'NEXT'
067.164  314 116 117           5043          DB      CT.NOT,'NOT'
067.170  232 117 114           5044          DB      CT.OLD,'OLD'
067.174  233 117 116           5045          DB      CT.ON,'ON'
067.177  234 117 120           5046          DB      CT.OPE,'OPEN'
067.204  315 117 122           5047          DB      CT.OR,'OR'
067.207  235 117 125           5048          DB      CT.OUT,'OUT'
067.213  333 120 101           5049          DB      CT.PAD,'PAD'
067.217  236 120 101           5050          DB      CT.PAU,'PAUSE'
```

```
067.225  334 120 105  5051          DB      CT.PEK,'PEEK'
067.232  335 120 111  5052          DB      CT.PIN,'PIN'
067.236  237 120 117  5053          DB      CT.POK,'POKE'
067.243  336 120 117  5054          DB      CT.POS,'POS'
067.247  240 120 122  5055          DB      CT.PRT,'PRINT'
067.255  241 122 105  5056          DB      CT.REA,'READ'
067.262  242 122 105  5057          DB      CT.REM,'REM'
067.266  205 122 105  5058          DB      CT.REP,'REPLACE'
067.276  243 122 105  5059          DB      CT.RES,'RESTORE'
067.306  244 122 105  5060          DB      CT.RET,'RETURN'
067.315  355 122 111  5061          DB      CT.RIG,'RIGHT$'
067.324  337 122 116  5062          DB      CT.RND,'RND'
067.330  206 122 125  5063          DB      CT.RUN,'RUN'
067.334  207 123 101  5064          DB      CT.SAV,'SAVE'
067.341  210 123 103  5065          DB      CT.SCR,'SCRATCH'
067.351  340 123 105  5066          DB      CT.SEG,'SEG'
067.355  341 123 107  5067          DB      CT.SGN,'SGN'
067.361  342 123 111  5068          DB      CT.SIN,'SIN'
067.365  343 123 120  5069          DB      CT.SPC,'SPC'
067.371  344 123 121  5070          DB      CT.SQR,'SQR'
067.375  345 123 124  5071          DB      CT.STR,'STR$'
070.002  211 123 124  5072          DB      CT.STE,'STEP'
070.007  255 123 124  5073          DB      CT.STP,'STOP'
070.014  346 124 101  5074          DB      CT.TAB,'TAB'
070.020  347 124 101  5075          DB      CT.TAN,'TAN'
070.024  316 124 110  5076          DB      CT.THN,'THEN'
070.031  317 124 117  5077          DB      CT.TO,'TO'
070.034  245 125 116  5078          DB      CT.UNF,'UNFREEZE'
070.045  246 125 116  5079          DB      CT.UNL,'UNLOCK'
070.054  247 125 116  5080          DB      CT.UNS,'UNSAVE'
070.063  356 126 101  5081          DB      CT.VAL,'VAL'
070.067  313 127 122  5082          DB      CT.WRI,'WRITE'
070.075  212 007 052  5083          DB      CT.SYE,BELL,'*ERR*'     HERE FOR LISTING VIA *EKA*, CANNOT BE MATCHED
070.105  377          5084          DB      377Q                    377Q = END OF TABLE
```

```
                        5088  **      ERROR PROCESSING.
                        5089  *
                        5090  *       THESE ERROR PROCESSORS ARE ENTERED WHEN AN ERROR IS DETECTED.
                        5091  *
                        5092  *       SINCE ALL TRACK OF CONTROL HAS BEEN LOST, EXECUTING CANNOT
                        5093  *       BE RESUMED.
                        5094  *
                        5095  *       THE USER MAY DISPLAY VARIABLES WHEN AN ERROR OCCURS, BUT MAY
                        5096  *       NOT 'CONTINUE'.
                        5097
                        5098
                        5099
                        5100
                        5101
070.106  076 200        5102  ERR.CC  MVI     A,BEC.CC                CONTROL-C
070.110  001              5103          DB      MI.LXIB
                        5104
070.111  076 201        5105  ERR.CB  MVI     A,BEC.CB                CTL-B
070.113  001              5106          DB      MI.LXIB
                        5107
070.114  076 202        5108  ERR.DE  MVI     A,BEC.DE                DATA EXHAUSTED
070.116  001              5109          DB      MI.LXIB
                        5110
070.117  076 203        5111  ERR.DO  MVI     A,BEC.DO                /0
070.121  001              5112          DB      MI.LXIB
                        5113
070.122  076 204        5114  ERR.IN  MVI     A,BEC.IN                ILLEGAL NUMBER
070.124  001              5115          DB      MI.LXIB
                        5116
070.125  076 205        5117  ERR.IU  MVI     A,BEC.IU                ILLEGAL USAGE
070.127  001              5118          DB      MI.LXIB
                        5119
070.130  076 206        5120  ERR.LK  MVI     A,BEC.LK                DATA LOCK ENGAGED
070.132  001              5121          DB      MI.LXIB
                        5122
070.133  076 207        5123  ERR.NV  MVI     A,BEC.NV                NEXT VARIABLE MISSING
070.135  001              5124          DB      MI.LXIB
                        5125
070.136  076 210        5126  ERR.OV  MVI     A,BEC.OV                OVERFLOW
070.140  001              5127          DB      MI.LXIB
                        5128
070.141  076 211        5129  ERR.RE  MVI     A,BEC.RE                RETURN ERROR
070.143  001              5130          DB      MI.LXIB
                        5131
070.144  076 212        5132  ERR.SL  MVI     A,BEC.SL                STRING LENGTH
070.146  001              5133          DB      MI.LXIB
                        5134
070.147  076 213        5135  ERR.SN  MVI     A,BEC.SN                STATEMENT NUMBER
070.151  001              5136          DB      MI.LXIB
                        5137
070.152  076 214        5138  ERR.SY  MVI     A,BEC.SY                SYNTAX ERROR
070.154  001              5139          DB      MI.LXIB
                        5140
070.155  076 215        5141  ERR.TC  MVI     A,BEC.TC                TYPE CONFLICH
070.157  001              5142          DB      MI.LXIB
                        5143
```

```
070.160  076 216      5144  ERR.TO  MVI   A,BEC.TO              TABLE OVERFLOW
070.162  001          5145          DB    MI.LXIB
                      5146
070.163  076 217      5147  ERR.SR  MVI   A,BEC.SR              SUBSCRIPT RANGE
070.165  001          5148          DB    MI.LXIB
                      5149
070.166  076 220      5150  ERR.SC  MVI   A,BEC.SC              SUBSCRIPT COUNT
070.170  001          5151          DB    MI.LXIB
                      5152
070.171  076 221      5153  ERR.ND  MVI   A,BEC.ND              NOT DIMENSIONED
070.173  001          5154          DB    MI.LXIB
                      5155
070.174  076 222      5156  ERR.IC  MVI   A,BEC.IC              ILLEGAL CHARACTER
070.176  001          5157          DB    MI.LXIB
                      5158
070.177  076 226      5159  ERR.FAE MVI   A,BEC.FAE     FILE ALREADY EXISTS
070.201  001          5160          DB    MI.LXIB
                      5161
070.202  076 227      5162  ERR.ILF MVI   A,BEC.ILF     ILLEGAL FILE NAME
070.204  001          5163          DB    MI.LXIB
                      5164
070.205  076 230      5165  ERR.AC  MVI   A,BEC.AC
070.207  001          5166          DB    MI.LXIB              ARG COUNT
                      5167
070.210  076 231      5168  ERR.FNO MVI   A,BEC.FNO             FILE NOT OPEN
070.212  001          5169          DB    MI.LXIB
                      5170
070.213  076 001      5171  ERR.EOF MVI   A,EC.EOF              END OF FILE
070.215  001          5172          DB    MI.LXIB
                      5173
070.216  076 223      5174  ERR.UD  MVI   A,BEC.UD              UNDEFINED FUNCTION
070.220  001          5175          DB    MI.LXIB                                  /80.01.GC/
                      5176
070.221  076 233      5177  ERR.CIU MVI   A,BEC.CIU             CHANNEL IN USE   /80.01.GC/
                      5178
070.223              5179  SERROR  EQU   *
070.223              5180  $FERROR EQU   *
                      5181
070.223  365          5182          PUSH  PSW                   SAVE ERROR CODE
070.224  041 123 112  5183          LXI   H,MTABIND+MT.LEN       (HL) = #LENGTH OF TXTTAB
070.227  136          5184          MOV   E,M
070.230  043          5185          INX   H
070.231  176          5186          MOV   A,M                   (AE) = LENGTH OF TABLE
070.232  247          5187          ANA   A
070.233  302 244 070  5188          JNZ   ERROR1                TABLE LENGTH > 3
070.236  173          5189          MOV   A,E
070.237  376 004      5190          CPI   4
070.241  334 320 077  5191          CC    SCRA                  TABLE LENGTH < 3
                      5192
070.244              5193  ERROR1  EQU   *
                      5194  *       CALL  FOP                   MAKE OVL RESIDENT
                      5195  *       CALL  CLF                   CLEAR FILE OPERATIONS
070.244  377 007      5196          DB    SYSCALL,.CLRCO        CLEAR CONSOLE
070.246  315 136 031  5197          CALL  $TYPTX
070.251  012 007 041  5198          DB    NL,BELL,'! ERROR -',' '+200Q
                      5199
```

```
                          5200  *       TYPE MESSAGE
                          5201
070.265 303 063 075       5202          JMP     ILM             ISSUE LINE MESSAGE
```

```
                                5205  **      MTL - MANAGE TEXT LINE.
                                5206  *
                                5207  *       MTL IS CALLED TI INSERT/REPLACE/DELETE A TEXT LINE FROM
                                5208  *       THE TEXT BUFFER.
                                5209  *
                                5210  *       ENTRY   *LINE* = TEXT LINE
                                5211  *       EXIT    LINE INSERTED/DELETED/REPLACED
                                5212  *               'CI' FLAG SET
                                5213  *       USES    ALL
                                5214
                                5215
        070.270                 5216  MTL     EQU     *
        070.270  315 313 075    5217          CALL    LFC             CHECK FOR DATA LOCK
        070.273  041 327 112    5218          LXI     H,LINE          CRACK NUMBER FROM LINE
        070.276  315 233 111    5219          CALL    DDN             DECODE DECIMAL NUMBER
        070.301  315 206 072    5220          CALL    CLN             CHECK FOR LEGAL NUMBER
                                5221
                                5222  *       DELETE LEADING BLANKS.
                                5223
        070.304  053            5224  MTL0    DCX     H
        070.305  076 040        5225          MVI     A,' '
        070.307  043            5226  MTL1    INX     H               SKIP LEADING BLANKS
        070.310  276            5227          CMP     M
        070.311  312 307 070    5228          JE      MTL1            STILL BLANKS
        070.314  315 335 111    5229          CALL    $CLL            COMPUTE LINE LENGTH
        070.317  075            5230          DCR     A               REMOVE END COUNT
        070.320  312 325 070    5231          JZ      MTL1.5          AM TO DELETE
        070.323  306 003        5232          ADI     3               LINE NUMBER + END-OF-LINE
        070.325  117            5233  MTL1.5  MOV     C,A             (C) = NEW LENGTH
        070.326  053            5234          DCX     H
        070.327  162            5235          MOV     M,D
        070.330  053            5236          DCX     H
        070.331  163            5237          MOV     M,E
        070.332  345            5238          PUSH    H               SAVE 'FROM' ADDRESS
        070.333  052 121 112    5239          LHLD    TXTTAB+MT.FWA
        070.336  345            5240          PUSH    H
        070.337  315 242 074    5241          CALL    FLN             FIND LINE BY NUMBER
        070.342  006 000        5242          MVI     B,0             (B) = OLD LENGTH
        070.344  332 361 070    5243          JC      MTL2            IS INSERT
        070.347  043            5244          INX     H
        070.350  043            5245          INX     H
        070.351  315 335 111    5246          CALL    $CLL
        070.354  306 002        5247          ADI     2
        070.356  107            5248          MOV     B,A             (B) = OLD LENGTH
        070.357  053            5249          DCX     H
        070.360  053            5250          DCX     H               (HL) = ADDRESS TO INSERT
        070.361  321            5251  MTL2    POP     D               (DE) = TABLE FWA
        070.362  175            5252          MOV     A,L
        070.363  223            5253          SUB     E
        070.364  157            5254          MOV     L,A
        070.365  174            5255          MOV     A,H
        070.366  232            5256          SBB     D
        070.367  147            5257          MOV     H,A             (HL) = INDEX
        070.370  171            5258          MOV     A,C             (A) = NEW LENGTH
        070.371  220            5259          SUB     B               (A) = NEW LENGTH - OLD
        070.372  137            5260          MOV     E,A
```

```
070.373  237           5261          SBB     A
070.374  127           5262          MOV     D,A             (DE) = NEEDED BYTES COUNT
070.375  315 213 104   5263          CALL    $IBT            MAKE OR DESTROY ROOM
071.000  121 112       5264          DW      TXTTAB+1        TABLE POINTER
071.002  353           5265          XCHG
071.003  052 121 112   5266          LHLD    TXTTAB+MT.FWA
071.006  031           5267          DAD     D               (HL) = *TO* ADDRESS
071.007  321           5268          POP     D               (DE) = *FROM* ADDRSS
071.010  006 000       5269          MVI     B,0             (BC) = NEW LENGTH
071.012  303 252 030   5270          JMP     $MOVE           COPY TEXT INTO BUFFER AND RETURN


                       5272  **      MOV - MOVE A BLOCK OF DATA.
                       5273  *
                       5274  *       MOV MOVES A BLOCK OF DATA IN MEMORY.
                       5275  *
                       5276  *       ENTRY   (DE) = FROM
                       5277  *               (HL) = TO
                       5278  *               (A) = COUNT
                       5279  *       EXIT    MOVED
                       5280  *               (DE) = FROM + COUNT
                       5281  *               (HL) = TO + COUNT
                       5282  *       USES    A,F
                       5283
                       5284
071.015  305           5285  MOV     PUSH    B
071.016  117           5286          MOV     C,A
071.017  006 000       5287          MVI     B,0
071.021  315 252 030   5288          CALL    $MOVE
071.024  301           5289          POP     B
071.025  311           5290          RET
```

```
                                5294  **      AMB - ALLOCATE MEMORY BYTES.
                                5295  *
                                5296  *       AMB ALLOCATES A BLOCK OF MEMORY TO THE END OF A TABLE.
                                5297  *       AND RETURNS THE FWA OF THE BLOCK.
                                5298  *
                                5299  *       ENTRY   (DE) = TABLE ADDRESS+1
                                5300  *               (HL) = BYTES WANTED
                                5301  *       EXIT    (DE) = TABLE ADDRESS+1
                                5302  *               (HL) = FWA (ABS) OF BLOCK
                                5303  *       USES    A,F,H,L
                                5304
                                5305
071.026                         5306  AMB     EQU     *
071.026  345                    5307          PUSH    H               SAVE COUNT
071.027  325                    5308          PUSH    D               SAVE TABLE ADDRESS
071.030  023                    5309          INX     D
071.031  023                    5310          INX     D
071.032  032                    5311          LDAX    D
071.033  157                    5312          MOV     L,A             (HL) = TABLE LENGTH
071.034  023                    5313          INX     D
071.035  032                    5314          LDAX    D
071.036  321                    5315          POP     D
071.037  147                    5316          MOV     H,A
071.040  343                    5317          XTHL                    (HL) = COUNT
071.041  315 244 103            5318          CALL    $ATS            ALLOCATE SPACE
071.044  341                    5319          POP     H               (HL) = ORIGINAL LENGTH
071.045  032                    5320          LDAX    D
071.046  205                    5321          ADD     L
071.047  157                    5322          MOV     L,A
071.050  023                    5323          INX     D
071.051  032                    5324          LDAX    D
071.052  214                    5325          ADC     H
071.053  147                    5326          MOV     H,A             (HL) = FWA OF BLOCK
071.054  033                    5327          DCX     D
071.055  311                    5328          RET


                                5330  **      ANT - ACCEPT NEXT TOKEN.
                                5331  *
                                5332  *       ANT ACCEPTS THE NEXT TEXT TOKEN.
                                5333  *
                                5334  *       ENTRY   (BC) = TEXT POINTER
                                5335  *       EXIT    (A) = TYPE
                                5336  *               (DE) = INDEX (IF VARIABLE)
                                5337  *       USES    A,F, (D,E IF VARIABLE)
                                5338
                                5339
071.056  315 072 076            5340  ANT     CALL    PNT             PEEK AT NEXT TOKEN
071.061  365                    5341          PUSH    PSW             SAVE TYPE
000.000                         5342          ERRNZ   MI.NOP
071.062  257                    5343          XRA     A
071.063  062 073 076            5344          STA     PNTA            CLEAR TYPE
071.066  303 130 076            5345          JMP     PNT1            CLEAR 'TOKEN ALREADY READ' FLAG
```

```
                                5347  **      ATP - ADJUST TABLE POINTERS.
                                5348  *
                                5349  *       $ATP IS CALLED BY THE MANAGED TABLE PACKAGE WHENEVER THE TABLES
                                5350  *       HAVE BEEN SHUFFLED. $ATP IS TO ADJUST ANY ABS POINTERS THAT MAY
                                5351  *       EXIST.
                                5352  *
                                5353  *       THE ONLY ABS POINTES ARE THE ONES IN THE FILE BUFFERS IN
                                5354  *       THE FILTAB TABLE.
                                5355  *
                                5356  *       SINCE THE FILE BUFFER IMMEDIATELY FOLLOWS THE FILE BLOCK, THE
                                5357  *       DISPLACEMENT FOR THE TABLE CAN BE COMPUTED  BY SUBTRACTING THE
                                5358  *       OLD BUFFER FWA (IN FB.FWA) FROM THE NEW ONE (FILTAB ENTRY + FB.NAM +
                                5359  *       FB.NAML)
                                5360  *
                                5361  *       NOTE THAT THE LAST BUFFER IN THE TABLE MAY NOT HAVE IT'S POINTERS SETUP
                                5362  *       CORRECTLY, IN WHICH CASE THE GARBAGE THERE JUST GETS STIRED UP A LITTLE.
                                5363  *
                                5364  *       ENTRY   NONE
                                5365  *       EXIT    NONE
                                5366  *       USES    ALL
                                5367
                                5368
071.071  052 226 042           5369  $ATP    LHLD    FBUFAD          (HL) = OLD FILTAB MT.FWA
071.074  353                   5370          XCHG
071.075  052 164 112           5371          LHLD    FILTAB+MT.FWA   (HL) = NEW FILTAB MT.FWA
071.100  042 226 042           5372          SHLD    FBUFAD          SAVE FOR NEXT TIME
071.103  175                   5373          MOV     A,L
071.104  223                   5374          SUB     E
071.105  137                   5375          MOV     E,A             (DE) = TABLE DISPLACEMENT
071.106  174                   5376          MOV     A,H
071.107  232                   5377          SBB     D
071.110  127                   5378          MOV     D,A
071.111  006 005               5379          MVI     B,CHANMAX       (B) = TABLES TO ADJUST
071.113  041 265 042           5380          LXI     H,FBLIST+FBENL+FB.FWA   START AT FIRST USER BLOCK
071.116  016 004               5381  ATP1    MVI     C,4             4 ADDRESSES IN EACH BLOCK
071.120  176                   5382  ATP2    MOV     A,M             RELOCATE ADDRESS
071.121  203                   5383          ADD     E
071.122  167                   5384          MOV     M,A
071.123  043                   5385          INX     H
071.124  176                   5386          MOV     A,M
071.125  212                   5387          ADC     D
071.126  167                   5388          MOV     M,A
071.127  043                   5389          INX     H
071.130  015                   5390          DCR     C
071.131  302 120 071           5391          JNZ     ATP2            RELOCATE ALL 4 ADDRESSES
071.134  076 023               5392          MVI     A,FBENL-8
071.136  315 101 030           5393          CALL    $DADA.          POINT TO NEXT BLOCK
071.141  005                   5394          DCR     B
071.142  302 116 071           5395          JNZ     ATP1            RELOCATE ALL BLOCKS
071.145  311                   5396          RET                     EXIT
```

```
                                5398  **      AYS - ASK 'ARE YOU SURE?'
                                5399  *
                                5400  *       AYS ASKS THE USER IF HE IS SURE. A LINE LINE ANSWER IS
                                5401  *       RECEIVED, AND ITS FIRST CHARACTER IS CHECKED.
                                5402  *
                                5403  *       ENTRY   NONE
                                5404  *       EXIT    'Z' SET IF REPLY STARTED WITH 'Y'
                                5405  *               (BC) = $ZERO
                                5406  *       USES    ALL
                                5407
                                5408
071.146  315 136 031            5409  AYS     CALL    $TYPTX
071.151  007 123 165            5410          DB      BELL,'Sure','?'+200Q
071.157  041 327 112            5411          LXI     H,LINE
071.162  315 075 077            5412          CALL    RIL
071.165  332 106 070            5413          JC      ERR.CC          CTL-C STRUCK
071.170  176                    5414          MOV     A,M             (A) = REPLY
071.171  315 107 112            5415          CALL    $MCU
071.174  376 131                5416          CPI     'Y'
071.176  001 007 115            5417          LXI     B,ZERO          POINT TO END OF LINE
071.201  311                    5418          RET


                                5420  **      AVV - ASSIGN VALUE TO VARIABLE.
                                5421  *
                                5422  *       AVV ASSIGNS THE VALUE IN (ACCX) TO A VARIABLE POINTED TO
                                5423  *       BY      (DE).
                                5424  *
                                5425  *       IF THE TYPES DO NOT MATCH, FLAG AN ERROR.
                                5426  *
                                5427  *
                                5428  *       ENTRY   (ACCX) = VALUE
                                5429  *               (A) = TARGET TYPE
                                5430  *               (DE) = TARGET POINTER
                                5431  *       EXIT    TO 'RET' IF OK
                                5432  *               TO ERR.TC IF MISMATCH.
                                5433  *       USES    A,F,D,E
                                5434
                                5435
071.202  345                    5436  AVV     PUSH    H               SAVE (HL)
071.203  147                    5437          MOV     H,A
                                5438
                                5439  *       DETERMINE ABSOLUTE ADDRESS OF TARGET.
                                5440
071.204  072 201 042            5441          LDA     ACCX-1
071.207  057                    5442          CMA
071.210  254                    5443          XRA     H
071.211  346 001                5444          ANI     CF.STR
071.213  312 155 070            5445          JZ      ERR.TC          MISMATCH
071.216  244                    5446          ANA     H
071.217  312 240 073            5447          JZ      CXV.
                                5448
                                5449  *       HAVE STRING
                                5450  *       (DE) = ADDRESS OF BLOCK
```

```
                                5451
071.222  315 000 073  5452  AVV1    CALL    CSI             (DE) = INDEX INTO SYMBOL
071.225  325          5453          PUSH    D               SAVE 'TO' DESCRIPTOR ADDRESS
071.226  315 366 072  5454          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL
071.231  315 315 074  5455          CALL    FSE             (HL) = TO ABS, (A) = TO LEN
071.234  127          5456          MOV     D,A
071.235  072 202 042  5457          LDA     ACCX
071.240  222          5458          SUB     D               (A) = NEWLEN-OLDLEN
071.241  137          5459          MOV     E,A
071.242  237          5460          SBB     A
071.243  127          5461          MOV     D,A             (DE) = COUNT CHANGE
071.244  325          5462          PUSH    D               SAVE TABLE DELTA
071.245  353          5463          XCHG                    (DE) = 'TO' ABS ADDRESS
071.246  052 152 112  5464          LHLD    STRTAB+MT.FWA
071.251  173          5465          MOV     A,E             COMPUTE INDEX OF 'TO'
071.252  225          5466          SUB     L
071.253  157          5467          MOV     L,A
071.254  172          5468          MOV     A,D
071.255  234          5469          SBB     H
071.256  147          5470          MOV     H,A
071.257  321          5471          POP     D
071.260  172          5472          MOV     A,D
071.261  027          5473          RAL                     MOVE SIGN BIT INTO CARRY
071.262  315 213 104  5474          CALL    $IBT
071.265  152 112      5475          DW      STRTAB+1
071.267  325          5476          PUSH    D               SAVE COUNT
071.270  353          5477          XCHG
071.271  052 152 112  5478          LHLD    STRTAB+MT.FWA
071.274  031          5479          DAD     D               (HL ) ABS ADDRESS OF ADDITION
071.275  321          5480          POP     D               (DE) = COUNT
071.276  172          5481          MOV     A,D
071.277  247          5482          ANA     A
071.300  364 351 100  5483          CP      ZRO             CLEAR IF WAS ADDITION
071.303  321          5484          POP     D               (DE) = 'TO' DESCRIPTOR ADDRESS(INDEX)
071.304  315 366 072  5485          CALL    CSA             (DE) = 'TO' DESCRIPTOR ADDRESS(ABS.)
071.307  072 202 042  5486          LDA     ACCX
071.312  022          5487          STAX    D               SET NEW COUNT
071.313  315 315 074  5488          CALL    FSE
071.316  345          5489          PUSH    H
071.317  021 202 042  5490          LXI     D,ACCX
071.322  315 315 074  5491          CALL    FSE             FIND 'FROM'
071.325  353          5492          XCHG                    (DE) = 'FROM' ADDRESS
071.326  341          5493          POP     H               (HL) = 'TO' ADDRESS
071.327  315 015 071  5494          CALL    MOV             MOVE STRING
071.332  341          5495          POP     H               RESTORE (HL)
071.333  311          5496          RET
```

```
                                5498  **      CAS - CALCULATE ARRAY SIZE.
                                5499  *
                                5500  *       CAS COMPUTES THE NUMBER OF ENTRYS IN AN ARRAY.
                                5501  *
                                5502  *       ENTRY   (DE) = HEADER POINTER
                                5503  *       EXIT    (HL) = COUNT
                                5504  *       USES    A,F,D,E,H,L
                                5505
                                5506
071.334  305                    5507  CAS     PUSH    B               SAVE (BC)
071.335  032                    5508          LDAX    D               (A) = SUBSCRIPT COUNT
071.336  023                    5509          INX     D
071.337  023                    5510          INX     D
071.340  023                    5511          INX     D
071.341  023                    5512          INX     D
071.342  325                    5513          PUSH    D               SAVE SYMTAB ADDRESS
071.343  041 001 000            5514          LXI     H,1             (HL) = ACCUMULATOR
                                5515
071.346                         5516  CAS1    EQU     *
071.346  343                    5517          XTHL                    (HL) = ADDRESS
071.347  136                    5518          MOV     E,M
071.350  043                    5519          INX     H
071.351  126                    5520          MOV     D,M             (DE) = BOUND
071.352  043                    5521          INX     H
071.353  301                    5522          POP     B               (BC) = ACCUMULATOR
071.354  345                    5523          PUSH    H               SAVE ADDRESS
071.355  365                    5524          PUSH    PSW             SAVE COUNT
071.356  315 337 030            5525          CALL    $MU66           (HL) = ACCUMULATION
071.361  302 122 070            5526          JNZ     ERR.IN          OVERFLOW
071.364  361                    5527          POP     PSW
071.365  075                    5528          DCR     A               DECREMENT COUNT
071.366  302 346 071            5529          JNZ     CAS1            IF MORE
071.371  321                    5530          POP     D               DISCARD ADDRESS
071.372  301                    5531          POP     B               RESTORE BC
071.373  311                    5532          RET


                                5534  **      CEF - CREATE EMPTY FILE BUFFER.
                                5535  *
                                5536  *       CEF CREATES AN EMPTY FILE BUFFER
                                5537  *       ON THE END OF FILTAB.
                                5538  *
                                5539  *       ENTRY   NONE
                                5540  *       EXIT    NONE
                                5541  *       USES    ALL
                                5542
                                5543
071.374  021 164 112            5544  CEF     LXI     D,FILTAB+1
071.377  041 000 001            5545          LXI     H,256
072.002  303 244 103            5546          JMP     $ATS            ALLOCATE TABLE SPACE            /80.01.GC/
```

```
                                5548  **      CFA - COMPUTE FILE BLOCK ADDRESS.
                                5549  *
                                5550  *       CFA COMPUTES THE ABS ADDRESS OF A FILE BLOCK.
                                5551  *
                                5552  *       ENTRY   (A) = FILE BLOCK NUMBER (IOCHAN-1)
                                5553  *       EXIT    TO ERR.SY IF NUMBER TOO LARGE
                                5554  *                'C' CLEAR IF OK
                                5555  *                 (HL) = ABS ADDRESS OF FILE BLOCK
                                5556  *                'C' SET IF NOT THERE
                                5557  *       USES    A,F,D,E,H,L
                                5558
                                5559
072.005  376 007               5560  CFA     CPI     CHANMAX+2       +1 FOR TEST, +1 FOR SKEWED ENTRY
072.007  322 122 070           5561          JNC     ERR.IN          TOO LARGE
072.012  127                   5562          MOV     D,A             (D) = CHANNEL NUMBER
072.013  036 000               5563          MVI     E,0             (DE) = 256*CHANNEL
                                5564  *       ANA     A                                       /80.20.GC/
                                5565  *       JNZ     CFA1            IS USER CHANNEL         /80.02.GC/
                                5566
                                5567  *       IS SYSTEM BUFFER. SETUP WRITE-ACCESS TO PROTECTED H17 RAM
                                5568
                                5569  *       ERRNZ   M.SYSM                                  /80.02.GC/
                                5570  *       LHLD    S.DLINK                                 /80.02.GC/
                                5571  *       MVI     M,1             SET M.SYSM NON-ZERO     /80.02.GC/
                                5572  *       CALL    $WER            WRITE ENABLE RAM        /80.02.GC/
072.015  052 166 112           5573  CFA1    LHLD    FILTAB+MT.LEN
072.020  175                   5574          MOV     A,L             SEE IF WE HAVE THAT MANY
072.021  223                   5575          SUB     E
072.022  174                   5576          MOV     A,H
072.023  232                   5577          SBB     D
072.024  330                   5578          RC                      FILE BLOCK NOT IN TABLE
072.025  172                   5579          MOV     A,D             (A) = CHANNEL NUMBER
072.026  021 033 000           5580          LXI     D,FBENL
072.031  315 007 031           5581          CALL    $MU86
072.034  021 230 042           5582          LXI     D,FBLIST
072.037  031                   5583          DAD     D               (HL) = ABS ADDRESS OF BLOCK
072.040  311                   5584          RET


                                5586  **      CFN - CRACK FILE NAME.
                                5587  *
                                5588  *       CFN DECODES A STRING FROM THE TEXT LINE INTO THE FILE
                                5589  *       NAME AREA OF THE SYSTEM FILE BLOCK.
                                5590  *
                                5591  *       ENTRY   (BC) = LINE POINTER
                                5592  *       EXIT    (BC) ADVANCED
                                5593  *                (HL) = FWA OF FILE BLOCK
                                5594  *       USES    ALL
                                5595
                                5596
072.041  315 053 072           5597  CFN.    CALL    CFN             CFN WITH FOP
072.044  315 217 074           5598          CALL    FOP             FILE OPEN PRESET
072.047  041 230 042           5599          LXI     H,FBLIST
072.052  311                   5600          RET
```

```
                              5601
072.053  315 244 055         5602  CFN     CALL    EVAL                                    /78.10.GC/
072.056  033                 5603          DCX     D                                       /78.10.GC/
072.057  032                 5604          LDAX    D               (A) = TYPE              /78.10.GC/
072.060  023                 5605          INX     D                                       /78.10.GC/
072.061  346 001             5606          ANI     CF.STR                                  /78.10.GC/
072.063  312 152 070         5607          JZ      ERR.SY                                  /78.10.GC/
072.066  315 315 074         5608          CALL    FSE             FIND STRING TABLE ENTRY
072.071  247                 5609          ANA     A
072.072  312 202 070         5610          JZ      ERR.ILF         ILLEGAL FILE NAME
072.075  353                 5611          XCHG                    (DE) = STRING ADDRESS
072.076  376 021             5612          CPI     FB.NAML
072.100  322 202 070         5613          JNC     ERR.ILF         TOO LONG A NAME
072.103  305                 5614          PUSH    B               SAVE (BC)
072.104  041 242 042         5615          LXI     H,FBLIST+FB.NAM
072.107  117                 5616          MOV     C,A
072.110  006 000             5617          MVI     B,0             (BC) = LEN
072.112  315 252 030         5618          CALL    $MOVE           MOVE IN NAME
072.115  257                 5619          XRA     A
072.116  167                 5620          MOV     M,A             TERMINATE NAME
072.117  062 231 042         5621          STA     FBLIST+FB.FLG   CLEAR STATUS
072.122  041 230 042         5622          LXI     H,FBLIST
072.125  301                 5623          POP     B               RESTORE REGS
072.126  311                 5624          RET                     EXIT


                              5626  **      $CFS - CALCULATE FREE SPACE.
                              5627  *
                              5628  *       $CFS COUNTS THE FREE SPACE AVAILABLE TO MANAGED TABLES.
                              5629  *
                              5630  *       ENTRY   NONE
                              5631  *       EXIT    (HL) = COUNT
                              5632  *       USES    A,F,D,E,H,L
                              5633
                              5634
072.127  041 123 112         5635  $CFS    LXI     H,MTABIND+MT.LEN
072.132  345                 5636          PUSH    H               SAVE POINTER ON STACK
072.133  041 010 115         5637          LXI     H,MTAREA        (HL) = ACCUMULATOR
072.136  076 010             5638          MVI     A,MTABL         (A) = NUMBER OF TABLES
072.140  343                 5639  CFS1    XTHL                    (HL) = ADDRESS OF NEXT TABLE
072.141  136                 5640          MOV     E,M
072.142  043                 5641          INX     H
072.143  126                 5642          MOV     D,M
072.144  043                 5643          INX     H
072.145  043                 5644          INX     H
072.146  043                 5645          INX     H
072.147  043                 5646          INX     H
072.150  343                 5647          XTHL
072.151  031                 5648          DAD     D               (HL) = LENGTH
072.152  075                 5649          DCR     A
072.153  302 140 072         5650          JNZ     CFS1            MORE TABLES TO ADD
                              5651
                              5652  *       (HL) = TABLE BYTE COUNT + TABLE FWA
                              5653
```

```
072.156  321           5654          POP     D               (DE) = ADDRESS OF MEML+2
072.157  033           5655          DCX     D
072.160  033           5656          DCX     D
072.161  032           5657          LDAX    D
072.162  225           5658          SUB     L
072.163  157           5659          MOV     L,A
072.164  023           5660          INX     D
072.165  032           5661          LDAX    D
072.166  234           5662          SBB     H
072.167  147           5663          MOV     H,A
072.170  311           5664          RET


                       5666  **      CLF - CLEAR FILE OPERATIONS.
                       5667  *
                       5668  *       CLF IS CALLED TO CLEAR FILE JUNK.
                       5669
                       5670
072.171                5671  CLF     EQU     *
072.171  041 000 000   5672          LXI     H,0
072.174  042 166 112   5673          SHLD    FILTAB+MT.LEN   EMPTY ALL BUT ONE FILE
072.177  377 056       5674          DB      SYSCALL,.CLEARA CLEAR ALL CHANNELS (BUT OVERLAY CHANNEL)
072.201  257           5675          XRA     A
072.202  062 231 042   5676          STA     FBLIST+FB.FLG   CLEAR STATUS OF INTERNAL BUFFER
072.205  311           5677          RET


                       5679  **      CLN - CHECK FOR LEGAL NUMBER.
                       5680  *
                       5681  *       CLN EXAMINES A LINE NUMBER TO SEE IF IT OCCURS IN THE
                       5682  *       LEGAL RANGE.
                       5683  *
                       5684  *       ENTRY   (DE) = LINE NUMBER
                       5685  *       EXIT    TO *RET* IF BAD
                       5686  *               TO ERR.SR IF BAD
                       5687  *       USES    A,F
                       5688
                       5689
072.206  172           5690  CLN     MOV     A,D
072.207  263           5691          ORA     E
072.210  312 122 070   5692          JZ      ERR.IN          IS 0
072.213  023           5693          INX     D
072.214  172           5694          MOV     A,D
072.215  263           5695          ORA     E
072.216  033           5696          DCX     D
072.217  312 122 070   5697          JZ      ERR.IN          IS 377377A
072.222  311           5698          RET                     IS OK
```

```
                                5700  **      CMA - CHECK FOR COMMA.
                                5701  *
                                5702  *       CMA REQUIRES A COMMA IN THE TEXT STREAM.
                                5703  *
                                5704  *       ENTRY   NONE
                                5705  *       EXIT    (BC) ADVANCED
                                5706  *       USES    A,F,B,C,D,E
                                5707
                                5708
072.223  315 305 077            5709  CMA     CALL    RNT
072.226  026                    5710          DB      CT.CMA
072.227  311                    5711          RET



                                5713  **      CNC - CLASSIFY NEXT CHARACTER.
                                5714  *
                                5715  *       CNC CLASSIFYS THE NEXT TEXT CHARACTER.
                                5716  *
                                5717  *       ENTRY   (BC) = TEXT POINTER
                                5718  *       EXIT    (A) = 'CT.' CODE
                                5719  *       USES    A,F
                                5720
                                5721
072.230  012                    5722  CNC     LDAX    B               (A) = CODE
072.231  247                    5723          ANA     A
072.232  370                    5724          RM                      IS KEYWORD
000.000                         5725          ERRNZ   CT.FIN
072.233  310                    5726          RZ                      IS FIN
072.234  376 060                5727          CPI     '0'
072.236  332 265 072            5728          JC      CNC1            NOT NUMERIC OR ALPHA
072.241  376 072                5729          CPI     '9'+1
072.243  076 002                5730          MVI     A,CT.NUM
072.245  330                    5731          RC                      IS NUMERIC
072.246  012                    5732          LDAX    B
072.247  315 107 112            5733          CALL    $MCU            MAP CHARACTER TO UPPER CASE
072.252  376 101                5734          CPI     'A'
072.254  332 265 072            5735          JC      CNC1            NOT ALPHA
072.257  376 133                5736          CPI     'Z'+1
072.261  076 001                5737          MVI     A,CT.ALP
072.263  330                    5738          RC                      IS ALPHABETIC
072.264  012                    5739          LDAX    B
                                5740
                                5741  *       NOT ALPHABETIC OR NUMERIC. FIND IN TABLE.
                                5742
072.265  345                    5743  CNC1    PUSH    H
072.266  041 302 072            5744          LXI     H,CNCA
072.271  315 033 112            5745          CALL    $TBLS           SEARCH TABLE
072.274  176                    5746          MOV     A,M             (A) = INDEX
072.275  341                    5747          POP     H               RESOTRE (HL)
072.276  310                    5748          RZ                      FOUND
072.277  076 003                5749          MVI     A,CT.SEP        SEPERATOR
072.301  311                    5750          RET
                                5751
                                5752  **      TABLE OF SPECIAL TERMINATORS.
```

```
                                5753
072.302                         5754  CNCA    EQU     *
072.302  053 021               5755          DB      '+',CT.PL
072.304  055 022               5756          DB      '-',CT.MI
072.306  050 017               5757          DB      '(',CT.PAL
072.310  051 020               5758          DB      ')',CT.PAR
072.312  052 023               5759          DB      '*',CT.MU
072.314  057 024               5760          DB      '/',CT.DI
072.316  136 025               5761          DB      '^',CT.EX
072.320  072 000               5762          DB      ':',CT.FIN
072.322  056 002               5763          DB      '.',CT.NUM
072.324  054 026               5764          DB      ',',CT.CMA
072.326  074 014               5765          DB      '<',CT.LT
072.330  075 011               5766          DB      '=',CT.EQ
072.332  076 012               5767          DB      '>',CT.GT
072.334  073 027               5768          DB      ';',CT.SEM
072.336  042 030               5769          DB      '"',CT.QUO
072.340  133 017               5770          DB      '[',CT.PAL
072.342  135 020               5771          DB      ']',CT.PAR
072.344  043 031               5772          DB      '#',CT.PS
072.346  000                   5773          DB      0               END OF TABLE


                                5775  **      COT - CHECK OPERAND TYPES.
                                5776  *
                                5777  *       COT CHECKS THE OPERANDS TO SEE IF THE TYPE IS CONSISTANT.
                                5778  *
                                5779  *       EXIT    (ACCX), (ACCY) = 2 OPERANDS
                                5780  *       EXIT    TO *RET* IF BOTH SAME TYPE
                                5781  *                'Z' SET IF NUMERIC
                                5782  *                TO ERR.TE IF OF DIFFERING TYPES
                                5783  *       USES    A,F,H,L
                                5784
                                5785
072.347  072 201 042           5786  COT     LDA     ACCX-1
072.352  041 207 042           5787          LXI     H,ACCY-1
072.355  057                   5788          CMA
072.356  256                   5789          XRA     M
072.357  346 001               5790          ANI     CF.STR
072.361  312 155 070           5791          JZ      ERR.TC          DIFFERENT TYPES
072.364  246                   5792          ANA     M               (A) = CODE
072.365  311                   5793          RET                     RETURN WITH CODE


                                5795  **      CSA - CALCULATE SYMTAB ABSOLUTE ADDR.
                                5796  *
                                5797  *       CSA CALCULATES AN ABSOLUTE ADDRESS FOR A GIVEN
                                5798  *       INDEX
                                5799  *
                                5800  *       ENTRY   (DE) = INDEX INTO SYMTAB
                                5801  *       EXIT    (DE) = ABSOLUTE ADDRESS
                                5802  *       USES    D,E
```

```
                          5803
                          5804
072.366                   5805  CSA     EQU     *
                          5806
072.366  365             5807          PUSH    PSW             SAVE (A)
072.367  345             5808          PUSH    H               SAVE (HL)
072.370  052 126 112      5809          LHLD    SYMTAB+MT.FWA   (HL) = #FWA OF SYMTAB
072.373  031             5810          DAD     D                                    /80.01.GC/
072.374  353             5811          XCHG                    DE = ABSOLUTE ADDRESS IN SYMTAB /80.01.GC/
072.375  341             5812          POP     H               RESTORE (HL)
072.376  361             5813          POP     PSW
072.377  311             5814          RET                     EXIT


                          5816  **      CSI - CALCULATE SYMTAB INDEX
                          5817  *
                          5818  *       CSI CALCULATES AN INDEX INTO THE SYMTAB
                          5819  *       FROM A GIVEN ABSOLUTE ADDRESS
                          5820  *
                          5821  *       ENTRY   (DE) = ABSOLUTE ADDRESS INTO SYMBOL
                          5822  *       EXIT    (DE) = INDEX INTO SYMTAB
                          5823  *       USES    D,E
                          5824
                          5825
073.000                   5826  CSI     EQU     *
                          5827
073.000  365             5828          PUSH    PSW             SAVE (A)
073.001  345             5829          PUSH    H               SAVE (HL)
073.002  052 126 112      5830          LHLD    SYMTAB+MT.FWA
073.005  173             5831          MOV     A,E
073.006  225             5832          SUB     L
073.007  137             5833          MOV     E,A
073.010  172             5834          MOV     A,D
073.011  234             5835          SBB     H
073.012  127             5836          MOV     D,A             (DE) = INDEX INTO SYMBOL TABLE
073.013  341             5837          POP     H               RESTORE (HL)
073.014  361             5838          POP     PSW             RESTORE (A)
073.015  311             5839          RET                     EXIT


                          5841  **      CSE - CREATE STRING TABLE ENTRY.
                          5842  *
                          5843  *       CSE CREATES A STRING TABLE ENTRY.
                          5844  *
                          5845  *       ENTRY   (DE) = POINTER BLOCK ADDRESS.
                          5846  *       EXIT    DESCRIPTOR SET IN BLOCK
                          5847  *               (DE) = POINTER BLOCK ADDRESS
                          5848  *               (HL) = ABS STRING ADDRESS
                          5849  *       USES    A,F,D,E,H,L
                          5850
                          5851
073.016  305              5852  CSE     PUSH    B
```

```
073.017  041 205 112  5853          LXI     H,STRVI
073.022  315 056 073  5854          CALL    CSE..
073.025  021 152 112  5855          LXI     D,STRTAB+1
073.030  303 045 073  5856          JMP     CSE1
                      5857
073.033  305         5858  CSE.    PUSH    B
073.034  041 207 112  5859          LXI     H,STRTI
073.037  315 056 073  5860          CALL    CSE..
073.042  021 157 112  5861          LXI     D,TSTTAB+1
                      5862
073.045  315 026 071  5863  CSE1    CALL    AMB             MAKE ROOM
073.050  160         5864          MOV     M,B
073.051  043         5865          INX     H
073.052  161         5866          MOV     M,C             SET NUMBER IN STRING
073.053  043         5867          INX     H               (HL) = ABS ADDRESS
073.054  301         5868          POP     B
073.055  311         5869          RET
                      5870
073.056  043         5871  CSE..   INX     H
073.057  064         5872          INR     M               INCREMENT INDEX
073.060  053         5873          DCX     H
073.061  302 065 073  5874          JNZ     CSE2            NOT OVERFLOW
073.064  064         5875          INR     M
073.065  106         5876  CSE2    MOV     B,M
073.066  043         5877          INX     H
073.067  116         5878          MOV     C,M             (BC) = STRING NAME
073.070  353         5879          XCHG                    (HL) = BLOCK ADDRESS + 2
073.071  136         5880          MOV     E,M
073.072  043         5881          INX     H
073.073  126         5882          MOV     D,M             (DE) = STRING LENGTH
073.074  043         5883          INX     H
073.075  160         5884          MOV     M,B
073.076  043         5885          INX     H
073.077  161         5886          MOV     M,C             STORE IN HEADER
073.100  023         5887          INX     D
073.101  023         5888          INX     D               +2 FOR HEADER
073.102  353         5889          XCHG
073.103  311         5890          RET


                      5892  **      CUF     -  CLEAR USER FUNCTION
                      5893  *
                      5894  *       CUF CLEARS THE USER-DEFINED FUNCTIONS FROM THE FUNCTION TABLE
                      5895  *       BY REMOVING THE ENTRIES FROM  *SYMTAB*.
                      5896  *
                      5897  *       ENTRY:  NONE
                      5898  *
                      5899  *       EXIT:   USER-DEFINED FUNCTIONS OUT OF THE SYMBOL TABLE ENTRY
                      5900  *
                      5901  *       USES:   ALL
                      5902  *
                      5903
073.104               5904  CUF     EQU     *
                      5905
```

```
073.104  021 000 000  5906          LXI     D,0             SET THE INDEX TO ZERO
                      5907
073.107  052 130 112  5908  CUF1    LHLD    SYMTAB+MT.LEN
073.112  023         5909          INX     D
073.113  315 010 112  5910          CALL    HLCPDE
073.116  033         5911          DCX     D
073.117  330         5912          RC                      ALL FINISHED (LENGTH <= INDEX+1)
                      5913
073.120  315 126 073  5914          CALL    CUF2            PROCESS THE ENTRY
                      5915
073.123  303 107 073  5916          JMP     CUF1
                      5917
                      5918  *       PROCESS A SYMBOL TABLE ENTRY
                      5919
073.126               5920  CUF2    EQU     *
                      5921
073.126  052 126 112  5922          LHLD    SYMTAB+MT.FWA
073.131  031         5923          DAD     D               HL = FWA OF SYMBOL TABLE ENTRY
073.132  043         5924          INX     H
073.133  176         5925          MOV     A,M             A  FLAG BYTE
073.134  043         5926          INX     H
073.135  346 002      5927          ANI     CF.VEC
073.137  312 163 073  5928          JZ      CUF3            NOT A VECTOR
                      5929
073.142  176         5930          MOV     A,M
073.143  247         5931          ANA     A
073.144  362 172 073  5932          JP      CUF4            IS A VECTOR
                      5933
                      5934  *       DELETE A FUNCTION
                      5935
073.147  325         5936          PUSH    D
073.150  353         5937          XCHG                    HL = INDEX INTO TABLE
073.151  021 006 000  5938          LXI     D,6             COUNT = 6
073.154  315 203 104  5939          CALL    $DBT            DELETE THE BYTES FROM THE TABLE
073.157  126 112      5940          DW      SYMTAB+1
073.161  321         5941          POP     D
073.162  311         5942          RET
                      5943
                      5944  *       PASS OVER A SCALAR
                      5945
073.163  001 006 000  5946  CUF3    LXI     B,6
073.166  353         5947          XCHG
073.167  011         5948          DAD     B
073.170  353         5949          XCHG                    INDEX = INDEX + 6
073.171  311         5950          RET
                      5951
                      5952  *       PASS OVER A VECTOR
                      5953
073.172  043         5954  CUF4    INX     H
073.173  043         5955          INX     H               SKIP 'DIM' AND '0' BYTES
                      5956
073.174  116         5957          MOV     C,M
073.175  043         5958          INX     H
073.176  106         5959          MOV     B,M
073.177  043         5960          INX     H               BC = ARRAY SIZE FOR VECTOR ENTRYS
                      5961
```

```
073.200  353            5962            XCHG
073.201  011            5963            DAD     B               INDEX = INDEX + SIZE
073.202  001 006 000     5964            LXI     B,6
073.205  011            5965            DAD     B               INDEX = INDEX + 6 (BYTES SKIPPED AT START)
073.206  353            5966            XCHG
073.207  311            5967            RET


                        5969  **        CVX - COPY VALUE INTO 'X' ACCUMULATOR.
                        5970  *
                        5971  *         CVX COPIES A 4 BYTE VALUE INTO THE X ACCUMULATOR.
                        5972  *
                        5973  *         ENTRY   (DE) = ADDRESS OF VALUE
                        5974  *         EXIT    COPIED
                        5975  *         USES    A,F
                        5976
                        5977
073.210  345            5978  CVX       PUSH    H
073.211  325            5979            PUSH    D
073.212  041 202 042    5980            LXI     H,ACCX
073.215  315 051 076    5981            CALL    MOV4            MOVE
073.220  321            5982            POP     D
073.221  341            5983            POP     H
073.222  311            5984            RET


                        5986  **        CXY - COPY (ACCX) INTO (ACCY)
                        5987  *
                        5988  *         ENTRY   NONE
                        5989  *         EXIT    NONE
                        5990  *         USES    A,F,D,E
                        5991
                        5992
073.223                 5993  CXY       EQU     *
073.223  345            5994            PUSH    H               SAVE (HL)
073.224  021 201 042    5995            LXI     D,ACCX-1        SOURCE
073.227  041 207 042    5996            LXI     H,ACCY-1        DESTINATION
073.232  315 045 076    5997            CALL    MOV5            MOVE (ACCX) TO (ACCY)
073.235  341            5998            POP     H               RESTORE (HL)
073.236  311            5999            RET                     EXIT


                        6001  **        CXV - COPY X TO VALUE.
                        6002  *
                        6003  *         CXV COPIES THE CONTENTS OF THE 'X' ACCUMULATOR INTO A MEMORY
                        6004  *         LOCATION.
                        6005  *
                        6006  *         ENTRY   (DE) = TARGET ADDRESS
                        6007  *         EXIT    COPIED
                        6008  *         USES    A,F
```

```
                                6009
                                6010
073.237  345                   6011  CXV       PUSH    H
073.240  325                   6012  CXV.      PUSH    D
073.241  353                   6013            XCHG
073.242  021 202 042           6014            LXI     D,ACCX
073.245  315 051 076           6015            CALL    MOV4            MOVE
073.250  321                   6016            POP     D               RESTORE DE
073.251  341                   6017            POP     H
073.252  311                   6018            RET


                                6020  **        DCN - DECODE CHANNEL NUMBER.
                                6021  *
                                6022  *         DCN DECODES A CHANNEL SPECIFICATION OF THE FORM:
                                6023  *
                                6024  *         #N                  OR
                                6025  *         #LNO(EXPR) ARCHAIC(THIS IS TACKY!)                      /80.01.GC/
                                6026  *
                                6027  *         IF THE CHANNEL EXPRESSION IS OMITTED, IOCHAN IS SETUP TO INDICATE THE
                                6028  *         SYSTEM CONSOLE.
                                6029  *
                                6030  *         ENTRY   (BC) = TEXT POINTER
                                6031  *         EXIT    (BC) ADVANCED
                                6032  *                  IOCHAN = 0 IF CONSOLE, = N+1 IF FILE
                                6033  *                 (A) = (IOCHAN)
                                6034  *         USES    ALL
                                6035
                                6036
073.253  257                   6037  DCN       XRA     A
073.254  062 202 112           6038            STA     IOCHAN          ASSUME NONE
073.257  315 072 076           6039            CALL    PNT
073.262  376 031               6040            CPI     CT.PS
073.264  300                   6041            RNE                     NONE
073.265  315 273 073           6042            CALL    DCN.            DECODE CHANNEL NUMBER
073.270  303 223 072           6043            JMP     CMA             REQUIRE COMMA AND EXIT
                                6044
                                6045  **        DCN. - DECODE CHANNEL NUMBER.
                                6046  *
                                6047  *         SAME AS DCN, BUT REQUIRES CHANNEL
                                6048  *         AND DOESNT CHECK FOR TRAILING COMMA
                                6049  *
                                6050
073.273  315 305 077           6051  DCN.      CALL    RNT
073.276  031                   6052            DB      CT.PS
073.277  315 036 057           6053            CALL    EVALI           EVALUATE AN EXPRESSION          /80.01.GC/
                                6054
                                6055  **        DCN.. - CHECK CHANNEL NUMBER.
                                6056  *
                                6057  *         CHECH (DE) FOR VALID CHANNEL NUMBER
                                6058  *         EXIT    (A) = CHANNEL VALUE
                                6059  *
                                6060
073.302  172                   6061  DCN..     MOV     A,D
```

```
073.303  267            6062           ORA     A
073.304  302 122 070    6063           JNZ     ERR.IN          TOO LARGE
073.307  173            6064           MOV     A,E
073.310  376 006        6065           CPI     CHANMAX+1                       /78.10.GC/
073.312  322 122 070    6066           JNC     ERR.IN          TOO LARGE
073.315  247            6067           ANA     A
073.316  312 322 073    6068           JZ      DCN1
073.321  074            6069           INR     A               (A) = 2+N
073.322  062 202 112    6070  DCN1     STA     IOCHAN
073.325  311            6071           RET                     EXIT


                        6073  **       DNF - DELETE NON-OPEN FILE BLOCKS.
                        6074  *
                        6075  *        DNF DELETES ALL THE NON-OPEN FILE BLOCKS THAT ARE AT THE
                        6076  *        END OF THE FILTAB. AS SOON AS AN OPEN FILE BLOCK BECOMES THE END ONE,
                        6077  *        NO MORE WILL BE DELETED. THUS, IF #5 IS OPEN, AND #4, #3, AND #2 ARE
                        6078  *        CLOSED, THEY WILL REMAIN UNRECOVERED UNTIL #5 IS CLOSED, AND THEN BE
                        6079  *        CLEANED OUT IN ONE SWOOP.
                        6080  *
                        6081  *        ENTRY   NONE
                        6082  *        EXIT    NONE
                        6083  *        USES    ALL
                        6084
                        6085
073.326  072 167 112    6086  DNF      LDA     FILTAB+MT.LEN+1 (A) = # OF BUFFERS
073.331  247            6087           ANA     A
073.332  310            6088           RZ                      NONE ELIGIBLE
073.333  021 033 000    6089           LXI     D,FBENL
073.336  315 007 031    6090           CALL    $MU86
073.341  021 231 042    6091           LXI     D,FBLIST+FB.FLG
073.344  031            6092           DAD     D               (HL) = ADDRESS OF FB.STA FOR LAST BLOCK W/BUFFER
073.345  176            6093           MOV     A,M
073.346  247            6094           ANA     A
073.347  300            6095           RNZ                     IS OPEN
073.350  041 167 112    6096           LXI     H,FILTAB+MT.LEN+1
073.353  065            6097           DCR     M               SHORTEN TABLE
073.354  303 326 073    6098           JMP     DNF             TRY AGAIN


                        6100  **       DTS - DELETE TEMP STRINGS.
                        6101  *
                        6102  *        DTS DELETES ANY TEMP STRINGS WHICH MAY HAVE BUILT UP
                        6103  *        IN THE STRING TABLE.
                        6104  *
                        6105  *        ENTRY   NONE
                        6106  *        EXIT    TABLE PARED.
                        6107  *        USES    H,L
                        6108
                        6109
073.357  041 000 000    6110  DTS      LXI     H,0
073.362  042 161 112    6111           SHLD    TSTTAB+MT.LEN
```

```
073.365  041 300 000  6112          LXI     H,300Q
073.366               6113  DTSA    EQU     *-2
073.370  042 207 112  6114          SHLD    STRTI           RESET STRING TEMP INDEX
073.373  311          6115          RET


                      6117  **      EKA - EXPAND KEYWORD INTO ASCII EQUIVALENT.
                      6118  *
                      6119  *       EKA EXPANDS A KEYWORD BYTE INTO THE ASCII EQUIVALENT.
                      6120  *
                      6121  *       ENTRY   (A) = TOKEN
                      6122  *               (DE) = ADDRESS FOR STRING
                      6123  *       EXIT    (A) = LAST CHARACTER OF ASCII
                      6124  *               (DE) = ADDRESS FOR LAST CHARACTER OF ASCII
                      6125  *       USES    A,F,B,C,D,E
                      6126
                      6127
073.374  001 240 066  6128  EKA     LXI     B,KEYTAB
073.377  325          6129          PUSH    D               SAVE ADDRESS
074.000  127          6130          MOV     D,A             (D) = PATTERN
074.001  012          6131  EKA1    LDAX    B
074.002  272          6132          CMP     D
074.003  003          6133          INX     B
074.004  302 001 074  6134          JNE     EKA1            NOT THERE, YET
074.007  321          6135          POP     D               (DE) = ADDRESS
074.010  365          6136          PUSH    PSW             SAVE KEYWORD BYTE
                      6137
                      6138  *       EXPAND IT.
                      6139
074.011  012          6140  EKA2    LDAX    B
074.012  022          6141          STAX    D
074.013  003          6142          INX     B
074.014  023          6143          INX     D
074.015  247          6144          ANA     A
074.016  362 011 074  6145          JP      EKA2            MORE TO GO
074.021  033          6146          DCX     D               REPLACE EXTRA BYTE WITH ' ' OR '('
074.022  361          6147          POP     PSW
074.023  376 320      6148          CPI     CT.FCN
074.025  076 040      6149          MVI     A,' '           ASSUME NOT FUNCTION
074.027  330          6150          RC                      NOT FUNCTION
074.030  076 050      6151          MVI     A,'('           IS FUNCTION
074.032  311          6152          RET


                      6154  **      ELN - EVALUATE LINE NUMBER.
                      6155  *
                      6156  *       ELN IS CALLED WHEN A LINE NUMBER IS TO BE EVALUATED.
                      6157  *
                      6158  *       THE LINE NUMBER CAN EITHER BE A DECIMAL INTEGER, OR
                      6159  *       THE EXPRESSION LNO(EXPR)
                      6160  *
                      6161  *       ENTRY   (BC) = LINE POINTER
```

```
                                6162  *       EXIT    (BC) UPDATED
                                6163  *               (DE) = LINE NUMBER
                                6164  *       USES    A,F,B,C,D,E
                                6165
                                6166
074.033  315 126 100           6167  ELN     CALL    SOB             SKIP BLANKS
                                6168
                                6169  *       MUST HAVE DECIMAL INTEGER, OR LNO(
                                6170
074.036  012                  6171          LDAX    B
074.037  376 327               6172          CPI     CT.LNO
074.041  312 102 074           6173          JE      ELN2            IS LNO(EXPR)
074.044  315 363 111           6174          CALL    $CVD.           SEE IF DIGIT
074.047  332 152 070           6175          JC      ERR.SY
074.052  021 000 000           6176          LXI     D,0             (DE) =ACCUM
                                6177
                                6178  *       HAVE DECIMAL INTERGER
                                6179
074.055  012                  6180  ELN1    LDAX    B
074.056  315 363 111           6181          CALL    $CVD.
074.061  330                  6182          RC                      END OF NUMBER
074.062  345                  6183          PUSH    H               SAVE (HL)
074.063  315 324 030           6184          CALL    $MUI0           (HL) = 10*ACCUM
074.066  315 072 030           6185          CALL    $DADA           (HL) = 10*ACCUM+DIGIT
074.071  353                  6186          XCHG
074.072  003                  6187          INX     B
074.073  341                  6188          POP     H               RESTORE (HL)
074.074  332 122 070           6189          JC      ERR.IN          ILLEGAL NUMBER IF OVERFLOW
074.077  303 055 074           6190          JMP     ELN1            TRY FOR MORE
                                6191
                                6192  *       IS LNO(EXPR)
                                6193
074.102  003                  6194  ELN2    INX     B               SKIP LNO( KEYWORD
074.103  315 036 057           6195          CALL    EVALI
074.106  325                  6196          PUSH    D               SAVE VALUE
074.107  315 305 077           6197          CALL    RNT
074.112  020                  6198          DB      CT.PAR          REQURE ')'
074.113  321                  6199          POP     D
074.114  311                  6200          RET


                                6202  **      FOC - FILE OPEN CLEANUP.
                                6203  *
                                6204  *       FOC IS CALLED TO CLEANUP AFTER FOP. FOC RESTORES AS MUCH
                                6205  *       MEMORY AS THE SYSTEM IS NOT USING (ALSO DEPENDS UPON CNTRL OPTION)
                                6206  *
                                6207  *       ENTRY   NONE
                                6208  *       EXIT    NONE
                                6209  *       USES    NONE
                                6210
                                6211
074.115  315 054 031           6212  FOC     CALL    $SAVALL         SAVE ALL
                                6213  *       LHLD    S.DLINK                                 /80.02.GC/
                                6214  *       ERRNZ   M.SYSM                                  /80.02.GC/
```

```
                                6215  *       MVI     M,0             CLEAR ARTIFICAL SYSTEM MODE (ALLOW H17 RAM TO
                                6216  *                                WRITE DISABLE))                 /80.02.GC/
074.120  052 320 040            6217          LHLD    S.SYSM
074.123  021 360 377            6218          LXI     D,-16
074.126  031                    6219          DAD     D               (HL) = LWA USABLE
074.127  072 203 112            6220          LDA     OVLMAN          (A) = OVERLAY MANGAEMENT FLAG
074.132  247                    6221          ANA     A
074.133  312 146 074            6222          JZ      FOC1            LEAVE OVERLAY OUT
074.136  353                    6223          XCHG
074.137  052 324 040            6224          LHLD    S.OMAX
074.142  315 224 030            6225          CALL    $CHL            (HL) = -OVLMAX
074.145  031                    6226          DAD     D               (HL) = LIMIT TO ALLOW OVL RESIDENT
074.146  353                    6227  FOC1    XCHG                    (DE) = PROSPECTIVE NEW MEML
074.147  052 171 112            6228          LHLD    MEML            (HL) = CURRENET MEML
074.152  173                    6229  FOC1.3  MOV     A,E                                     /80.01.GC/
074.153  225                    6230          SUB     L               MAKE SURE GETTING LARGER, NOT SMALLER!
074.154  172                    6231          MOV     A,D
074.155  234                    6232          SBB     H
074.156  332 210 074            6233          JC      FOC3            NOT ENOUGH MEMORY
074.161  353                    6234          XCHG                    (HL) = NEW LIMIT
074.162  042 171 112            6235  FOC1.5  SHLD    MEML            NOTIFY TABLES
074.165  353                    6236          XCHG                    (DE) = NEW MEML
074.166  052 322 040            6237          LHLD    S.USRM
074.171  315 216 030            6238          CALL    $CDEHL
074.174  312 205 074            6239          JE      FOC2            NO NEED TO REQUEST, ALREADY GOT IT
074.177  353                    6240          XCHG                    (HL) = REQUEST
074.200  377 052                6241          DB      SYSCALL+.SETTP  SET TOP
074.202  332 223 070            6242          JC      SERROR          ERROR
074.205  303 047 031            6243  FOC2    JMP     $RSTALL         RESTORE ALL AND EXIT
                                6244
                                6245  *       NOT ENOUGH MEMORY TO RESIDE OVERLAY
                                6246
074.210  257                    6247  FOC3    XRA     A
074.211  062 203 112            6248          STA     OVLMAN          CLEAR RESIDE
074.214  303 160 070            6249          JMP     ERR.TO          TABLE OVERFLOW


                                6251  **      FOP - FILE OPEN PRESET.
                                6252  *
                                6253  *       FOP IS CALLED BEFORE FILE OPENING AND CLOSING IS TO TAKE
                                6254  *       PLACE. SINCE THE SYSTEM WILL LOAD AN OVERLAY, AND MAY
                                6255  *       NEED TO LOAD A DEVICE DRIVER, FOP WILL SQUEEZE THE TABLES
                                6256  *       UP TO TAKE AS LITTLE SPACE AS POSSIBLE. LATER ON, FOC WILL BE
                                6257  *       USED TO RESTORE THE TABLES INTO ANY OPEN SPACE
                                6258  *       LEFT AFTER THE OPERATIONS.
                                6259  *
                                6260  *       ENTRY   NONE
                                6261  *       EXIT    NONE
                                6262  *       USES    NONE
                                6263
                                6264
074.217  315 054 031            6265  FOP     CALL    $SAVALL         SAVE REGS
074.222  315 230 074            6266          CALL    FOP.                                    /80.01.GC/
074.225  303 162 074            6267          JMP     FOC1.5                                  /80.01.GC/
```

```
                                6268
074.230  315 127 104           6269  FOP.      CALL    MTD             MOVE TABLES DOWN
074.233  325                    6270            PUSH    D               SAVE LWA
074.234  315 071 071            6271            CALL    $ATP            ADJUST TABLE POINTERS
074.237  341                    6272            POP     H               (HL) = LWA
074.240  043                    6273            INX     H
074.241  311                    6274            RET                                             /80.01.GC/


                                6276  **        FLN - FIND LINE BY NUMBER.
                                6277  *
                                6278  *         FLN SEARCHES THE TEXT BUFFER FOR THE SPECIFIED LINE.
                                6279  *
                                6280  *         ENTRY   (DE) = LINE NUMBER
                                6281  *         EXIT    TO ERR.SN IF LINE NUMBER = 65535
                                6282  *                 'C' SET IF NOT FOUND
                                6283  *                 (HL) = ADDRESS OF LINE IF FOUND, ADDRESS IF LINE+1 IF NOT
                                6284  *         USES    A,F,H,L
                                6285
                                6286
074.242  305                    6287  FLN       PUSH    B
074.243  041 010 115            6288            LXI     H,MTAREA
                                6289
                                6290  *         CHECK IF LINE NUMBER = 65535
                                6291
074.246  172                    6292            MOV     A,D
074.247  074                    6293            INR     A
074.250  302 260 074            6294            JNZ     FLN1            HIGH ORDER BYTE <> 377
074.253  173                    6295            MOV     A,E
074.254  074                    6296            INR     A
074.255  312 147 070            6297            JZ      ERR.SN          LINE NUMBER = 65535; ERROR
                                6298
074.260  173                    6299  FLN1      MOV     A,E
074.261  226                    6300            SUB     M
074.262  107                    6301            MOV     B,A             (B) = LOW LETER
074.263  172                    6302            MOV     A,D
074.264  043                    6303            INX     H
074.265  236                    6304            SBB     M
074.266  332 312 074            6305            JC      FLN3            RAN PAST
074.271  302 300 074            6306            JNZ     FLN1.5          NOT THERE YET
074.274  260                    6307            ORA     B
074.275  312 312 074            6308            JZ      FLN3            FOUND IT
074.300  043                    6309  FLN1.5    INX     H
074.301  176                    6310  FLN2      MOV     A,M             SKIP THIS LINE
074.302  043                    6311            INX     H
074.303  247                    6312            ANA     A
074.304  302 301 074            6313            JNZ     FLN2            NOT END OF LINE
074.307  303 260 074            6314            JMP     FLN1
                                6315
                                6316  *         FOUND LINE. 'C' CLEAR IF FOUND
                                6317
074.312  053                    6318  FLN3      DCX     H
074.313  301                    6319            POP     B
074.314  311                    6320            RET
```

```
                                6322  **      FSE - FIND STRINGTAB ENTRY.
                                6323  *
                                6324  *       FSE FINDS A SPECIFED STRING IN THE TABLE.
                                6325  *
                                6326  *       ENTRY   (DE) = DESCRIPTOR ADDRESS
                                6327  *       EXIT    (HL) = ABS ADDRESS
                                6328  *               (A) = LENGTH
                                6329  *       USES    A,F,D,E,H,L
                                6330
                                6331
074.315  032                   6332  FSE     LDAX    D               (A) = LENGTH
074.316  365                   6333          PUSH    PSW
074.317  023                   6334          INX     D
074.320  023                   6335          INX     D
074.321  353                   6336          XCHG
074.322  126                   6337          MOV     D,M
074.323  043                   6338          INX     H
074.324  136                   6339          MOV     E,M             (DE) = INDEX
                                6340
                                6341  *       CHECK FOR WHICH STRING TABLE
                                6342
074.325  172                   6343          MOV     A,D
074.326  346 100               6344          ANI     100Q
074.330  312 341 074           6345          JZ      FSE0
074.333  052 157 112           6346          LHLD    TSTTAB+MT.FWA
074.336  303 344 074           6347          JMP     FSE1
                                6348
074.341  052 152 112           6349  FSE0    LHLD    STRTAB+MT.FWA
000.000                         6350          ERRNZ   *-FSE1
                                6351
                                6352  *       SEE IF WE HAVE IT YET
                                6353
074.344  172                   6354  FSE1    MOV     A,D
074.345  247                   6355          ANA     A
074.346  362 374 074           6356          JP      FSE3            NOT LOOKING FOR A VALID STRING ID
074.351  276                   6357          CMP     M
074.352  043                   6358          INX     H
074.353  302 363 074           6359          JNE     FSE2            NO MATCH
074.356  173                   6360          MOV     A,E
074.357  276                   6361          CMP     M
074.360  312 374 074           6362          JE      FSE3            FOUND
074.363  043                   6363  FSE2    INX     H               NOT FOUND
074.364  176                   6364          MOV     A,M
074.365  247                   6365          ANA     A
074.366  362 363 074           6366          JP      FSE2            SKIP TO NEXT INDEX
074.371  303 344 074           6367          JMP     FSE1            TRY AGAIN
                                6368
                                6369  *       FOUND IT.
                                6370
074.374  043                   6371  FSE3    INX     H
074.375  361                   6372          POP     PSW             (A) = LEN
074.376  311                   6373          RET
```

```
                                6375  **      IFIX - SPLIT NUMBER INTO INTEGER AND FRACTION.
                                6376  *
                                6377  *       IFIX FIXES ((DE)) INTO AN INTEGER.
                                6378  *
                                6379  *       ENTRY   (DE) = ADDRESS OF NUMBER
                                6380  *       EXIT    (DE) = INTEGERAL PART OF 0<=N<=65535
                                6381  *                      TO ERR.IN OTHERWISE
                                6382
                                6383
074.377  021 202 042           6384  IFIX.   LXI     D,ACCX          USE ACCX
075.002  305                   6385  IFIX    PUSH    B
075.003  345                   6386          PUSH    H
075.004  353                   6387          XCHG
075.005  315 250 107           6388          CALL    LDD             (BCDE) = NUMBER
075.010  171                   6389          MOV     A,C
075.011  247                   6390          ANA     A
075.012  372 122 070           6391          JM      ERR.IN          TOO LARGE
075.015  076 220               6392          MVI     A,220Q
075.017  220                   6393          SUB     B
075.020  332 122 070           6394          JC      ERR.IN          TOO LARGE
075.023  306 007               6395          ADI     7               (A) = SHIFT COUNT
075.025  107                   6396          MOV     B,A
075.026  315 231 107           6397  IFIX1   CALL    SRS
075.031  005                   6398          DCR     B
075.032  302 026 075           6399          JNZ     IFIX1           NOT DONE YET
075.035  341                   6400          POP     H
075.036  301                   6401          POP     B
075.037  311                   6402          RET


                                6404  **      IFLT - FLOAT NUMBER.
                                6405  *
                                6406  *       ENTRY   (DE) = VALUE
                                6407  *       EXIT    (ACCX) = NUMBER VALUE
                                6408  *                       (DE) = #ACCX-1
                                6409
                                6410
075.040  305                   6411  IFLT    PUSH    B
075.041  345                   6412          PUSH    H
075.042  001 000 227           6413          LXI     B,200Q+23*256
075.045  315 213 105           6414          CALL    NRM             NORMALIZE
075.050  315 245 106           6415          CALL    STX             STORE IN ACCX
075.053  076 300               6416          MVI     A,CT.SNV        SCALAR NUMERIC VALUE
075.055  062 201 042           6417          STA     ACCX-1          SET TYPE
075.060  341                   6418          POP     H
075.061  301                   6419          POP     B
075.062  311                   6420          RET
```

```
                                    6422  **      ILM - ISSUE LINE MESSAGE.
                                    6423  *
                                    6424  *       ILM ISSUES A MESSAGE OF THE FORM
                                    6425  *
                                    6426  *       XXXXXX AT LINE NNNNN
                                    6427  *
                                    6428  *       WHERE XXXXXX = SUPPLIED TEXT,
                                    6429  *                NNNNN = (CURNUM)
                                    6430  *
                                    6431  *       NOTE THAT ILM ALSO CLEARS THE BASIC WORKING CHANNEL
                                    6432  *       (CHANNEL 0). THIS IS A KLUDGE SO THAT THE CHANNEL DOESNT REMAIN
                                    6433  *       OPEN IF AN ERROR OCCOURS WHILE USING IT.
                                    6434  *
                                    6435  *       ENTRY   (SP+0) = ERROR CODE
                                    6436  *       EXIT    NONE
                                    6437  *       USES    A,F,H,L
                                    6438
                                    6439
075.063  315 217 074                6440  ILM     CALL    FOP             ALLOW OVERLAY TO RESIDE
075.066  257                        6441          XRA     A
075.067  377 055                    6442          DB      SYSCALL,.CLEAR
075.071  361                        6443          POP     PSW             (A) = CODE
075.072  046 040                    6444          MVI     H,' '
075.074  377 057                    6445          DB      SYSCALL,.ERROR  LOOKUP ERROR
                                    6446
075.076  041 124 043                6447          LXI     H,RESTART       (HL) = EXIT PROCESSOR ADDRESS
075.077                             6448  ILMA    EQU     *-2             SET BY CALLER
075.101  345                        6449          PUSH    H               SET AS 'RETURN ADDRESS'
075.102  041 343 114                6450          LXI     H,RUNMOD
075.105  176                        6451          MOV     A,M             (A) = OLD RUN MODE
075.106  366 200                    6452          ORI     RM.HLT          SET HALT FLAG
075.110  167                        6453          MOV     M,A
075.111  376 200                    6454          CPI     RM.IMM+RM.HLT
075.113  310                        6455          RE                      DONT PRINT LINE NUMBER IF IMMEDIATE
075.114  315 136 031                6456          CALL    $TYPTX
075.117  101 164 040                6457          DB      'At Line',' '+200Q
075.127  052 175 112                6458          LHLD    CURNUM
075.132  353                        6459          XCHG
075.133  303 206 100                6460          JMP     TDI             TYPE LINE NUMBER


                                    6462  **      IST - INSERT IN SYMBOL TABLE.
                                    6463  *
                                    6464  *       IST LOOKS UP THE ADDRESS HOLDING THE VALUE FOR
                                    6465  *       A VARIABLE. IF THE VARIABLE IS A MATRIX OR VECTOR,
                                    6466  *       THE SUBSCRIPT IS EVALUATED AND THE PARTICULAR ENTRY
                                    6467  *       IS RETURED AS A SCALAR VALUE.
                                    6468  *
                                    6469  *       IF THE VARIABLE IS NOT YET DEFINED, AND IT IS NOT A
                                    6470  *       VECTOR, IT IS DEFINED WITH A VALUE OF 0 (OR A NULL STRING).
                                    6471  *
                                    6472  *       ENTRY   (BC) = TEXT POINTER
                                    6473  *       EXIT    (BC) UPDATED
                                    6474  *               (DE) = INDEX OF SYMBOL ADDRESS
```

```
                                6475  *          (A) = TYPE
                                6476  *     USES ALL
                                6477
                                6478
075.136  315 172 075            6479  IST  CALL  ISTO       INSERT SYMBOL IN TABLE
075.141  365                    6480       PUSH  PSW        SAVE TYPE
075.142  315 000 073            6481       CALL  CSI        (DE) = INDEX INTO SYMBOL TABLE
075.145  346 001                6482       ANI   CF.STR
075.147  312 170 075            6483       JZ    ISTOO      IS NOT STRING TYPE
075.152  325                    6484       PUSH  D          SAVE INDEX
075.153  315 366 072            6485       CALL  CSA        (DE) = ABS. ADDR. INTO SYMBOL
075.156  325                    6486       PUSH  D
075.157  023                    6487       INX   D
075.160  023                    6488       INX   D
075.161  032                    6489       LDAX  D          (A) = STRING ID
075.162  321                    6490       POP   D          RESTORE DE
075.163  247                    6491       ANA   A
075.164  314 016 073            6492       CZ    CSE        CREATE STRING TABLE ENTRY IF NOT THERE
075.167  321                    6493       POP   D
075.170  361                    6494  ISTOO POP   PSW        RESTORE TYPE
075.171  311                    6495       RET
                                6496
075.172                         6497  ISTO EQU   *
075.172  315 056 071            6498       CALL  ANT        ACCEPT NEXT TOKEN
075.175  376 300                6499       CPI   CT.VARL    SEE IF HAVE VARIABLE
075.177  332 152 070            6500       JC    ERR.SY     NOT VARIABLE
075.202  376 310                6501       CPI   CT.VARH+1
075.204  322 152 070            6502       JNC   ERR.SY     NOT VARIABLE
075.207  041 151 335            6503       LXI   H,-LEXLIM-1
075.212  031                    6504       DAD   D
075.213  332 257 075            6505       JC    IST2       IS PRE-DEFINED
075.216  365                    6506  IST1 PUSH  PSW        SAVE TYPE
                                6507
                                6508  *          NEVER BEFORE DEFINED.
                                6509
075.217  346 002                6510       ANI   CF.VEC
075.221  302 171 070            6511       JNZ   ERR.ND     NOT DECLARED
075.224  021 126 112            6512       LXI   D,SYMTAB+1
075.227  041 006 000            6513       LXI   H,6
075.232  315 026 071            6514       CALL  AMB        ALLOCATE 6 BYTES
075.235  021 000 000            6515       LXI   D,0        (DE) = NAME
075.236                         6516  LEXA EQU   *-2        UNDEFINED NAME
075.240  162                    6517       MOV   M,D
075.241  043                    6518       INX   H
075.242  163                    6519       MOV   M,E        STORE NAME
075.243  043                    6520       INX   H
075.244  124                    6521       MOV   D,H
075.245  135                    6522       MOV   E,L        (DE) = ADDRESS OF VALUE
075.246  257                    6523       XRA   A          (A) = 0
075.247  167                    6524       MOV   M,A        CLEAR VALUE
075.250  043                    6525       INX   H
075.251  167                    6526       MOV   M,A
075.252  043                    6527       INX   H
075.253  167                    6528       MOV   M,A
075.254  043                    6529       INX   H
075.255  167                    6530       MOV   M,A        000 000 000 000
```

```
075.256  361          6531          POP     PSW             RESTORE TYPE
                      6532
                      6533  *       IS NOW DEFINED.
                      6534
075.257  334 107 055  6535  IST2    CC      VARIAB.         PROCESS VARIABLE IF NOT JUST DEFINED
075.262  311          6536          RET
```

```
                      6538  **      IVT - INSERT VECTOR IN TABLE.
                      6539  *
                      6540  *       IVT INSERTS A SYMBOL OF TYPE VECTOR IN THE SYMBOL TABLE.
                      6541  *       THE SYMBOL MUST NOT BE PREVIOUSLY DEFINED.
                      6542  *
                      6543  *       ENTRY   (BC) = TEXT POINTER
                      6544  *       EXIT    (BC) UPDATED
                      6545  *               (DE) = SYMBOL ADDRESS
                      6546  *               (A) = TYPE - CF.VEC
                      6547  *       USES    A,F,B,C,D,E
                      6548
                      6549
075.263  315 056 071  6550  IVT     CALL    ANT             ACCEPT NEXT TOKEN
075.266  041 151 335  6551          LXI     H,-LEXLIM-1
075.271  031          6552          DAD     D
075.272  332 125 070  6553          JC      ERR,IU          ALREADY DEFINED
075.275  356 002      6554          XRI     CF.VEC          TOGGLE VECTOR FLAG
075.277  303 216 075  6555          JMP     IST1            PROCESS AS IST
```

```
                      6557  **      LCC - LOCATE CHANNEL COLUMN COUNTER.
                      6558  *
                      6559  *       LCC IS CALLED TO LOCATE THE BYTE CONTAINING THE COLUMN COUNTER
                      6560  *       FOR THIS CHANNEL ((IOCHAN)). SINCE PRINTING
                      6561  *       CAN BE IN PROGRESS ON SEVERAL CHANNELS AT ONCE, A SEPERATE COLUNE
                      6562  *       COUNTER IS KEPT FOR EACH ONE.
                      6563  *
                      6564  *       ENTRY   NONE
                      6565  *       EXIT    (HL) = ADDRESS OF RIGHT ENTRY IN *COLCNTS*
                      6566  *       USES    A,F,H,L
                      6567
                      6568
075.302  041 315 112  6569  LCC     LXI     H,COLCNTS
075.305  072 202 112  6570          LDA     IOCHAN
075.310  303 101 030  6571          JMP     $DADA.          (HL) = ADDRESS
```

```
                                6573  **      LFC - LOCK FLAG CHECK
                                6574  *
                                6575  *       LFC CHECKS IF THE DATA LOCK IS INVOKED
                                6576  *
                                6577  *       ENTRY   NONE
                                6578  *       EXIT    TO ERR.LK IF DATA LOCK IN FORCE
                                6579  *               TO (RET) IF NORMAL
                                6580  *       USES    A,F
                                6581
                                6582
075.313  072 201 112           6583  LFC     LDA     LCKFLG          (A) = DATA LOCK FLAG
075.316  247                   6584          ANA     A
075.317  302 130 070           6585          JNZ     ERR.LK          DATA LOCK IN FORCE
075.322  311                   6586          RET                     EXIT
```

```
                                6588  **      LVS     -  LOOK-UP VARIABLE IN SYMBOL TABLE
                                6589  *
                                6590  *       LVS LOOKS UP THE SPECIFIED VARIABLE IN THE SYMBOL TABLE.
                                6591  *       THE VARIABLE IS SPECIFIED BY THE VARIABLE NAME AND TYPE
                                6592  *       IN THE  *DE*  REGISTER PAIR AS PER THE  *SYMTAB*  FORMAT.
                                6593  *
                                6594  *       ENTRY:  DE      =  SYMTAB KEY
                                6595  *
                                6596  *       EXIT:   PSW     =  'Z' CLEAR IF NOT FOUND
                                6597  *                       =  'Z' SET   IF     FOUND
                                6598  *                          DE  =  SYMTAB ADDRESS
                                6599  *
                                6600  *       USES:   PSW,DE
                                6601  *
                                6602
075.323                         6603  LVS     EQU     *
                                6604
075.323  345                   6605          PUSH    H
075.324  305                   6606          PUSH    B
075.325  052 130 112           6607          LHLD    SYMTAB+MT.LEN
075.330  104                   6608          MOV     B,H
075.331  115                   6609          MOV     C,L             BC = SYMTAB LENGTH
075.332  052 126 112           6610          LHLD    SYMTAB+MT.FWA   HL = SYMTAB FWA
                                6611
075.335  170                   6612  LVS1    MOV     A,B
075.336  261                   6613          ORA     C
075.337  312 040 076           6614          JZ      LVS4            NOT FOUND
                                6615
075.342  176                   6616          MOV     A,M
075.343  043                   6617          INX     H
075.344  272                   6618          CMP     D
075.345  176                   6619          MOV     A,M
075.346  302 363 075           6620          JNE     LVS2            NO MATCH
075.351  273                   6621          CMP     E
075.352  302 363 075           6622          JNE     LVS2            NO MATCH
                                6623
                                6624  *       HAVE A MATCH
                                6625
```

```
075.355  053          6626          DCX     H
075.356  353          6627          XCHG                    DE = SYMTAB ADDRESS
075.357  257          6628          XRA     A               SET THE ZERO FLAG
075.360  301          6629          POP     B
075.361  341          6630          POP     H
075.362  311          6631          RET
                      6632
                      6633  *       HAVE NO MATCH
                      6634
075.363  013          6635  LVS2    DCX     B
075.364  013          6636          DCX     B
075.365  013          6637          DCX     B
075.366  013          6638          DCX     B
075.367  013          6639          DCX     B
075.370  013          6640          DCX     B               BC = BC - 6
                      6641
075.371  176          6642          MOV     A,M
075.372  043          6643          INX     H
075.373  346 002      6644          ANI     CF.VEC
075.375  312 027 076  6645          JZ      LVS3            IS NOT A VECTOR
                      6646
076.000  176          6647          MOV     A,M
076.001  247          6648          ANA     A
076.002  372 027 076  6649          JM      LVS3            IS JUST A FUNCTION
                      6650
                      6651  *       PROCESS A VECTOR
                      6652
076.005  043          6653          INX     H
076.006  043          6654          INX     H               SKIP 'DIM' AND '0' BYTES
                      6655
076.007  325          6656          PUSH    D
076.010  136          6657          MOV     E,M
076.011  043          6658          INX     H
076.012  126          6659          MOV     D,M
076.013  043          6660          INX     H               DE = ARRAY SIZE
                      6661
076.014  171          6662          MOV     A,C
076.015  223          6663          SUB     E
076.016  117          6664          MOV     C,A
076.017  170          6665          MOV     A,B
076.020  232          6666          SBB     D
076.021  107          6667          MOV     B,A             BC = BC - SIZE
                      6668
076.022  031          6669          DAD     D               HL = HL + ARRAY SIZE
076.023  321          6670          POP     D
                      6671
076.024  303 335 075  6672          JMP     LVS1
                      6673
                      6674  *       PROCESS A NORMAL SCALAR
                      6675
076.027  305          6676  LVS3    PUSH    B
076.030  001 004 000  6677          LXI     B,6-2
076.033  011          6678          DAD     B               HL = HL + 6-2
076.034  301          6679          POP     B
                      6680
076.035  303 335 075  6681          JMP     LVS1            DO IT AGAIN
```

```
                                6682
                                6683  *       PROCESS AN UNFOUND VARIABLE
                                6684
076.040  366 001               6685  LVS4    ORI     1               CLEAR ZERO FLAG
076.042  301                   6686          POP     B
076.043  341                   6687          POP     H
076.044  311                   6688          RET


                                6690  **      MOVX - MOVE X BYTES OF DATA
                                6691  *
                                6692  *       MOVX CONSISTS OF TWO ROUTINES
                                6693  *
                                6694  *       MOV4 MOVES 4 BYTES OF DATA
                                6695  *
                                6696  *       MOV5 MOVES 5 BYTES OF DATA
                                6697  *
                                6698  *       ENTRY   (HL) = DESTINATION ADDRESS
                                6699  *               (DE) = SOURCE ADDRESS
                                6700  *       EXIT    (HL) = (HL) + COUNT
                                6701  *               (DE) = (DE) + COUNT
                                6702  *       USES    A,F,D,E,H,L
                                6703
                                6704
076.045                         6705  MOV5    EQU     *               ENTRY POINT TO MOVE 4 BYTES
076.045  032                   6706          LDAX    D
076.046  167                   6707          MOV     M,A
076.047  023                   6708          INX     D
076.050  043                   6709          INX     H
076.051  032                   6710  MOV4    LDAX    D
076.052  167                   6711          MOV     M,A
076.053  023                   6712          INX     D
076.054  043                   6713          INX     H
076.055  032                   6714          LDAX    D
076.056  167                   6715          MOV     M,A
076.057  023                   6716          INX     D
076.060  043                   6717          INX     H
076.061  032                   6718          LDAX    D
076.062  167                   6719          MOV     M,A
076.063  023                   6720          INX     D
076.064  043                   6721          INX     H
076.065  032                   6722          LDAX    D
076.066  167                   6723          MOV     M,A
076.067  023                   6724          INX     D
076.070  043                   6725          INX     H
076.071  311                   6726          RET
```

```
                          6728  **      PNT - PREVIEW NEXT TOKEN.
                          6729  *
                          6730  *       PNT READS THE NEXT TEXT TOKEN. HOWEVER, THE TOKEN POINTER
                          6731  *       IS NOT ADVANCED, SO THAT IT CAN BE PREVIEWED OVER
                          6732  *       AND OVER, (AND ACCEPTED ONCE).
                          6733  *
                          6734  *       ENTRY   (BC) = TEXT POINTER
                          6735  *       EXIT    (BC) UPDATED
                          6736  *               (A) = TYPE
                          6737  *               (DE) = CODE (IF VARIABLE)
                          6738  *       USES    A,F (D,E IF VARIABLE)
                          6739
                          6740
076.072  076 000          6741  PNT     MVI     A,0             (A) = TYPE
076.073                   6742  PNTA    EQU     *-1             TYPE OF CURRENT TOKEN
076.074  376 300          6743          CPI     CT.VARL
076.076  332 111 076      6744          JC      PNT2            IS NOT VARIABLE
076.101  376 310          6745          CPI     CT.VARH+1
076.103  322 111 076      6746          JNC     PNT2            IS NOT VARIABLE
076.106  021 000 000      6747          LXI     D,0             (DE) = INDEX
076.107                   6748  PNTB    EQU     *-2
076.111                   6749  PNT2    EQU     *
076.111  000               6750  PNTC    NOP                     'RET' IF VALUE IN PNTA, PNTB
076.112  315 131 054      6751          CALL    LEXCAL
076.115  353               6752          XCHG
076.116  042 107 076      6753          SHLD    PNTB            SET INDEX
076.121  353               6754          XCHG
076.122  062 073 076      6755          STA     PNTA            SET TYPE
076.125  365               6756          PUSH    PSW
076.126  076 311          6757          MVI     A,MI.RET        VALUE IS IN PNTA, PNTB
076.130  062 111 076      6758  PNT1    STA     PNTC            SET FLAG
076.133  361               6759          POP     PSW
076.134  311               6760          RET


                          6762  **      PVI - PERFORM VALUE INPUT.
                          6763  *
                          6764  *       PVI READS A LIST OF VARIABLES FROM THE TEXT AT (BC), AND
                          6765  *       ASSIGNS THEM THE VALUES OF THE EXPRESSIONS AT (HL).
                          6766  *
                          6767  *       ENTRY   (BC) = VARIABLE LIST
                          6768  *               (HL) = TEXT LIST
                          6769  *       EXIT    (BC) UPDATED
                          6770  *               (HL) UPDATED
                          6771  *               'Z' SET IF VARIABLE LIST SATISFIED
                          6772  *       USES    ALL
                          6773
                          6774
076.135  315 072 076      6775  PVI     CALL    PNT             PEEK AT NEXT TOKEN
000.000                   6776          ERRNZ   CT.FIN
076.140  247               6777          ANA     A
076.141  310               6778          RZ                      NO MORE VARIABLES
076.142  326 020          6779          SUI     CT.PAR          SEE IF )
076.144  310               6780          RZ                      NO MORE VARIABLES
```

```
076.145  315 372 111  6781          CALL    $SOB            SKIP BLANKS
076.150  176          6782          MOV     A,M             (A) = NEXT NON-BLANK
076.151  247          6783          ANA     A
076.152  312 236 076  6784          JZ      PVI3            NO DATA
                      6785
                      6786  *       WE KNOW WE HAVE DATA (OR A SPECIFED NULL VALUE)
                      6787
076.155  345          6788          PUSH    H
076.156  315 136 075  6789          CALL    IST             INSERT SYMBOL IN TABLE
076.161  341          6790          POP     H
076.162  365          6791          PUSH    PSW             SAVE TYPE
076.163  315 062 077  6792          CALL    RCE             REQUIRE DELIMITER, CLEAR RNT
076.166  361          6793          POP     PSW             (A) = TYPE OF VARIABLE
076.167  305          6794          PUSH    B
076.170  325          6795          PUSH    D               SAVE INDEX
076.171  365          6796          PUSH    PSW             SAVE TYPE
076.172  104          6797          MOV     B,H
076.173  115          6798          MOV     C,L             (BC) = VALUE LIST ADDRESS
076.174  012          6799          LDAX    B
076.175  376 054      6800          CPI     ','
076.177  302 210 076  6801          JNE     PVI1            IS NOT NULL VALUE
076.202  321          6802          POP     D
076.203  361          6803          POP     PSW             SKIP ASSIGNING VALUE
076.204  003          6804          INX     B
076.205  303 230 076  6805          JMP     PVI2
                      6806
                      6807  *       STORE VALUE.
                      6808
076.210  361          6809  PVI1    POP     PSW             (A) = TYPE
076.211  365          6810          PUSH    PSW
076.212  315 240 076  6811          CALL    PVI5            EVALUATE VALUE
076.215  315 062 077  6812          CALL    RCE             REQUIRE COMMA OR END
076.220  361          6813          POP     PSW             RESTORE TYPE
076.221  321          6814          POP     D               (DE) = VARIABLE POINTER
076.222  315 366 072  6815          CALL    CSA             (DE) = ABS. ADDR. INTO SYMBOL TABLE
076.225  315 202 071  6816          CALL    AVV             ASSIGN VALUE TO VARIABLE
076.230              6817  PVI2    EQU     *
076.230  140          6818          MOV     H,B
076.231  151          6819          MOV     L,C             (BC) = VALUE LIST ADDRESS
076.232  301          6820          POP     B
076.233  303 135 076  6821          JMP     PVI             PROCESS ANOTHER
                      6822
                      6823  *       RAN OUT OF VALUES.
                      6824
076.236  264          6825  PVI3    ORA     H               CLEAR 'Z'
076.237  311          6826          RET
                      6827
                      6828  *       CRACK VALUE.
                      6829  *
                      6830  *       (A) = TYPE OF INPUT VARIABLE
                      6831
000.000               6832          ERRNZ   CF.STR-1        ASSUME 1 BIT FOR STRING
076.240  037          6833  PVI5    RAR
076.241  332 261 076  6834          JC      PVI7            IS STRING VARIABLE
                      6835
                      6836  *       REQUIRE NUMBER.
```

```
                        6837  *       IF NOT VALID NUMBER, *ATF* WONT LEAVE POINTER AT DELIMITER
                        6838
076.244  140            6839  PVI6    MOV     H,B
076.245  151            6840          MOV     L,C
076.246  315 323 107    6841          CALL    ATF             ASCII TO FLOATING
076.251  076 300        6842          MVI     A,CT.SNV
076.253  062 201 042    6843          STA     ACCX-1          SET SCALAR NUMBERIC VALUE
076.256  104            6844          MOV     B,H
076.257  115            6845          MOV     C,L             UPDATE (BC)
076.260  311            6846          RET
                        6847
                        6848  *       MUST BE STRING. IF QUOTES, GOBBLE IT ALL. IF NONE, GO TO COMMA
                        6849
076.261  012            6850  PVI7    LDAX    B               (A) = FIRST DATA CHARACTER
076.262  376 042        6851          CPI     '"'
076.264  003            6852          INX     B               ASSUME HAVE QUOTE
076.265  312 325 076    6853          JE      PVI10           INPUT AS STRING
                        6854
                        6855  *       DOSENT HAVE QUOTES. COPY INTO LINE2, AND ADD THE QUOTES
                        6856
076.270  041 335 113    6857          LXI     H,LINE2
076.273  013            6858          DCX     B               POINT TO 1ST CHARACTER
076.274  012            6859  PVI8    LDAX    B
076.275  167            6860          MOV     M,A
076.276  003            6861          INX     B
076.277  043            6862          INX     H
076.300  012            6863          LDAX    B               CHECK NEXT CHARACTER
076.301  376 054        6864          CPI     ','
076.303  312 312 076    6865          JE      PVI9            GOT THE END
000.000                 6866          ERRNZ   CT.FIN
076.306  247            6867          ANA     A
076.307  302 274 076    6868          JNZ     PVI8            NOT AT END OF LINE
                        6869
                        6870  *       ALL DONE COPYING STRING. ADD CLOSE QUOTE
                        6871
076.312  066 042        6872  PVI9    MVI     M,'"'
076.314  305            6873          PUSH    B               SAVE (BC)
076.315  001 335 113    6874          LXI     B,LINE2
076.320  315 325 076    6875          CALL    PVI10           BUILD STRING
076.323  301            6876          POP     B               RESTORE (BC)
076.324  311            6877          RET
                        6878
076.325  315 015 055    6879  PVI10   CALL    LEX12           READ STRING TYPE
076.330  305            6880          PUSH    B
076.331  001 005 000    6881          LXI     B,5
076.334  033            6882          DCX     D               POINT TO VALUE-1
076.335  041 201 042    6883          LXI     H,ACCX-1
076.340  315 252 030    6884          CALL    $MOVE           MOVE DESCRIPTOR INTO ACCX
076.343  301            6885          POP     B
076.344  311            6886          RET
```

```
                                6888  **      PBO - PRESET BOOLEAN OPERATORS
                                6889  *
                                6890  *       PBO INSURES THAT BOTH VALUES ARE NUMERIC, AND THEN
                                6891  *       FIXES BOTH TO INTEGERS.
                                6892  *
                                6893  *       ENTRY   (ACCX) = VALUE 1
                                6894  *               (ACCY) = VALUE 2
                                6895  *       EXIT    (HL) = IFIX(ACCX)
                                6896  *               (DE) = IFIX(ACCY)
                                6897  *       USES    A,F,D,E,H,L
                                6898
                                6899
076.345  315 177 077           6900  PBO     CALL    RNO             REQUIRE NUMERIC OPERANDS
076.350  315 002 075           6901          CALL    IFIX            (DE) = IFIX(ACCX)
076.353  353                   6902          XCHG
076.354  303 377 074           6903          JMP     IFIX.           (DE) = IFIX(ACCX)


                                6905  **      POPX - POP VALUE INTO *X* ACCUMULATOR.
                                6906  *
                                6907  *       ENTRY   NONE
                                6908  *       EXIT    NONE
                                6909  *       USES    H,L
                                6910
                                6911
076.357  041 201 042           6912  POPX    LXI     H,ACCX-1
076.362  303 373 076           6913          JMP     POP


                                6915  **      POPY - POP VALUE INTO *Y* ACCUMULATOR.
                                6916  *
                                6917  *       ENTRY   NONE
                                6918  *       EXIT    (DE) = *ACCY
                                6919  *       USES    A,F,D,E,H,L
                                6920
                                6921
                                6922  **      POPY. - POP VALUE INTO *Y* ACCUMULATOR.
                                6923  *
                                6924  *       ENTRY   NONE
                                6925  *       EXIT    NONE
                                6926  *       USES    D,E,H,L
                                6927
                                6928
076.365  021 210 042           6929  POPY    LXI     D,ACCY
076.370  041 207 042           6930  POPY.   LXI     H,ACCY-1        STORE AREA
```

```
                                6932  **      POP - POP VALUE FROM WRKTAB.
                                6933  *
                                6934  *       ENTRY   (HL) = ADDRESS OF 5 BYTE AREA
                                6935  *       EXIT    DATA IN AREA
                                6936  *       USES    H,L
                                6937
                                6938
076.373  365                   6939  POP     PUSH    PSW             SAVE PSW
076.374  325                   6940          PUSH    D
076.375  345                   6941          PUSH    H
076.376  052 147 112           6942          LHLD    WRKTAB+MT.LEN
077.001  021 373 377           6943          LXI     D,-5
077.004  031                   6944          DAD     D
077.005  042 147 112           6945          SHLD    WRKTAB+MT.LEN   DECREASE SIZE
077.010  322 117 070           6946          JNC     ERR.DO          SHOULD NOT OCCUR
077.013  353                   6947          XCHG
077.014  052 145 112           6948          LHLD    WRKTAB+MT.FWA
077.017  031                   6949          DAD     D               (HL) = ABS ADDRESS OF 5 BYTES
077.020  353                   6950          XCHG
077.021  341                   6951          POP     H               (HL) = TO
077.022  315 045 076           6952          CALL    MOV5            MOVE DATA
077.025  321                   6953          POP     D
077.026  361                   6954          POP     PSW
077.027  311                   6955          RET


                                6957  **      PSHX - PUSH (ACCX) ONTO STACK.
                                6958  *
                                6959  *       ENTRY   NONE
                                6960  *       USES    A,F,D,E,H,L
                                6961
                                6962
077.030  315 056 071           6963  PSHX.   CALL    ANT             ACCEPT OPERATION
077.033  041 201 042           6964  PSHX    LXI     H,ACCX-1
077.036  303 044 077           6965          JMP     PSH             PUSH ACCX
                                6966
                                6967
                                6968  **      PSHY - PUSH (ACCY) ONTO WORK STACK.
                                6969  *
                                6970  *       ENTRY   NONE
                                6971  *       EXIT    (ACCY) ON STACK
                                6972  *       USES    A,F,D,E,H,L
                                6973
                                6974
077.041  041 207 042           6975  PSHY    LXI     H,ACCY-1
                                6976
                                6977
                                6978  **      PSH - PUSH MEMORY VALUE INTO WORK STAC.
                                6979  *
                                6980  *       ENTRY   (HL) = ADDRESS OF 5 BYTES
                                6981  *       EXIT    ON WRKTAB
                                6982  *       USES    A,F,D,E,H,L
                                6983
                                6984
```

```
077.044  345           6985  PSH     PUSH    H
077.045  041 005 000   6986          LXI     H,5
077.050  021 145 112   6987          LXI     D,WRKTAB+1
077.053  315 026 071   6988          CALL    AMB             ALLOCATE 5 BYTES
077.056  321           6989          POP     D               (DE) = FROM
077.057  303 045 076   6990          JMP     MOV5            COPY AND EXIT


                       6992  **      RCE - REQUIRE COMMA OR END.
                       6993  *
                       6994  *       RCE REQUIRES EITHER A COMMA, OR END OF STATEMENT.
                       6995  *
                       6996  *       ENTRY   (BC) = TEXT POINTER
                       6997  *       EXIT    TO *RET* IF OK
                       6998  *                (BC) UPDATED
                       6999  *                (A) = TYPE CODE
                       7000  *                TO ERR.SY IF NOT ',' OR CT.FIN
                       7001  *       USES    A,F,B,C
                       7002
                       7003
077.062  315 056 071   7004  RCE     CALL    ANT             ACCEPT NEXT TOKEN
077.065  247           7005          ANA     A
000.000                7006          ERRNZ   CT.FIN
077.066  310           7007          RZ                      IS FIN
077.067  376 026       7008          CPI     CT.CMA
077.071  310           7009          RE                      COMMA
077.072  303 152 070   7010          JMP     ERR.SY          SYNTAX ERROR


                       7012  **      RIL - READ INPUT LINE.
                       7013  *
                       7014  *       RIL READS A LINE FROM THE SYSTEM CONSOLE.
                       7015  *
                       7016  *       ENTRY   (HL) = LINE FWA
                       7017  *       EXIT    'C' SETE IF CTL-C STRUCK
                       7018  *                'C' CLEAR IF GOT LINE
                       7019  *                (A) = LINE LENGTH
                       7020  *       USES    A,F,D,E
                       7021
                       7022
077.075  345           7023  RIL     PUSH    H               SAVE START ADDRESS
077.076  072 204 112   7024  RIL1    LDA     CTLFLAG
000.000                7025          ERRNZ   CFCTLC-1
077.101  037           7026          RAR
077.102  332 137 077   7027          JC      RIL3            CTL-C
077.105  377 001       7028          DB      SYSCALL,.SCIN
077.107  332 076 077   7029          JC      RIL1
077.112  376 012       7030          CPI     NL
077.114  302 120 077   7031          JNE     RIL2            NOT END OF LINE
077.117  257           7032          XRA     A               USE 00 AS END OF LINE
077.120  167           7033  RIL2    MOV     M,A
077.121  043           7034          INX     H
```

```
077.122  302 076 077  7035          JNZ     RIL1            MORE TO GO
077.125  074          7036          INR     A               (A) = 1
077.126  062 315 112  7037          STA     COLCNTS         SET CONSOLE COLUMN AT FRONT OF LINE
077.131  321          7038          POP     D               (DE) = LINE FWA
077.132  175          7039          MOV     A,L             (A) = LENGTH OF LINE
077.133  223          7040          SUB     E
077.134  247          7041          ANA     A               CLEAR CARRY
077.135  353          7042          XCHG                    (HL) = LINE FWA
077.136  311          7043          RET
                      7044
                      7045  *       CTL-C HIT
                      7046
077.137  341          7047  RIL3    POP     H
077.140  311          7048          RET



                      7050  **      RLF - READ LINE FROM FILE.
                      7051  *
                      7052  *       RLF READS A LINE FROM THE FILE NUMBER SPECIFIED IN IOCHAN.
                      7053  *       THIS MAY BE THE CONSOLE, OR IT MAY BE A FILE BLOCK.
                      7054  *
                      7055  *       ENTRY   (HL) = LINE ADDRESS
                      7056  *       EXIT    'C' SET IF CTL-C (WAS CONSOLE INPUT)
                      7057  *       USES    A,F,D,E
                      7058
                      7059
077.141  072 202 112  7060  RLF     LDA     IOCHAN
077.144  247          7061          ANA     A
077.145  312 075 077  7062          JZ      RIL             IS CONSOLE, READ LINE
                      7063
                      7064  *       IS FROM FILE
                      7065
077.150  345          7066          PUSH    H               SAVE TEXT FWA
077.151  075          7067          DCR     A
077.152  315 005 072  7068          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
077.155  332 210 070  7069          JC      ERR,FNO         FILE NOT OPEN
077.160  321          7070          POP     D               (DE) = LINE FWA
077.161  305          7071          PUSH    B               SAVE BC
077.162  001 005 001  7072          LXI     B,LINEL+6       MAX.CHAR.TO.READ+6  FOR LINE #  /78.10.GC/
077.165  325          7073          PUSH    D               SAVE LINE FWA
077.166  315 161 101  7074          CALL    $FREAL          READ LINE
077.171  332 213 070  7075          JC      ERR,EOF         EOF ON DEVICE
077.174  341          7076          POP     H               (HL) = LINE FWA
077.175  301          7077          POP     B               RESTORE (BC)
077.176  311          7078          RET
```

```
                                7080  **      RNO - REQUIRE NUMERIC OPERANDS.
                                7081  *
                                7082  *       RNO REQUIRES THAT (ACCX) AND (ACCY) ARE BOTH NUMERIC.
                                7083  *
                                7084  *       ENTRY   NONE
                                7085  *       EXIT    TO *RET* IF NUMERIC
                                7086  *               TO ERR.TE IF NOT
                                7087  *       USES    A,F,H,L
                                7088
                                7089
077.177  315 347 072            7090  RNO     CALL    COT
077.202  310                    7091          RZ                      NUMERIC
077.203  303 155 070            7092          JMP     ERR.TC          TYPE ERROR


                                7094  **      RNP - READ NEW PROGRAM.
                                7095  *
                                7096  *       RNP IS CALLED TO READ A NEW SOURCE PROGRAM INTO THE TEXT TABLE (TXTTAB)
                                7097  *
                                7098  *       ALL TXTTAB DEPENDANT TABLES ARE CLEARED.
                                7099  *
                                7100  *       RNP EXPECTS THAT THE PROPER FILE NAME IS ALREADY INSTALLED
                                7101  *       IN THE FIRST FILE BLOCK IN FILTAB. NOTE THAT AS THE PROGRAM TEXT
                                7102  *       IS INSERTED, FILTAB MAY MOVE BETWEEN LINES.  THUS, RNP RE-LOADS
                                7103  *       THIS ADDRESS BEFORE EVERY OPERATION.
                                7104  *
                                7105  *       ENTRY   NONE
                                7106  *       EXIT    (BC) = #ZERO
                                7107  *       USES    ALL
                                7108
                                7109
077.206  315 320 077            7110  RNP     CALL    SCRA            CLEAR TEXT TABLE
077.211  315 104 073            7111          CALL    CUF             CLEAR USER-DEFINED FUNCTIONS FROM SYMTAB
077.214  315 021 045            7112          CALL    CLR1            CLEAR TEXT TABLE REFERENCES
077.217  315 217 074            7113          CALL    FOP             FILE OPEN PRESET
077.222  041 230 042            7114          LXI     H,FBLIST        (HL) = TABLE FWA
077.225  021 072 043            7115          LXI     D,DEFALTP
077.230  315 021 101            7116          CALL    $FOPER          OPEN FOR READ
077.233  315 115 074            7117          CALL    FOC             RESTORE MEMORY SPACE
077.236  001 005 001            7118  RNP1    LXI     B,LINEL+6
077.241  021 330 112            7119          LXI     D,LINE+1
077.244  041 230 042            7120          LXI     H,FBLIST        (HL) = FB FWA
077.247  325                    7121          PUSH    D               SAVE ADDRESS
077.250  315 161 101            7122          CALL    $FREAL          READ LINE INTO BUFFER+1
077.253  341                    7123          POP     H               (HL) = #BUFFER+1
077.254  332 273 077            7124          JC      RNP2            ALL DONE
077.257  315 373 065            7125          CALL    ICL.            COMPRESS LINE INTO BUFFER+0
077.262  332 273 077            7126          JC      RNP2            CTL-C HIT
077.265  315 270 070            7127          CALL    MTL             MERGE TEXT LINEE
077.270  303 236 077            7128          JMP     RNP1            GET NEXT
                                7129
                                7130  *       END OF TEXT
                                7131
077.273  041 230 042            7132  RNP2    LXI     H,FBLIST        (HL) = FB FWA
```

```
077.276  315 335 102  7133          CALL    $FCLO           CLOSE INPUT FILE
077.301  001 007 115  7134          LXI     B,ZERO          (BC) = TEXT POINTER = NO MORE
077.304  311          7135          RET                     EXIT


                      7137  **      RNT - REQUIRE NEXT TOKEN.
                      7138  *
                      7139  *       RNT CHECSK TO SEE IF THE NEXT TOKEN IS THE REQUIRED VALUE,
                      7140  *       AND FLAGS A SYNTAX ERROR IF NOTL
                      7141  *
                      7142  *       ENTRY   (RET) = REQUIRED TOKEN
                      7143  *       EXIT    TO (RET)+1 IF MATCH
                      7144  *                (DE) = SYMBOL POINTER (IF VARIABLE)
                      7145  *                (A) = VARIABLE TYPE
                      7146  *               TO ERR.SY IF NOT
                      7147  *       USES    A,F, (DE, IF VARIABLE)
                      7148
                      7149
077.305  315 056 071  7150  RNT     CALL    ANT             ACCEPT NEXT TOKEN
077.310  343          7151          XTHL
077.311  276          7152          CMP     M
077.312  043          7153          INX     H
077.313  343          7154          XTHL
077.314  310          7155          RE                      OK
077.315  303 152 070  7156          JMP     ERR.SY          NO GOOD


                      7158  **      SCRA - SCRATCH TEXT BUFFER
                      7159  *
                      7160  *       SCRA INSERTS THE DUMMY LAST LINE
                      7161  *       AT THE BEGINNING OF THE BUFFER
                      7162  *
                      7163  *       ENTRY   NONE
                      7164  *       EXIT    (BC) = #0 BYTE
                      7165  *       USES     A,F,B,C,H,L
                      7166
                      7167
077.320  041 003 000  7168  SCRA    LXI     H,3             SCRATCH STORE
077.323  042 123 112  7169          SHLD    TXTTAB+MT.LEN           LENGTH = 3
077.326  041 377 377  7170          LXI     H,377377A
077.331  042 010 115  7171          SHLD    MTAREA
077.334  001 012 115  7172          LXI     B,MTAREA+2
077.337  257          7173          XRA     A
077.340  002          7174          STAX    B               CLEAR TEXT, SET ((BC)) = 0
                      7175
077.341  311          7176          RET                     EXIT
```

```
                                7178  **      SES - SKIP TO END OF STATEMENT.
                                7179  *
                                7180  *       SES SKIPS OVER TEXT UNTIL AN END-OF-LINE IS DETECTED
                                7181  *
                                7182  *       ENTRY   (BC) = TEXT POINTER
                                7183  *       EXIT    (BC) UPATED
                                7184  *       USES    A,F,B,C
                                7185
                                7186
077.342  315 056 071            7187  SES     CALL    ANT             ACCEPT NEXT TOKEN
000.000                         7188          ERRNZ   CT.FIN
077.345  247                    7189          ANA     A
077.346  302 342 077            7190          JNZ     SES             NOT YET
077.351  311                    7191          RET


                                7193  **      SFS - SEARCH 'FOR' STACK.
                                7194  *
                                7195  *       SFS SEARCHES A FOR STACK FOR AN ENTRY MATCHEDING A
                                7196  *       SUPPLIED ONE.
                                7197  *
                                7198  *
                                7199  *       ENTRY   (DE) = INDEX VARIABLE INDEX
                                7200  *
                                7201  *       EXIT    'Z' SET IF FOUND
                                7202  *               (HL) = INDEX OF IND+2 (IF ANY)
                                7203  *               (DE) = ABS. ADDRESS OF INDEX IN SYMTAB
                                7204  *
                                7205  *       USES    A,F,H,L
                                7206  *
                                7207
077.352  315 072 076            7208  SFS.    CALL    PNT             ALLOW NULL AS THE LAST IN STACK
000.000                         7209          ERRNZ   CT.FIN
077.355  127                    7210          MOV     D,A             ASSUME (A) = 0 = CT.FIN
077.356  137                    7211          MOV     E,A
077.357  312 375 077            7212          JZ      SFS0            (DE) = 0 = INDEX
                                7213
077.362  315 136 075            7214  SFS     CALL    IST             INSERT SYMBOL IN TABLE
077.365  376 300                7215          CPI     CT.SNV
077.367  302 152 070            7216          JNE     ERR.SY          MUST BE SCALAR NUMERIC VARIABLE
077.372  315 366 072            7217          CALL    CSA             DE = ABS. SYMTAB ADDRESS        /80.01.GC/
                                7218
077.375  305                    7219  SFS0    PUSH    B               SAVE TEXT POINTER
077.376  052 135 112            7220          LHLD    FORTAB+MT.LEN
100.001  104                    7221          MOV     B,H
100.002  115                    7222          MOV     C,L
100.003  052 133 112            7223          LHLD    FORTAB+MT.FWA   (HL) = TABLE FWA
100.006  345                    7224          PUSH    H
100.007  011                    7225          DAD     B               (HL) = LWA+1
                                7226
100.010  172                    7227          MOV     A,D                                     /80.01.GC/
100.011  263                    7228          ORA     E                                       /80.01.GC/
100.012  312 025 100            7229          JZ      SFS1                                    /80.01.GC/
                                7230
```

```
100.015 353          7231          XCHG                                                  /80.01.GC/
100.016 053          7232          DCX     H                                             /80.01.GC/
100.017 053          7233          DCX     H                                             /80.01.GC/
100.020 176          7234          MOV     A,M                                           /80.01.GC/
100.021 043          7235          INX     H                                             /80.01.GC/
100.022 156          7236          MOV     L,M                                           /80.01.GC/
100.023 147          7237          MOV     H,A                                           /80.01.GC/
100.024 353          7238          XCHG                  DE = SYMBOL KEY                 /80.01.GC/
                     7239
100.025 170          7240  SFS1    MOV     A,B
100.026 261          7241          ORA     C             CHECK COUNT
100.027 312 073 100  7242          JZ      SFS3          NOT FOUND
100.032 305          7243          PUSH    B
100.033 001 365 377  7244          LXI     B,-11
100.036 011          7245          DAD     B             (HL) = ADDRESS OF LAST ELEMENT
100.037 301          7246          POP     B
100.040 172          7247          MOV     A,D
100.041 263          7248          ORA     E
100.042 176          7249          MOV     A,M
100.043 053          7250          DCX     H
100.044 312 120 100  7251          JZ      SFS6          WILL TAKE THE LAST
100.047 273          7252          CMP     E             SEE IF FOUND                    /80.01.GC/
100.050 302 060 100  7253          JNE     SFS2          NOT FOUND
100.053 176          7254          MOV     A,M
100.054 272          7255          CMP     D                                             /80.01.GC/
100.055 312 074 100  7256          JE      SFS4          FOUND
100.060 171          7257  SFS2    MOV     A,C
100.061 326 014      7258          SUI     12
100.063 117          7259          MOV     C,A
100.064 170          7260          MOV     A,B           COUNT = COUNT-12
100.065 336 000      7261          SBI     0
100.067 107          7262          MOV     B,A
100.070 303 025 100  7263          JMP     SFS1          TRY AGAIN
                     7264
                     7265  *       NOT FOUND.
                     7266
100.073 264          7267  SFS3    ORA     H
                     7268
                     7269  *       FOUND. COMPUTE FORTAB INDEX FROM ABS.
                     7270
100.074 043          7271  SFS4    INX     H
100.075 043          7272  SFS5    INX     H
100.076 104          7273          MOV     B,H
100.077 115          7274          MOV     C,L           (BC) = ABS ADDRESS
100.100 341          7275          POP     H             (HL) = FORTAB FWA
100.101 365          7276          PUSH    PSW           SAVE CODE
100.102 171          7277          MOV     A,C
100.103 225          7278          SUB     L
100.104 157          7279          MOV     L,A
100.105 170          7280          MOV     A,B
100.106 234          7281          SBB     H
100.107 147          7282          MOV     H,A
100.110 315 323 075  7283          CALL    LVS           GET AN ABS. ADDRESS             /80.01.GC/
100.113 023          7284          INX     D                                             /80.01.GC/
100.114 023          7285          INX     D                                             /80.01.GC/
100.115 361          7286          POP     PSW           RESTORE CODE
```

```
100.116  301           7287          POP     B               RESTORE (BC)
100.117  311           7288          RET
                       7289
                       7290  *       NO VARIABLE SUPPLIED, JUST TAKE THE LAST ONE.
                       7291
100.120  126           7292  SFS6    MOV     B,M                                     /80.01.GC/
100.121  043           7293          INX     H
100.122  136           7294          MOV     E,M             (DE) = VARIABLE INDEX   /80.01.GC/
100.123  303 075 100   7295          JMP     SFS5



                       7297  **      SOB - SKIP OVER BLANKS.
                       7298  *
                       7299  *       ENTRY   (BC) = TEXT POINTER
                       7300  *       EXIT    (BC) = ADDRESS OF NEXT NON-BLANK CHARACTER
                       7301  *       USES    A,F,B,C
                       7302
                       7303
100.126  012           7304  SOB     LDAX    B
100.127  376 040       7305          CPI     ' '
100.131  312 137 100   7306          JE      SOB1            IS BLANK
100.134  376 011       7307          CPI     TAB
100.136  300           7308          RNE                     NOT TAB, EITHER
100.137  003           7309  SOB1    INX     B
100.140  303 126 100   7310          JMP     SOB



                       7312  **      SRA - STACK RETURN ADDRESS.
                       7313  *
                       7314  *       SRA STACKS THE TEXT RETURN ADDRESS (END OF CURRENT STATEMENT)
                       7315  *       AND THE CURRENT LINE NUMBER ON STACK 'GOSTAB'.
                       7316  *
                       7317  *       ENTRY   (BC) = TEXT POINTER
                       7318  *       EXIT    (BC) UNCHANGED.
                       7319  *       USES    A,F,D,E
                       7320
                       7321
100.143  305           7322  SRA     PUSH    B               SAVE TEXT ADDRESS
100.144  345           7323          PUSH    H               SAVE (HL)
100.145  315 342 077   7324          CALL    SES             SKIP TO END OF STATEMENT
100.150  041 004 000   7325          LXI     H,4
100.153  021 140 112   7326          LXI     D,GOSTAB+1
100.156  315 026 071   7327          CALL    AMB             ALLOCATE ROOM
100.161  353           7328          XCHG
100.162  052 175 112   7329          LHLD    CURNUM          (HL) = CURRENT LINE NUMBER
100.165  353           7330          XCHG
100.166  161           7331          MOV     M,C             SAVE RETURN ADDRESS
100.167  043           7332          INX     H
100.170  160           7333          MOV     M,B
100.171  043           7334          INX     H
100.172  163           7335          MOV     M,E             SET CURNUM
100.173  043           7336          INX     H
```

```
100.174  162            7337          MOV     M,D
100.175  341            7338          POP     H
100.176  301            7339          POP     B               RESTORE (BC)
100.177  311            7340          RET


                        7342  **      TCS - TYPE CHARACTER STRING.
                        7343  *
                        7344  *       TCS TYPES A CHARACTER STRING ON THE CONSOLE.
                        7345  *
                        7346  *       ENTRY   (DE) = STRING DESCRIPTOR
                        7347  *       EXIT    TYPED
                        7348  *       USES    A,F,D,E,H,L
                        7349
                        7350
100.200                 7351  TCS     EQU     *
100.200  315 315 074    7352          CALL    FSE             FIND STRINGTAB ENTRY
100.203  303 251 100    7353          JMP     WLF.            WRITE LINE TO FILE


                        7355  **      TDI - TYPE DECIMAL INTEGER.
                        7356  *
                        7357  *       TDI TYPES AN INTEGER AS A 5 PLACE NUMBER. LEADING ZEROS ARE
                        7358  *       SUPPRESSED.
                        7359  *
                        7360  *       ENTRY   (DE) = NUMBER
                        7361  *       EXIT    TYPED
                        7362  *       USES    A,F,D,E
                        7363
                        7364
100.206  345            7365  TDI     PUSH    H
100.207  315 040 075    7366          CALL    IFLT            FLOAT IT
100.212  041 335 113    7367          LXI     H,LINE2
100.215  315 301 110    7368          CALL    FTA             FLOATING TO ASCII
100.220  315 217 103    7369          CALL    $TYPCC          TYPE NUMBER MINUS .
100.223  341            7370          POP     H
100.224  311            7371          RET


                        7373  **      WEL - WRITE END OF LINE.
                        7374  *
                        7375  *       WEL WRITES AN END OF LINE CHARACTER TO THE CURRENT OUTPUT FILE.
                        7376  *
                        7377  *       ENTRY   NONE
                        7378  *       EXIT    NONE
                        7379  *       USES    A,F,D,E,H,L
                        7380
                        7381
100.225  315 302 075    7382  WEL     CALL    LCC
100.230  257            7383          XRA     A
```

```
100.231  167            7384          MOV     M,A             SET COLUMN # = 1(-1 FOR THE NL CHARACTER)
100.232  041 241 100    7385          LXI     H,WELA
100.235  074            7386          INR     A               (A) = 1
100.236  303 251 100    7387          JMP     WLF.            WRITE CHARACTER AND EXIT
                        7388
100.241  012            7389  WELA    DB      NL


                        7391  **      WLF - WRITE LINE TO FILE.
                        7392  *
                        7393  *       WLF WRITES A LINE IN "LINE2" TO THE INDICATED (IOCHAN) FILE.
                        7394  *       THIS MAY BE THE SYSTEM CONSOLE, AND THE "LINE" MAY NOT
                        7395  *       BE A COMPLETE LINE (I.E., HAVE NO NL CHARACTER)
                        7396  *
                        7397  *       ENTRY   (LINE2) = TEXT
                        7398  *       EXIT    NONE
                        7399  *       USES    A,F,D,E,H,L
                        7400
                        7401
100.242  041 335 113    7402  WLF     LXI     H,LINE2
100.245  315 335 111    7403          CALL    $CLL            COMPUTE LINE LENGTH
100.250  075            7404          DCR     A               REMOVE 00 BYTE COUNT
                        7405
                        7406  **      WLF. - WRITE LINE TO FILE.
                        7407  *
                        7408  *       ENTRY   (A) = LINE LENGTH
                        7409  *               (HL) = LINE FWA
                        7410  *       EXIT    NONE
                        7411  *       USES    A,F,D,E,H,L
                        7412
                        7413
100.251  305            7414  WLF.    PUSH    B               SAVE (BC)
100.252  117            7415          MOV     C,A
100.253  006 000        7416          MVI     B,0             (BC) = COUNT
100.255  353            7417          XCHG                    (DE) = TEXT ADDRESS
100.256  315 302 075    7418          CALL    LCC             LOCATE COLCNT FOR THIS CHANNEL
100.261  176            7419          MOV     A,M
100.262  201            7420          ADD     C
100.263  167            7421          MOV     M,A             UPDATE COLUME NUMBER
100.264  353            7422          XCHG                    (HL) = TEXT ADDRESS
100.265  072 202 112    7423          LDA     IOCHAN
100.270  247            7424          ANA     A
100.271  302 301 100    7425          JNZ     WLF2            WRITE TO DISK
                        7426
                        7427  *       TO CONSOLE
                        7428
100.274  171            7429          MOV     A,C
100.275  301            7430          POP     B               RESTORE (BC)
100.276  303 217 103    7431          JMP     $TYPCC          WRITE TO CONSOLE AND EXIT
                        7432
                        7433  *       WRITE TO DISK BUFFER, LOCATE FILE BLOCK
                        7434
100.301  345            7435  WLF2    PUSH    H               SAVE TEXT FWA
100.302  075            7436          DCR     A               (A) = FILE BLOCK NUMBER
```

```
100.303  315 005 072  7437          CALL    CFA             COMPUTE FILE BLOCK ADDRESS
100.306  332 210 070  7438          JC      ERR.FNO         FILE NOT OPEN
100.311  321          7439          POP     D               (DE) = DATA FWA
100.312  315 047 102  7440          CALL    $FWRIB          WRITE DATA TO BUFFER
100.315  301          7441          POP     B               RESTORE (BC)
100.316  311          7442          RET


                      7444  **      XCY - EXCHANGE (ACCX) WITH (ACCY)
                      7445  *
                      7446  *       ENTRY   NONE
                      7447  *       EXIT    NONE
                      7448  *       USES    A,F
                      7449
                      7450
100.317  305          7451  XCY     PUSH    B
100.320  325          7452          PUSH    D
100.321  345          7453          PUSH    H
100.322  021 201 042  7454          LXI     D,ACCX-1
100.325  041 207 042  7455          LXI     H,ACCY-1
100.330  016 005      7456          MVI     C,5             (C) = BYTE COUNT
100.332  032          7457  XCY1    LDAX    D
100.333  106          7458          MOV     B,M
100.334  167          7459          MOV     M,A             (A) = NEXT BYTE IN LIST
100.335  170          7460          MOV     A,B
100.336  022          7461          STAX    D
100.337  023          7462          INX     D
100.340  043          7463          INX     H
100.341  015          7464          DCR     C
100.342  302 332 100  7465          JNZ     XCY1            MORE TO GO
100.345  341          7466          POP     H
100.346  321          7467          POP     D
100.347  301          7468          POP     B
100.350  311          7469          RET


                      7471  **      ZRO - ZERO MEMORY.
                      7472  *
                      7473  *       ZRO ZEROS A FIELD OF MEMORY.
                      7474  *
                      7475  *       ENTRY   (HL) = ADDRESS
                      7476  *               (DE) = COUNT
                      7477  *       EXIT    NONE
                      7478  *       USES    A,F,D,E,H,L
                      7479
                      7480
100.351  172          7481  ZRO     MOV     A,D
100.352  263          7482          ORA     E
100.353  310          7483          RZ                      DONE
100.354  066 000      7484          MVI     M,0
100.356  043          7485          INX     H
100.357  033          7486          DCX     D
```

```
100.360  303 351 100  7487          JMP     ZRO
```

```
                                7490  ***     THE FOLLOWING SUBROUTINES ARE ENVOKED BY INTERRUPTS.
                                7491  *
                                7492

                                7494  **      CCINT - PROCESS CTL-C INTERRUPT.
                                7495  *
                                7496  *       ENTRY   NONE
                                7497  *       EXIT    TO CALLER (INTERRUPT)
                                7498  *       USES    A,F
                                7499
                                7500
100.363  076 001               7501  CCINT   MVI     A,CFCTLC
100.365  303 372 100           7502          JMP     CBINT1          PROCESS AS CTL-B


                                7504  **      CBINT - CONTROL-B INTERRUPT.
                                7505  *
                                7506  *       ENTRY   NONE
                                7507  *       EXIT    NONE
                                7508  *       USES    A,F
                                7509
                                7510
100.370  076 002               7511  CBINT   MVI     A,CFCTLB
100.372  345                   7512  CBINT1  PUSH    H
100.373  365                   7513          PUSH    PSW             SAVE FLAG BIT
100.374  315 136 031           7514          CALL    $TYPTX
100.377  336                   7515          DB      '^'+200Q
101.000  361                   7516          POP     PSW
101.001  365                   7517          PUSH    PSW             (A) = CODE
000.000                        7518          ERRNZ   CFCTLB-2        02 IF CTL-B
000.000                        7519          ERRNZ   CFCTLC-1        01 IF CTL-C
101.002  366 002               7520          ORI     2               =3 IF CTLC, =2 IF CTL-B
101.004  306 100               7521          ADI     '@'
101.006  315 241 103           7522          CALL    $WCHAR          ECHO CHARACTER
101.011  361                   7523          POP     PSW             (A) = CODE
101.012  041 204 112           7524          LXI     H,CTLFLAG
101.015  266                   7525          ORA     M
101.016  167                   7526          MOV     M,A
101.017  341                   7527          POP     H
101.020  311                   7528          RET                     RETURN
```

```
101.021                  7531           XTEXT   FOPE


                         7533X **       $FOPEx - OPEN FILE BLOCK FOR I/O
                         7534X *
                         7535X *        $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                         7536X *        FILE BLOCK.  $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                         7537X *        THE FILE VIA *HDOS*.
                         7538X *
                         7539X *        ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                         7540X *                (HL) = ADDRESS OF FILE BLOCK
                         7541X *        EXIT    TO $FERROR IF ERROR
                         7542X *                TO CALLER IF OK
                         7543X *        USES    A,F,B,C,D,E
                         7544X
                         7545X
101.021  315 046 101    7546X $FOPER   CALL    $FOPER.
101.024  320            7547X          RNC
101.025  303 223 070    7548X          JMP     $FERROR         IN ERROR
                         7549X
101.030  315 051 101    7550X $FOPEW   CALL    $FOPEW.
101.033  320            7551X          RNC
101.034  303 223 070    7552X          JMP     $FERROR         IN ERROR
                         7553X
101.037  315 054 101    7554X $FOPEU   CALL    $FOPEU.
101.042  320            7555X          RNC
101.043  303 223 070    7556X          JMP     $FERROR         IN ERROR
                         7557X
                         7558X
101.046  076 002        7559X $FOPER.  MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
101.050  001            7560X          DB      001Q            LXI,B TO SKIP NEXT MVI
101.051  076 004        7561X $FOPEW.  MVI     A,FT.OW         OPEN FOR WRITE
101.053  001            7562X          DB      001Q            LXI,B TO SKIP NEXT MIV
101.054  076 006        7563X $FOPEU.  MVI     A,FT.OR+FT.OW
                         7564X
                         7565X *        (A) = FILE FLAGS
                         7566X
101.056  345            7567X          PUSH    H               SAVE FILE BLOCK ADDRESS
101.057  365            7568X          PUSH    PSW             SAVE NEW FLAGS
000.000                  7569X          ERRNZ   FB.CHA
101.060  106            7570X          MOV     B,M             (B) = CHANNEL NUMBER
101.061  305            7571X          PUSH    B               SAVE HANNEL NUMBER
000.000                  7572X          ERRNZ   FB.FLG-FB.CHA-1
101.062  043            7573X          INX     H
101.063  117            7574X          MOV     C,A             (C) = NEW FILE FLAGS
101.064  176            7575X          MOV     A,M             (A) = CURRENT TYPE
101.065  247            7576X          ANA     A
101.066  171            7577X          MOV     A,C             (A) = NEW FLAGS TO BE SET
101.067  312 101 101     7578X          JZ      $FOPE1          NOT ALREADY OPEN
                         7579X
                         7580X *        ALREADY OPEN.  SQUACK
                         7581X
101.072  301            7582X          POP     B               RESTORE (BC)
101.073  361            7583X          POP     PSW             DISCARD NEW FLAGS
```

```
101.074  341           7584X          POP     H               (HL) = FB ADDRESS
101.075  076 031       7585X          MVI     A,EC.FAO        FILE ALREADY OPEN
101.077  067           7586X          STC
101.100  311           7587X          RET
                       7588X
000.000                7589X          ERRNZ   FB.FWA-FB.FLG-1
101.101  043           7590X $FOPE1   INX     H               (HL) = $FB.FWA
101.102  116           7591X          MOV     C,M
101.103  043           7592X          INX     H
101.104  106           7593X          MOV     B,M             (BC) = FB.FWA
101.105  043           7594X          INX     H
000.000                7595X          ERRNZ   FB.PTR-FB.FWA-2
101.106  161           7596X          MOV     M,C             SET FB.PTR = FB.FWA
101.107  043           7597X          INX     H
101.110  160           7598X          MOV     M,B
101.111  043           7599X          INX     H
000.000                7600X          ERRNZ   FB.LIM-FB.PTR-2
101.112  161           7601X          MOV     M,C             SET FB.LIM = FB.FWA
101.113  043           7602X          INX     H
101.114  160           7603X          MOV     M,B
101.115  043           7604X          INX     H
000.000                7605X          ERRNZ   FB.NAM-FB.LIM-4
101.116  043           7606X          INX     H
101.117  043           7607X          INX     H               (HL) = $FB.NAM
                       7608X
                       7609X *        FILE BLOCK POINTERS SETUP, OPEN FILE
                       7610X
101.120  345           7611X          PUSH    H               SAVE NEW ADDRESS FOR NAME
101.121  041 152 101   7612X          LXI     H,$FOPEB
101.124  247           7613X          ANA     A                                       /78.10.GC/
101.125  312 134 101   7614X          JZ      $FOPE2
000.000                7615X          ERRNZ   .EXIT
101.130  315 033 112   7616X          CALL    $TBLS           FIND CODE
101.133  176           7617X          MOV     A,M
101.134  062 142 101   7618X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
101.137  341           7619X          POP     H               (HL) = $FB.NAM
101.140  361           7620X          POP     PSW             (A) = CHANNEL NUMBER
101.141  377 000       7621X          DB      SYSCALL,.EXIT
101.142                7622X $FOPEA   EQU     *-1             SYSCALL CODE
101.143  321           7623X          POP     D               (D) = NEW FLAG
101.144  341           7624X          POP     H               (HL) = FILE BLOCK ADDRESS
101.145  330           7625X          RC                      EXIT IF ERROR
101.146  043           7626X          INX     H
000.000                7627X          ERRNZ   FB.FLG-1
101.147  162           7628X          MOV     M,D             SET NEW FLAGS
101.150  053           7629X          DCX     H               RESTORE (HL)
101.151  311           7630X          RET
                       7631X
101.152  002 042       7632X $FOPEB   DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
101.154  004 043       7633X          DB      FT.OW,.OPENW
101.156  006 044       7634X          DB      FT.OR+FT.OW,.OPENU
101.160  000           7635X          DB      0               SHOULD NOT OCCUR
101.161                7636           XTEXT   FREAL
```

```
                                7638X **      $FREAL - READ BYTES FROM FILE BUFFER.
                                7639X *
                                7640X *       $FREAL IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                                7641X *
                                7642X *       ENTRY   (BC) = BYTE COUNT
                                7643X *               (DE) = FWA FOR BYTES
                                7644X *               (HL) = ADDRESS OF FILE BUFFER
                                7645X *       EXIT    TO *FERROR* IF ERROR
                                7646X *               TO CALLER IF OK
                                7647X *                (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
                                7648X *                (DE) = ADDRESS OF FIRST UNUSED BYTE
                                7649X *                'C' SET IF EOF DURING READ
                                7650X *       USES    A,F,B,C,D,E
                                7651X
                                7652X
      101.161  315 174 101      7653X $FREAL  CALL    $FREAL.
      101.164  320              7654X         RNC                       RETURN IF OK
      101.165  376 001          7655X         CPI     EC.EOF
      101.167  302 223 070      7656X         JNE     $FERROR           ERROR IS NOT EOF
      101.172  067              7657X         STC
      101.173  311              7658X         RET                       ERROR IS SIMPLY EOF
                                7659X
                                7660X
      101.174                   7661X $FREAL. EQU     *
      101.174  013              7662X         DCX     B                 (BC) = COUNT NOT ENCLUDING 00 BYTE
      101.175  257              7663X         XRA     A
      101.176  062 216 103      7664X         STA     EOFFLG            CLEAR EOF FLAG
      101.201  345              7665X         PUSH    H
      101.202  315 042 103      7666X         CALL    CBT               COPY BUFFER POINTERS TO TEMP CELLS
                                7667X
                                7668X *       COPY DATA FROM BUFFER TO TARGET
                                7669X
      101.205  325              7670X $REAL2  PUSH    D                 SAVE TARGET ADDRESS
      101.206  072 205 103      7671X         LDA     T.FLG
      101.211  346 002          7672X         ANI     FT.OR
      101.213  076 011          7673X         MVI     A,EC.FNO
      101.215  067              7674X         STC                       ASSUME FILE NOT OPEN
      101.216  312 352 101      7675X         JZ      $REAL8            ERROR
      101.221  170              7676X         MOV     A,B
      101.222  261              7677X         ORA     C
      101.223  312 352 101      7678X         JZ      $REAL8            ALL DONE
                                7679X
                                7680X *       COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                                7681X
      101.226  052 210 103      7682X $REAL3  LHLD    T.PTR
      101.231  353              7683X         XCHG                      (DE) = (FB.PTR) = ADDRESS OF DATA
      101.232  052 212 103      7684X         LHLD    T.LIM             (HL) = LIMIT ADDRESS
      101.235  175              7685X         MOV     A,L
      101.236  223              7686X         SUB     E
      101.237  157              7687X         MOV     L,A
      101.240  174              7688X         MOV     A,H
      101.241  232              7689X         SBB     D
      101.242  147              7690X         MOV     H,A               (HL) = NUMBER OF BYTES IN BUFFER
      101.243  171              7691X         MOV     A,C
      101.244  225              7692X         SUB     L                 COMPARE TO REQUESTED COUNT
      101.245  170              7693X         MOV     A,B
```

```
101.246  234           7694X          SBB     H
101.247  322 254 101   7695X          JNC     $REAL4          LESS THAN REQUESTED COUNT
101.252  140           7696X          MOV     H,B
101.253  151           7697X          MOV     L,C             DONT TRANSFER MORE THAN LIMIT
101.254  174           7698X $REAL4   MOV     A,H
101.255  265           7699X          ORA     L
101.256  302 272 101   7700X          JNZ     $REAL6          SOME IN BUFFER
                       7701X
                       7702X *        BUFFER IS EMPTY. RE-FILL IT
                       7703X
101.261  315 122 103   7704X          CALL    $FFB            FILL FILE BUFFER
101.264  332 352 101   7705X          JC      $REAL8          ERROR CONDITION
101.267  303 226 101   7706X          JMP     $REAL3          COUNT THE DATA
                       7707X
                       7708X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                       7709X *
                       7710X *        (BC) = LIMIT COUNT
                       7711X *        (DE) = FROM
                       7712X *        (HL) = COUNT
                       7713X *        ((SP)) = TO
                       7714X
101.272  171           7715X $REAL6   MOV     A,C
101.273  225           7716X          SUB     L
101.274  117           7717X          MOV     C,A
101.275  170           7718X          MOV     A,B
101.276  234           7719X          SBB     H
101.277  107           7720X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
101.300  305           7721X          PUSH    B
101.301  343           7722X          XTHL                    (HL) = REMAINING REQUEST COUNT
101.302  301           7723X          POP     B               (BC) = COUNT FOR THIS COPY
101.303  343           7724X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
101.304  032           7725X $REAL7   LDAX    D
101.305  023           7726X          INX     D
101.306  167           7727X          MOV     M,A
101.307  043           7728X          INX     H
101.310  247           7729X          ANA     A               SEE IF 00 BYTE
101.311  302 320 101   7730X          JNZ     $REL7.3         NOT 00
                       7731X
                       7732X *        IS 00 BYTE. IGNORE IT
                       7733X
101.314  343           7734X          XTHL
101.315  043           7735X          INX     H               ADD ONE TO UNREQUITED COUNT
101.316  343           7736X          XTHL
101.317  053           7737X          DCX     H               BACKSPACE OVER CHARACTER
101.320  013           7738X $REL7.3  DCX     B
101.321  376 012       7739X          CPI     NL
101.323  312 343 101   7740X          JE      $REL7.5         IS END OF LINE
101.326  170           7741X          MOV     A,B
101.327  261           7742X          ORA     C
101.330  302 304 101   7743X          JNZ     $REAL7          MORE TO GO
101.333  353           7744X          XCHG
101.334  042 210 103   7745X          SHLD    T.PTR           UPDATE POINTER
101.337  301           7746X          POP     B               (BC) = REMAINING COUNT
101.340  303 205 101   7747X          JMP     $REAL2          SEE IF MORE IN BUFFER
                       7748X
                       7749X *        END OF CODED LINE
```

```
                                7750X
      101.343  353               7751X $REL7.5 XCHG
      101.344  033               7752X         DCX     D               BACK OVER NL CHARACTER
      101.345  042 210 103       7753X         SHLD    T.PTR           UPDATE POINTER
      101.350  301               7754X         POP     B               (BC) = REMAINING COUNT
      101.351  325               7755X         PUSH    D               SAVE TARGET LWA
                                7756X
                                7757X *       READ COMPLETE.
                                7758X *
                                7759X *       (PSW) = COMPLETION FLAGS
                                7760X
      101.352  321               7761X $REAL8  POP     D               RESTORE TARGET ADDRESS
      101.353  365               7762X         PUSH    PSW             SAVE RETURN CODE
      101.354  257               7763X         XRA     A
      101.355  022               7764X         STAX    D               FLAG END OF LINE
      101.356  361               7765X         POP     PSW             RESTORE RESULT FLAGS
      101.357  023               7766X         INX     D               POINT TO NEXT FREE
      101.360  341               7767X $REAL9  POP     H
      101.361  303 070 103       7768X         JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
      101.364                   7769          XTEXT   FREAO


                                7771X **      $FREAD - READ ONE BYTE FROM FILE BUFFER.
                                7772X *
                                7773X *       $FREAD IS CALLED TO READ ONE BYTE FROM A FILE BUFFER.
                                7774X *
                                7775X *       ENTRY   (HL) = ADDRESS OF FILE BUFFER
                                7776X *       EXIT    TO *FERROR* IF ERROR
                                7777X *                 TO CALLER IF OK
                                7778X *                  (A) = CHARACTER
                                7779X *                 "C" SET IF EOF DURING READ
                                7780X *       USES    A,F,B,C,D,E
                                7781X
                                7782X
      101.364  315 377 101       7783X $FREAD  CALL    $FREAO;
      101.367  320               7784X         RNC                     RETURN IF OK
      101.370  376 001           7785X         CPI     EC:EOF
      101.372  302 223 070       7786X         JNE     $FERROR         ERROR IS NOT EOF
      101.375  067               7787X         STC
      101.376  311               7788X         RET                     ERROR IS SIMPLY EOF
                                7789X
                                7790X
      101.377                   7791X $FREAO. EQU     *
      101.377  257               7792X         XRA     A
      102.000  062 216 103       7793X         STA     EOFFLG          CLEAR EOF FLAG
      102.003  345               7794X         PUSH    H
      102.004  315 042 103       7795X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
      102.007  052 212 103       7796X $READ1  LHLD    T.LIM
      102.012  353               7797X         XCHG
      102.013  052 210 103       7798X         LHLD    T.PTR
      102.016  315 216 030       7799X         CALL    $CDEHL          SEE IF ANY TO READ
      102.021  302 035 102       7800X         JNE     $READ2          GOT DATA
      102.024  315 122 103       7801X         CALL    $FFB            FILL FILE BUFFER
      102.027  332 043 102       7802X         JC      $READ8          ERROR CONDITION
```

```
102.032  303 007 102  7803X           JMP     $READ1          TRY AGAIN
                      7804X
102.035  176          7805X $READ2    MOV     A,M             (A) = CHARACTER
102.036  247          7806X           ANA     A               CLEAR CARRY
102.037  043          7807X           INX     H
102.040  042 210 103  7808X           SHLD    T.PTR
                      7809X
                      7810X *         READ COMPLETE
                      7811X *
                      7812X *         (PSW) = COMPLETION FLAGS
                      7813X
102.043  341          7814X $READ8    POP     H
102.044  303 070 103  7815X           JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT
102.047               7816            XTEXT   FWRIB



                      7818X **        $FWRIB - WRITE BYTES FROM FILE BUFFER.
                      7819X *
                      7820X *         $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                      7821X *
                      7822X *         ENTRY   (BC) = BYTE COUNT
                      7823X *                 (DE) = FWA FOR BYTES
                      7824X *                 (HL) = ADDRESS OF FILE BUFFER
                      7825X *         EXIT    TO *FERROR* IF ERROR
                      7826X *                 TO CALLER IF OK
                      7827X *                 (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                      7828X *         USES    A,F,B,C,D,E
                      7829X
                      7830X
102.047  315 056 102  7831X $FWRIB   CALL    $FWRIB.
102.052  320          7832X           RNC                     RETURN IF OK
102.053  303 223 070  7833X           JMP     $FERROR         ERROR
                      7834X
                      7835X
102.056               7836X $FWRIB.   EQU     *
102.056  345          7837X           PUSH    H
102.057  315 042 103  7838X           CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                      7839X
                      7840X *         COPY DATA FROM USER AREA TO BUFFER
                      7841X
102.062  325          7842X $WRIB2    PUSH    D               SAVE AREA ADDRESS
102.063  072 205 103  7843X           LDA     T.FLG
102.066  346 004      7844X           ANI     FT.OW           SEE IF OPEN FOR WRITE
102.070  312 224 102  7845X           JZ      $WRIB8          FILE NOT OPEN FOR WRITE
102.073  170          7846X           MOV     A,B
102.074  261          7847X           ORA     C
102.075  312 224 102  7848X           JZ      $WRIB8          ALL DONE
                      7849X
                      7850X *         COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                      7851X
102.100  052 210 103  7852X $WRIB3    LHLD    T.PTR
102.103  353          7853X           XCHG                    (DE) = (FB.PTR) = ADDRESS OF ROOM
102.104  052 214 103  7854X           LHLD    T.LWA           (HL) = LIMIT ADDRESS
102.107  175          7855X           MOV     A,L
```

```
102.110  223           7856X          SUB     E
102.111  157           7857X          MOV     L,A
102.112  174           7858X          MOV     A,H
102.113  232           7859X          SBB     D
102.114  147           7860X          MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
102.115  171           7861X          MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
102.116  225           7862X          SUB     L
102.117  170           7863X          MOV     A,B
102.120  234           7864X          SBB     H
102.121  322 126 102   7865X          JNC     $WRIB4          MORE REQUESTED THEN ROOM
102.124  140           7866X          MOV     H,B
102.125  151           7867X          MOV     L,C             USE REQUESTED COUNT
102.126  174           7868X $WRIB4   MOV     A,H
102.127  265           7869X          ORA     L
102.130  302 170 102   7870X          JNZ     $WRIB6          SOME ROOM IN BUFFER
                       7871X
                       7872X *        BUFFER IS FULL. EMPTY IT
                       7873X
102.133  305           7874X          PUSH    B               SAVE COUNT
102.134  052 206 103   7875X          LHLD    T.FWA
102.137  042 210 103   7876X          SHLD    T.PTR           CLEAR REMOVAL POINTER
102.142  353           7877X          XCHG
102.143  052 214 103   7878X          LHLD    T.LWA
102.146  175           7879X          MOV     A,L
102.147  223           7880X          SUB     E
102.150  117           7881X          MOV     C,A
102.151  174           7882X          MOV     A,H
102.152  232           7883X          SBB     D
102.153  107           7884X          MOV     B,A             (BC) = DATA IN BUFFER
102.154  072 204 103   7885X          LDA     T.CHA
102.157  377 005       7886X          DB      SYSCALL+.WRITE  WRITE BUFFER
102.161  301           7887X          POP     B               (BC) = DESIRED COUNT
102.162  322 100 102   7888X          JNC     $WRIB3          GOT THE DATA
                       7889X
                       7890X *        ERROR ON WRITE.
                       7891X
102.165  303 224 102   7892X          JMP     $WRIB8          HAVE ERROR
                       7893X
                       7894X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                       7895X *
                       7896X *        (BC) = REQUEST COUNT
                       7897X *        (DE) = TO
                       7898X *        (HL) = COUNT
                       7899X *        ((SP)) = FROM
                       7900X
102.170  171           7901X $WRIB6   MOV     A,C
102.171  225           7902X          SUB     L
102.172  117           7903X          MOV     C,A
102.173  170           7904X          MOV     A,B
102.174  234           7905X          SBB     H
102.175  107           7906X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
102.176  305           7907X          PUSH    B
102.177  343           7908X          XTHL                    (HL) = REMAINING REQUEST COUNT
102.200  301           7909X          POP     B               (BC) = COUNT FOR THIS COPY
102.201  343           7910X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
102.202  176           7911X $WRIB7   MOV     A,M
```

```
102,203  022           7912X          STAX    D
102,204  023           7913X          INX     D
102,205  043           7914X          INX     H
102,206  013           7915X          DCX     B
102,207  170           7916X          MOV     A,B
102,210  261           7917X          ORA     C
102,211  302 202 102   7918X          JNZ     $WRIB7          MORE TO GO
102,214  353           7919X          XCHG
102,215  042 210 103   7920X          SHLD    T.PTR           UPDATE POINTER
102,220  301           7921X          POP     B               (BC) = REMAINING COUNT
102,221  303 062 102   7922X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                       7923X
                       7924X *        WRITE COMPLETE.
                       7925X *
                       7926X *        (PSW) = COMPLETION FLAGS
                       7927X
102,224  321           7928X $WRIB8   POP     D               RESTORE TARGET ADDRESS
102,225  341           7929X          POP     H
102,226  303 070 103   7930X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                       7932X **       $FWBRK - BREAKOUTPUT                                    /80.02.GC/
                       7933X *
                       7934X *        $FWBRK empties the specified buffer by filling it with NULLs
                       7935X *        and then writing it.  Note this is used to insure that block
                       7936X *        mode I/O is output if it is not really a serial device (eg,
                       7937X *        writing to AT: from  *EDIT*.
                       7938X *
                       7939X *
                       7940X *        ENTRY:  HL      =  FILE BLOCK POINTER
                       7941X *
                       7942X *        EXIT:   HL      =  FILE BLOCK POINTER
                       7943X *                TO  $FERROR IF ERROR
                       7944X *
                       7945X *        USES:   PSW,BC,DE
                       7946X *
                       7947X
102,231  315 240 102   7948X $FWBRK   CALL    $FWBRK.
102,234  320           7949X          RNC                     NO ERROR
                       7950X
102,235  303 223 070   7951X          JMP     $FERROR
                       7952X
102,240  345           7953X $FWBRK.  PUSH    H
102,241  315 042 103   7954X          CALL    CBT             COPY BUFFER TO TEMPORARY
102,244  315 254 102   7955X          CALL    $FWBRK1
102,247  341           7956X          POP     H
102,250  315 070 103   7957X          CALL    CTB             COPY TEMPORARY TO BUFFER
102,253  311           7958X          RET
                       7959X
102,254  052 214 103   7960X $FWBRK1  LHLD    T.LWA
102,257  353           7961X          XCHG                    DE = BUFFER LWA
102,260  052 210 103   7962X          LHLD    T.PTR           HL = BUFFER PTR
102,263  173           7963X          MOV     A,E
102,264  225           7964X          SUB     L
```

```
102.265  117          7965X          MOV     C,A
102.266  172          7966X          MOV     A,D
102.267  234          7967X          SBB     H
102.270  107          7968X          MOV     B,A             BC = DE - HL
102.271  261          7969X          ORA     C
102.272  310          7970X          RZ                      THE BUFFER IS ALREADY FLUSHED
                      7971X
                      7972X *        FILL THE BUFFER WITH NULLS
                      7973X
102.273  170          7974X FWBRK2   MOV     A,B
102.274  261          7975X          ORA     C
102.275  312 307 102   7976X          JZ      FWBRK3          NO MORE LEFT TO FILL
                      7977X
102.300  066 000       7978X          MVI     M,0
102.302  043          7979X          INX     H
102.303  013          7980X          DCX     B
102.304  303 273 102   7981X          JMP     FWBRK2
                      7982X
102.307  052 206 103   7983X FWBRK3   LHLD    T.FWA
102.312  042 210 103   7984X          SHLD    T.PTR
102.315  353          7985X          XCHG                    DE = BUFFER FWA
102.316  052 214 103   7986X          LHLD    T.LWA           HL = BUFFER LWA
102.321  175          7987X          MOV     A,L
102.322  223          7988X          SUB     E
102.323  117          7989X          MOV     C,A
102.324  174          7990X          MOV     A,H
102.325  232          7991X          SBB     D
102.326  107          7992X          MOV     B,A             BC = HL - DE  ( BC = COUNT )
102.327  072 204 103   7993X          LDA     T.CHA
102.332  377 005       7994X          DB      SYSCALL,.WRITE
102.334  311          7995X          RET
102.335               7996           XTEXT   FCLO


                      7998X **       $FCLO - CLOSE FILE BLOCK.
                      7999X *
                      8000X *        $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                      8001X *        BLOCK.
                      8002X *
                      8003X *        ENTRY   (HL) = FILE BLOCK ADDRESS
                      8004X *        EXIT    TO $FERROR IF ERROR
                      8005X *                 TO CALLER IF OK
                      8006X *        USES    A,F,B,C,D,E
                      8007X
                      8008X
102.335  315 344 102   8009X $FCLO    CALL    $FCLO.
102.340  320          8010X          RNC                     NO ERROR
102.341  303 223 070   8011X          JMP     $FERROR
                      8012X
102.344  345          8013X $FCLO.   PUSH    H               SAVE FILE BLOCK ADDRESS
000.000               8014X          ERRNZ   FB.FLG-1
102.345  043          8015X          INX     H               (HL) = $FB.FLG
102.346  176          8016X          MOV     A,M
102.347  066 000       8017X          MVI     M,0             CLEAR FLAG
```

```
102.351  247          8018X          ANA     A
102.352  312 040 103  8019X          JZ      $FCLO4          FILE NOT OPEN
102.355  346 004      8020X          ANI     FT.OW
102.357  312 032 103  8021X          JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                      8022X
                      8023X *        WAS OPEN FOR WRITE. SEE IF NEED FLUSH THE LAST SECTOR
                      8024X
102.362  315 234 030  8025X          CALL    $INDL
102.365  003 000      8026X          DW      FB.PTR-FB.FLG
102.367  325          8027X          PUSH    D               SAVE (FB.PTR)
102.370  315 234 030  8028X          CALL    $INDL           (DE) = (FB.FWA)
102.373  001 000      8029X          DW      FB.FWA-FB.FLG
102.375  341          8030X          POP     H               (HL) = (FB.PTR)
102.376  175          8031X          MOV     A,L
102.377  223          8032X          SUB     E
103.000  117          8033X          MOV     C,A
103.001  174          8034X          MOV     A,H
103.002  232          8035X          SBB     D
103.003  107          8036X          MOV     B,A             (BC) = AMOUNT IN BLOCK
103.004  261          8037X          ORA     C
103.005  312 032 103  8038X          JZ      $FCLO3          NONE TO FLUSH
                      8039X
                      8040X *        NEED TO FLUSH BUFFER
                      8041X *
                      8042X *        (BC) = DATA AMOUNT
                      8043X *        (DE) = FWA
                      8044X *        (HL) = LWA+1
                      8045X
103.010  171          8046X          MOV     A,C
103.011  247          8047X          ANA     A
103.012  312 025 103  8048X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                      8049X
                      8050X *        ZERO FILL PARTIAL SECTOR
                      8051X
103.015  066 000      8052X $FCLO1   MVI     M,0
103.017  043          8053X          INX     H
103.020  014          8054X          INR     C
103.021  302 015 103  8055X          JNZ     $FCLO1
103.024  004          8056X          INR     B               COUNT ANOTHER FULL SECTOR
103.025  341          8057X $FCLO2   POP     H               (HL) = FB.FWA
103.026  176          8058X          MOV     A,M             (A) = CHANNEL NUMBER
000.000                8059X          ERRNZ   FB.CHA
103.027  345          8060X          PUSH    H
103.030  377 005       8061X          DB      SYSCALL+.WRITE          FLUSH
                      8062X
                      8063X *        READY TO CLOSE FILE.
                      8064X *
                      8065X *        'C' SET IF ERROR
                      8066X *         (A) = ERROR CODE
                      8067X
103.032  341          8068X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
103.033  330          8069X          RC                      ERROR
000.000                8070X          ERRNZ   FB.CHA
103.034  176          8071X          MOV     A,M             (A) = CHANNEL NUMBER
103.035  345          8072X          PUSH    H
103.036  377 046       8073X          DB      SYSCALL+.CLOSE  CLOSE CHANNEL
```

```
                                8074X $FCLO4  POP     H               (HL) = FILE BLOCK ADDRESS
                                8075X         RET
                                8076          XTEXT   FUTIL


                                8078X **      $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                                8079X
                                8080X **      CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                                8081X *
                                8082X *       ENTRY   (HL) = FILE BLOK FWA
                                8083X *       EXIT    NONE
                                8084X *       USES    A,F,H,L
                                8085X
                                8086X CBT     PUSH    D
                                8087X         PUSH    B               SAVE REGISTERS
                                8088X         ERRNZ   TLEN-10         ASSUME 10 BYTES TO MOVE
                                8089X         LXI     D,T.CHA         (DE) = TARGET FOR MOVE
                                8090X         MVI     B,10/2
                                8091X CBT1    MOV     A,M             COPY FILE BUFFER INTO WORK AREA
                                8092X         STAX    D
                                8093X         INX     H
                                8094X         INX     D
                                8095X         MOV     A,M
                                8096X         STAX    D
                                8097X         INX     H
                                8098X         INX     D
                                8099X         DCR     B
                                8100X         JNZ     CBT1            MORE TO GO
                                8101X         POP     B
                                8102X         POP     D               (DE) = DATA TARGET ADDRESS
                                8103X         RET
                                8104X
                                8105X
                                8106X **      CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                                8107X *
                                8108X *       ENTRY   (HL) = FILE BLOCK ADDRESS
                                8109X *       EXIT    NONE
                                8110X *       USES    NONE
                                8111X
                                8112X CTB     PUSH    PSW
                                8113X         PUSH    D
                                8114X         PUSH    B
                                8115X         PUSH    H               SAVE REGISTERS
                                8116X         MVI     B,8/2
                                8117X         LXI     D,T.CHA
                                8118X CTB1    LDAX    D
                                8119X         MOV     M,A
                                8120X         INX     D
                                8121X         INX     H
                                8122X         LDAX    D
                                8123X         MOV     M,A
                                8124X         INX     D
                                8125X         INX     H
                                8126X         DCR     B
```

```
103.112  302 101 103  8127X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
103.115  341          8128X          POP     H
103.116  301          8129X          POP     B
103.117  321          8130X          POP     D
103.120  361          8131X          POP     PSW
103.121  311          8132X          RET


                      8134X **       $FFB - FILE FILE BUFFER.
                      8135X *
                      8136X *        $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE.
                      8137X *
                      8138X *        ENTRY   NONE
                      8139X *        EXIT    'C' SET IF READ INCOMPLETE
                      8140X *                 (A) = ERROR CODE
                      8141X *                'C' CLEAR IF READ COMPLETEE
                      8142X *                 DATA IN BUFFER
                      8143X *        USES    A,F,D,E,H,L
                      8144X
                      8145X
103.122  072 216 103  8146X $FFB     LDA     EOFFLG
103.125  037          8147X          RAR
103.126  330          8148X          RC                      EOF
                      8149X
                      8150X *        CAN READ MORE. DO SO
                      8151X
103.127  305          8152X          PUSH    B               SAVE COUNT
103.130  052 206 103  8153X          LHLD    T.FWA
103.133  042 210 103  8154X          SHLD    T.PTR           CLEAR REMOVAL POINTER
103.136  353          8155X          XCHG
103.137  052 214 103  8156X          LHLD    T.LWA
103.142  042 212 103  8157X          SHLD    T.LIM           SET DATA LIMIT
103.145  175          8158X          MOV     A,L
103.146  223          8159X          SUB     E
103.147  117          8160X          MOV     C,A
103.150  174          8161X          MOV     A,H
103.151  232          8162X          SBB     D
103.152  107          8163X          MOV     B,A             (BC) = ROOM IN BUFFER
103.153  072 204 103  8164X          LDA     T.CHA
103.156  377 004      8165X          DB      SYSCALL,,READ   READ BUFFER
103.160  120          8166X          MOV     D,B             (D) = SECTORS UNREAD
103.161  301          8167X          POP     B               (BC) = DESIRED COUNT
103.162  320          8168X          RNC                     GOT THE DATA
                      8169X
                      8170X *        ERROR ON READ. SEE IF EOF
                      8171X
103.163  027          8172X          RAL
103.164  062 216 103  8173X          STA     EOFFLG          SET EOF, WE HOPE
103.167  376 003      8174X          CPI     EC.EOF*2+1
103.171  037          8175X          RAR
103.172  300          8176X          RNE                     IS NOT EOF, RETURN NOW!
103.173  072 213 103  8177X          LDA     T.LIM+1
103.176  222          8178X          SUB     D
103.177  062 213 103  8179X          STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
```

```
103.202  247          8180X          ANA     A
103.203  311          8181X          RET                     EXIT WITH DATA
                      8182X
                      8183X
                      8184X **       TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                      8185X
000.000               8186X          ERRNZ   FB.CHA
103.204  000          8187X T.CHA    DB      0               CHANNEL NUMBER
000.000               8188X          ERRNZ   *-T.CHA-FB.FLG
103.205  000          8189X T.FLG    DB      0               FLAG BYTE
000.000               8190X          ERRNZ   *-T.CHA-FB.FWA
103.206  000 000      8191X T.FWA    DW      0
000.000               8192X          ERRNZ   *-T.CHA-FB.PTR
103.210  000 000      8193X T.PTR    DW      0
000.000               8194X          ERRNZ   *-T.CHA-FB.LIM
103.212  000 000      8195X T.LIM    DW      0
000.000               8196X          ERRNZ   *-T.CHA-FB.LWA
103.214  000 000      8197X T.LWA    DW      0
000.012               8198X TLEN     EQU     *-T.CHA         LENGTH OF TEMP CELLS
                      8199X
103.216  000          8200X EOFFLG   DB      0
103.217               8201           XTEXT   TYPCC
```

```
                      8203X **       $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                      8204X *
                      8205X *        $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                      8206X *        THE CHARACTER ADDRESS AND COUNT.
                      8207X *
                      8208X *        ENTRY   (HL) = ADDRESS
                      8209X *                (A) = COUNT
                      8210X *        EXIT    (HL) = LAST CHARACTER ADDRESS+1
                      8211X *        USES    A,F,H,L
                      8212X
                      8213X
103.217               8214X $TYPCC   EQU     *
103.217  247          8215X          ANA     A
103.220  310          8216X          RZ                      NOTHING TO TYPE
103.221  365          8217X          PUSH    PSW             SAVE COUNT
103.222  176          8218X          MOV     A,M             (A) = CHARACTER
103.223  043          8219X          INX     H
103.224  377 002      8220X          DB      SYSCALL+.SCOUT
103.226  361          8221X          POP     PSW
103.227  075          8222X          DCR     A
103.230  303 217 103  8223X          JMP     $TYPCC
103.233               8224           XTEXT   RCHAR
```

```
                                8226X **      $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                8227X *
                                8228X *       ENTRY   NONE
                                8229X *       EXIT    (A) = CHARACTER
                                8230X *       USES    A,F
                                8231X
                                8232X
103.233  377 001                8233X $RCHAR  DB      SYSCALL,,SCIN
103.235  332 233 103            8234X         JC      $RCHAR          NOT READY
103.240  311                    8235X         RET
                                8236X
103.241  377 002                8237X $WCHAR  DB      SYSCALL,,SCOUT
103.243  311                    8238X         RET
                                8239          LON     C
103.244                         8240          XTEXT   ATS


                                8242X **      $ATS - ALLOCATE TABLE SPACE.
                                8243X *
                                8244X *       ATS IS CALLED TO ALLOCATE ADDITIONAL SPACE TO A MANAGED TABLE.
                                8245X *
                                8246X *       IF NO MOVING IS REQUIRED, $ATS REQUIRES ABOUT 150 MICROSECONDS.
                                8247X *
                                8248X *       ENTRY (HL) = BYTES TO ALLOCATE
                                8249X *              (DE) = ADDRESS OF TABLE INDEX+1
                                8250X *       EXIT    SPACE ALLOCATED (IF ENOUGH ROOM)
                                8251X *               TO *ERR.TO* IF NO MORE ROOM
                                8252X *       USES    A,F,H,L
                                8253X
                                8254X
103.244                         8255X $ATS    EQU     *               ENTRY POINT
103.244  305                    8256X         PUSH    B               SAVE REGISTERS
103.245  325                    8257X         PUSH    D
103.246  345                    8258X         PUSH    H
103.247  353                    8259X         XCHG                    (DE) = BN (BYTES NEEDED)
103.250  042 356 103            8260X         SHLD    ATSA            SAVE FOR LATER
103.253  116                    8261X         MOV     C,M
103.254  043                    8262X         INX     H
103.255  106                    8263X         MOV     B,M             (BC) = TFWA (TABLE FWA)
103.256  043                    8264X         INX     H
103.257  305                    8265X         PUSH    B               SAVE TFWA
103.260  116                    8266X         MOV     C,M
103.261  043                    8267X         INX     H
103.262  106                    8268X         MOV     B,M             (BC) = TL (TABLE LENGTH)
103.263  043                    8269X         INX     H
103.264  353                    8270X         XCHG                    (HL) = BN
103.265  011                    8271X         DAD     B               (HL) = NEW TABLE LENGTH
103.266  104                    8272X         MOV     B,H
103.267  115                    8273X         MOV     C,L             (BC) = NEW TABLE LENGTH
103.270  341                    8274X         POP     H
103.271  011                    8275X         DAD     B               (HL) = NEW TABLE LWA
103.272  332 321 103            8276X         JC      ATS1
103.275  353                    8277X         XCHG                    (DE) = NEW LWA, (HL) = INDEX ENTRY ADDRESS
103.276  043                    8278X         INX     H               SPACE OVER ALLOCATION FACTOR
```

```
103.277  173          8279X          MOV     A,E
103.300  226          8280X          SUB     M               COMPARE NEW LWA WITH NEXT TABLE FWA
103.301  043          8281X          INX     H
103.302  172          8282X          MOV     A,D
103.303  236          8283X          SBB     M
103.304  322 321 103  8284X          JNC     ATS1            OVERFLOW
                      8285X
                      8286X *        HAVE ENOUGH ROOM WITHOUT TABLE MOVES. UPDATE INDEX
                      8287X
103.307  053          8288X          DCX     H
103.310  053          8289X          DCX     H
103.311  053          8290X          DCX     H
103.312  160          8291X          MOV     M,B             SET NEW LENGTH
103.313  053          8292X          DCX     H
103.314  161          8293X          MOV     M,C
103.315  341          8294X          POP     H               RESTORE REGISTERS
103.316  321          8295X          POP     D
103.317  301          8296X          POP     B
103.320  311          8297X          RET

                      8299X **       THE TABLE OVERFLOWED IT'S FREE SPACE. REALLOCATE FREE SPACE
                      8300X *        AMONG STACKS.
                      8301X *
                      8302X *        (ATSA) = TABLE INDEX FWA
                      8303X *        (STACK TOP) = BN (BYTES NEEDED)
                      8304X
103.321  315 127 104  8305X ATS1     CALL    MTD             MOVE TABLES DOWN
103.324  041 012 000  8306X          LXI     H,10
103.327  031          8307X          DAD     D
103.330  321          8308X          POP     D
103.331  325          8309X          PUSH    D               (DE) = BN
103.332  031          8310X          DAD     D
103.333  332 160 070  8311X          JC      ERR.TO          TABLE OVERFLOW
103.336  353          8312X          XCHG                    (DE) = FIRST FREE SPACE AFTER ALLOCATION
103.337  052 171 112  8313X          LHLD    MEML            (HL) = MEMORY LIMIT ADDRESS
103.342  173          8314X          MOV     A,E
103.343  225          8315X          SUB     L
103.344  157          8316X          MOV     L,A
103.345  172          8317X          MOV     A,D
103.346  234          8318X          SBB     H
103.347  147          8319X          MOV     H,A             (HL) = -SPACE LEFT
103.350  322 160 070  8320X          JNC     ERR.TO          TABLE OVERFLOW
                      8321X
                      8322X *        THE ROOM EXISTS. ADD REQESTED SPACE TO PROPER TABLE.
                      8323X
103.353  301          8324X          POP     B               (BC) = BN (BYTES NEEDED)
103.354  345          8325X          PUSH    H               SAVE -(SPACE LEFT)
103.355  041 000 000  8326X          LXI     H,0             (HL) = TABLE INDEX FWA
103.356               8327X ATSA     EQU     *-2
103.360  043          8328X          INX     H
103.361  043          8329X          INX     H
103.362  136          8330X          MOV     E,M
103.363  043          8331X          INX     H
103.364  126          8332X          MOV     D,M             (DE) = CURRENT SIZE
103.365  353          8333X          XCHG
```

```
103.366  011          8334X           DAD     B
103.367  353          8335X           XCHG                    (DE) = NEW SIZE
103.370  162          8336X           MOV     M,D
103.371  053          8337X           DCX     H
103.372  163          8338X           MOV     M,E             SET NEW SIZE
                      8339X
                      8340X *         TABLES ARE ALL PACKED TOGETHER AT THE BOTTOM OF THE TABLE
                      8341X *         AREA. DECIDE HOW MUCH SPACE IS TO BE GIVEN TO EACH TABLE,
                      8342X *         AND MOVE THEM ONE BY ONE INTO POSITION, STARTING WITH THE
                      8343X *         HIGHEST TABLE, WORKING DOWN TO TABLE 2.
                      8344X
103.373  301          8345X           POP     B               (B) = -(SPACE LEFT)
103.374  046 003      8346X           MVI     H,3             DIVIDE BY 8
                      8347X
                      8348X *         DIVIDE SPACE LEFT BY 8
                      8349X
103.376  067          8350X ATS2      STC
103.377  170          8351X           MOV     A,B
104.000  037          8352X           RAR                     SHIFT RIGHT WITH SIGN EXTEND
104.001  107          8353X           MOV     B,A
104.002  171          8354X           MOV     A,C
104.003  037          8355X           RAR
104.004  117          8356X           MOV     C,A
104.005  045          8357X           DCR     H
104.006  302 376 103   8358X           JNZ     ATS2
104.011  003          8359X           INX     B               (BC) = 1/8 FREE SPACE
104.012  170          8360X           MOV     A,B
104.013  247          8361X           ANA     A
104.014  362 160 070   8362X           JP      ERR.TO          TABLE OVERFLOW
                      8363X
                      8364X *         (BC) = 1/8 FREE SPACE AVAILABLE.
                      8365X *
                      8366X *         MOVE TABLES INTO FINAL POSITION.
                      8367X
104.017  072 124 104   8368X           LDA     ATSB            (A) = TABLE COUNT-1
104.022  041 172 112   8369X           LXI     H,MEML+1
104.025  126          8370X           MOV     D,M
104.026  053          8371X           DCX     H
104.027  136          8372X           MOV     E,M             (DE) = (MEML)
104.030  053          8373X           DCX     H
                      8374X
104.031  365          8375X ATS3      PUSH    PSW             SAVE COUNT
104.032  305          8376X           PUSH    B
104.033  053          8377X           DCX     H
104.034  106          8378X           MOV     B,M
104.035  053          8379X           DCX     H
104.036  116          8380X           MOV     C,M             (BC) = TABLE LENGTH
104.037  053          8381X           DCX     H
104.040  173          8382X           MOV     A,E
104.041  221          8383X           SUB     C
104.042  137          8384X           MOV     E,A
104.043  172          8385X           MOV     A,D
104.044  230          8386X           SBB     B
104.045  127          8387X           MOV     D,A             (DE) = MEM TOP - TABLE SIZE
104.046  053          8388X           DCX     H
104.047  053          8389X           DCX     H
```

```
104.050  176          8390X          MOV     A,M             (A) = NUMBER OF 1/8'S TO GIVE THIS TABLE
104.051  343          8391X          XTHL                    (HL) = 1/8TH -SPACE
104.052  353          8392X          XCHG                    (HL) = MEM ADDRESS
104.053  247          8393X          ANA     A
104.054  312 064 104  8394X          JZ      ATS5            NO SPACE FOR THIS TABLE
104.057  031          8395X ATS4     DAD     D               DECREMENT BY FREE SPACE AMOUNT
104.060  075          8396X          DCR     A
104.061  302 057 104  8397X          JNZ     ATS4            GIVE SPECIFIED NUMBER OF 1/8THS
104.064  353          8398X ATS5     XCHG                    (DE) = TARGET ADDRESS
104.065  343          8399X          XTHL                    (HL) = TABLE ENTRY ADDRESS
104.066  345          8400X          PUSH    H
104.067  043          8401X          INX     H
104.070  176          8402X          MOV     A,M
104.071  163          8403X          MOV     M,E             SET NEW ADDRESS
104.072  365          8404X          PUSH    PSW
104.073  043          8405X          INX     H
104.074  176          8406X          MOV     A,M
104.075  162          8407X          MOV     M,D
104.076  147          8408X          MOV     H,A
104.077  361          8409X          POP     PSW
104.100  157          8410X          MOV     L,A
104.101  353          8411X          XCHG                    (BC) = COUNT, (DE) = FROM, (HL) = TO
104.102  345          8412X          PUSH    H
104.103  315 252 030  8413X          CALL    $MOVE           MOVE TABLE
104.106  321          8414X          POP     D               (DE) = NEW MEMORY LIMIT
104.107  341          8415X          POP     H
104.110  301          8416X          POP     B
104.111  361          8417X          POP     PSW
104.112  075          8418X          DCR     A
104.113  302 031 104  8419X          JNZ     ATS3            IF MORE TABLES TO MOVE
104.116  315 071 071  8420X          CALL    $ATP            ADJUST TABLE POINTERS
104.121  321          8421X          POP     D
104.122  301          8422X          POP     B
104.123  311          8423X          RET                     RETURN
                      8424X
104.124  007          8425X ATSB     DB      MTABL-1         TABLE COUNT-1
104.125  120 112      8426X ATSC     DW      MTABIND         ADDRESS OF 1ST TABLE TO MANAGE


                      8428X **       MTD - MOVE TABLES DOWN.
                      8429X *
                      8430X *        MTD IS CALLED TO MOVE ALL THE MANAGED TABLES DOWN INTO THE LOW
                      8431X *        PART OF THE MEMORY AREA, SO THAT ALL OF THE FREE SPACE IS CONCEN
                      8432X *        AFTER THE LAST TABLE.
                      8433X *
                      8434X *        ENTRY NONE
                      8435X *        EXIT    (DE) = FIRST FREE BYTE (LAST TABLE LWA+1)
                      8436X *        USES    ALL
                      8437X
                      8438X
104.127  052 125 104  8439X MTD      LHLD    ATSC
104.132  072 124 104  8440X          LDA     ATSB
                      8441X
                      8442X *        WONT NEED TO MOVE FIRST TABLE, FIND ITS LWA.
```

```
                          8443X
        104,135  043      8444X           INX     H
        104.136  116      8445X           MOV     C,M
        104.137  043      8446X           INX     H
        104.140  106      8447X           MOV     B,M             (BC) = FWA
        104.141  043      8448X           INX     H
        104.142  136      8449X           MOV     E,M
        104.143  043      8450X           INX     H
        104.144  126      8451X           MOV     D,M             (DE) = TABLE LEN
        104.145  043      8452X           INX     H
        104.146  353      8453X           XCHG
        104.147  011      8454X           DAD     B
        104.150  353      8455X           XCHG                    (DE) = TABLE LWA+1
                          8456X
                          8457X *         MOVE NEXT TABLE DOWN.
                          8458X
        104.151  365      8459X MTD1      PUSH    PSW
        104.152  043      8460X           INX     H
        104.153  116      8461X           MOV     C,M
        104.154  163      8462X           MOV     M,E             SET NEW START ADDRESS
        104.155  043      8463X           INX     H
        104.156  106      8464X           MOV     B,M             (BC) = TABLE FWA
        104.157  162      8465X           MOV     M,D
        104.160  043      8466X           INX     H
        104.161  305      8467X           PUSH    B
        104.162  116      8468X           MOV     C,M
        104.163  043      8469X           INX     H
        104.164  106      8470X           MOV     B,M             (BC) = TABLE LENGTH
        104.165  043      8471X           INX     H
        104.166  343      8472X           XTHL                    (HL) = TABLE FWA
        104.167  353      8473X           XCHG                    (DE) = FWA, (HL) = NEW ADDRESS
        104.170  315 252 030 8474X        CALL    $MOVE           MOVE DOWN
        104.173  353      8475X           XCHG
        104.174  341      8476X           POP     H               (DE) = NEXT FREE BYTE, (HL) = INDEX POINTER
        104.175  361      8477X           POP     PSW
        104.176  075      8478X           DCR     A
        104.177  302 151 104 8479X        JNZ     MTD1
        104.202  311      8480X           RET                     EXIT
        104.203           8481            XTEXT   DBT


                          8483X **        $DBT - DELETE BYTES FROM TABLE.
                          8484X *
                          8485X *         DBT DELETES BYTES FROM A MANAGED TABLE.
                          8486X *
                          8487X *         ENTRY   (DE) = BYTES TO DELETE
                          8488X *                 (HL) = POINTER TO PLACE (IN TABLE) TO BEGIN DELETEING (PTR
                          8489X *                 (RET+1, RET+2) = TABLE INDEX ADDRESS+1
                          8490X *         EXIT    BYTES DELETED.
                          8491X *         USES    A,F
                          8492X
                          8493X
        104.203           8494X $DBT      EQU     *
        104.203  173      8495X           MOV     A,E
```

```
104.204  057          8496X          CMA
104.205  137          8497X          MOV     E,A
104.206  172          8498X          MOV     A,D
104.207  057          8499X          CMA
104.210  127          8500X          MOV     D,A
104.211  023          8501X          INX     D               (DE) = -(BYTES TO DELETE)
104.212  067          8502X          STC                     SET CARRY


                      8504X **       $IBT - INSERT BYTES INTO TABLE.
                      8505X *
                      8506X *        $IBT IS CALLED TO MAKE A FREE SPACE IN A MANAGED TABLE. THIS
                      8507X *        FREE SPACE MAY BE CREATED ANYWHERE IN A TABLE: AT THE FRONT,
                      8508X *        AT THE BACK, OR IN THE MIDDLE.
                      8509X *
                      8510X *        ENTRY   (DE) = BYTES NEEDED (BN)
                      8511X *                (IF 'C' SET, DELETE BYTES)
                      8512X *                (HL) = POINTER TO INSERT AREA IN TABLE (PTR)
                      8513X *                (RET+1, RET+2) = TABLE ADDRESS
                      8514X *        EXIT    BYTES INSERTED
                      8515X *                TO (RET)+2
                      8516X *        USES    A,F
                      8517X
                      8518X
104.213  042 244 104  8519X $IBT     SHLD    IBTA            SAVE PTR
104.216  353          8520X          XCHG
104.217  343          8521X          XTHL                    (HL) = RETURN ADDRES
104.220  136          8522X          MOV     E,M
104.221  043          8523X          INX     H
104.222  126          8524X          MOV     D,M             (DE) = TABLE ADDRESS
104.223  043          8525X          INX     H
104.224  343          8526X          XTHL                    (HL) = BYTES NEEDED (BN)
104.225  345          8527X          PUSH    H               SAVE ENTRY (DE)
104.226  305          8528X          PUSH    B
104.227  332 301 104  8529X          JC      IBT2            IF TO DELETE
104.232  345          8530X          PUSH    H               SAVE BN
104.233  315 244 103  8531X          CALL    $ATS            ALLOCATE TABLE SPACE
                      8532X
                      8533X *        MOVE (TL-PTR) BYTES FROM (TFWA+PTR) TO (TFWA+PTR+BN)
                      8534X *        MOVE (TL-PTR-BN) BYTES FROM (TFWA+PTR) TO (FTWA+BTR+BN)
                      8535X
104.236  353          8536X          XCHG                    (HL) = TABLE ADDRESS
104.237  136          8537X          MOV     E,M
104.240  043          8538X          INX     H
104.241  126          8539X          MOV     D,M             (DE) = TABLE FWA
104.242  043          8540X          INX     H
104.243  001 000 000  8541X          LXI     B,0             (BC) = POINTER
104.244               8542X IBTA     EQU     *-2
104.246  353          8543X          XCHG
104.247  011          8544X          DAD     B               (HL) = TFWA+PTR
104.250  353          8545X          XCHG                    (DE) = TFWA+PTR
104.251  176          8546X          MOV     A,M
104.252  221          8547X          SUB     C
104.253  117          8548X          MOV     C,A
```

```
104.254  043          8549X          INX     H
104.255  176          8550X          MOV     A,M
104.256  230          8551X          SBB     B
104.257  107          8552X          MOV     B,A             (BC) = TL-PTR
104.260  341          8553X          POP     H               (HL) = BN
104.261  171          8554X          MOV     A,C
104.262  225          8555X          SUB     L
104.263  117          8556X          MOV     C,A
104.264  170          8557X          MOV     A,B
104.265  234          8558X          SBB     H
104.266  107          8559X          MOV     B,A             (BC) = TL-PTR-BN
104.267  031          8560X          DAD     D               (HL) = TFWA+PTR+BN
104.270  315 252 030   8561X IBT1     CALL    $MOVE           MOVE BLOCK
104.273  301          8562X          POP     B               RESTORE REGISTERS
104.274  321          8563X          POP     D
104.275  052 244 104   8564X          LHLD    IBTA            RESTORE (HL)
104.300  311          8565X          RET

                      8567X **       DELETE BYTES FROM TABLE.
                      8568X
104.301  174          8569X IBT2     MOV     A,H
104.302  057          8570X          CMA
104.303  147          8571X          MOV     H,A
104.304  175          8572X          MOV     A,L
104.305  057          8573X          CMA
104.306  157          8574X          MOV     L,A
104.307  043          8575X          INX     H               (HL) = BYTES TO DELETE
                      8576X
                      8577X *        MOVE (TL-PTR-BN) BYTES FROM (PTR+BN+TFWA) TO (PTR+TFWA)
                      8578X
104.310  353          8579X          XCHG                    (HL) = ADDRESS, (DE) = BN
104.311  116          8580X          MOV     C,M
104.312  043          8581X          INX     H
104.313  106          8582X          MOV     B,M             (BC) = TFWA
104.314  305          8583X          PUSH    B
104.315  043          8584X          INX     H
104.316  176          8585X          MOV     A,M
104.317  223          8586X          SUB     E
104.320  117          8587X          MOV     C,A
104.321  167          8588X          MOV     M,A
104.322  043          8589X          INX     H
104.323  176          8590X          MOV     A,M
104.324  232          8591X          SBB     D
104.325  107          8592X          MOV     B,A             (BC) = TL-BN
104.326  167          8593X          MOV     M,A             SET NEW LEN IN TABLE
104.327  052 244 104   8594X          LHLD    IBTA            (HL) = PTR
104.332  171          8595X          MOV     A,C
104.333  225          8596X          SUB     L
104.334  117          8597X          MOV     C,A
104.335  170          8598X          MOV     A,B
104.336  234          8599X          SBB     H
104.337  107          8600X          MOV     B,A             (BC) = TL-PTR-BN
104.340  353          8601X          XCHG                    (DE) = PTR, (HL) = BN
104.341  343          8602X          XTHL
```

```
104.342  031          8603X        DAD     D              (HL) = PTR+TFWA
104.343  353          8604X        XCHG                   (DE) = PTR+TFWA
104.344  341          8605X        POP     H              (HL) = BN
104.345  031          8606X        DAD     D
                      8607X
                      8608X *      (BC) = TL-PTR-BN
                      8609X *      (DE) = BTF+TFWA
                      8610X *      (HL) = PTR+TFWA+BN
                      8611X
104.346  353          8612X        XCHG
104.347  303 270 104  8613X        JMP     IBT1           MOVE DATA AND EXIT
104.352               8614         XTEXT   FPP
```

```
                        8618X **      FPADD - FLOATING POINT ADD.
                        8619X *
                        8620X *       ACCX = ACCX + (DE)
                        8621X *
                        8622X *       ENTRY (DE) = POINTER TO 4 BYTE FP VALUE
                        8623X *       EXIT    ACCX = RESULT
                        8624X *               SUPPLIED VALUE UNCHANGED
                        8625X *       USES    A,F
                        8626X
                        8627X
104.352  315 215 107    8628X FPADD   CALL    SPE             SETUP PACKAGE ENTRY
104.355  353            8629X         XCHG                    (HL) = ADDRESS OF VALUE


                        8631X **      ADD - PERFORM FLOATING POINT ADD.
                        8632X *
                        8633X *       ACCX = ACCX + (HL)
                        8634X *
                        8635X *       ENTRY (HL) = POINTER TO 4 BYTE FP VALUE
                        8636X *               RESULT STORED IN ACCX
                        8637X *       USES    ALL
                        8638X
                        8639X
104.356                 8640X ADD     EQU     *
104.356  315 250 107    8641X         CALL    LDD             (BCDE) = Y
104.361  041 205 042    8642X         LXI     H,ACCX+3
                        8643X
                        8644X *       CHECK FOR X+0, 0+Y
                        8645X
104.364  170            8646X         MOV     A,B             (A) = EXP(Y)
104.365  267            8647X         ORA     A
104.366  310            8648X         RZ                      IF Y=0
                        8649X
104.367  176            8650X ADD0    MOV     A,M             (A) = EXP(X)
104.370  267            8651X         ORA     A
104.371  312 160 105    8652X         JZ      ADD5            X = 0; RESULT = (BCDE)          /80.02.BC/
                        8653X
                        8654X *       COMPARE EXPONENTS, TO SEE IF SIGNIFICANT
                        8655X
104.374  220            8656X         SUB     B
104.375  322 022 105    8657X         JNC     ADD1            EXPX GT EXPY
105.000  052 202 042    8658X         LHLD    ACCX            SWAP (BCDE) WITH ACCX
105.003  353            8659X         XCHG
105.004  042 202 042    8660X         SHLD    ACCX
105.007  052 204 042    8661X         LHLD    ACCX+2
105.012  305            8662X         PUSH    B
105.013  343            8663X         XTHL
105.014  301            8664X         POP     B
105.015  042 204 042    8665X         SHLD    ACCX+2
105.020  057            8666X         CMA
105.021  074            8667X         INR     A               (A) = SHIFT COUNT
                        8668X
                        8669X *       (A) = SHIFT COUNT FOR JUSTIFICATION
                        8670X
```

```
105.022  312 074 105  8671X ADD1   JZ      ADD3            NONE TO SHIFT
105.025  376 030      8672X         CPI     24
105.027  332 057 105  8673X         JC      ADD2.5          IS LESS THAN 24
                      8674X
                      8675X *       WOULD NEED TO SHIFT INTO INSIGNIFACANCE. JUST ADD 0
                      8676X
105.032  021 000 000  8677X         LXI     D,0             (DE) = 0
105.035  112          8678X         MOV     C,D             (C) = 0
105.036  303 074 105  8679X         JMP     ADD3
                      8680X
                      8681X *       DO JUSTIFYING RIGHT SHIFT
                      8682X
105.041  132          8683X ADD2    MOV     E,D
105.042  121          8684X         MOV     D,C
105.043  171          8685X         MOV     A,C
105.044  027          8686X         RAL
105.045  076 000      8687X         MVI     A,0
105.047  237          8688X         SBB     A
105.050  117          8689X         MOV     C,A
105.051  174          8690X         MOV     A,H
105.052  326 010      8691X         SUI     8
105.054  312 074 105  8692X         JZ      ADD3            IF NO MORE
105.057  147          8693X ADD2.5  MOV     H,A             (H) = SHIFT COUNT
105.060  376 010      8694X         CPI     8
105.062  322 041 105  8695X         JNC     ADD2            IF MORE THAN 8
105.065  315 231 107  8696X ADD2.7  CALL    SRS             SHIFT RIGHT WITH SIGN EXTEND
105.070  045          8697X         DCR     H
105.071  302 065 105  8698X         JNZ     ADD2.7
                      8699X
                      8700X *       NUMBERS ALLIGNED. PERFORM ADD
                      8701X
105.074  041 202 042  8702X ADD3    LXI     H,ACCX
105.077  171          8703X         MOV     A,C
105.100  365          8704X         PUSH    PSW             SAVE OLD Y SIGN
105.101  176          8705X         MOV     A,M
105.102  213          8706X         ADC     E               ADD WITH ROUND
105.103  137          8707X         MOV     E,A
105.104  043          8708X         INX     H
105.105  176          8709X         MOV     A,M
105.106  212          8710X         ADC     D
105.107  127          8711X         MOV     D,A
105.110  043          8712X         INX     H
105.111  176          8713X         MOV     A,M
105.112  211          8714X         ADC     C
105.113  117          8715X         MOV     C,A             (CDE) = NEW SUM
105.114  176          8716X         MOV     A,M             (A) = X SIGN
105.115  043          8717X         INX     H
105.116  106          8718X         MOV     B,M             (B) = NEW EXPONENT
105.117  037          8719X         RAR
105.120  147          8720X         MOV     H,A             (H) 200 BIT = CARRY, 100 BIT = X SIGN
105.121  361          8721X         POP     PSW             (A) = Y SIGN
105.122  037          8722X         RAR
105.123  254          8723X         XRA     H               (A) 100 BIT = XSIGN XOR YSIGN
105.124  027          8724X         RAL
105.125  251          8725X         XRA     C               (A) 200 BIT = XSIGN XOR YSIGN XOR SUMSIGN
105.126  254          8726X         XRA     H               (A) = XSIGN XOR YSIGN XOR SUMSIGN XOR CARRY
```

```
105.127  362 142 105  8727X          JP      ADD4            IS NOT OVERFLOW
                      8728X
                      8729X *        IS OVERFLOW. SHIFT RIGHT 1
                      8730X
105.132  174          8731X          MOV     A,H
105.133  315 232 107  8732X          CALL    SRS.            SHIFT RIGHT
105.136  004          8733X          INR     B
105.137  312 136 070  8734X          JZ      ERR.OV          IF OVERFLOW
                      8735X
                      8736X *        RESULT IN (B,C,D,E)
                      8737X
105.142  305          8738X ADD4     PUSH    B               SAVE OLD EXPONENT
105.143  315 213 105  8739X          CALL    NRM
105.146  361          8740X          POP     PSW             (A) = OLD EXPONENT
105.147  220          8741X          SUB     B
105.150  376 025      8742X          CPI     21
105.152  324 221 105  8743X          CNC     NRM0            USE 0 IF HAVE LOST 21 BITS OF SIGNIFICANCE
105.155  303 245 106  8744X          JMP     STX             STORE AND EXIT
                      8745X
                      8746X *        NORMALIZE RESULT = (BCDE)                       /80.02.GC/
                      8747X
105.160  315 213 105  8748X ADD5     CALL    NRM             NORMALIZE               /80.02.GC/
105.163  303 245 106  8749X          JMP     STX             STORE AND EXIT          /80.02.GC/


                      8751X **       FPSUB - FLOATING POINT SUBTRACT.
                      8752X *
                      8753X *        FPSUB COMPUTES (DE) - ACCX
                      8754X *
                      8755X *        ENTRY (DE) = POINTER TO 4 BYTE FP VALUE
                      8756X *        EXIT    ACCX = RESULT
                      8757X *                SUPPLIED VALUE UNCHANGED
                      8758X *        USES    A,F
                      8759X
                      8760X
105.166  315 215 107  8761X FPSUB    CALL    SPE             SETUP PACKAGE ENTRY/EXIT
105.171  353          8762X          XCHG                    (HL) = ADDRESS
105.172  345          8763X SUB      PUSH    H               SAVE
105.173  315 305 105  8764X          CALL    NEG             NEGATE (ACCX)
105.176  341          8765X          POP     H               (HL) = ADDRESS OF VALUE
105.177  303 356 104  8766X          JMP     ADD             ADD, RESTORE, RETURN
```

```
                                8770X **      FPNRM - FLOATING POINT NORMALIZE.
                                8771X *
                                8772X *       FPNRM NORMALIZES THE CONTENTS OF (ACCX).
                                8773X *
                                8774X *       ENTRY NONE
                                8775X *       EXIT    (ACCX) NORMALIZED
                                8776X *       USES    A;F
                                8777X
                                8778X
105.202  315 215 107            8779X FPNRM   CALL    SPE             SETUP PACKAGE ENTRY
105.205  315 245 107            8780X NRM.    CALL    LDX             (BCDE) = (ACCX)
105.210  303 142 105            8781X         JMP     ADD4            NORMALIZE AND STORE


                                8783X **      NRM - NORMALIZE NUMBER.
                                8784X *
                                8785X *       ENTRY (B;C;D;E) = NUMBER
                                8786X *       EXIT    NORMAILLIZED
                                8787X *       USES    H;L
                                8788X
                                8789X
105.213                         8790X NRM     EQU     *
105.213  171                    8791X         MOV     A;C
105.214  262                    8792X         ORA     D
105.215  263                    8793X         ORA     E
105.216  302 242 105            8794X         JNZ     NRM2            IF NON-ZERO
                                8795X
                                8796X *       NUMBER IS ZERO
                                8797X
105.221  001 000 000            8798X NRM0    LXI     B;0
105.224  120                    8799X         MOV     D;B
105.225  130                    8800X         MOV     E;B             (BCDE) = 0
105.226  311                    8801X         RET
                                8802X
                                8803X *       NUMBER IS NON-ZERO
                                8804X
105.227  112                    8805X NRM1    MOV     C;D
105.230  123                    8806X         MOV     D;E
105.231  137                    8807X         MOV     E;A
105.232  170                    8808X         MOV     A;B
105.233  326 011                8809X         SUI     9
105.235  332 136 070            8810X         JC      ERR.OV          IF OVERFLOW
105.240  074                    8811X         INR     A
105.241  107                    8812X         MOV     B;A
                                8813X
105.242  171                    8814X NRM2    MOV     A;C
105.243  027                    8815X         RAL
105.244  251                    8816X         XRA     C
105.245  027                    8817X         RAL
105.246  330                    8818X         RC                      IF NORMALIZED
105.247  172                    8819X         MOV     A;D
105.250  027                    8820X         RAL
105.251  171                    8821X         MOV     A;C
105.252  027                    8822X         RAL
```

```
105.253  322 257 105  8823X          JNC     NRM3            IF PL
105.256  074         8824X          INR     A
105.257  247         8825X NRM3     ANA     A
105.260  312 227 105  8826X          JZ      NRM1            IF A FULL WORD TO SHIFT
                      8827X
                      8828X *        SHIFT LEFT UNTIL NORMALIZED
                      8829X
105.263  315 101 107  8830X NRM4     CALL    LSH             LSFT SHIFT
105.266  005         8831X          DCR     B
105.267  312 221 105  8832X          JZ      NRM0            UNDERFLOW
105.272  171         8833X          MOV     A,C
105.273  027         8834X          RAL
105.274  251         8835X          XRA     C
105.275  027         8836X          RAL
105.276  322 263 105  8837X          JNC     NRM4            IF NORE TO SHIT
105.301  311         8838X          RET                     EXIT
```

```
                                8842X **      FPNEG - FLOATING POINT NEGATE.
                                8843X *
                                8844X *       FPNEG NEGATES THE CONTENTS OF ACCX.
                                8845X *
                                8846X *       ENTRY NONE
                                8847X *       EXIT    (ACCX) = -(ACCX)
                                8848X *       USES    A,F
                                8849X
                                8850X
      105.302  315 215 107     8851X FPNEG   CALL    SPE             SETUP PACKAGE ENTRY
      105.305  315 245 107     8852X NEG     CALL    LDX             (BCDE) = (ACCX)
      105.310  315 260 107     8853X         CALL    TCV             TWO'S COMPLEMENT IT
      105.313  303 245 106     8854X         JMP     STX             STORE AND RETURN


                                8857X **      FPTST - FLOATING POINT TEST.
                                8858X *
                                8859X *       FPTST TESTS THE SIGN AND VALUE OF (ACCX).
                                8860X *
                                8861X *       ENTRY NONE
                                8862X *       EXIT    'Z' SET IF (ACSX) = 0
                                8863X *               'M' SET IF (ACCX) < 0
                                8864X *       USES    A,F
                                8865X
                                8866X
      105.316  072 204 042     8867X FPTST   LDA     ACCX+2
      105.321  247             8868X         ANA     A               SET CONDITION CODE
      105.322  311             8869X         RET
```

```
                                8872X **      FPMUL - FLOATING POINT MULTIPLY.
                                8873X *
                                8874X *       ENTRY (DE) = ADDRESS OF Y
                                8875X *       EXIT    ACCX = ACCX * Y
                                8876X *       USES    A,F
                                8877X
                                8878X
105.323  315 215 107           8879X FPMUL   CALL    SPE             SETUP PACKAGE ENTRY
105.326  353                   8880X         XCHG                    (HL) = ADDRESS OF VALUE


                                8882X **      MUL - FLOATING POINT MULTIPLY.
                                8883X *
                                8884X
                                8885X
105.327                         8886X MUL     EQU     *
105.327  021 200 000           8887X         LXI     D,MI.ADDB       (DE) = 'ADD B', 'NOP'
105.332  315 114 107           8888X         CALL    FMD             PREPARE MULTIPLY
105.335  312 240 106           8889X         JZ      MUL5            IS ZERO
105.340  332 136 070           8890X         JC      ERR.OV          IS OVERFLOW
105.343  147                   8891X         MOV     H,A             SAVE NEW EXPONEHT
105.344  345                   8892X         PUSH    H               SAVE NEW EXP AND SIGN
105.345  171                   8893X         MOV     A,C
105.346  062 034 106           8894X         STA     MULA            SETUP MULTIPLICAND
105.351  062 076 106           8895X         STA     MULD
105.354  062 143 106           8896X         STA     MULH
105.357  172                   8897X         MOV     A,D
105.360  062 072 106           8898X         STA     MULC
105.363  062 137 106           8899X         STA     MULG
105.366  173                   8900X         MOV     A,E
105.367  062 133 106           8901X         STA     MULF
105.372  041 204 042           8902X         LXI     H,ACCX+2
105.375  176                   8903X         MOV     A,M
105.376  062 121 106           8904X         STA     MULE
106.001  053                   8905X         DCX     H
106.002  176                   8906X         MOV     A,M
106.003  062 051 106           8907X         STA     MULB
106.006  053                   8908X         DCX     H
106.007  106                   8909X         MOV     B,M
106.010  046 007               8910X         MVI     H,7
106.012  154                   8911X         MOV     L,H
106.013  021 000 000           8912X         LXI     D,0
106.016  112                   8913X         MOV     C,D             ZERO ACCUMULATOR
106.017  170                   8914X         MOV     A,B
106.020  247                   8915X         ANA     A
106.021  312 047 106           8916X         JZ      MUL2.5
106.024  170                   8917X L1      MOV     A,B             (A) = MULTIPLICAND
106.025  037                   8918X         RAR
106.026  107                   8919X         MOV     B,A
106.027  171                   8920X         MOV     A,C
106.030  322 035 106           8921X         JNC     L2              BIT NOT PRESENT
106.033  306 000               8922X         ADI     0
106.034                         8923X MULA    EQU     *-1
106.035  037                   8924X L2      RAR
```

```
106.036  117            8925X          MOV     C,A
106.037  045            8926X          DCR     H
106.040  362 024 106    8927X          JP      L1
106.043  322 047 106    8928X          JNC     MUL2.5          NOT CARRY
106.046  014            8929X          INR     C
                        8930X
                        8931X *        2ND PARTIAL PRODUCT
                        8932X
106.047  145            8933X MUL2.5   MOV     H,L
106.050  006 000        8934X          MVI     B,0
106.051                 8935X MULB     EQU     *-1
106.052  072 202 042    8936X          LDA     ACCX
106.055  260            8937X          ORA     B
106.056  312 120 106    8938X          JZ      L4.5            NONE IN LOW TWO BYTES
                        8939X
106.061  170            8940X MUL3     MOV     A,B
106.062  037            8941X          RAR
106.063  107            8942X          MOV     B,A
106.064  171            8943X          MOV     A,C
106.065  322 077 106    8944X          JNC     L4              NOT SET
106.070  172            8945X          MOV     A,D
106.071  306 000        8946X          ADI     0
106.072                 8947X MULC     EQU     *-1
106.073  127            8948X          MOV     D,A
106.074  171            8949X          MOV     A,C
106.075  316 000        8950X          ACI     0
106.076                 8951X MULD     EQU     *-1
106.077  037            8952X L4       RAR
106.100  117            8953X          MOV     C,A
106.101  172            8954X          MOV     A,D
106.102  037            8955X          RAR
106.103  127            8956X          MOV     D,A
106.104  045            8957X          DCR     H
106.105  362 061 106    8958X          JP      MUL3
106.110  322 120 106    8959X          JNC     L4.5            NOT CARRY
106.113  024            8960X          INR     D
106.114  302 120 106    8961X          JNZ     L4.5
106.117  014            8962X          INR     C
106.120  006 000        8963X L4.5     MVI     B,0
106.121                 8964X MULE     EQU     *-1
                        8965X
106.122  170            8966X L5       MOV     A,B
106.123  037            8967X          RAR
106.124  107            8968X          MOV     B,A
106.125  171            8969X          MOV     A,C
106.126  322 144 106    8970X          JNC     L6              NOT SET
106.131  173            8971X          MOV     A,E
106.132  306 000        8972X          ADI     0
106.133                 8973X MULF     EQU     *-1
106.134  137            8974X          MOV     E,A
106.135  172            8975X          MOV     A,D
106.136  316 000        8976X          ACI     0
106.137                 8977X MULG     EQU     *-1
106.140  127            8978X          MOV     D,A
106.141  171            8979X          MOV     A,C
106.142  316 000        8980X          ACI     0
```

```
106.143                  8981X MULH   EQU    *-1
106.144  037           8982X L6     RAR
106.145  117           8983X        MOV    C,A
106.146  172           8984X        MOV    A,D
106.147  037           8985X        RAR
106.150  127           8986X        MOV    D,A
106.151  173           8987X        MOV    A,E
106.152  037           8988X        RAR
106.153  137           8989X        MOV    E,A
106.154  055           8990X        DCR    L
106.155  302 122 106   8991X        JNZ    L5
106.160  322 174 106   8992X        JNC    L7             NOT TO ROUND UP
106.163  034           8993X        INR    E
106.164  302 174 106   8994X        JNZ    L7
106.167  024           8995X        INR    D
106.170  302 174 106   8996X        JNZ    L7
106.173  014           8997X        INR    C
                       8998X *      NORMALIZE
                       8999X
106.174  171           9000X L7     MOV    A,C
106.175  341           9001X        POP    H              (HL) = EXPONENT AND SIGN
106.176  104           9002X        MOV    B,H
106.177  027           9003X        RAL
106.200  247           9004X        ANA    A
106.201  372 216 106   9005X        JM     MUL3.5         NORMALIZED
                       9006X
106.204  173           9007X        MOV    A,E
106.205  027           9008X        RAL
106.206  137           9009X        MOV    E,A
106.207  172           9010X        MOV    A,D
106.210  027           9011X        RAL
106.211  127           9012X        MOV    D,A
106.212  171           9013X        MOV    A,C
106.213  027           9014X        RAL
106.214  117           9015X        MOV    C,A
106.215  005           9016X        DCR    B              ADJUST EXPONENT
106.216  004           9017X MUL3.5 INR    B              ADJUST EXPONENT
106.217  312 136 070   9018X        JZ     ERR,OV
                       9019X
                       9020X *      NEGATE IF NECESSARY
                       9021X
106.222  175           9022X MUL4   MOV    A,L
106.223  247           9023X        ANA    A
106.224  374 260 107   9024X        CM     TCV            TWOS COMP VALUE
106.227  170           9025X        MOV    A,B
106.230  247           9026X        ANA    A
106.231  312 245 106   9027X        JZ     STX            VALUE IS 0
106.234  005           9028X        DCR    B
106.235  302 245 106   9029X        JNZ    STX            NOT UNDERFLOW
                       9030X
                       9031X *      RESULT = 0
                       9032X
106.240  001 000 000   9033X MUL5   LXI    B,0
106.243  120           9034X        MOV    D,B
106.244  130           9035X        MOV    E,B            OBCDE) = 0
                       9036X *      JMP    STX
```

```
                                9038X **      STX - STORE REGISTERS INTO X VALUE.
                                9039X *
                                9040X *       ENTRY (B,C,D,E) = VALUES
                                9041X *       EXIT    STORED IN REG.X
                                9042X
                                9043X
      106.245  041 202 042      9044X STX     LXI     H,ACCX
      106.250  163              9045X STO     MOV     M,E
      106.251  043              9046X         INX     H
      106.252  162              9047X         MOV     M,D
      106.253  043              9048X         INX     H
      106.254  161              9049X         MOV     M,C
      106.255  043              9050X         INX     H
      106.256  160              9051X         MOV     M,B
      106.257  311              9052X         RET
```

```
                                9055X **      FPDIV - FLOATING POINT DIVIDE.
                                9056X *
                                9057X *       ACCX = ACCX/Y
                                9058X *
                                9059X *       ENTRY (DE) = POINTER TO Y
                                9060X *       EXIT    (ACCX) = RESULT
                                9061X *       USES    A,F
                                9062X
                                9063X
      106.260  315 215 107      9064X FPDIV   CALL    SPE             SETUP PACKAGE ENTRY
      106.263  353              9065X         XCHG                    (HL) = ADDRESS OF VALUE


                                9067X **      DIV - FLOATING POINT DEVIDE.
                                9068X *
                                9069X *       X=Y/X
                                9070X
                                9071X
                                9072X
      106.264                   9073X DIV     EQU     *
      106.264  021 220 077      9074X         LXI     D,MI.CMC*256+MI.SUBB    (DE) = 'SUB B', 'CMC'
      106.267  315 114 107      9075X         CALL    PMD             PRESET FOR DEVICDE
      106.272  302 305 106      9076X         JNZ     DIV0            IF NEIGHER ZERO
      106.275  170              9077X         MOV     A,B
      106.276  247              9078X         ANA     A
      106.277  312 117 070      9079X         JZ      ERR.DO          (Y) = 0
      106.302  303 240 106      9080X         JMP     MUL5            (X) = 0
                                9081X
      106.305  332 136 070      9082X DIV0    JC      ERR.OV          IF OVERFLOW
      106.310  074              9083X         INR     A
      106.311  312 136 070      9084X         JZ      ERR.OV          IF OVERFLOW
      106.314  147              9085X         MOV     H,A             (H) = RESULT EXP, (L) = RESULT SIGN
      106.315  345              9086X         PUSH    H
      106.316  173              9087X         MOV     A,E
      106.317  062 367 106      9088X         STA     DIVA
      106.322  172              9089X         MOV     A,D
      106.323  062 373 106      9090X         STA     DIVB
      106.326  171              9091X         MOV     A,C
      106.327  062 377 106      9092X         STA     DIVC
      106.332  171              9093X         MOV     A,C
      106.333  062 016 107      9094X         STA     PMAC+1
      106.336  172              9095X         MOV     A,D
      106.337  062 012 107      9096X         STA     PMAB+1
      106.342  173              9097X         MOV     A,E
      106.343  062 006 107      9098X         STA     PMAA+1
      106.346  315 245 107      9099X         CALL    LDX             (BCDE) = X VALUE
      106.351  053              9100X         DCX     H
      106.352  345              9101X         PUSH    H
      106.353  056 003          9102X         MVI     L,3             (L) = LOOP COUNT
      106.355  076 002          9103X DIV1    MVI     A,2
      106.357  275              9104X         CMP     L
      106.360  336 372          9105X         SBI     -6
      106.362  147              9106X         MOV     H,A             (H) = 7 IF FIRST, 8 IF 2ND OR 3RD
      106.363  006 000          9107X         MVI     B,0             (B) = RESULT
```

```
106.365  173          9108X DIV2    MOV     A,E
106.366  326 000      9109X         SUI     0
106.367               9110X DIVA    EQU     *-1
106.370  137          9111X         MOV     E,A
106.371  172          9112X         MOV     A,D
106.372  336 000      9113X         SBI     0
106.373               9114X DIVB    EQU     *-1
106.374  127          9115X         MOV     D,A
106.375  171          9116X         MOV     A,C
106.376  336 000      9117X         SBI     0
106.377               9118X DIVC    EQU     *-1
107.000  117          9119X         MOV     C,A
107.001  322 020 107  9120X         JNC     DIV3
107.004  173          9121X         MOV     A,E
107.005  306 000      9122X PMAA    ADI     0
107.007  137          9123X         MOV     E,A
107.010  172          9124X         MOV     A,D
107.011  316 000      9125X PMAB    ACI     0
107.013  127          9126X         MOV     D,A
107.014  171          9127X         MOV     A,C
107.015  316 000      9128X PMAC    ACI     0
107.017  117          9129X         MOV     C,A
107.020  077          9130X DIV3    CMC
                      9131X
                      9132X *       SET RESULT BIT IN ACCUMULATOR
                      9133X
107.021  170          9134X         MOV     A,B
107.022  027          9135X         RAL
107.023  107          9136X         MOV     B,A
                      9137X
                      9138X *       SHIFT REMAINDER VALUE LEFT 1
                      9139X
107.024  315 101 107  9140X         CALL    LSH
107.027  045          9141X         DCR     H
107.030  302 365 106  9142X         JNZ     DIV2
                      9143X
                      9144X *       STORE SUBVALUE
                      9145X
107.033  343          9146X         XTHL
107.034  160          9147X         MOV     M,B
107.035  053          9148X         DCX     H
107.036  343          9149X         XTHL
107.037  055          9150X         DCR     L
107.040  302 355 106  9151X         JNZ     DIV1
107.043  341          9152X         POP     H
107.044  043          9153X         INX     H
107.045  043          9154X         INX     H
107.046  130          9155X         MOV     E,B
107.047  126          9156X         MOV     D,M
107.050  043          9157X         INX     H
107.051  171          9158X         MOV     A,C           (A) = REMAINDER HIGH ORDER
107.052  116          9159X         MOV     C,M           (BCDE) = RESULT
107.053  341          9160X         POP     H
107.054  104          9161X         MOV     B,H           (B) = RESULTANT EXPONENT
107.055  147          9162X         MOV     H,A
107.056  072 377 106  9163X         LDA     DIVC
```

```
107.061  224           9164X  SUB   H          SEE IF NEXT RESULT BIT WOULD BE 1 (OR CLOSE
107.062  334 315 107   9165X  CC    RVU        ROUND VALUE UP IF SO
107.065  171           9166X  MOV   A,C
107.066  346 100       9167X  ANI   100Q
107.070  302 216 106   9168X  JNZ   MUL3.5     IF NOT TO NORMALIZE
107.073  315 101 107   9169X  CALL  LSH
107.076  303 222 106   9170X  JMP   MUL4
```

```
                                9174X **      LSH - LEFT SHIFT VALUE.
                                9175X *
                                9176X *       ENTRY (C,D,E) = VALUE
                                9177X *       EXIT    (C,D,E) SHIFTED RIGHT 1
                                9178X
                                9179X
     107.101  247               9180X LSH     ANA     A               CLEAR CARRY
     107.102  173               9181X         MOV     A,E
     107.103  027               9182X         RAL
     107.104  137               9183X         MOV     E,A
     107.105  172               9184X         MOV     A,D
     107.106  027               9185X         RAL
     107.107  127               9186X         MOV     D,A
     107.110  171               9187X         MOV     A,C
     107.111  027               9188X         RAL
     107.112  117               9189X         MOV     C,A
     107.113  311               9190X         RET


                                9192X **      PMD - PRESET MULTIPLY/DIVIED
                                9193X *
                                9194X *       ENTRY (DE) = EXPONENT INSTRUCTIONS (FOR MULTIPLY OR DIVIDE)
                                9195X *             (HL) = ADDRESS OF "Y" VALUE
                                9196X *       EXIT    (C,D,E) = X VALUES
                                9197X *             'Z' SET IF VALUE ZERO
                                9198X *             'C' SET OF OVERFLOW
                                9199X *             'L' = NEW SIGN
                                9200X *             (A) = NEW EXPONENT
                                9201X
                                9202X
     107.114                    9203X PMD     EQU     *
     107.114  345               9204X         PUSH    H
     107.115  353               9205X         XCHG                    (HL) = EXPONENT INSTRUCTIONS
     107.116  042 164 107       9206X         SHLD    PMDB
     107.121  041 202 042       9207X         LXI     H,ACCX
     107.124  072 204 042       9208X         LDA     ACCX+2
     107.127  062 152 107       9209X         STA     PMDA            SET SIGN
     107.132  247               9210X         ANA     A
     107.133  374 204 107       9211X         CM      PMD2            IF MUST COMPLEMENT X
     107.136  341               9212X         POP     H               (HL) = ADDRESS OF Y
     107.137  315 250 107       9213X         CALL    LDD             LOAD NUMBER
     107.142  171               9214X         MOV     A,C
     107.143  157               9215X         MOV     L,A             (L) = SIGN
     107.144  247               9216X         ANA     A
     107.145  374 260 107       9217X         CM      TCV             IS NEGATIVE
     107.150  175               9218X         MOV     A,L             (A) = SIGN
     107.151  356 000           9219X         XRI     0               COMPARE SIGNS
     107.152                    9220X PMDA    EQU     *-1             SIGN OF X
     107.153  157               9221X         MOV     L,A
     107.154  170               9222X         MOV     A,B
     107.155  247               9223X         ANA     A
     107.156  310               9224X         RZ                      IF ZERO
     107.157  072 205 042       9225X         LDA     ACCX+3
     107.162  247               9226X         ANA     A
```

```
107.163  310            9227X          RZ                      IF ZERO
107.164  200            9228X PMDB     ADD     B               IF DIVIDE, = 'SUB  B'
107.165  000            9229X          NOP                     = 'CMC'
107.166  107            9230X          MOV     B,A             (B) = SUM OF 2
                        9231X
                        9232X *        SEE IF EXPONENT OVERFLOW
                        9233X
107.167  037            9234X          RAR
107.170  250            9235X          XRA     B
107.171  170            9236X          MOV     A,B             (A) = SUM OF EXPONENTS
107.172  362 200 107     9237X          JP      PMD1            OVERFLOW OR UNDERFLOW
107.175  356 200        9238X          XRI     200Q
107.177  311            9239X          RET                     'Z' SET IF UNDERFLOW
                        9240X
                        9241X *        OVERFLOW OR UNDERFLOW.
                        9242X
107.200  007            9243X PMD1     RLC                     'C' IF OVERFLOW
107.201  330            9244X          RC
107.202  257            9245X          XRA     A               UNDERFLOW. SET *Z*
107.203  311            9246X          RET                     EXIT
                        9247X
                        9248X
                        9249X *        COMPLEMENT ACCX TO A POSITIVE NUMBER
                        9250X
107.204  315 245 107     9251X PMD2     CALL    LDX
107.207  315 260 107     9252X          CALL    TCV
107.212  303 245 106     9253X          JMP     STX             STORE AND RETURN


                        9255X **       SPE - SETUP PACKAGE ENTRY.
                        9256X *
                        9257X *        SPE IS CALLED UPON ENTRY TO THE FLOATING POINT PACKAGE.
                        9258X *
                        9259X *        IT SAVES THE REGISTERS ON THE STACK, SETS UP A RETURN ADDRESS
                        9260X *        TO A RESTORE REGISTER ROUTINE, AND THEN ENTERS THE SELECTED
                        9261X *        ROUTINE.
                        9262X *
                        9263X *        ENTRY (SP+0) = ADDRESS TO RETURN CONTROL TO
                        9264X *        EXIT    REGISTERS ON STACK, *SPEX* SET AS RETURN ADDRESS
                        9265X *        USES    B,C,H,L
                        9266X
                        9267X
107.215  343            9268X SPE      XTHL                    SAVE H
107.216  325            9269X          PUSH    D               SAVE D
107.217  305            9270X          PUSH    B               SAVE B
107.220  001 225 107     9271X          LXI     B,SPEX
107.223  305            9272X          PUSH    B               SET 'RETURN ADDRESS'
107.224  351            9273X          PCHL                    ENTER ROUTINE
                        9274X
                        9275X *        RETURN FROM ROUTINE, RESTORE REGISTERS AND RETURN TO CALLER.
                        9276X
107.225  301            9277X SPEX     POP     B
107.226  321            9278X          POP     D
107.227  341            9279X          POP     H
```

```
107.230  311          9280X          RET


                       9282X **       SRS - SHIFT RIGHT WITH SIGN EXTEND.
                       9283X *
                       9284X *        ENTRY (C,D,E) = VALUE
                       9285X *        EXIT    (C,D,E) SHIFTED RIGHT 1 BIT
                       9286X *        USES    A
                       9287X
107.231                9288X SRS      EQU     *
107.231  171          9289X          MOV     A,C
107.232  027          9290X SRS.     RAL
107.233  171          9291X SRS..    MOV     A,C
107.234  037          9292X          RAR
107.235  117          9293X          MOV     C,A
107.236  172          9294X          MOV     A,D
107.237  037          9295X          RAR
107.240  127          9296X          MOV     D,A
107.241  173          9297X          MOV     A,E
107.242  037          9298X          RAR
107.243  137          9299X          MOV     E,A
107.244  311          9300X          RET


                       9302X **       LDX - LOAD X VALUE INTO REGISTERS
                       9303X *
                       9304X *        ENTRY NONE
                       9305X *        EXIT    (BCDE) = (ACCX)
                       9306X *        USES    ALL
                       9307X
                       9308X
107.245  041 202 042  9309X LDX      LXI     H,ACCX


                       9311X **       LDD - LOAD VALUE INTO REGISTERS.
                       9312X *
                       9313X *        ENTRY (HL) = ADDRESS OF VALUE
                       9314X *        EXIT    (B,C,D,E) = X VALUE
                       9315X
                       9316X
107.250  136          9317X LDD      MOV     E,M
107.251  043          9318X          INX     H
107.252  126          9319X          MOV     D,M
107.253  043          9320X          INX     H
107.254  116          9321X          MOV     C,M
107.255  043          9322X          INX     H
107.256  106          9323X          MOV     B,M
107.257  311          9324X          RET
```

```
                                9326X **      TCV - TWOS COMPLEMENT VALUE.
                                9327X *
                                9328X *       ENTRY (BCDE) = VALUE
                                9329X
        107.260                 9330X TCV     EQU     *
        107.260  171           9331X         MOV     A,C
        107.261  057           9332X         CMA
        107.262  117           9333X         MOV     C,A
        107.263  172           9334X         MOV     A,D
        107.264  057           9335X         CMA
        107.265  127           9336X         MOV     D,A
        107.266  173           9337X         MOV     A,E
        107.267  057           9338X         CMA
        107.270  137           9339X         MOV     E,A
        107.271  034           9340X         INR     E
        107.272  300           9341X         RNZ
        107.273  024           9342X         INR     D
        107.274  300           9343X         RNZ
        107.275  171           9344X         MOV     A,C             (A) = SIGN
        107.276  014           9345X         INR     C
        107.277  247           9346X         ANA     A
        107.300  372 213 105   9347X         JM      NRM             IF POSITIVE TO NEGATIVE, NORMALIZE
        107.303  171           9348X         MOV     A,C             WAS NEGATIVE TO POSITIVE, MAY NEED RIGHT SH
        107.304  247           9349X         ANA     A
        107.305  360           9350X         RP                      DONT NEED SHIFT
        107.306  004           9351X         INR     B
        107.307  312 136 070   9352X         JZ      ERR.OV          IF OVERFLOW
        107.312  303 233 107   9353X         JMP     SRS.,           SHIFT RIGHT AND EXIT


                                9355X **      RVU - ROUND VALUE UP.
                                9356X *
                                9357X *       RVU IS CALLED TO ADD ONE BIT TO THE VALUE.
                                9358X *
                                9359X *       ENTRY (BCDE) = VALUE
                                9360X *       EXIT    (BCDE) ADJUSTED
                                9361X *       USES    A,F,B,C,D,E
                                9362X
                                9363X
        107.315  034           9364X RVU     INR     E
        107.316  300           9365X         RNZ                     NO CARRY
        107.317  024           9366X         INR     D
        107.320  300           9367X         RNZ                     NO CARRY
        107.321  014           9368X         INR     C
        107.322  311           9369X         RET
        107.323                9370          XTEXT   FPC
```

```
                                9373X **      ATF - ASCII TO FLOATING.
                                9374X *
                                9375X *       ATF CONVERTS AN ASCII STRING INTO A FLOATING POINT VALUE
                                9376X *       IN ACCX.
                                9377X *
                                9378X *       SYNTAX
                                9379X *
                                9380X *        NNNN [.NNN] [E [+-] NN]
                                9381X *
                                9382X *       NO LEADING BLANKS ALLOWED, A SINGLE LEADING
                                9383X *       '-' IS ALLOWED, AND PROCESSED.
                                9384X *
                                9385X *       ENTRY   (HL) = ADDRESS OF TEXT
                                9386X *       EXIT    (HL) UPDATED
                                9387X *               (ACCX) = VALUE
                                9388X *       USES    A,F,H,L
                                9389X
                                9390X
      107.323                   9391X ATF     EQU     *
      107.323  305             9392X         PUSH    B               SAVE REGISTERS
      107.324  325             9393X         PUSH    D
      107.325  176             9394X         MOV     A,M             SEE IF '-'
      107.326  376 055         9395X         CPI     '-'
      107.330  365             9396X         PUSH    PSW             SAVE RESULTS UNTIL THE VERY END
      107.331  302 335 107     9397X         JNE     ATF0            NOT -
      107.334  043             9398X         INX     H               SKIP '-'
                                9399X
                                9400X *       SET FLAG AS TO WHETHER THERE IS A NUMBER PRESENT        /WCZ080880/
                                9401X *       BEFORE A POSSIBLE [E+-NN].                              /WCZ080880/
                                9402X
      107.335  345             9403X ATF0    PUSH    H               SAVE TEXT POINTER
      107.336  076 001         9404X         MVI     A,1             ASSUME DIGITS APPEAR BEFORE     /WCZ080880/
      107.340  062 300 110     9405X         STA     ATFB             POSSIBLE [E+-NN]               /WCZ080880/
      107.343  176             9406X         MOV     A,M                                             /WCZ080880/
      107.344  376 056         9407X         CPI     '.'             CHECK FOR '.'                   /WCZ080880/
      107.346  302 352 107     9408X         JNE     ATF0.1                                          /WCZ080880/
      107.351  043             9409X         INX     H               SKIP OVER '.'                   /WCZ080880/
      107.352                   9410X ATF0.1  EQU     *                                               /WCZ080880/
      107.352  315 362 111     9411X         CALL    $CVD            CHECK FOR VALID DIGIT           /WCZ080880/
      107.355  322 364 107     9412X         JNC     ATF0.2            BR IF THERE IS ONE            /WCZ080880/
      107.360  257             9413X         XRA     A               INDICATE NO DIGITS APPEAR       /WCZ080880/
      107.361  062 300 110     9414X         STA     ATFB              BEFORE POSSIBLE [E+-NN]       /WCZ080880/
                                9415X
      107.364                   9416X ATF0.2  EQU     *                                               /WCZ080880/
      107.364  341             9417X         POP     H               RESTORE TEXT POINTER            /WCZ080880/
      107.365  345             9418X         PUSH    H               CONTINE TO SAVE IT              /WCZ080880/
      107.366  006 006         9419X         MVI     B,6             DIGIT COUNT+2
                                9420X
                                9421X *       COUNT # OF SIGNIFICANT DIGITS
                                9422X
      107.370  005             9423X ATF1    DCR     B
      107.371  312 103 110     9424X         JZ      ATF3            TOO MANY DIGITS
      107.374  176             9425X         MOV     A,M
      107.375  043             9426X         INX     H
      107.376  376 056         9427X         CPI     '.'
      110.000  312 370 107     9428X         JE      ATF1            DONT COUNT DECIMAL POINT
```

```
110.003  376 060      9429X          CPI     '0'
110.005  332 015 110  9430X          JC      ATF1.5          NOT DIGIT
110.010  376 072      9431X          CPI     '9'+1
110.012  332 370 107  9432X          JC      ATF1            IS DIGIT
                      9433X
                      9434X *        WILL DECODE NUMBER AS DECIMAL INTEGER
                      9435X
110.015  341          9436X ATF1.5   POP     H               (HL) = START OF NUMBER
110.016  021 000 000  9437X          LXI     D,0
110.021  315 244 111  9438X          CALL    DDN1            DECODE DECIMAL NUMBER
110.024  006 000      9439X          MVI     B,0             ZERO AFTER-DECIMAL COUNT
110.026  076 056      9440X          MVI     A,'.'
110.030  276          9441X          CMP     M
110.031  314 275 111  9442X          CE      DDN2            DECODE FRACTIONAL, IF ANY
110.034  305          9443X          PUSH    B               SAVE DP COUNT
110.035  112          9444X          MOV     C,D
110.036  123          9445X          MOV     D,E
110.037  257          9446X          XRA     A               CLEAR CARRY
110.040  137          9447X          MOV     E,A             (E) = 0
110.041  103          9448X          MOV     B,E             (B) = 0
110.042  171          9449X          MOV     A,C
110.043  262          9450X          ORA     D
110.044  312 065 110  9451X          JZ      ATF2.5          IS 0
110.047  006 217      9452X          MVI     B,217Q
                      9453X
                      9454X *        NORMALIZE
                      9455X
110.051  172          9456X ATF2     MOV     A,D
110.052  027          9457X          RAL
110.053  127          9458X          MOV     D,A
110.054  171          9459X          MOV     A,C
110.055  027          9460X          RAL
110.056  117          9461X          MOV     C,A
110.057  005          9462X          DCR     B
110.060  346 100      9463X          ANI     100Q
110.062  312 051 110  9464X          JZ      ATF2            MORE TO GO
110.065  353          9465X ATF2.5   XCHG
110.066  042 202 042  9466X          SHLD    ACCX            SET LOW-ORDER
110.071  140          9467X          MOV     H,B
110.072  151          9468X          MOV     L,C
110.073  042 204 042  9469X          SHLD    ACCX+2          SET HIGH-ORDER
110.076  353          9470X          XCHG                    (HL) = NEXT BYTE ADDR
110.077  301          9471X          POP     B               (B) = SCALE COUNT
110.100  303 132 110  9472X          JMP     ATF5            CHECK FOR EXPONENT
                      9473X
                      9474X *        MUST DECODE VIA FLOATING NUMBERS.
                      9475X
110.103  315 240 106  9476X ATF3     CALL    MUL5            CLEAR ACCX
110.106  006 207      9477X          MVI     B,227Q-16
110.110  041 210 042  9478X          LXI     H,ACCY
110.113  315 250 106  9479X          CALL    STO             SETUP Y
110.116  341          9480X          POP     H               (HL) = NUMBER START
110.117  315 301 111  9481X ATF4     CALL    DFD             DECODE FLOATING DECIMAL
110.122  006 000      9482X          MVI     B,0             CLEAR DP COUNT
110.124  076 056      9483X          MVI     A,'.'
110.126  276          9484X          CMP     M
```

```
110.127  314 331 111  9485X          CE      DFDI            IF FRACTIONAL PART
                      9486X
                      9487X *        HAVE FLOATING VALUE. LOOK FOR E+-NN
                      9488X *        (B) = DP SCALE COUNT
                      9489X
110.132  076 105      9490X ATF5     MVI     A,'E'
110.134  276          9491X          CMP     M
110.135  076 000      9492X          MVI     A,0             ASSUME HAVE ONE
110.137  302 221 110  9493X          JNE     ATF8            HAVE NONE
110.142  127          9494X          MOV     D,A             SAVE REG A                      /WCZ080880/
110.143  072 300 110  9495X          LDA     ATFB            0, DID A NUMBER APPEAR          /WCZ080880/
110.146  247          9496X          ANA     A                   BEFORE [E+-NN]              /WCZ080880/
110.147  172          9497X          MOV     A,D             RESTORE REG A                   /WCZ080880/
110.150  312 221 110  9498X          JZ      ATF8            BR IF NOT                       /WCZ080880/
110.153  043          9499X          INX     H               INCREMENT PAST 'E'
                      9500X
                      9501X *        DECODE EXPONENT.
                      9502X
110.154  176          9503X          MOV     A,M             (A) = NEXT CHARACTER
110.155  326 053      9504X          SUI     '+'
110.157  312 173 110  9505X          JZ      ATF6            IS +
110.162  376 002      9506X          CPI     '-'-'+'
110.164  076 200      9507X          MVI     A,80H           ASSUME -
110.166  312 173 110  9508X          JE      ATF6            IS -
110.171  257          9509X          XRA     A               IS NONE. USE +
110.172  053          9510X          DCX     H
110.173  043          9511X ATF6     INX     H               ADVANCE PAST + OR -
110.174  365          9512X          PUSH    PSW             SAVE SIGN
110.175  110          9513X          MOV     C,B             (C) = DP COUNT
110.176  315 233 111  9514X          CALL    DDN             DECODE DECIMAL DIGITS
110.201  101          9515X          MOV     B,C             RESTORE DP COUNT
110.202  343          9516X          XTHL                    SAVE (HL), (H) = EXPONENT SIGN
110.203  172          9517X          MOV     A,D
110.204  247          9518X          ANA     A
110.205  302 122 070  9519X          JNZ     ERR.IN          IF TOO LARGE
110.210  174          9520X          MOV     A,H
110.211  027          9521X          RAL                     'C' SET IF NEGATIVE
110.212  173          9522X          MOV     A,E
110.213  322 220 110  9523X          JNC     ATF7            NOT NEGATIVE
110.216  057          9524X          CMA
110.217  074          9525X          INR     A
110.220  341          9526X ATF7     POP     H
110.221  200          9527X ATF8     ADD     B               (A) = SCALE COUNT
110.222  312 271 110  9528X          JZ      ATF11           NO SCALEING
110.225  345          9529X          PUSH    H               SAVE (HL)
110.226  041 327 105  9530X          LXI     H,MUL           ASSUME *                        /78.10.GC/
110.231  042 260 110  9531X          SHLD    ATFA                                            /78.10.GC/
110.234  041 215 112  9532X          LXI     H,FP10.         ASSUME *10
110.237  362 254 110  9533X          JP      ATF9            IS POSITIVE
110.242  345          9534X          PUSH    H                                               /78.10.GC/
110.243  041 264 106  9535X          LXI     H,DIV                                           /78.10.GC/
110.246  042 260 110  9536X          SHLD    ATFA                                            /78.10.GC/
110.251  341          9537X          POP     H                                               /78.10.GC/
110.252  057          9538X          CMA
110.253  074          9539X          INR     A               (A) = COUNT
110.254  117          9540X ATF9     MOV     C,A             (C) = SCALE COUNT
```

```
110.255  305           9541X ATF10   PUSH    B
110.256  345           9542X         PUSH    H
110.257  315 327 105   9543X         CALL    MUL             SCALE
110.260               9544X ATFA    EQU     *-2
110.262  341           9545X         POP     H
110.263  301           9546X         POP     B
110.264  015           9547X         DCR     C
110.265  302 255 110   9548X         JNZ     ATF10           IF MORE TO GO
110.270  341           9549X         POP     H               RESTORE (HL)
                       9550X
                       9551X *       DONE.
                       9552X
110.271  361           9553X ATF11   POP     PSW             (PSW) = RESULTS OF EARLY '-' CHECK
110.272  314 302 105   9554X         CE      FPNEG           MUST NEGATE
110.275  321           9555X         POP     D
110.276  301           9556X         POP     B
110.277  311           9557X         RET
                       9558X
110.300               9559X ATFB    DS      1 FLAG := <>0 DIGITS APPEAR BEFORE POSSIBLE E+-NN /WCZ080880/
```

```
                                9562X **      FTA - FLOATING TO ASCII.
                                9563X *
                                9564X *       FTA CONVERTS A FLOATING POINT NUMBER INTO AN ASCII
                                9565X *       REPRESENTATION.
                                9566X *
                                9567X *       ENTRY   (ACCX) = VALUE
                                9568X *               (HL) = ADDRESS TO STORE TEXT
                                9569X *       EXIT    (A) = LENGTH OF STRING DECODED
                                9570X *               (DE) = ADDRESS OF LAST BYTE
                                9571X *       USES    A,F,D,E
                                9572X
                                9573X
     110.301                    9574X FTA     EQU     *
     110.301  305               9575X         PUSH    B
     110.302  345               9576X         PUSH    H
     110.303  066 040           9577X         MVI     M,' '           INSURE LEADING BLANK
     110.305  072 204 042       9578X         LDA     ACCX+2
     110.310  247               9579X         ANA     A               TEST VALUE
     110.311  362 323 110       9580X         JP      FTA1
     110.314  043               9581X         INX     H               ADD MINUS SIGN
     110.315  066 055           9582X         MVI     M,'-'
     110.317  315 302 105       9583X         CALL    FPNEG           INVERT IT
     110.322  264               9584X         ORA     H               CLEAR 'Z'
     110.323  043               9585X FTA1    INX     H
     110.324  006 001           9586X         MVI     B,1             (B) = EXPONENT
     110.326  312 020 111       9587X         JZ      FTA2.7          IS 0
                                9588X
                                9589X *       SCALE NUMBER
                                9590X
     110.331  021 331 110       9591X FTA2    LXI     D,FTA2
     110.334  325               9592X         PUSH    D               SET 'RETURN ADDRESS'
     110.335  021 215 112       9593X         LXI     D,FP10.
     110.340  072 205 042       9594X         LDA     ACCX+3
     110.343  005               9595X         DCR     B
     110.344  376 201           9596X         CPI     201Q
     110.346  332 323 105       9597X         JC      FPMUL           ACCX = ACCX * 10
     110.351  004               9598X         INR     B
     110.352  004               9599X         INR     B
     110.353  326 205           9600X         SUI     205Q
     110.355  322 260 106       9601X         JNC     FPDIV           ACCX = ACCX / 10
     110.360  074               9602X         INR     A
     110.361  372 374 110       9603X         JM      FTA2.5          IS SCALED
     110.364  072 204 042       9604X         LDA     ACCX+2
     110.367  376 120           9605X         CPI     120Q
     110.371  322 260 106       9606X         JNC     FPDIV
     110.374  005               9607X FTA2.5  DCR     B
     110.375  321               9608X         POP     D               DISCARD 'RETURN ADDRESS'
                                9609X
                                9610X *       ROUND NUMBER
                                9611X
     110.376  072 204 042       9612X         LDA     ACCX+2
     111.001  365               9613X         PUSH    PSW             SAVE HIGH ORDER PART
     111.002  021 227 111       9614X         LXI     D,FTAA
     111.005  315 352 104       9615X         CALL    FPADD           ROUND UP
     111.010  321               9616X         POP     D               (D) = OLD MANTISSA
     111.011  072 204 042       9617X         LDA     ACCX+2
```

```
111.014  272          9618X          CMP     D
111.015  302 331 110  9619X          JNE     FTA2            CAUSED MAJOR CHANGE, ROUND AGAIN
                      9620X
                      9621X *        SCALED. (B) = DECIMAL PLACE
                      9622X
111.020               9623X FTA2.7   EQU     *
111.020  170          9624X          MOV     A,B
111.021  376 007      9625X          CPI     7
111.022               9626X FTAC     EQU     *-1             SCIENTIFIC/FIXED FLAG
111.023  365          9627X          PUSH    PSW
111.024  332 031 111  9628X          JC      FTA3            USED FIXED NOTATION
111.027  006 001      9629X          MVI     B,1             USE SCIENTIFIC NOATATION
111.031  016 007      9630X FTA3     MVI     C,7             (C) = DIGIT COUNT (+1 FOR .)
111.032               9631X FTAD     EQU     *-1
111.033  004          9632X          INR     B               (B) = DIGITS BEFORE DP (+1)
                      9633X
                      9634X *        SEE IF TO PLACE DECIMAL POINT
                      9635X
111.034  005          9636X FTA4     DCR     B
111.035  302 043 111  9637X          JNZ     FTA4.5          NOT TIME FOR DECIMAL POINT
111.040  066 056      9638X          MVI     M,'.'
111.042  043          9639X          INX     H
111.043  015          9640X FTA4.5   DCR     C
111.044  312 130 111  9641X          JZ      FTA8.5          IF ROOM FOR NO MORE DIGITS
                      9642X
                      9643X *        DECODE DIGIT
                      9644X
111.047  345          9645X FTA5     PUSH    H
111.050  041 205 042  9646X          LXI     H,ACCX+3
111.053  126          9647X          MOV     D,M
111.054  172          9648X          MOV     A,D
111.055  376 201      9649X          CPI     201Q
111.057  076 000      9650X          MVI     A,0
111.061  332 107 111  9651X          JC      FTA7.5          IF NO DIGIT FOR THIS PLACE
111.064  257          9652X          XRA     A
111.065  067          9653X FTA6     STC
111.066  037          9654X          RAR
111.067  025          9655X          DCR     D
111.070  372 065 111  9656X          JM      FTA6            GENERATE MASK OF 1S FOR SIG BITS
111.073  126          9657X          MOV     D,M             (D) = EXPONENT
111.074  053          9658X          DCX     H
111.075  246          9659X          ANA     M               (A) = VALUE
111.076  137          9660X          MOV     E,A
111.077  256          9661X          XRA     M
111.100  167          9662X          MOV     M,A
111.101  173          9663X          MOV     A,E
111.102  007          9664X FTA7     RLC
111.103  025          9665X          DCR     D
111.104  372 102 111  9666X          JM      FTA7            ROTATE VALUE LOW ORDER
111.107  306 060      9667X FTA7.5   ADI     '0'             (A) = DIGIT
111.111  341          9668X          POP     H
111.112  167          9669X          MOV     M,A
111.113  043          9670X          INX     H
111.114  315 202 105  9671X          CALL    FPNRM           NORMALIZE
111.117               9672X FTA8     EQU     *
111.117  021 215 112  9673X          LXI     D,FP10.
```

```
111.122  315 323 105  9674X          CALL    FPMUL
111.125  303 034 111  9675X          JMP     FTA4            HAVE NOT PRINTED DECIMAL YET
                      9676X
                      9677X *        ADD EXPONENT, IF NECESSARY
                      9678X
111.130  361          9679X FTA8.5   POP     PSW
111.131  332 177 111  9680X          JC      FTA12           NOT SCIENTIFIC
                      9681X
                      9682X *        ADD E+-NN
                      9683X
111.134  066 105      9684X          MVI     M,'E'
111.136  043          9685X          INX     H
111.137  066 053      9686X          MVI     M,'+'
111.141  075          9687X          DCR     A
111.142  362 151 111  9688X          JP      FTA9
111.145  066 055      9689X          MVI     M,'-'
111.147  057          9690X          CMA
111.150  074          9691X          INR     A
111.151  043          9692X FTA9     INX     H
111.152  066 057      9693X          MVI     M,'0'-1
111.154  064          9694X FTA10    INR     M               DECODE 10S DIGIT
111.155  326 012      9695X          SUI     10
111.157  362 154 111  9696X          JP      FTA10
111.162  306 012      9697X          ADI     10
111.164  043          9698X          INX     H
111.165  066 057      9699X          MVI     M,'0'-1
111.167  064          9700X FTA11    INR     M
111.170  075          9701X          DCR     A
111.171  362 167 111  9702X          JP      FTA11
111.174  303 214 111  9703X          JMP     FTA13           DONT TRIM TRAILING THINGS
                      9704X
                      9705X *        DONE STRIP TRAILING ZEROS.
                      9706X
111.177  053          9707X FTA12    DCX     H
111.200  176          9708X          MOV     A,M
111.201  376 060      9709X          CPI     '0'
111.203  312 177 111  9710X          JE      FTA12
111.206  376 056      9711X          CPI     '.'
111.210  302 214 111  9712X          JNE     FTA13           NOT .
111.213  053          9713X          DCX     H
111.214  043          9714X FTA13    INX     H
111.215  066 040      9715X          MVI     M,' '           ADD TRAILING BLANK
111.217  043          9716X          INX     H
                      9717X
111.220  321          9718X          POP     D               (DE) = NUMBER FWA
111.221  175          9719X          MOV     A,L
111.222  223          9720X          SUB     E
111.223  353          9721X          XCHG
111.224  033          9722X          DCX     D
111.225  301          9723X          POP     B
111.226  311          9724X          RET
                      9725X
111.227  051 000 000  9726X FTAA     DB      51Q,0,0,200Q    5.E-7
```

```
                                9728X **    DDN - DECODE DECIMAL NUMBER.
                                9729X *
                                9730X *     ENTRY   (HL) = TEXT POINTER
                                9731X *     EXIT    (DE) = VALUE (IF NON-NULL)
                                9732X *             (HL) UPDATED
                                9733X *             TO 'DDNERR' IF NULL
                                9734X *     USES    ALL
                                9735X
                                9736X
111,233                         9737X DDN   EQU     *
111,233  315 362 111            9738X       CALL    $CVD            CHECK DECIMAL VALUE
111,236  332 122 070            9739X       JC      DDNERR          HAVE NO DECIMAL DIGITS
111,241  021 000 000            9740X       LXI     D,0             (DE) = ACCUMULATOR
111,244  315 362 111            9741X DDN1  CALL    $CVD            CHECK DECIMAL VALUE
111,247  330                    9742X       RC                      NO MORE DIGITS
111,250  345                    9743X       PUSH    H               SAVE TEXT POINTER
111,251  353                    9744X       XCHG            (HL) =MULTIPLIER
111,252  051                    9745X       DAD     H               (HL) = X*2
111,253  124                    9746X       MOV     D,H
111,254  135                    9747X       MOV     E,L
111,255  051                    9748X       DAD     H               (HL) = X*4
111,256  051                    9749X       DAD     H               (HL) = X*8
111,257  031                    9750X       DAD     D               (HL) = X*10
111,260  332 122 070            9751X       JC      DDNERR          OVERFLOW
111,263  137                    9752X       MOV     E,A
111,264  026 000                9753X       MVI     D,0             (DE) = DIGIT VALUE
111,266  031                    9754X       DAD     D
111,267  332 122 070            9755X       JC      DDNERR          NO GOOD
111,272  353                    9756X       XCHG                    (DE) = VALUE
111,273  341                    9757X       POP     H
111,274  005                    9758X       DCR     B               COUNT DP
111,275  043                    9759X DDN2  INX     H
111,276  303 244 111            9760X       JMP     DDN1            ACCEPT ANOTHER


                                9762X **    DFD - DECODE FLOATING DECIMAL.
                                9763X *
                                9764X *     DFD PERFORMS THE EQUIVALENT TO DDN, BUT DOES IT IN
                                9765X *     THE FLOATING POINT ACCUMULATOR.
                                9766X *
                                9767X
                                9768X
111,301  315 362 111            9769X DFD   CALL    $CVD            CHECK VALID DEC
111,304  330                    9770X       RC                      NO GOOD
111,305  062 212 042            9771X       STA     ACCY+2
111,310  345                    9772X       PUSH    H
111,311  305                    9773X       PUSH    B               SAVE (B)
111,312  041 215 112            9774X       LXI     H,FP10.
111,315  315 327 105            9775X       CALL    MUL             SCALE ACCUM
111,320  041 210 042            9776X       LXI     H,ACCY
111,323  315 356 104            9777X       CALL    ADD             ADD VALUE
111,326  301                    9778X       POP     B
111,327  341                    9779X       POP     H
111,330  005                    9780X       DCR     B               COUNT DIGIT
```

```
                                9781X DFD1   INX     H
111.331  043                    9782X        JMP     DFD             ANOTHER DIGIT
111.332  303 301 111            9783         LOF     C
111.335                         9784         XTEXT   WER


                                9786X **     $WER - WRITE ENABLE RAM.
                                9787X *
                                9788X *      $WER IS CALLED TO ENABLE WRITTING TO THE H17 CONTROLLER'S
                                9789X *      RAM AREA.
                                9790X *
                                9791X *      ENTRY   NONE
                                9792X *      EXIT    NONE
                                9793X *      USES    NONE
                                9794X
                                9795X
031.241                         9796X $WER   EQU     31241A          IN H17 ROM


                                9798X **     $WDR - WRITE DISABLE RAM.
                                9799X *
                                9800X *      $WDR IS CALLED TO DISABLE WRITTING TO THE H17 CONTROLLER'S
                                9801X *      RAM AREA.
                                9802X *
                                9803X *      ENTRY   NONE
                                9804X *      EXIT    NONE
                                9805X *      USES    NONE
                                9806X
                                9807X
031.222                         9808X $WDR   EQU     31222A          IN H17 ROM
070.122                         9809  DDNERR EQU     ERR.IN
111.335                         9810         XTEXT   CLL


                                9812X **     CLL - COMPUTE LINE LENGTH.
                                9813X *
                                9814X *      CLL COUNTS THE NUMBER OF CHARACTERS IN A SOURCE LINE.
                                9815X *      THE LINE IS TERMINATED BY A 00 BYTE; THE 00 BYTE IS ENCLUDED
                                9816X *      IN THE COUNT.
                                9817X *
                                9818X *      ENTRY   (HL) = FWA OF LINE
                                9819X *      EXIT    (HL) UNCHANGED
                                9820X *              (A) = LENGTH OF LINE
                                9821X *      USES    A,F
                                9822X
                                9823X
111.335  345                    9824X $CLL   PUSH    H               SAVE STARTING ADDRESS
111.336  325                    9825X        PUSH    D
111.337  026 000                9826X        MVI     D,0
                                9827X
```

```
111.341  176          9828X CLL1   MOV     A,M
111.342  024          9829X        INR     D
111.343  247          9830X        ANA     A
111.344  043          9831X        INX     H
111.345  302 341 111  9832X        JNZ     CLL1          SCAN FOR END
111.350  172          9833X        MOV     A,D
111.351  321          9834X        POP     D
111.352  341          9835X        POP     H
111.353  311          9836X        RET
111.354               9837         XTEXT   CRLF


                      9839X **     $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                      9840X *
                      9841X *      $CRLF IS USED TO GENERATE PADDED CRLF'S.
                      9842X *
                      9843X *      ENTRY   NONE
                      9844X *      EXIT    (A) = 0
                      9845X *      USES    A,F
                      9846X
                      9847X
111.354  076 012      9848X $CRLF  MVI     A,NL
111.356  377 002      9849X        DB      SYSCALL+.SCOUT
111.360  257          9850X        XRA     A
111.361  311          9851X        RET
111.362               9852         XTEXT   CVD


                      9854X **     $CVD - CHECK FOR VALID DIGIT.
                      9855X *
                      9856X *      CVD EXAMINES A DIGIT TO SEE IF IT IS A VALID DECIMAL DIGIT.
                      9857X *
                      9858X *      ENTRY   (HL) = ADDRESS OF CHARACTER
                      9859X *      EXIT    'C' SET IF ILLEGAL
                      9860X *                (A) = VALUE
                      9861X *      USES    A,F
                      9862X
                      9863X
111.362  176          9864X $CVD   MOV     A,M           (A) = CHARACTER
111.363  326 060      9865X $CVD.  SUI     '0'
111.365  330          9866X        RC                    ILLEGAL
111.366  376 012      9867X        CPI     9+1
111.370  077          9868X        CMC
111.371  311          9869X        RET
111.372               9870         XTEXT   ZERO
```

```
                                9872X **      $ZERO - ZERO MEMORY
                                9873X *
                                9874X *       $ZERO ZEROS A BLOCK OF MEMORY.
                                9875X *
                                9876X *       ENTRY   (HL) = ADDRESS
                                9877X *               (B) = COUNT
                                9878X *       EXIT    (A) = 0
                                9879X *       USES    A,B,F,H,L
                                9880X
                                9881X
        031,212                 9882X $ZERO   EQU     31212A          IN H17 ROM
        111,372                 9883          XTEXT   SOB


                                9885X **      $SOB - SKIP OVER BLANKS.
                                9886X *
                                9887X *       $SOB IS CALLED TO SKIP AN ARBITRARILY LONG STRING OF BLANKS AND TABS.
                                9888X *
                                9889X *       ENTRY   (HL) = FWA OF (POSSIBLE) BLANK STRING
                                9890X *       EXIT    (HL) = LWA+1 OF BLANK STRING (UNCHANGED IF NO BLANKS)
                                9891X *               (A) = FIRST NON-BLANK, NON-TAB CHARACTER EEN
                                9892X *       USES    A,F,H,L
                                9893X
                                9894X
        111,372  053           9895X $SOB    DCX     H               PRE-DECREMENT
        111,373  043           9896X $SOB1   INX     H
        111,374  176           9897X         MOV     A,M
        111,375  376 040       9898X         CPI     ' '
        111,377  312 373 111   9899X         JE      $SOB1           GOT BLANK
        112,002  376 011       9900X         CPI     TAB
        112,004  312 373 111   9901X         JE      $SOB1           GOT TAB
        112,007  311           9902X         RET
        112,010                 9903          XTEXT   CDEHL


                                9905X **      $CDEHL - COMPARE (DE) TO (HL)
                                9906X *
                                9907X *       $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                                9908X *
                                9909X *       ENTRY   NONE
                                9910X *       EXIT    'Z' SET IF (DE) = (HL)
                                9911X *       USES    A,F
                                9912X
                                9913X
        030,216                 9914X $CDEHL  EQU     30216A          IN H17 ROM
        112,010                 9915          XTEXT   HLCPDE
                                9916X **      HLCPDE  - (HL) COMPARED TO (DE)
                                9917X *
                                9918X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
                                9919X *
                                9920X *       ENTRY:  (HL)&(DE) SET UP
                                9921X *
```

```
                        9922X *       EXIT:   (PSW)   =
                        9923X *                        'Z' SET   IF (HL) =  (DE)
                        9924X *                        'C' SET   IF (HL) <  (DE)
                        9925X *                        'C' CLEAR IF (HL) >= (DE)
                        9926X *
                        9927X *
                        9928X *       USES:   (PSW)
                        9929X *
                        9930X
    112.010  174        9931X HLCPDE  MOV     A,H
    112.011  272        9932X         CMP     D       'C' SET => (A) < (D)
    112.012  300        9933X         RNZ
    112.013  175        9934X         MOV     A,L
    112.014  273        9935X         CMP     E       'C' SET => (L) < (E)
    112.015  311        9936X         RET
    112.016             9937          XTEXT   DU66



                        9939X **      $DU66 - UNSIGNED 16 / 16 DIVIDE.
                        9940X *
                        9941X *       (HL) = (BC)/(DE)
                        9942X *
                        9943X *       ENTRY   (BC), (DE) PRESET
                        9944X *       EXIT    (HL) = RESULT
                        9945X *               (DE) = REMAINDER
                        9946X *       USES    ALL
                        9947X
                        9948X
    030.106             9949X $DU66   EQU     30106A          IN H17 ROM
    112,016             9950          XTEXT   MU86



                        9952X **      $MU86 - MULTIPLY 8X16 UNSIGNED.
                        9953X *
                        9954X *       $MU86 MULTIPLIES A 16 BIT VALUE BY A 8
                        9955X *       BIT VALUE.
                        9956X *
                        9957X *       ENTRY   (A) = MULTIPLIER
                        9958X *               (DE) = MULTIPLICAND
                        9959X *       EXIT    (HL) = RESULT
                        9960X *               'Z' SET IF NOT OVERFLOW
                        9961X *       USES    A,F,H,L
                        9962X
                        9963X
    031.007             9964X $MU86   EQU     31007A          IN H17 ROM
    112.016             9965          XTEXT   ZEROS
```

```
                        9967X **      8 CONSTANT ZERO BYTES.
                        9968X
031.320                 9969X $ZEROS  EQU     31320A          IN H17 ROM
112.016                 9970          XTEXT   UDD


                        9972X **      $UDD - UNPACK DECIMAL DIGITS.
                        9973X *
                        9974X *       UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                        9975X *       DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                        9976X *
                        9977X *       ENTRY   (B,C) = ADDRESS VALUE
                        9978X *               (A) = DIGIT COUNT
                        9979X *               (H,L) = MEMORY ADDRESS
                        9980X *       EXIT    (HL) = (HL) + (A)
                        9981X *       USES    ALL
                        9982X
                        9983X
031.157                 9984X $UDD    EQU     31157A          IN H17 ROM
112.016                 9985          XTEXT   CCO


                        9987X **      $CCO - CLEAR CONTROL-O
                        9988X *
                        9989X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                        9990X *
                        9991X *       ENTRY   NONE
                        9992X *       EXIT    NONE
                        9993X *       USES    NONE
                        9994X
                        9995X
112.016  315 054 031   9996X $CCO    CALL    $SAVALL         SAVE REGISTERS
112.021  076 004       9997X         MVI     A,I.CONFL
112.023  001 001 000   9998X         LXI     B,CO.FLG        CLEAR CO.FLG
112.026  377 006       9999X         DB      SYSCALL,.CONSL
112.030  303 047 031  10000X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
112.033               10001          XTEXT   DADA


                       10003X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                       10004X *
                       10005X *       ENTRY   (H,L) = BEFORE VALUE
                       10006X *               (A) = BEFORE VALUE
                       10007X *       EXIT    (H,L) = (H,L) + (0,A)
                       10008X *               'C' SET IF OVERFLOW
                       10009X *       USES    F,H,L
                       10010X
                       10011X
030.072                10012X $DADA   EQU     30072A          IN H17 ROM
112.033                10013          XTEXT   DADA2
```

```
                        10015X **       $DADA. - ADD (0,A) TO (H,L)
                        10016X *
                        10017X *        ENTRY   NONE
                        10018X *        EXIT    (HL) = (HL) + (0A)
                        10019X *        USES    A,F,H,L
                        10020X
                        10021X
030.101                 10022X $DADA.   EQU     30101A          IN H17 ROM
112.033                 10023           XTEXT   MOVE


                        10025X **       $MOVE - MOVE DATA
                        10026X *
                        10027X *        $MOVE MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        10028X *        IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        10029X *        FIRST TO LAST.
                        10030X *
                        10031X *        IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        10032X *        LAST TO FIRST.
                        10033X *
                        10034X *        THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        10035X *
                        10036X *        ENTRY   (BC) = COUNT
                        10037X *                (DE) = FROM
                        10038X *                (HL) = TO
                        10039X *        EXIT    MOVED
                        10040X *                (DE) = ADDRESS OF NEXT FROM BYTE
                        10041X *                (HL) = ADDRESS OF NEXT *TO* BYTE
                        10042X *                'C' CLEAR
                        10043X *        USES    ALL
                        10044X
                        10045X
030.252                 10046X $MOVE    EQU     30252A          IN H17 ROM
112.033                 10047           XTEXT   MU66


                        10049X **       $MU66 - UNSIGNED 16X16 MULTIPLY.'
                        10050X *
                        10051X *        ENTRY (BC) = MULTIPLICAND
                        10052X *                (DE) = MULTIPLIER
                        10053X *        EXIT    (HL) = RESULT
                        10054X *                'Z' SET IF NOT OVERFLOW
                        10055X *        USES    ALL
                        10056X
                        10057X
030.337                 10058X $MU66    EQU     30337A          IN H17 ROM
112.033                 10059           XTEXT   TBLS
```

```
                                10061X **       $TBLS - TABLE SEARCH
                                10062X *
                                10063X *        TABLE FORMAT
                                10064X *
                                10065X *        DB      KEY1,VAL1,
                                10066X *        .       .
                                10067X *        .       .
                                10068X *        DB      KEYN,VALN
                                10069X *        DB      0
                                10070X *
                                10071X *        ENTRY   (A) = PATTERN
                                10072X *                (H,L) = TABLE FWA
                                10073X *        EXIT    (A) = PATTERN IF FOUND
                                10074X *                'Z' SET IF FOUND
                                10075X *                'Z' CLEAR IF NOT FOUND OR PATTERN=0             /78.10.GC/
                                10076X *        USES    A,F,H,L
                                10077X
                                10078X
112.033  305                    10079X $TBLS    PUSH    B
112.034  376 000                10080X          CPI     0                                               /78.10.GC/
112.036  312 060 112            10081X          JZ      TBL2                                            /78.10.GC/
112.041  107                    10082X          MOV     B,A
112.042  176                    10083X TBL1     MOV     A,M             (A) = CHARACTER
112.043  043                    10084X          INX     H
112.044  270                    10085X          CMP     B
112.045  312 062 112            10086X          JZ      TBL3            IF MATCH
112.050  247                    10087X          ANA     A
112.051  043                    10088X          INX     H               SKIP PAST
112.052  302 042 112            10089X          JNZ     TBL1            IF NOT END OF TABLE
112.055  053                    10090X          DCX     H
112.056  053                    10091X          DCX     H
112.057  257                    10092X          XRA     A               SET TO ZERO FOR OLD USERS       /78.10.GC/
112.060  376 001                10093X TBL2     CPI     1               CLEAR ZERO                      /78.10.GC/
                                10094X
                                10095X *        DONE
                                10096X
112.062  301                    10097X TBL3     POP     B
112.063  311                    10098X          RET
112.064                         10099           XTEXT   TBRA


                                10101X **       $TBRA - BRANCH RELATIVE THOUGH TABLE.
                                10102X *
                                10103X *        $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                                10104X *        JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                                10105X *        ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                                10106X *
                                10107X *        CALL    $TBRA
                                10108X *        DB      LAB1-*          INDEX = 0 FOR LAB1
                                10109X *        DB      LAB2-*          INDEX = 1 FOR LAB2
                                10110X *        DB      LABN-*          INDEX = N-1 FOR LABN
                                10111X *
                                10112X *        ENTRY   (A) = INDEX
                                10113X *                (RET) = TABLE FWA
```

```
                        10114X *        EXIT    TO COMPUTED ADDRESS
                        10115X *        USES    F,H,L
                        10116X
                        10117X
    031.076             10118X $TBRA    EQU     31076A          IN H17 ROM
    112.064             10119           XTEXT   TJMP


                        10121X **       $TJMP - TABLE JUMP.
                        10122X *
                        10123X *        USAGE
                        10124X *
                        10125X *        CALL    $TJMP           (A) = INDEX
                        10126X *        DW      ADDR1
                        10127X *        .         .
                        10128X *        .         .
                        10129X *        .         .
                        10130X *        DW      ADDRN
                        10131X *
                        10132X *        ENTRY   (A) = INDEX
                        10133X *        EXIT    TO PROCESSOR
                        10134X *                (A) = INDEX*2
                        10135X *        USES    NONE.
                        10136X
                        10137X
    031.061             10138X $TJMP    EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        10139X
    031.062             10140X $TJMP.   EQU     31062A          IN H17 ROM
    112.064             10141           XTEXT   TYPCH


                        10143X **       $TYPCH - TYPE SINGLE CHARACTER.
                        10144X *
                        10145X *        ENTRY   (RET) = CHARACTER
                        10146X *        EXIT    TO (RET)+1
                        10147X *                (A) = CHARACTER TYPED
                        10148X
                        10149X
    112.064  343        10150X $TYPCH   XTHL                    (HL) = RETURN ADDRESS
    112.065  176        10151X          MOV     A,M             (A) = CHARACTER
    112.066  043        10152X          INX     H
    112.067  343        10153X          XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        10154X
                        10155X **       $TYPC. - TYPE SINGLE CHARACTER.
                        10156X *
                        10157X *        ENTRY   (A) = CHARACTER
                        10158X *        EXIT    TO (RET)
                        10159X
    112.070  377 002    10160X $TYPC.   DB      SYSCALL,.SCOUT
    112.072  311        10161X          RET
    112.073             10162           XTEXT   TYPTX
```

```
                        10164X **       $TYPTX - TYPE TEXT.
                        10165X *
                        10166X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        10167X *
                        10168X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        10169X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        10170X *
                        10171X *       ENTRY   (RET) = TEXT
                        10172X *       EXIT    TO (RET+LENGTH)
                        10173X *       USES    A,F
                        10174X
                        10175X
031.136                 10176X $TYPTX  EQU     31136A          IN H17 ROM
                        10177X
031.144                 10178X $TYPTX. EQU     31144A          IN H17 ROM
112.073                 10179          XTEXT   GNL


                        10181X **       $GNL - GUARANTEE NEW LINE.
                        10182X *
                        10183X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                        10184X *       IF THE CURSOR IS NOT AT COLUMN 1..
                        10185X *
                        10186X *       ENTRY   NONE
                        10187X *       EXIT    NONE
                        10188X *       USES    ALL
                        10189X
                        10190X
112.073  076 002       10191X $GNL     MVI     A,I.CUSOR
112.075  001 000 000   10192X          LXI     B,0
112.100  377 006       10193X          DB      SYSCALL+.CONSL          READ CURSOR
112.102  075           10194X          DCR     A
112.103  310           10195X          RZ                      AT COLUMN 1
112.104  303 354 111   10196X          JMP     $CRLF           NEW LINE
112.107                10197          XTEXT   CHL


                        10199X **       $CHL - COMPLEMENT (HL).
                        10200X *
                        10201X *       (HL) = -(HL)            TWO'S COMPLEMENT
                        10202X *
                        10203X *       ENTRY   NONE
                        10204X *       EXIT    NONE
                        10205X *       USES    A,F,H,L
                        10206X
                        10207X
030.224                 10208X $CHL    EQU     30224A          IN H17 ROM
112.107                 10209          XTEXT   COMP
```

```
                        10211X **       $COMP - COMPARE TWO CHARACTER STRINGS.
                        10212X *
                        10213X *        $COMP COMPARES TWO BYTE STRINGS.
                        10214X *
                        10215X *        ENTRY   (C) = COMPARE COUNT
                        10216X *                (DE) = FWA OF STRING #1
                        10217X *                (HL) = FWA OF STRING #2
                        10218X *        EXIT    'Z' CLEAR, IS MIS-MATCH
                        10219X *                 (C) = LENGTH REMAINING
                        10220X *                 (DE) = ADDRESS OF MISMATCH IN STRING#1
                        10221X *                 (HL) = ADDRESS OF MISMATCH IN STRING #2
                        10222X *                'C' SET, HAVE MATCH
                        10223X *                 (C) = 0
                        10224X *                 (DE) = (DE) + (OC)
                        10225X *                 (HL) = (HL) + (OC)
                        10226X *        USES    A,F,C,D,E,H,L
                        10227X
                        10228X
030.060                 10229X $COMP    EQU     30060A          IN H17 ROM
112.107                 10230           XTEXT   MCU


                        10232X **       MCU - MAP LOWER CASE TO UPPER CASE.
                        10233X *
                        10234X *        MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                        10235X *        CASE.
                        10236X *
                        10237X *        ENTRY   (A) = CHARACTER
                        10238X *        EXIT    (A) = CHARACTER RESULT
                        10239X *        USES    A,F
                        10240X
                        10241X
112.107  376 141        10242X $MCU     CPI     'a'
112.111  330            10243X          RC                      NOT LOWER CASE
112.112  376 173        10244X          CPI     'z'+1
112.114  320            10245X          RNC                     NOT LOWER CASE
112.115  326 040        10246X          SUI     'a'-'A'
112.117  311            10247X          RET
112.120                 10248           XTEXT   MU10


                        10250X **       $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                        10251X *
                        10252X *        (HL) = (DE)*10
                        10253X *
                        10254X *        ENTRY   (DE) = MULTIPLIER
                        10255X *        EXIT    'C' CLEAR IF OK
                        10256X *                 (HL) = PRODUCT
                        10257X *                'C' SET IF ERROR
                        10258X *        USES    D,E,H,L,F
                        10259X
                        10260X
```

```
030.324          10261X $MU10   EQU    30324A          IN H17 ROM
112.120          10262          XTEXT  SAVALL

                 10264X **      $RSTALL - RESTORE ALL REGISTERS.
                 10265X *
                 10266X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                 10267X *       RETURNS TO THE PREVIOUS CALLER.
                 10268X *
                 10269X *       ENTRY  (SP) = PSW
                 10270X *              (SP+2) = BC
                 10271X *              (SP+4) = DE
                 10272X *              (SP+6) = HL
                 10273X *              (SP+8) = RET
                 10274X *       EXIT   TO *RET*, REGISTERS RESTORED
                 10275X *       USES   ALL
                 10276X
                 10277X
031.047          10278X $RSTALL EQU    31047A          IN H17 ROM

                 10280X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                 10281X *
                 10282X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                 10283X *
                 10284X *       ENTRY  NONE
                 10285X *       EXIT   (SP) = PSW
                 10286X *              (SP+2) = BC
                 10287X *              (SP+4) = DE
                 10288X *              (SP+6) = HL
                 10289X *       USES   H,L
                 10290X
                 10291X
031.054          10292X $SAVALL EQU    31054A          IN H17 ROM
112.120          10293          XTEXT  INDL

                 10295X **      $INDL - INDEXED LOAD.
                 10296X *
                 10297X *       $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                 10298X *
                 10299X *       THIS ACTS AS AN INDEXED FULL WORD LOAD.
                 10300X *
                 10301X *       (DE) = ( (HL) + DSPLACEMENT )
                 10302X *
                 10303X *       ENTRY  ((RET)) = DISPLACMENT (FULL WORD)
                 10304X *              (HL) = TABLE ADDRESS
                 10305X *       EXIT   TO (RET+2)
                 10306X *       USES   A,F,D,E
                 10307X
                 10308X
030.234          10309X $INDL   EQU    30234A          IN H17 ROM
```

```
112.120          10310          XTEXT   MTDOC

                 10312X ***     MANAGED TABLES.
                 10313X *
                 10314X *       THE FOLLOWING STRUCTURES ARE MANAGED TABLES.
                 10315X *
                 10316X *       SEVERAL TABLES OF DATA ARE 'MANAGED' BY A SUBROUTINE
                 10317X *       PACKAGE SO THAT THEIR SIZES MAY VARY INDEFINITELY.
                 10318X *
                 10319X *       EACH TABLE HAS A CERTAIN AMOUNT OF FREE SPACE LOCATED AFTER
                 10320X *       IT. WHEN A TABLE NEEDS TO BE ENLARGED, $ATS (ALLOCATE
                 10321X *       TABLE SPACE) PERFORMS THE ALLOCATION. IF SUFFICIENT FREE S PCE
                 10322X *       FOLLOWS THE TABLE, IT IS SIMPLY ALLOCATED.
                 10323X *
                 10324X *       IF THE FREE SPACE FOLLOWING THE TABLE IS INSUFFICIENT, ALL
                 10325X *       TABLES ARE MOVED, REDUCING THE FREE SPACE BEHIND EACH ONE, IN
                 10326X *       ORDER TO CONCENTRATE SUFFICIENT FREE SPACE BEHIND THE ONE
                 10327X *       NEEDING IT. THUS, WHEN TABLE OVERFLOW OCCURS, ALL TABLES HAVE
                 10328X *       OVERFLOWED, IN THAT THERE IS NO MORE FREE SPACE AVAILABLE
                 10329X *       BEHIND ANY OF THEM.
                 10330X *
                 10331X *       STORAGE USED:
                 10332X *
                 10333X *       THE MANAGED TABLE PACKAGE USES MEMORY STARTING AT SYMBOL 'MTAREA
                 10334X *       EXTENDING TO THE VALUE IN (MEML). (MEML) MAY BE INCREASED DURING
                 10335X *       EXECUTION, BUT IT SHOULD NOT BE DECREASED.
                 10336X *
                 10337X *       FREE SPACE ALLOCATION:
                 10338X *
                 10339X *       WHEN THE TABLES MUST BE MOVED, $ATS DIVIDES UP THE MEMORY FREE
                 10340X *       SPACE AMONG THE TABLES, HOWEVER, THIS SPLITTING IS NOT NECESSARI
                 10341X *       EVEN. EACH TABLE CONTAINS A ONE BYTE ALLOCATION FACTOR,
                 10342X *       INDICATING HOW MANY 1/16THS SHARES IT WILL RECEIVE. THESE
                 10343X *       NUMBERS MUST ALL ADD UP TO 16 (THUS, THE NEX NUMBER OF TABLES
                 10344X *       IS 16, SINCE NO ALLOCATION FACTOR MAY BE 0).
                 10345X *
                 10346X *       TABLE USAGE:
                 10347X *
                 10348X *       NO TABLE ITEM (EXCEPT ITEMS IN THE 1ST TABLE) MAY BE REFERENCED
                 10349X *       BY ADDRESS, SINCE THE ADDRESS MAY BE CHANGED (VIA TABLE MOVES)
                 10350X *       AT ANY TIME. INSTEAD, THE ITEMS SHOULD BE REFERENCED BY
                 10351X *       TABLE INDEX, THAT IS, THEIR SEQUENTIAL POSITION WITHIN THE
                 10352X *       TABLE.
```

```
                        10354X **       TABLE INDEX.
                        10355X *
                        10356X *       THE FOLLOWING INDEXES ARE USED TO KEEP TRACK OF TABLES. ALL
                        10357X *       TABLE INDIXES MUST APPEAR CONTIGUOUSLY IN MEMORY.
                        10358X *
                        10359X *
                        10360X * MTABIND EQU *
                        10361X *       CINDEX FOR TABLE 1]
                        10362X *          .
                        10363X *          .
                        10364X *       CINDEX FOR TABLE N]
                        10365X *       DB      0               DUMY ALLOCATION BYTE
                        10366X * MEML  DW      0               MEMORY LIMIT
                        10367X *MTABL  EQU     *-MTABIND/5     NUMBER OF TABLES.
                        10368X *
                        10369X *       INDEX FORMAT:
                        10370X *
                        10371X *       DB      FACT            ALLOCATION FACTOR (NUMBER OF 1/16THS)
                        10372X *       DW      FWA             TABLE FWA
                        10373X *       DW      LEN             TABLE LENGTH
                        10374X
    000.000             10375X MT.AFC  EQU     0               ALLOCATION FACTOR
    000.001             10376X MT.FWA  EQU     1               FWA INDEX
    000.003             10377X MT.LEN  EQU     3               LENGTH FIELD
    112.120             10378  MTABIND EQU     *               FWA OF 1ST TABLE HEADER
```

```
                        10382  **      TXTTAB - USER SOURCE TEXT TABLE.
                        10383  *
                        10384  *       FORMAT:
                        10385  *
                        10386  *       DW      LINE            LINE NUMBER
                        10387  *       DB      'TEXT'          LINE TEXT
                        10388  *       DB      0               END OF LINE
                        10389  *
                        10390  *       LINE NUMBER 65535 (377377A) IS ALWAYS PRESENT IN THE TABLE,
                        10391  *       BUT MAY NOT BE ALTERED OR DISPLAYED.
                        10392
112.120  001           10393  TXTTAB  DB      1               ALLOCATION COUNT
112.121  010 115       10394          DW      MTAREA          FWA
112.123  003 000       10395          DW      3               LENGTH



                        10397  **      SYMTAB - SYMBOL TABLE.
                        10398  *
                        10399  *       SYMTAB CONTAINS THE USER SYMBOL TABLE.
                        10400  *       AN ENTRY IS PRESENT FOR EACH
                        10401  *
                        10402  *       1) SCALAR NUMERIC VARIABLE
                        10403  *       2) SCALAR NUMERIC FUNCTION
                        10404  *       3) SCALAR STRING VARIABLE
                        10405  *       4) SCALAR STRING FUNCTION
                        10406  *       5) NUMERIC VECTOR
                        10407  *       6) STRING VECTOR
                        10408  *
                        10409  *       ENTRY   FORMAT:
                        10410  *
                        10411  *       THE ENTRY FORMAT DEPENDS UPON THE SYMBOL TYPE.
                        10412  *       ALL SCALAR ENTRYS ARE 6 BYTES LONG WITH VECTORS BEING LONGER.(SEE BELOW)
                        10413  *       THE FIRST TWO BYTES OF ALL ENTRYS ARE ALWAYS FORMATTED:
                        10414  *
                        10415  *       DB      'C'             1ST CHARACTER OF VARIABLE NAME
                        10416  *       DB      N+F             N = 2ND CHARACTER INDEX
                        10417  *                                (0=NONE, 0001B='0',...,1010B='9')
                        10418  *                               F=00000000 SCALAR NUMERIC VARIABLE
                        10419  *                                =00000001 SCALAR STRING VARIABLE
                        10420  *                               =00000010 NUMERIC VECTOR
                        10421  *                                =00000011 STRING VECTOR
                        10422  *
                        10423  *       THE REMAINING  BYTES ARE FORMATTED:
                        10424  *
                        10425  *       1) SCALAR NUMERIC VARIABLE:
                        10426  *
                        10427  *       DW      V1              4 BYTE FLOATING POINT VALUE
                        10428  *       DW      V2
                        10429  *
                        10430  *       2) SCALAR NUMERIC FUNCTION
                        10431  *
                        10432  *       DB      201*    DB      2010            FUNCTION FLAG
                        10433  *       DW      ADDR            TEXT ADDRESS OF FUNCTION LINE
                        10434  *       DB      0               UNUSED
```

```
                                10435  *
                                10436  *       3) SCALAR STRING VARIABLE
                                10437  *
                                10438  *       DB      LEN,0           LENGTH
                                10439  *       DW      STRNAM          STRING NAME ((LABEL) SEE STRTAB)
                                10440  *
                                10441  *       4) SCALAR STRING FUNCTION
                                10442  *
                                10443  *       DB      201Q            FUNCTION FLAG
                                10444  *       DW      ADDR            TEXT ADDRESS OF FUNCTION LINE
                                10445  *       DB      0               UNUSED
                                10446  *
                                10447  *       5) NUMERIC VECTOR
                                10448  *
                                10449  *       DB      DIM,0           NUMBER OF DIMENSIONS
                                10450  *       DW      SIZE            SIZE OF ARRAY(# OF BYTES FROM L1 TO NEXT ENTRY)
                                10451  * L1    DW      DIM 1           DIMENSION 1
                                10452  *       .       .
                                10453  *       .       .
                                10454  *       .       .
                                10455  *       DW      DIM N           DIMENSION N
                                10456  *       DW      V1              4 BYTE FLOATING POINT VALUE
                                10457  *       DW      V2
                                10458  *       .       .
                                10459  *       .       .
                                10460  *       .       .
                                10461  *       DW      V M-1
                                10462  *       DW      V M
                                10463  *
                                10464  *       6) STRING VECTOR
                                10465  *
                                10466  *       DB      DIM,0           NUMBER OF DIMENSIONS
                                10467  *       DW      SIZE            SIZE OF ARRAY
                                10468  *       DW      DIM 1           DIMENSION 1
                                10469  *       .       .
                                10470  *       .       .
                                10471  *       .       .
                                10472  *       DW      DIM N           DIMENSION N
                                10473  *       DW      LABEL 1         STRING LABEL
                                10474  *       DW      LEN 1           LENGTH OF STRING
                                10475  *       .       .
                                10476  *       .       .
                                10477  *       .       .
                                10478  *       DW      LABEL M
                                10479  *       DW      LEN M
                                10480
                                10481
112.125  001                    10482  SYMTAB  DB      1               ALLOCATION FACTOR
112.126  013 115                10483          DW      MTAREA+3
112.130  000 000                10484          DW      0
```

```
                        10486  **      FORTAB - FOR/NEXT LOOP TABLE.
                        10487  *
                        10488  *       FORTAB IS USED TO KEEP TRACK OF THE INDEX VARIABLE FOR
                        10489  *       'FOR/NEXT' LOOPS.
                        10490  *
                        10491  *       ENTRY   FORMAT:
                        10492  *
                        10493  *       DB      'C',N+F         SYMBOL TABLE KEY (SEE SYMTAB)   /80.01.GC/
                        10494  *       DW      INC,INC         INCREMENT VALUE
                        10495  *       DW      TRM,TRM         TERMINATEION VALUE
                        10496  *       DW      LOOPADR         ADDRESS FOR 'FOR' LOOP
                        10497
                        10498
112.132  001            10499  FORTAB  DB      1               ALLOCATION FACTOR
112.133  013 115        10500          DW      MTAREA+3
112.135  000 000        10501          DW      0               LENGTH


                        10503  **      GOSTAB - SOSUB TABLE.
                        10504  *
                        10505  *       GOSTAB CONTAINS THE RETURN ADDRESSES (AND LINE NUMBERS)
                        10506  *       FOR GOSUB CONSTRUCTS.
                        10507  *
                        10508  *       ENTRY   FORMAT:
                        10509  *
                        10510  *       DW      ADDR            RETURN TEXT ADDRESS
                        10511  *       DW      STATNO          RETURN LINE NUMBER
                        10512
                        10513
112.137  001            10514  GOSTAB  DB      1               ALLOCATION FACTOR
112.140  013 115        10515          DW      MTAREA+3
112.142  000 000        10516          DW      0


                        10518  **      WRKTAB - WORKING STORAGE TABLE.
                        10519  *
                        10520  *       WRKTAB IS USED BY THE EXPRESSION EVAULATOR TO STORE
                        10521  *       (ON A STACK) WORKING VALUES.
                        10522  *
                        10523  *       EACH ENTRY CONSISTS OF 5 BYTES, USUALLY A DESCRIPTOR BYTE
                        10524  *       AND 4 VALUE BYTES.
                        10525
                        10526
112.144  001            10527  WRKTAB  DB      1               ALLOCATION INDEX
112.145  013 115        10528          DW      MTAREA+3
112.147  000 000        10529          DW      0
```

```
                        10531  **      STRTAB - PERMANENT STRING TABLE.
                        10532  *
                        10533  *       STRTAB HOLDS PERMANENT STRING VARIABLES USED IN BASIC.
                        10534  *
                        10535  *       EACH STRING IS INDEXED BY AN ENTRY IN SYMTAB OR VECTAB.
                        10536  *
                        10537  *       ENTRY   FORMAT:
                        10538  *
                        10539  *       EACH STRING APPEARS CONTIGUOUSLY IN MEMORY, NO TRAILING
                        10540  *       CHARACTER IS USED SINCE LENGTHS ARE KNOWN IN THE POINTER.
                        10541  *       EXAMPLE:
                        10542  *
                        10543  *       DS      2               STRING LABEL(2NN NNN FOR PERM. STRING, 3NN NNN
                        10544  *                               FOR A TEMPORARY STRING)
                        10545  *       DS      N               ASCII STRING ( N=1 TO 256 )
                        10546  *       .       .
                        10547  *       .       .
                        10548  *       .       .
                        10549  *       DS      2               NTH LABEL
                        10550  *       DS      N
                        10551
                        10552
112.151  002           10553  STRTAB  DB      2               ALLOCATION INDEX
112.152  013 115       10554          DW      MTAREA+3
112.154  000 000       10555  STRLEN  DW      0


                        10557  **      TSTTAB - TEMPORARY STRING TABLE
                        10558  *
                        10559  *       TSTTAB HOLDS ALL TEMPORARY STRING VARIABLES USED IN BASIC.
                        10560  *
                        10561  *       THE FORMAT USED IS SIMILAR TO THAT OF STRTAB.
                        10562
                        10563
112.156  001           10564  TSTTAB  DB      1
112.157  013 115       10565          DW      MTAREA+3
112.161  000 000       10566          DW      0


                        10568  **      FILE TABLE.
                        10569  *
                        10570  *       CONTAINS BUFFER FOR EACH OPEN FILE.
                        10571
112.163  000           10572  FILTAB  DB      0               ALLOCATION INDEX
112.164  013 115       10573          DW      MTAREA+3        FWA
112.166  000 000       10574          DW      0               LWA
```

```
                        10576  **      DUMY LAST TABLE.
                        10577  *
                        10578  *       FORMATTED LIKE REGULAR TABLE, BUT CONTAINS
                        10579  *       MOVE COUNT, AND MEMORY LIMIT VALUES.
                        10580
000.010                 10581  MTABL   EQU     *-MTABIND/5     NUMBER OF TABLES
112.170  000            10582          DB      0               STORAGE MOVES (IN ALLOCATING INDEX CELL)
112.171  013 115        10583  MEML    DW      MTAREA+3        MEMORY LIMIT ADDRESS (IN FWA CELL)
112.173                 10584          DS      2               TABLE LENGTH CELL NOT USED
                        10585
000.005                 10586  MTABLEN EQU     5               LENGTH OF EACH TABLE HEADER


                        10588  **      POINTERS TO CURRENT INFORMATION ABOUT RUN.
                        10589
112.175  000 000        10590  CURNUM  DW      0               CURRENT LINE NUMBER
112.177  000 000        10591  CURADR  DW      0               CURRENT LINE ADDRESS
112.201  000            10592  LCKFLG  DB      0               DATA LOCK FLAG
                        10593
                        10594  **      CURRENT I.O CHANNEL.
                        10595  *
                        10596  *       =0      SYSTEM CONSOLE
                        10597  *       =1      INTERNAL FILE
                        10598  *       =1+N    BASIC CHANNEL N (N=1 TO X, IF N=0 THEN IOCHAN=0)
                        10599
112.202  000            10600  IOCHAN  DB      0
                        10601
112.203  000            10602  OVLMAN  DB      0               <>0 IF TO LOCK OVERLAY
                        10603
                        10604
112.204  000            10605  CTLFLAG DB      0               CTL CHARACTERS FLAG BYTE
000.001                 10606  CFCTLC  EQU     001Q            CTL-C HIT
000.002                 10607  CFCTLB  EQU     002Q            CTL-B HIT


                        10609  **      STRING INDEXES.
                        10610  *
                        10611
112.205  200 000        10612  STRVI   DW      000200A
112.207  300 000        10613  STRTI   DW      000300A


                        10615  **      FLOATING POINT VALUES.
                        10616  *
                        10617
112.211  000 000 100    10618  FP1.0   DB      0,0,100Q,201Q
112.215  000 000 120    10619  FP10.   DB      0,0,120Q,204Q
112.221  146 146 146    10620  FP0.1   DB      146Q,146Q,146Q,175Q
031.320                 10621  FP0.0   EQU     $ZEROS
112.225  022 170 233    10622  NPI.2   DB      022Q,170Q,233Q,201Q     -PI/2
112.231  022 170 233    10623  NPI2    DB      022Q,170Q,233Q,203Q     -PI*2
112.235  022 170 233    10624  NPI     DB      022Q,170Q,233Q,202Q     -PI
112.241  022 170 233    10625  NPI.4   DB      022Q,170Q,233Q,200Q     -PI/4
```

```
112.245  356 207 144 10626  PI.4    DB      3560,2070,1440,2000       PI/4
                     10627
112.251  040          10628  SPACE   DB      ' '                   SPACE CHARACTER
```

```
                                10631  **      PRS - PRESET BASIC.
                                10632  *
                                10633  *       PRS PERFORMS PRESET INITIALIZATION.
                                10634  *
                                10635
112,252                         10636  PRS     EQU     *
                                10637
                                10638  *       CHECK THE HDOS VERSION
                                10639
112,252  377 011                10640          DB      SYSCALL,.VERS
112.254  332 056 113            10641          JC      PRSERR1         NO SYSTEM CALL
112,257  376 040                10642          CPI     VERS
112.261  302 056 113            10643          JNZ     PRSERR1         NOT THE CORRECT VERSION
                                10644
                                10645  *       REQUEST MINIMAL MEMORY
                                10646
112.264  041 010 115            10647          LXI     H,MTAREA
112,267  377 052                10648          DB      SYSCALL,.SETTP
112.271  332 060 113            10649          JC      PRSERR          NOT EVEN ENOUGH MEMORY TO START
                                10650
                                10651  *       SET UP THE INTERNAL WORK FILE BLOCK
                                10652
112.274  052 121 041            10653          LHLD    S.SCR           HL = ADDRESS OF  *HDOS*  SCRATCH BUFFER
112,277  042 232 042            10654          SHLD    FBSCR+2+0
112.302  042 234 042            10655          SHLD    FBSCR+2+2
112,305  042 236 042            10656          SHLD    FBSCR+2+4
112.310  021 000 002            10657          LXI     D,512
112,313  031                    10658          DAD     D
112.314  042 240 042            10659          SHLD    FBSCR+2+6
                                10660
                                10661  *       PROCEED WITH INITIALIZATION
                                10662
112.317  315 357 073            10663          CALL    DTS             DELETE TEMP. STRINGS
112,322  072 033 040            10664          LDA     .TICCNT         INITIALIZE RANDOM NUMBER SEED
112.325  147                    10665          MOV     H,A
112,326  157                    10666          MOV     L,A
112.327  042 101 061            10667          SHLD    RNDA            INITIALIZE SEED
112,332  021 016 000            10668          LXI     D,14                                    /80.01.GC/
112.335  315 003 046            10669          CALL    CNTL4           SET TAB-FIELD WIDTH TO 14       /80.01.GC/
112,340  041 370 100            10670          LXI     H,CBINT
112.343  076 002                10671          MVI     A,CTLB
112,345  377 041                10672          DB      SYSCALL,.CTLC   SETUP CTL-B HANDLER
112.347  041 363 100            10673          LXI     H,CCINT
112,352  076 003                10674          MVI     A,CTLC
112.354  377 041                10675          DB      SYSCALL,.CTLC   SETUP CTL-C HANDLER
112,356  315 136 031            10676          CALL    $TYPTX
112.361  012 012 105            10677  PRSA    DB      NL,NL,'Extended Benton Harbor BASIC #110.06.00',ENL
113,033  315 115 074            10678          CALL    FOC             SET TABLES TO MAXIMUM AREA
113.036  315 360 044            10679          CALL    SCR.            SCRATCH TEXT
                                10680
                                10681  ********************************************************************
                                10682  ********************************************************************
                                10683  **                                                              **
                                10684  **      Note:   Be very careful about the followins initializations.    **
                                10685  **              Be sure that the instructions do not destroy thems-     **
                                10686  **              selves.                                         **
```

```
                        10687  **                                                                  **
                        10688  **      Note:   If you don't understand the followins error messages**
                        10689  **              neither do I, just love it and leave it.            **
                        10690  **                                                                  **
                        10691  **********************************************************************
                        10692  **********************************************************************
                        10693
      113.041  257      10694          XRA     A
                        10695
      113.042  062 007 115 10696          STA     ZERO            CLEAR LINE-1
      115.007           10697  .       SET     ZERO
      000.000           10698          IF      *-1/.           *-1 < .
      001.331           10699          ERRMI   .-PRSB           .  < PRSB
                        10700          ENDIF
                        10701
      113.045  062 334 113 10702          STA     LINE+LINEL+6    INSURE 0 AT END OF LINE
      113.334           10703  .       SET     LINE+LINEL+6
      000.000           10704          IF      *-1/.           *-1 < .
      000.256           10705          ERRMI   .-PRSB           .  < PRSB
                        10706          ENDIF
                        10707
      113.050  062 342 114 10708          STA     LINE2+LINEL+6   INSURE 0 AT END OF LINE
      114.342           10709  .       SET     LINE2+LINEL+6
      000.000           10710          IF      *-1/.           *-1 < .
      001.264           10711          ERRMI   .-PRSB           .  < PRSB
                        10712          ENDIF
                        10713
      113.053  303 124 043 10714          JMP     RESTART         START PROGRAM
      113.056           10715  PRSB    EQU     *
                        10716
      113.056  076 050  10717  PRSERR1 MVI     A,EC:NCV        NOT THE CORRECT VERSION OF  *HDOS*
                        10718
      113.060  046 012  10719  PRSERR  MVI     H,NL
      113.062  377 057  10720          DB      SYSCALL,.ERROR  OUTPUT THE ERROR
      113.064  257      10721          XRA     A
      113.065  377 000  10722          DB      SYSCALL,.EXIT   QUIT BEFORE PROBLEMS ARISE
                        10723
      113.067           10724  PRSLIM  EQU     *               LWA OF PRS CODE
                        10725
      113.067           10726  LOADL   EQU     *               LOAD LWA
                        10727
                        10728
                        10729  **      OVERLAID BUFFER AREA
                        10730
      112.315           10731          ORG     PRSLIM-106
                        10732
                        10733  **      COLUMN COUNTERS.
                        10734  *
                        10735  *       SINCE SEVERAL CHANNELS MAY BE PRINTED ON, INTERMINGLED,
                        10736  *       A SEPERATE COLUMN COUNTER IS KEPT FOR EACH.
                        10737  *       THIS TABLE IS INDEXED BY THE CONTENTS OF IOCHAN
                        10738
      112.315           10739  COLCNTS DS      CHANMAX+1+2     ONE FOR EACH CHANNEL, +2 FOR TTY AND INTERNAL
                        10740
                        10741
      112.325           10742          DS      2               USED BY ITL (WHEN CALLED BY BUILD)
```

```
    112.327  10743  LINE    DS      0               LINE BUFFER                    /80.01.GC/
    112.327  10744          DS      255                                            /80.01.GC/
    000.377  10745  LINEL   EQU     *-LINE          LINE LENGTH
    000.237  10746          ERRMI   *-PRSLIM        FOLLOWING CELLS CHANGED BY PRS CODE
    113.326  10747          DS      6               ROOM FOR EXPANDED LINE NUMBER  /78.10.GC/
    113.334  10748          DS      1               ALWAYS 0 TO GUARANTEE END OF LINE
             10749
    113.335  10750  LINE2   DS      LINEL           WORK AREA
    114.334  10751          DS      6               ROOM FOR EXPANDED LINE NUMBER  /78.10.GC/
    114.342  10752          DS      1               ALWAYS ZERO TO GUARANTEE END OF LINE
             10753
    112.327  10754  FNRMA   EQU     LINE            FNRM WORK AREA
             10755
    114.343  10756  RUNMOD  DS      1
    114.344  10757  STATE   DS      1
    114.345  10758  DATPTR  DS      2


             10760  **      PATCH AREA.
             10761
    114.347  10762  PATCH   DS      32


             10764  **      BEGINNING OF MANAGED TABLE ADDRESS.
             10765  *
             10766
    115.007  10767  ZERO    DS      1               DUMY END OF FIRST LINE -1
    115.010  10768  MTAREA  EQU     *               BEGINNING OF MANAGED TABLES AREA
             10769
    115.010  10770          DS      100             AUX. PATCH AREA
             10771
    115.154  10772          END
ASSEMBLY COMPLETE
10772 STATEMENTS
    0 ERRORS DETECTED
18388 BYTES FREE
```

$ATP	071071	5369L	6271	8420									
$ATS	103244	5318	5546	8255E	8531								
$CCO	112016	998	2027	9996L									
$CDEHL	030216	6238	7799	9914E									
$CFS	072127	1880	5635L										
$CHL	030224	1596	2526	2532	6225	10208E							
$CLL	111335	5229	5246	7403	9824L								
$COMP	030060	3775	10229E										
$CRLF	111354	1887	9848L	10196									
$CVD	111362	9411	9738	9741	9769	9864L							
$CVD.	111363	6174	6181	9865L									
$DADA	030072	3709	6185	10012E									
$DADA.	030101	3754	5393	6571	10022E								
$DBT	104203	1392	1799	5939	8494E								
$DU66	030106	2634	9949E										
$FCLO	102335	1470	1926	2784	7133	8009L							
$FCLO.	102344	8009	8013L										
$FCLO1	103015	8052L	8055										
$FCLO2	103025	8048	8057L										
$FCLO3	103032	8021	8038	8068L									
$FCLO4	103040	8019	8074L										
$FERROR	070223	2769	2839	2856	5180E	7548	7552	7556	7656	7786	7833	7951	8011
$FFB	103122	7704	7801	8146L									
$FOPE1	101101	7578	7590L										
$FOPE2	101134	7614	7618L										
$FOPEA	101142	7618	7622E										
$FOPEB	101152	7612	7632L										
$FOPER	101021	2463	7116	7546L									
$FOPER.	101046	2766	7546	7559L									
$FOPEU	101037	7554L											
$FOPEU.	101054	7554	7563L										
$FOPEW	101030	1913	2464	2774	7550L								
$FOPEW.	101051	7550	7561L										
$FREAL	101161	7074	7122	7653L									
$FREAL.	101174	7653	7661E										
$FREAO	101364	3546	7783L										
$FREAO.	101377	7783	7791E										
$FWBRK	102231	7948L											
$FWBRK.	102240	7948	7953L										
$FWBRK1	102254	7955	7960L										
$FWRIB	102047	1922	1925	7440	7831L								
$FWRIB.	102056	7831	7836E										
$GNL	112073	999	10191L										
$IBT	104213	1657	5263	5474	8519L								
$INDL	030234	8025	8028	10309E									
$MCU	112107	2937	4874	4887	4905	5415	5733	10242L					
$MOVE	030252	2090	2452	2869	3068	5270	5288	5618	6884	8413	8474	8561	10046E
$MU10	030324	6184	10261E										
$MU66	030337	1688	3124	5525	10058E								
$MU86	031007	5581	6090	9964E									
$RCHAR	103233	2516	8233L	8234									
$REAL2	101205	7670L	7747										
$REAL3	101226	7682L	7706										
$REAL4	101254	7695	7698L										
$REAL6	101272	7700	7715L										
$REAL7	101304	7725L	7743										
$REAL8	101352	7675	7678	7705	7761L								
$REAL9	101360	7767L											

```
$READ1  102007  7796L  7803
$READ2  102035  7800  7805L
$READ8  102043  7802  7814L
$REL7.3 101320  7730  7738L
$REL7.5 101343  7740  7751L
$RSTALL 031047  6243  10000  10278E
$SAVALL 031054  1577  6212  6265  9996  10292E
$SOB    111372  4020  4949  6781  9895L
$SOB1   111373  9896L  9899  9901
$TBLS   112033  5745  7616  10079L
$TBRA   031076  1508  10118E
$TJMP   031061  1151  3389  10138E
$TJMP.  031062  10140E
$TYPC.  112070  10160L
$TYPCC  103217  7369  7431  8214E  8223
$TYPCH  112064  10150L
$TYPTX  031136  1000  1273  1864  2812  5197  5409  6456  7514  10176E  10676
$TYPTX. 031144  10178E
$UDD    031157  2182  9984E
$WCHAR  103241  7522  8237L
$WDR    031222  9808E
$WER    031241  9796E
$WRIB2  102062  7842L  7922
$WRIB3  102100  7852L  7888
$WRIB4  102126  7865  7868L
$WRIB6  102170  7870  7901L
$WRIB7  102202  7911L  7918
$WRIB8  102224  7845  7848  7892  7928L
$ZERO   031212  3793  9882E
$ZEROS  031320  9969E  10621
.       114342  1238S  1239  10697S  10698  10699  10703S  10704  10705  10709S  10710  10711
.ABUSS  040024  360E
.ALARM  002136  333E
.ALEDS  040013  358E
.CHFLG  000060  603L
.CLEAN  000205  618L
.CLEAR  000055  600L  6442
.CLEARA 000056  601L  5674
.CLOSE  000046  593L  8073
.CLRCO  000007  577L  1007  5196
.CONSL  000006  576L  9999  10193
.CRC    002347  341E
.CRCSUM 040027  361E
.CTC    002172  335E
.CTL2FL 040066  367E
.CTLC   000041  588L  974  977  10672  10675
.CTLFLG 040011  357E
.DAD    000206  619L
.DECODE 000053  598L
.DELET  000050  595L  2853
.DISMT  000061  604L
.DLEDS  040021  359E
.DLY    000053  330E
.DMNMS  000203  616L
.DMOUN  000201  614L
.DOD    003122  344E
.DODA   003356  346E  3962
.DSPMOD 040007  355E
```

```
.DSPROT 040006  354E
.DUMP   001374  332E
.ERROR  000057  602L    1277    6445    10720
.EXIT   000000  570L    1292    7615    7621    10722
.HORN   002140  334E
.IDENT  000000  329E
.IOWRK  040002  352E    2474    2476    3883    3884
.LINK   000040  587L    2838
.LOAD   001267  331E
.LOADD  000062  605L
.LOADO  000010  578L
.MFLAG  040010  356E    1092
.MONMS  000202  615L
.MOUNT  000200  613L
.NAME   000054  599L
.NMIRET 040064  366E
.OPEN   000063  606L
.OPENC  000045  592L
.OPENR  000042  589L    7632
.OPENU  000044  591L    7634
.OPENW  000043  590L    7633
.PCHL   002264  337E
.POSIT  000047  594L
.PRINT  000003  573L    1863    2054
.RCK    003260  345E    3858
.READ   000004  574L    8165
.REGI   040005  353E
.REGPTR 040035  364E
.RENAM  000051  596L
.RESET  000204  617L
.RNB    002331  340E
.RNP    002325  339E
.SCIN   000001  571L    3530    7028    8233
.SCOUT  000002  572L    8220    8237    9849    10160
.SETTP  000052  597L    6241    10648
.SRS    002265  338E
.START  040000  351E
.SYSRES 000012  580L
.TICCNT 040033  363E    2522    2525    10664
.TPERR  002205  336E
.TPERRX 040031  362E
.UIVEC  040037  365E
.VERS   000011  579L    10640
.WNB    003024  343E
.WNP    003017  342E
.WRITE  000005  575L    7886    7994    8061
ABS     057055  3390    3470L
ABS.COD 000010  883L    887
ABS.ENT 000006  881L
ABS.ID  000000  877L
ABS.LDA 000002  879L
ABS.LEN 000004  880L
ACCX    042202  897L    2089    2304    3204    3311    3341    3378    3433    3470    3483    3498    3499
                3507    3555    3601    3698    3700    3731    3791    3839    3872    3912    3919    3943    3976
                3995    3996    4005    4013    4059    4107    4133    4187    4204    4250    4261    4355    4383
                4409    4459    4492    4505    4533    4553    4578    4641    4645    4680    4760    4813    5441
                5457    5486    5490    5786    5980    5995    6014    6384    6417    6843    6883    6912    6964
                7454    8642    8658    8680    8681    8685    8702    8867    8902    8936    9044    9207    9208
```

9225 9309 9466 9469 9578 9594 9604 9612 9617 9646

ACCY	042210	899L	3642	3688	3704	3837	4137	4189	4198	4256	4370	4435	4500	
		4732	4774	4791	5787	5996	6929	6930	6975	7455	9478	9721	9776	
ACTLB	044216	1216E	1338	1521										
ADD	104356	4434	4436	4504	4551	4598	4612	4667	4736	4811	8640E	8766	9777	
ADD0	104367	8650L												
ADD1	105022	8657	8671L											
ADD2	105041	8683L	8695											
ADD2.5	105057	8673	8693L											
ADD2.7	105065	8696L	8698											
ADD3	105074	8671	8679	8692	8702L									
ADD4	105142	8727	8738L	8781										
ADD5	105160	8652	8748L											
AIO.CGN	041047	822L												
AIO.CHA	041116	837L												
AIO.CNT	041111	833L												
AIO.CSI	041050	823L												
AIO.DDA	041041	818E												
AIO.DES	041055	827L												
AIO.DEV	041057	828L												
AIO.DIR	041062	831L												
AIO.DTA	041053	826L												
AIO.EOF	041113	835L												
AIO.EOM	041112	834L												
AIO.FLG	041043	819L												
AIO.GRT	041044	820L												
AIO.LGN	041051	824L												
AIO.LSI	041052	825L												
AIO.SPC	041046	821L												
AIO.TFP	041114	836L												
AIO.UNI	041061	829L												
AIO.VEC	041040	817L												
AMB	071026	1695	1740	1807	5306E	5863	6514	6988	7327					
ANT	071056	1060	1149	1319	1700	1754	1837	1973	2033	2044	2361	2372	2402	
		2597	2618	2647	3093	3148	3156	3306	3355	3666	4278	4301	4320	5340L
		6498	6550	6963	7004	7150	7187							
ASC	057065	3417	3483L											
ATAN1	065052	4650L												
ATAN2	065137	4665	4669L											
ATF	107323	2921	4021	6841	9391E									
ATF0	107335	9397	9403L											
ATF0.1	107352	9408	9410E											
ATF0.2	107364	9412	9416E											
ATF1	107370	9423L	9428	9432										
ATF1.5	110015	9430	9436L											
ATF10	110255	9541L	9548											
ATF11	110271	9528	9553L											
ATF2	110051	9456L	9464											
ATF2.5	110065	9451	9465L											
ATF3	110103	9424	9476L											
ATF4	110117	9481L												
ATF5	110132	9472	9490L											
ATF6	110173	9505	9508	9511L										
ATF7	110220	9523	9526L											
ATF8	110221	9493	9498	9527L										
ATF9	110254	9533	9540L											
ATFA	110260	9531	9536	9544E										
ATFB	110300	9405	9414	9495	9559L									

```
ATN     065026   3391     4639E
ATP1    071116   5381L    5395
ATP2    071120   5382L    5391
ATS1    103321   8276     8284    8305L
ATS2    103376   8350L    8358
ATS3    104031   8375L    8419
ATS4    104057   8395L    8397
ATS5    104084   8394     8398L
ATSA    103356   8260     8327E
ATSB    104124   8368     8425L   8440
ATSC    104125   8426L    8439
AVV     071202   2095     2124    4319    5436L   6816
AVV1    071222   5452L
AYS     071146   1289     1301    5409L
BAS1    043203   1003     1010L
BAS2    043240   1016     1028L
BAS3    043233   1014     1025L
BEC.AC  000230    426L    5165
BEC.CB  000201    403L    5105
BEC.CC  000200    402L    5102
BEC.CIU 000233    429L    5177
BEC.DO  000203    405L    5111
BEC.DE  000202    404L    5108
BEC.EN  000224    422L    1772
BEC.FAE 000226    424L    5159
BEC.FND 000231    427L    5168
BEC.IC  000222    420L    5156
BEC.ILF 000227    425L    5162
BEC.IN  000204    406L    5114
BEC.IU  000205    407L    5117
BEC.LK  000206    408L    5120
BEC.LTL 000232    428L
BEC.ND  000221    419L    5153
BEC.NV  000207    409L    5123
BEC.OV  000210    410L    5126
BEC.RE  000211    411L    5129
BEC.SC  000220    418L    5150
BEC.SL  000212    412L    5132
BEC.SN  000213    413L    5135
BEC.SR  000217    417L    5147
BEC.ST  000225    423L    2823
BEC.SY  000214    414L    1275    5138
BEC.TC  000215    415L    5141
BEC.TO  000216    416L    5144
BEC.UD  000223    421L    5174
BELL    000007    376E    1274    5083    5198    5410
BKSP    000010    378E
BLD1    044255   1253L    1269    1280
BLD2    044320   1260     1273L
BOOT.P  000001    797E
BUILD   044247   1153     1250E
BYE     044337   1154     1287E
C.STX   000002    380E
C.SYN   000026    379E
CAS     071334   5507L
CAS1    071346   5516E    5529
CB.CLI  000100    275E     298
CB.MTL  000040    274E
```

```
CB.SPK  000200  276E
CB.SSI  000020  273E
CB2.CLI 000002  279E
CB2.ORG 000040  280E
CB2.SID 000100  281E
CB2.SSI 000001  278E
CBINT   100370  972     7511L   10670
CBINT1  100372  7502    7512L
CBT     103042  7666    7795    7838    7954    8086L
CBT1    103051  8091L   8100
CCINT   100363  975     7501L   10673
CDB.H84 000001  740E
CDB.H85 000000  739E
CEF     071374  2427    5544L
CF.FCN  000004  178E    188     190     194     195
CF.STR  000001  176E    189     190     192     194     195     2607    2960    3312    3381    3434    3732
                5444    5606    5790    6482    6832
CF.VEC  000002  177E    191     192     194     195     1365    2076    2967    3095    3171    4279    5927
                6510    6554    6644
CFA     072005  1468    1911    2424    2429    2455    3544    5560L   7068    7437
CFA1    072015  5573L
CFCTLB  000002  1224    7511    7518    10607E
CFCTLC  000001  2207    7025    7501    7519    10606E
CFN     072053  1431    2336    2399    2861    5597    5602L
CFN.    072041  1909    2724    2763    5597L
CFS1    072140  5639L   5650
CHAIN   045205  1165    1428E
CHAIN1  045231  1435    1439L
CHANMAX 000005  39E     5379    5560    6065    10739
CHR$    057103  3392    3495L
CIN     057140  3393    3518E
CIN0    057160  3531L   3547
CIN1    057170  3533    3535L
CIN2    057202  3526    3544L
CLEAR   044363  1166    1317E
CLEAR.  044375  1323L   1404
CLF     072171  1329    5671E
CLL1    111341  9828L   9832
CLN     072206  1256    5220    5690L
CLOSE   045260  1167    1463E   1478
CLOSE1  045304  1469    1472L
CLR1    045021  1020    1334L   7112
CLR2    045062  1322    1355L
CLR3    045132  1366    1369    1381L
CMA     072223  1436    1477    1637    2155    2483    3625    3728    3736    3832    5709L   6043
CN.170M 000014  316E
CN.174M 000003  315E
CN.ABO  000200  320E
CN.BAU  000100  319E
CN.MEM  000040  318E
CN.PRI  000020  317E
CNC     072230  1012    2895    5722L
CNC1    072265  5728    5735    5743L
CNCA    072302  5744    5754E
CND.H17 000000  322E
CND.H47 000001  324E
CND.NDI 000000  323E
CNTL1   045341  1509    1519L
```

```
CNTL2    045353   1510     1526L
CNTL3    045371   1511     1536L
CNTL4    046003   1512     1544L  10669
CNTL43   046017   1551L    1554
CNTL46   046033   1552     1555   1557L
CNTL5    046041   1513     1565L
CNTL52   046126   1590     1599L
CNTLA    045334   1509L    1514
CNTLMX   000005   1506     1514E
CNTRL    045320   1168     1503E
CO.FLG   000001    717E    9998
COLCNTS  112315    997     3893   6569   7037   10739L
CONT     045163   1155     1410L  1444   1452
CONT1    045165   1411L    2806
COS      064125   3394     4530E
COS1     064175   4549     4553L
COS2     064233   4565L    4633   4669
COSA     064234   4548     4566E  4583   4594   4643
COT      072347   4101     4182   5786L  7090
CR       000015    372E
CS.FLG   000200    718E
CSA      072366   1713     1809   2093   2122   2277   4318   5454   5485   5805E  6485   6815   7217
CSE      073016   5852L    6492
CSE.     073033   3065     3500   3701   3997   4196   5858L
CSE..    073056   5854     5860   5871L
CSE1     073045   5856     5863L
CSE2     073065   5874     5876L
CSI      073000   1673     1793   1823   2264   5452   5826E  6481
CSL.CHR  000001    694E
CSL.ECH  000200    691E
CSL.RAW  000004    692E
CSL.WRP  000002    693E
CT.ABS   000320    213L     4999
CT.ALP   000001     76L     2907   5737
CT.AND   000310    200L     3224   5000
CT.AS    000311    201L     2409   5002
CT.ASC   000350    241L     5001
CT.ATN   000321    214L     5003
CT.BLD   000200    111L     1135   5004
CT.BYE   000201    112L     5005
CT.CHA   000213    125L     5006
CT.CHR   000322    215L     5007
CT.CIN   000323    216L     5008
CT.CLO   000215    127L     5010
CT.CLR   000214    126L     5009
CT.CMA   000026    101L     1703   1755   2382   2598   2669   3149   3667   4303   5710   5764   7008
CT.CMD   000256    164E     1138
CT.CNT   000202    113L     5012
CT.COS   000324    217L     5013
CT.CTL   000216    128L     5011
CT.DAT   000251    158L     2711   4942   5014
CT.DEF   000252    159L     5015
CT.DEL   000203    114L     5016
CT.DI    000024     99L     3274   4228   5760
CT.DIM   000217    129L     5017
CT.END   000253    160L     1116   5018
CT.EQ    000011     87L     2119   3238   4119   4331   5766
CT.EX    000025    100L     3289   5761
```

```
CT.EXP  000325  218L  5019
CT.FCN  000320  211E  3371  3377  3633  4931  6148
CT.FIL  000312  202L  2411  5020
CT.FIN  000000   75L  1057  1130  1320  1434  1474  2145  2152  2380  2514  2588  2797
                2801  2896  4334  5725  5762  6776  6866  7006  7188  7209
CT.FN   000220  130L  1612  3366  5021
CT.FNM  000357  248L
CT.FOR  000221  131L  2401  5022
CT.FRE  000222  132L  5024
CT.FRZ  000223  133L  5023
CT.GE   000013   89L  3026
CT.GOS  000224  134L  2364  5025
CT.GOT  000225  135L  1974  2362  5026
CT.GT   000012   88L  2905  3028  5767
CT.IF   000226  136L  5027
CT.INP  000254  161L  2014  5028
CT.INT  000326  219L  5029
CT.IUA  000250  155E  1025
CT.LCK  000230  138L  5035
CT.LE   000015   91L  3009
CT.LEF  000351  242L  3632  5031
CT.LEN  000352  243L  5030
CT.LET  000227  137L  5032
CT.LIN  000250  157L  5033
CT.LIS  000204  115L  5034
CT.LNO  000327  220L  5036  6172
CT.LOG  000330  221L  5037
CT.LT   000014   90L  2903  3017  5765
CT.MAT  000353  244L  5038
CT.MAX  000331  222L  5039
CT.MI   000022   97L  3257  3302  3315  5756
CT.MID  000354  245L  3632  3633  3636  5040
CT.MIN  000332  223L  5041
CT.MU   000023   98L  3272  5759
CT.NE   000016   92L  3015  3240
CT.NOT  000314  204L  3304  5043
CT.NUM  000002   77L  1015  2909  5730  5763
CT.NXT  000231  139L  5042
CT.OLD  000232  140L  5044
CT.ON   000233  141L  5045
CT.OPE  000234  142L  5046
CT.OR   000315  205L  3211  5047
CT.OUT  000235  143L  5048
CT.PAD  000333  224L  5049
CT.PAL  000017   94L  3356  5757  5770
CT.PAR  000020   95L  1705  2651  3157  3363  3664  3830  4305  4324  5758  5771  6198
                6779
CT.PAU  000236  144L  5050
CT.PEK  000334  225L  5051
CT.PIN  000335  226L  5052
CT.PL   000021   96L  3255  4179  5755
CT.POK  000237  145L  5053
CT.POS  000336  227L  5054
CT.PRT  000240  146L  5055
CT.PS   000031  104L  5772  6040  6052
CT.QUO  000030  103L  2901  5769
CT.REA  000241  147L  2403  2462  5056
CT.REM  000242  148L  4940  5057
```

```
CT.REP  000205    116L    5058
CT.RES  000243    149L    5059
CT.RET  000244    150L    5060
CT.RIG  000355    246L    3636  5061
CT.RND  000337    228L    5062
CT.RUA  000212    122E    1141
CT.RUN  000206    117L    5063
CT.SAV  000207    118L    5064
CT.SCR  000210    119L    5065
CT.SEG  000340    229L    5066
CT.SEM  000027    102L    2031  2046  2596  2671  5768
CT.SEP  000003     78L    2911  5749
CT.SGN  000341    230L    5067
CT.SIN  000342    231L    5068
CT.SNF  000304    188L
CT.SNV  000300    187L     193  1825  3871  4060  6416  6842  7215
CT.SPC  000343    232L    2594  2653  5069
CT.SQR  000344    233L    5070
CT.SRA  000350    240E    3377
CT.SSF  000305    190L    1357
CT.SSV  000301    189L    2036  2077  2094  3072  4004
CT.STE  000211    120L    1839  5072
CT.STP  000255    162L    5073
CT.STR  000345    234L    5071
CT.SYE  000212    124L    4969  5083
CT.TAB  000346    235L    2592  5074
CT.TAN  000347    236L    5075
CT.THN  000316    206L    1976  5076
CT.TO   000317    207L    1828  5077
CT.UNF  000245    151L    5078
CT.UNL  000246    152L    5079
CT.UNS  000247    153L    5080
CT.VAL  000356    247L    5081
CT.VARH 000307    194E    2062  3335  6501  6745
CT.VARL 000300    193E    1355  2060  3333  6499  6743
CT.VNV  000302    191L
CT.VSV  000303    192L
CT.WRI  000313    203L    2405  5082
CTB     103070   7768     7815  7930  7957  8112L
CTB1    103101   8118L    8127
CTLA    000001    387E
CTLB    000002    388E     973  10671
CTLC    000003    389E     976  10674
CTLD    000004    390E
CTLFLAG 112204    991     1069  2206  2548  7024  7524  10605L
CTLO    000017    391E
CTLP    000020    392E
CTLQ    000021    393E
CTLS    000023    394E
CTLZ    000032    395E
CTP.2SB 000010    703E
CTP.BKM 000002    704E
CTP.BKS 000200    699E
CTP.FF  000100    700E
CTP.MLI 000040    701E
CTP.MLO 000020    702E
CTP.TAB 000001    705E
CUF     073104   5904E    7111
```

```
CUF1    073107  5908L   5916
CUF2    073126  5914    5920E
CUF3    073163  5928    5946L
CUF4    073172  5932    5954L
CURADR  112177  1210    1342    1411    1451    1771    1959    10591L
CURNUM  112175  1113    2752    2814    6458    7329    10590L
CVX     073210  2263    2297    4210    4726    4790    5978L
CXV     073237  2278    2926    6011L
CXV.    073240  5447    6012L
CXY     073223  4361    4430    4498    4721    4767    4788    4809    5993E
D.CON   040110   653L
D.RAM   040240   656L
D.VEC   040130   655L
DATPTR  114345  1350    2693    2695    10758L
DCN     073253  2025    2136    2584    6037L
DCN.    073273  1464    2412    6042    6051L
DCN.,   073302  6061L
DCN1    073322  6068    6070L
DDN     111233  5219    9514    9737E
DDN1    111244  9438    9741L   9760
DDN2    111275  9442    9759L
DDNERR  070122  9739    9751    9755    9809E
DEF     046133  1197    1610E
DEFALTD 043100   966L    2457
DEFALTP 043072   965L    2765    2773    2851    7115
DELETE  046162  1156    1634E
DF.CLR  000376   501E
DF.EMP  000377   500E
DFD     111301  9481    9769L   9782
DFD1    111331  9485    9781L
DIM     046236  1169    1666E   1757
DIM2    046265  1680L   1704
DIM3    047011  1745L   1750
DIM5    047033  1669    1761L
DIMA    047034  1668    1762E
DIR.ALD 000025   516L
DIR.CLU 000015   509L
DIR.CRD 000023   515L
DIR.EXT 000010   504L
DIR.FGN 000020   512L
DIR.FLG 000016   510L
DIR.LGN 000021   513L
DIR.LSI 000022   514L
DIR.NAM 000000   503L
DIR.PRO 000013   505L
DIR.VER 000014   506L
DIRELEN 000027   518E    550     831
DIRIDL  000015   507E
DIV     106264  4502    4616    4700    4770    4792    9073E   9535
DIV0    106305  9076    9082L
DIV1    106355  9103L   9151
DIV2    106365  9108L   9142
DIV3    107020  9120    9130L
DIVA    106367  9088    9110E
DIVB    106373  9090    9114E
DIVC    106377  9092    9118E   9163
DM.MR   000000   288E
DM.MW   000001   289E
```

```
DM.RR    000002    290E
DM.RW    000003    291E
DNF      073326    1471      6086L   6098
DTS      073357    1131      1323    6110L  10663
DTSA     073366    1340      6113E
EC.CNA   000004    439L
EC.DDA   000027    458L
EC.DIF   000017    450L
EC.DIW   000035    464L
EC.DNI   000045    472L
EC.DNR   000046    473L
EC.DNS   000005    440L
EC.DSC   000047    474L
EC.EOF   000001    436L      5171    7655    7785    8174
EC.EOM   000002    437L
EC.FAO   000031    460L      7585
EC.FAP   000026    457L
EC.FL    000030    459L
EC.FNF   000014    447L      2768
EC.FNO   000011    444L      7673
EC.FNR   000034    463L
EC.FOD   000043    470L
EC.FUC   000013    446L
EC.ICN   000016    449L
EC.IDN   000006    441L
EC.IFC   000020    451L
EC.IFN   000007    442L
EC.ILC   000003    438L
EC.ILO   000040    467L
EC.ILR   000012    445L
EC.ILV   000037    466L
EC.IOI   000052    477L
EC.IS    000032    461L
EC.NCV   000050    475L     10717
EC.NEM   000021    452L
EC.NOS   000051    476L
EC.NPM   000044    471L
EC.NRD   000010    443L
EC.NVM   000042    469L
EC.OTL   000053    478L
EC.RF    000022    453L
EC.UNA   000036    465L
EC.UND   000015    448L
EC.UUN   000033    462L
EC.VPM   000041    468L
EC.WF    000023    454L
EC.WP    000025    456L
EC.WPV   000024    455L
EKA      073374    2190      6128L
EKA1     074001    6131L     6134
EKA2     074011    6140L     6145
ELN      074033    1437      1951    2371    6167L
ELN1     074055    6180L     6190
ELN2     074102    6173      6194L
END      047044    1198      1770L
ENL      000212    385E     10677
EOFFLG   103216    7664      7793    8146    8173    8200L
ERR.AC   070205    4323      5165L
```

```
ERR.CB  070111  1219    5105L
ERR.CC  070106  1072    2067    2209    5102L   5413
ERR.CIU 070221  2444    5177L
ERR.DO  070117  5111L   6946    9079
ERR.DE  070114  2709    5108L
ERR.EOF 070213  5171L   7075
ERR.FAE 070177  2767    5159L
ERR.FNO 070210  3545    5168L   7069    7438
ERR.IC  070174  2913    5156L
ERR.ILF 070202  5162L   5610    5613
ERR.IN  070122  1268    1507    1529    1538    1546    1567    1570    1718    1720    2521    3461    3523
                3661    3752    4412    4462    5114L   5526    5561    5692    5697    6063    6066    6189    6391
                6394    9519    9809
ERR.IU  070125  1026    1136    1147    5117L   6553
ERR.LK  070130  5120L   6585
ERR.ND  070171  3106    5153L   6511
ERR.NV  070133  2260    5123L
ERR.OV  070136  4367    4386    4413    5126L   8734    8810    8890    9018    9082    9084    9352
ERR.RE  070141  2737    5129L
ERR.SC  070166  3150    3158    5150L
ERR.SL  070144  3061    4191    5132L
ERR.SN  070147  1449    1520    1953    5135L   6297
ERR.SR  070163  3136    5147L
ERR.SY  070152  1010    1164    1170    1356    1358    1616    1651    1706    1977    2061    2063    2078
                2365    2383    2406    3105    3372    3409    3412    3668    4280    4306    5138L   5607    6175
                6500    6502    7010    7156    7216
ERR.TC  070155  3313    3382    3435    3733    4061    5141L   5445    5791    7092
ERR.TO  070160  1689    2430    5144L   6249    8311    8320    8362
ERR.UD  070216  4283    5174L
ERROR1  070244  5188    5193E
ESC     000033  383E
EVAL    055244  2120    2603    3200E   3358    3374    3432    3730    4313    4332    5602
EVALI   057036  1636    1640    1682    1969    2148    2156    2359    2481    2484    2518    2799    3129
                3446L   3457    6053    6195
EVALI8  057044  2649    3457L   3627    3671    3738
EVALN   057022  1829    1840    3432L   3446    3834
EXEC    043241  1047E   1414
EXEC1   043244  1052L   1099    1121    1988
EXEC10  044231  1226L
EXEC2   043270  1064E   1132
EXEC3   043372  1059    1125L
EXEC3.5 043375  1129L   1230
EXEC4   044007  1125    1135L
EXEC5   044035  1142    1149L
EXEC6   044040  1117    1150L
EXEC7   044201  1065    1118    1204L   2822
EXEC8   044215  1074    1215L
EXEC9   044225  1223L
EXECA   044244  1088    1234L   1239
EXP     063075  3395    4269    4353E
EXP1    063133  4365    4368L
EXPA    063221  4359    4392L
FB.CHA  000000  484L    7569    7572    8059    8070    8186
FB.FLG  000001  485L    2432    2441    2446    5621    5676    6091    7572    7589    7627    8014    8026
                8029    8188
FB.FWA  000002  486L    5380    7589    7595    8029    8190
FB.LIM  000006  488L    7600    7605    8194
FB.LWA  000010  489L    8196
```

FB.NAM	000012	490L	491	2446	2448	2449	2862	5615	7605				
FB.NAML	000021	491E	933	939	945	951	957	963	2450	2451	2867	5612	
FB.PTR	000004	487L	7595	7600	8026	8192							
FBENL	000033	492E	5380	5392	5580	6089							
FBLIST	042230	929L	2448	5380	5582	5599	5615	5621	5622	5676	6091	7114	7120
		7132											
FBSCR	042230	931L	10654	10655	10656	10659							
FBUFAD	042226	927L	5369	5372									
FF	000014	386E											
FILTAB	112163	2428	5371	5544	5573	5673	6086	6096	10572L				
FLN	074242	1448	1519	1638	1642	1952	2157	2164	5241	6287L			
FLN1	074260	6294	6299L	6314									
FLN1.5	074300	6306	6309L										
FLN2	074301	6310L	6313										
FLN3	074312	6305	6308	6318L									
FNRMA	112327	10754E											
FOC	074115	995	1573	2460	2786	6212L	7117	10678					
FOC1	074146	6222	6227L										
FOC1.3	074152	1601	6229L										
FOC1.5	074162	6235L	6267										
FOC2	074205	6239	6243L										
FOC3	074210	6233	6247L										
FOP	074217	2453	5598	6265L	6440	7113							
FOP.	074230	1578	6266	6269L									
FOR	047060	1171	1791E										
FOR1	047104	1795	1804E										
FORTAB	112132	1335	1800	1806	2270	2271	2322	2325	7220	7223	10499L		
FP0.0	031320	10621E											
FP0.1	112221	10620L											
FP1.0	112211	1838	4260	4789	10618L								
FP10.	112215	9532	9593	9673	9774	10619L							
FPADD	104352	2274	3561	4183	4528	8628L	9615						
FPDIV	106260	4235	9064L	9601	9606								
FPMODE	043316	1087E	1539										
FPMUL	105323	4233	4268	8879L	9597	9674							
FPNEG	105302	3320	3472	3538	3559	3591	3982	8851L	9554	9583			
FPNRM	105202	3951	8779L	9671									
FPSUB	105166	2302	3838	4106	4219	8761L							
FPTST	105316	8867L											
FRC	061262	3501	4004L										
FREE	047213	1172	1856E										
FREE1	047225	1863L	1878										
FREEA	047272	1861	1889E										
FREEZE	047336	1173	1908E										
FREZEA	050016	1919	1930L	1934									
FREZEAL	000010	1921	1934E										
FREZEB	050022	1918	1932L										
FSE	074315	3484	3705	3742	3746	4014	4134	4138	4199	4205	5455	5488	5491
		5608	6332L	7352									
FSE0	074341	6345	6349L										
FSE1	074344	6347	6350	6354L	6367								
FSE2	074363	6359	6363L	6366									
FSE3	074374	6356	6362	6371L									
FT.ABS	000000	867E	888	1930									
FT.BAC	000003	870E											
FT.DD	000001	529E											
FT.OC	000020	533E											
FT.OR	000002	530E	7559	7563	7632	7634	7672						

FT.OU	000010	532E						
FT.OW	000004	531E	7561	7563	7633	7634	7844	8020
FT.PIC	000001	868E						
FT.REL	000002	869E						
FTA	110301	2613	3990	7368	9574E			
FTA1	110323	9580	9585L					
FTA10	111154	9694L	9696					
FTA11	111167	9700L	9702					
FTA12	111177	9680	9707L	9710				
FTA13	111214	9703	9712	9714L				
FTA2	110331	9591	9591L	9619				
FTA2.5	110374	9603	9607L					
FTA2.7	111020	9587	9623E					
FTA3	111031	9628	9630L					
FTA4	111034	9636L	9675					
FTA4.5	111043	9637	9640L					
FTA5	111047	9645L						
FTA6	111065	9653L	9656					
FTA7	111102	9664L	9666					
FTA7.5	111107	9651	9667L					
FTA8	111117	9672E						
FTA8.5	111130	9641	9679L					
FTA9	111151	9688	9692L					
FTAA	111227	9614	9726L					
FTAC	111022	1530	9626E					
FTAD	111032	1531	9631E					
FWBRK2	102273	7974L	7981					
FWBRK3	102307	7976	7983L					
GOSTAB	112137	1336	2732	2734	2742	7326	10514L	
GOSUB	050026	1174	1941E					
GOTO	050031	1175	1950E	1986	2003			
GOTO1	050034	1952L	2377					
GOTO2	050042	1229	1955E					
HLCPDE	112010	5910	9931L					
I.CONFL	000004	720E	721	9997				
I.CONTY	000001	707E	708					
I.CONWI	000003	713E	714					
I.CSLMD	000000	696E						
I.CUSOR	000002	710E	711	10191				
IBT1	104270	8561L	8613					
IBT2	104301	8529	8569L					
IBTA	104244	8519	8542E	8564	8594			
ICL	065364	1002	1258	4850E				
ICL.	065373	4859L	7125					
ICL1	066000	4867L	4878	4880	4889	4891	4964	
ICL1.5	066001	4868L	4945	4951				
ICL10	066234	4873	4990L					
ICL2	066056	4896L	4921					
ICL3	066062	4900L	4910					
ICL4	066104	4915L	4918					
ICL5	066122	4902	4928L					
ICL5.5	066142	4936	4938L					
ICL6	066163	4932	4949L					
ICL7	066176	4876	4956L	4963				
ICL8	066216	4922	4952	4961	4969L			
ICL9	066223	4941	4943	4977L	4982			
IF	050051	1176	1968E					
IF0	050114	1984	1987L					

```
IF1       050120    1992L     1995
IF2       050121    1188      1980      1993L     2002
IF3       050127    1975      2000L
IFIX      075002    3447      3495      3519      3866      3880      3892      3961      6385L     6901
IFIX.     074377    4062      4362      6384L     6903
IFIX1     075026    6397L     6399
IFLT      075040    3536      3807      3870      3978      3980      3981      4069      4369      4432      6411L     7366
ILM       075063    1774      2825      5202      6440L
ILMA      075077    1053      1670      6448E
INP1      050151    2016      2024L
INP2      050205    2032      2036L
INP4      050233    2034      2037      2051L     2102
INP4.5    050244    2053      2055E
INP5      050304    2075L
INP6      050356    2071      2099L
INPUT     050150    1199      2023L
INPUTA    050370    2024      2068      2104L
INPUTB    050371    2030      2101      2105L
INPUTC    050373    2047      2106L
INT       057216    3396      3554E     4771
INT1      057245    3571L     3575
INT2      057260    3572      3577L
INTA      057302    3560      3594L
IOC.CGN   000010     538L
IOC.CSI   000011     539L
IOC.DDA   000002     526L      534       548
IOC.DES   000016     545L
IOC.DEV   000020     546L
IOC.DIL   000021     548E
IOC.DIR   000023     550L
IOC.DRL   000010     542E
IOC.DTA   000014     544L
IOC.FLG   000004     528L      542
IOC.GRT   000005     536L
IOC.LGN   000012     540L
IOC.LNK   000000     525L
IOC.LSI   000013     541L
IOC.SPG   000007     537L
IOC.SQL   000003     534E
IOC.UNI   000022     547L
IOCCTD    000001     554E
IOCELEN   000052     552E
IOCHAN    112202    1055      1466      2041      2051      2418      2780      6038      6070      6570      7060      7423      10600L
IP.CON    000362     264E
IP.PAD    000360     260E
IST       075136    2080      2113      4297      6479L     6789      7214
IST0      075172    6479      6497E
IST00     075170    6483      6494L
IST1      075216    6506L     6555
IST2      075257    6505      6535L
IVT       075263    1613      1671      6550L
KEYTAB    066240    4895      4998E     6128
L1        106024    8917L     8927
L2        106035    8921      8924L
L4        106077    8944      8952L
L4.5      106120    8938      8959      8961      8963L
L5        106122    8966L     8991
L6        106144    8970      8982L
```

```
L7        106174   8992    8994    8996    9000L
LCC       075302   2413    2619    2656    6569L   7382    7418
LCKFLG    112201   2247    6583   10592L
LDD       107250   6388    8641    9213    9317L
LDX       107245   8780    8852    9099    9251    9309L
LEFT$     057314   3418    3621E
LEFT$1    060032   3635    3683E
LEN       057306   3419    3601L
LET       050374   1139    1177    2113L
LET.      050377   1826    2115L
LEV1      055255   3202    3210L
LEV11     055260   3211L   3219
LEV2      055304   3210    3214    3223L
LEV21     055307   3224L   3232
LEV3      055333   3223    3227    3233E
LEV4      055333   3237L
LEV41     055336   3238L   3250
LEV5      055367   3237    3244    3254L
LEV51     055372   3255L   3267
LEV52     056002   3256    3259L
LEV6      056025   3254    3261    3271L
LEV61     056030   3272L   3284
LEV62     056040   3273    3276L
LEV7      056063   3271    3278    3288L
LEV71     056066   3289L   3297
LEV8      056112   3288    3292    3301L
LEV81     056127   3303    3306L
LEV82     056163   3316    3326L
LEV9      056170   3305    3308    3332L
LEV90     056307   3379    3381L
LEV92     056234   3334    3336    3355L
LEV93     056256   3357    3366L
LEV94     056247   3362L   3384
LEV95     056220   3346L   3351
LEX0      054202   2910    2917L
LEX1      054231   2908    2934E
LEX10     054354   2904    3006L
LEX11     054377   2906    3023L
LEX11.5   055012   2086    3040L
LEX12     055015   2902    3043L   6879
LEX13     055027   3049L   3056
LEX14     055044   3054    3059L
LEX2      054264   2943    2945    2956L
LEX3      054274   2958    2962L
LEX3.5    054312   2965    2973L
LEX7      054334   2978    2993L
LEXA      075236   2993    6516E
LEXB      042222    911L   2925    2929    3063    3064    3073
LEXC      042215    908L   2995
LEXCAL    054131   2893E   6751
LEXD      055033   3045    3053E
LEXLIM    042226    913E   1360    6503    6551
LF        000012    373E
LFC       075313   1251    1288    1300    1318    1403    1635    2835    5217    6583L
LINE      112327   1011    1241    2065    2070    4851    5218    5411    7119   10702   10703   10743L  10745
                   10754
LINE2     113335   2180    2612    2865    3989    4016    4019    6857    6874    7367    7402   10708   10709
                   10750L
```

```
LINEL    000377    7072     7118   10702  10703  10708  10709  10745E 10750
LINPUT   050137    1195     2012E
LIST     051020    1157     2134E
LIST.    051023    2140L    2782
LIST1    051065    2147     2154   2161L
LIST2    051073    2168L    2210
LIST3    051126    2187L    2194
LIST8    051170    2177     2214L
LNO      057064    3397     3476L
LOADL    113067     890     10726E
LOCK     051175    1178     2244L
LOG      063225    3398     4266   4407E
LOGA     063350    4433     4442L
LOGB     063354    4437     4443L
LSH      107101    8830     9140   9169   9180L
LVS      075323    2975     6603E  7283
LVS1     075335    6612L    6672   6681
LVS2     075363    6620     6622   6635L
LVS3     076027    6645     6649   6676L
LVS4     076040    6614     6685L
M.CDCA   000017     857L
M.CDLY   000016     856L
M.CFWA   000012     854L
M.CIN    000006     852L
M.CINT   000005     851L
M.CLWA   000014     855L
M.COUT   000010     853L
M.CPRE   000003     849L
M.CRUB   000004     850L
M.CSLC   000002     848L
M.FOX    000303     308E
M.PAMB   000021     307E
M.SALO   000001     847L
M.SUNI   000021     858L
M.SYDD   000022     859L
M.SYSM   000000     846L
MATC2.3  060257    3757     3789L
MATC2.5  060260    3761     3790L
MATCH    060111    3420     3725E
MATCH1   060225    3768L    3787
MATCH2   060247    3770     3783L
MATCH3   060273    3779     3799L
MATCHA   060307    3726     3745   3809L
MATCHB   060276    3749     3802E
MATCHC   060313    3734     3741   3810L
MAX      060317    3399     3822L
MAX1     060325    3829L    3846
MAX2     061002    3831     3850L
MEML     112171    1915     6228   6235   8313   8369   10583L
MI.ADDB  000200      53E     8887
MI.CMC   000077      55E     9074
MI.IN    000333      62E     3882
MI.JMP   000303      54E
MI.LDA   000072      56E
MI.LXIB  000001      65E     5103   5106   5109   5112   5115   5118   5121   5124   5127   5130   5133
                    5136     5139   5142   5145   5148   5151   5154   5157   5160   5163   5166   5169   5172
                    5175
MI.LXID  000021      64E     3042
```

```
MI.LXIH 000041    63E
MI.MVIA 000076    57E    2244   3822
MI.NOP  000000    60E     989   5342
MI.OUT  000323    58E    2473
MI.RET  000311    61E    6757
MI.SUBB 000220    59E    9074
MID$    057314  3421     3623E
MID$1   057365  3634     3657E
MID$2   060032  3651     3665   3684E
MID$3   060043  3691     3693L
MID$4   060050  3694     3696L
MIN     060320  3400     3823L
MOV     071015  4000     4017   4202   4208   5285L  5494
MOV4    076051  1834     1841   3727   3735   4262   5981   6015   6710L
MOV5    076045  5997     6705E  6952   6990
MT.AFC  000000  10375E
MT.FWA  000001  1384     2271   2732   5239   5266   5371   5464   5478   5809   5830   5922   6346
                6349     6610   6948   7223   10376E
MT.LEN  000003  1325     1326   1335   1336   1337   1667   1763   1858   2270   2322   2325   2428
                2734     2742   5183   5573   5635   5673   5908   6086   6096   6111   6607   6942   6945
                7169     7220   10377E
MTABIND 112120  1858     5183   5635   8426   10378E 10581
MTABL   000010  1860     5638   8425   10581E
MTABLEN 000005  10586E
MTAREA  115010   927      938    938    938    938    944    944    944    944    950    950    950
                 950      956    956    956    956    962    962    962    962   1341   1349   1653   1770
                5637     6288   7171   7172   10394  10483  10500  10515  10528  10554  10565  10573  10583
                10647    10768E
MTD     104127  6269     8305   8439L
MTD1    104151  8459L    8479
MTL     070270  1021     5216E  7127
MTL0    070304  1264     5224L
MTL1    070307  5226L    5228
MTL1.5  070325  5231     5233L
MTL2    070361  5243     5251L
MUL     105327  4360     4438   4554   4628   4663   4681   4733   4773   8886E  9530   9543   9775
MUL2,5  106047  8916     8928   8933L
MUL3    106061  8940L    8958
MUL3,5  106216  9005     9017L  9168
MUL4    106222  9022L    9170
MUL5    106240  8889     9033L  9080   9476
MULA    106034  8894     8923E
MULB    106051  8907     8935E
MULC    106072  8898     8947E
MULD    106076  8895     8951E
MULE    106121  8904     8964E
MULF    106133  8901     8973E
MULG    106137  8899     8977E
MULH    106143  8896     8981E
NEG     105305  4358     4535   4552   4568   4584   4599   4613   4644   4668   8764   8852L
NEXT    051203  1179     2258E
NL      000012   384E     385   1276   2201   5198   7030   7389   7739   9848  10677  10677  10719
NPI     112235  4585     10624L
NPI.2   112225  4527     4544   4550   4590   4597   4666  10622L
NPI.4   112241  4603     4611  10625L
NPI2    112231  4539     4768  10623L
NRM     105213  6414     8739   8748   8790E  9347
NRM.    105205  8780L
```

```
NRM0      105221    8743      8798L   8832
NRM1      105227    8805L     8826
NRM2      105242    8794      8814L
NRM3      105257    8823      8825L
NRM4      105263    8830L     8837
NUL2      000000     375E
NULL      000200     374E
NXT1      051271    2293      2302L
NXT1.5    051307    2306      2311L
NXT2      051317    2307      2322L
OLD       051332    1180      2335E
ON        051355    1181      2359L
ON1       052001    2363      2367E
ON2       052001    2371L     2384
ONA       052035    2373      2378    2386L
OP.CTL    000360     261E
OP.DIG    000360     262E
OP.SEG    000361     263E
OP2.CTL   000362     265E
OPEN      052036    1182      2398E
OPEN1     052062    2404      2407L
OPEN2     052112    2423L     2435
OPEN3     052144    2425      2440L
OPEN4     052210    2458      2462L
OUT       052220    1183      2471L
OUT1      052235    1252      1504    2471    2481L   2559
OVL.COD   000000     626L
OVL.ENS   000010     631E
OVL.ENT   000004     628L
OVL.FLB   000006     629L      1583
OVL.IN    000001     764E      1586
OVL.NUM   000014     768E
OVL.RES   000002     765E
OVL.SIZ   000002     627L
OVL.UCS   000200     767E
OVL0      000000     637L      1582
OVL1      000001     638L
OVLMAN    112203    1571      2337    6220    6248    10602L
P.ADD     062134    3265      4179L
P.ADDA    062235    4194      4195    4209    4212L
P.AND     061336    3230      4045L
P.CMP     061375    3248      4100L
P.CMP1    062013    4109L     4173
P.CMP13   062030    4113      4115    4117L
P.CMP2    062047    4102      4132L
P.CMP3    062073    4147L     4156
P.CMP4    062115    4150      4160L
P.CMP5    062121    4148      4165L
P.CMP6    062130    4153      4161    4166    4172L
P.EXP     062270    3295      4248E
P.EXP1    062321    4252      4265L
P.MUL     062247    3282      4228L
P.NOT     061351    3326      4059L
P.NOT1    061371    4037      4051    4068L
P.OR      061323    3217      4031L
P.SUB     062241    4181      4218L
PAD       061006    3401      3858L
PATCH     114347   10762L
```

```
PAUSE   052251   1184     2512E
PAUSE1  052275   2525L    2551
PAUSE2  052320   2528     2540L
PAUSE3  052326   2536     2548L
PBO     076345   4031     4045    6900L
PEEK    061014   3402     3866L
PEEK1   061020   3486     3488    3602    3859    3868L   3885    3900    3966
PEEK1.5 061023   3870L    4122    4124
PEEK2   061026   3871L    4022
PI.4    112245   4615     10626L
PIN     061034   3403     3880L
PLY     065204   4692     4711    4721L
PLY0    065207   4696     4722L
PLY1    065222   4730L    4744
PLY2    065240   4728     4738L
PMAA    107005   9098     9122L
PMAB    107011   9096     9125L
PMAC    107015   9094     9128L
PMD     107114   8888     9075    9203E
PMD1    107200   9237     9243L
PMD2    107204   9211     9251L
PMDA    107152   9209     9220E
PMDB    107164   9206     9228L
PNT     076072   1056     1432    1473    2029    2059    2075    2144    2151    2513    2587    2794    3301
                 3332     3353    3364    3663    3829    4336    5340    6039    6741L   6775    7208
PNT1    076130   5345     6758L
PNT2    076111   6744     6746    6749E
PNTA    076073   5344     6742E   6755
PNTB    076107   6748E    6753
PNTC    076111    990     6750L   6758
POKE    052336   1185     2558E
POLY    065177   4372     4483    4555    4710L
POLY2   065245   4743L
POLYQ   065153   4419     4691L
POP     076373   6913     6939L
POPX    076357   3841     4499    4512    6912L
POPY    076365   3215     3228    3245    3262    3279    3293    3835    4267    4626    4661    4698    6929L
POPY.   076370   3630     3672    6930L
POS     061053   3404     3892L
POS1    061067   3896     3898L
PRI1    052352   2585     2585L
PRI2    053027   2612L
PRI3    053040   2599     2618L
PRI5    053074   2639L    2643    2655    2661
PRI6    053105   2593     2595    2647L
PRI7    053142   2590     2666L
PRI8    053156   2641     2677L
PRIA    053144   2583     2591    2668E
PRIB    053050   1560     2622E
PRIC    053062   1547     2633E   2636
PRINT   052343   1186     2581E
PRS     112252    891     10636E
PRSA    112361   10677L
PRSB    113056   10699    10705   10711   10715E
PRSERR  113060   10649    10719L
PRSERR1 113056   10641    10643   10717L
PRSLIM  113067   10724E   10731   10746
PSH     077044   6965     6985L
```

```
PSHX    077033  3626    3836    4458    4679    4694    6964L
PSHX.   077030  3213    3226    3243    3260    3277    3291    6963L
PSHY    077041  3670    3833    4265    6975L
PTS     065252  4532    4577    4760L
PVI     076135  2099    2694    6775L   6821
PVI1    076210  6801    6809L
PVI10   076325  6853    6875    6879L
PVI2    076230  6805    6817E
PVI3    076236  6784    6825L
PVI5    076240  6811    6833L
PVI6    076244  6839L
PVI7    076261  6834    6850L
PVI8    076274  6859L   6868
PVI9    076312  6865    6872L
QUOTE   000047   381E
RAR     065331  4540    4545    4586    4591    4604    4806L
RAR1    065332  4808L   4818
RAR2    065357  4815    4820L
RCE     077062  6792    6812    7004L
RCX     065312  4388    4632    4648    4788L
REA1    053174  2694L   2714
REA2    053203  2700L   2703    2712
READ    053171  1187    2692E
REPLACE 053233  1158    2723E
RESTART 043124   981E    986    1052    1259    1764    6447   10714
RESTORE 045053  1189    1349L
RETURN  053242  1190    2732L
RIGHT$  057314  3422    3622E
RIGHT$1 057356  3645    3647L
RIL     077075  4852    5412    7023L   7062
RIL1    077076  7024L   7029    7035
RIL2    077120  7031    7033L
RIL3    077137  7027    7047L
RLF     077141  2066    7060L
RM.CON  000004    47E   1410
RM.HLT  000200    48E   1081    1102    1205    6452    6454
RM.IMM  000000    45E    993    1029    1145    1206    1416    6454
RM.STE  000001    46E   1119    2805
RND     061074  3405    3911E
RND1    061116  3918    3920L   3939
RND2    061150  3917    3942L
RNDA    061101  3915E   3940   10667
RNO     077177  4218    4231    6900    7090L
RNP     077206  1441    2346    7110L
RNP1    077236  7118L   7128
RNP2    077273  7124    7126    7132L
RNT     077305  1129    1611    1827    2013    2045    2118    2400    2408    2410    2650    2800    3362
                4330    4333    5709    6051    6197    7150L
ROMBOOT 030000   648E
RUBOUT  000177   377E
RUN     045155  1159    1403L
RUNMOD  114343   994    1048    1078    1144    1204    1418    6450   10756L
RVU     107315  9165    9364L
S.BAUD  040344   741L
S.BDA   041120   839L
S.BOOTF 041034   796L
S.CAADR 040333   724L
S.CACC  041006   780L
```

```
S.CCTAB 040335    725L
S.CDB   040343    738L
S.CFWA  040352    748L
S.CODE  041007    781L
S.CONFL 040332    722L
S.CONTY 040327    709L
S.CONWI 040331    715L    1550
S.CSLMD 040326    697L     708     711     714     721     992
S.CUSOR 040330    712L
S.DATC  040310    678L
S.DATE  040277    677L
S.DCS   041033    794L
S.DDDTA 040366    759L
S.DDGRP 040364    756L
S.DDLDA 040360    754L
S.DDLEN 040362    755L
S.DDOPC 040370    760L
S.DFWA  040354    749L
S.DIREA 041016    788L
S.DLINK 040346    746L
S.FASER 041013    787L
S.FCI   041021    789L
S.GRT0  024000    644E
S.GRT1  025000    645E
S.GRT2  026000    646E
S.GUP   041027    791L
S.HIMEM 040316    680L
S.INT   040343    658L     734
S.JUMPS 041010    785L
S.MOUNT 041032    793L
S.OFWA  040350    747L    1581
S.OMAX  040324    686L    1595    6224
S.OSN   041004    776L
S.OVLE  041000    773L
S.OVLFL 040371    769L
S.OVLS  040376    772L
S.OVSTK 041035    801L
S.RFWA  040356    750L
S.SCI   041024    790L
S.SCR   041121    840L   10653
S.SDD   041010    786L
S.SOVR  041146    660L     662
S.SSN   041002    775L
S.SYSM  040320    682L    1587    6217
S.TIME  040312    679L
S.UCSF  040372    770L
S.UCSL  040374    771L
S.USRM  040322    684L    6237
S.VAL   040277    657L     675
SAVE    053302   1160    2762E
SAVE1   053324   2726    2773L
SCR.    044360   1303L   2342   10679
SCRA    077320   1303    5191    7110    7168L
SCRATCH 044351   1161    1299E
SEG     061170   3406    3961L
SERROR  070223   5179E   6242
SES     077342   1196    1627    7187L   7190    7324
SFS     077362   1792    7214L
```

```
SFS.    077352   2259     7208L
SFS0    077375   7212     7219L
SFS1    100025   7229     7240L   7263
SFS2    100060   7253     7257L
SFS3    100073   7242     7267L
SFS4    100074   7256     7271L
SFS5    100075   7272L    7295
SFS6    100120   7251     7292L
SGN     061205   3407     3975L
SIN     064117   3408     4526E
SOB     100126   1981     2894    2962    6167    7304L   7310
SOB1    100137   7306     7309L
SPACE   112251   2678     10628L
SPE     107215   8628     8761    8779    8851    8879    9064    9268L
SPEX    107225   9271     9277L
SQR     063360   3410     4456E
SQRT1   064011   4477L
SQRT2   064015   4473     4480L
SQRT3   064112   4463     4512L
SRA     100143   1228     1942    2366    7322L
SRS     107231   6397     8696    9288E
SRS.    107232   8732     9290L
SRS..   107233   3574     9291L   9353
STACK   042200    684E     985
STACKL  001032    662E
START   043106    971E     1933
STATE   114344   10757L
STEP    053356   1162     2794L
STEP1   053374   2804L    2811
STO     106250   9045L    9479
STOP    054030   1200     2822L
STR$    061231   3411     3989L
STR$1   061257   3713     4000L
STRLEN  112154   10555L
STRTAB  112151   1325     5464    5475    5478    5855    6349    10553L
STRTI   112207   5859     6114    10613L
STRVI   112205   1328     5853    10612L
STX     106245   6415     8744    8749    8854    9027    9029    9044L   9253
SUB     105172   4371     4775    8763L
SYDD    040130    654E
SYMTAB  112125   1326     1384    1393    1667    1694    1738    1763    5809    5830    5908    5922    5940
                 6512     6607    6610    10482L
SYSCALL 000377    563E     974     977     1007    1277    1292    1863    2054    2838    2853    3530    5196
                 5674     6241    6442    6445    7028    7621    7886    7994    8061    8073    8165    8220    8233
                 8237     9849    9999    10160   10193   10640   10648   10672   10675   10720   10722
T.CHA   103204   7885     7993    8089    8117    8164    8187L   8188    8190    8192    8194    8196    8198
T.FLG   103205   7671     7843    8189L
T.FWA   103206   7875     7983    8153    8191L
T.LIM   103212   7684     7796    8157    8177    8179    8195L
T.LWA   103214   7854     7878    7960    7986    8156    8197L
T.PTR   103210   7682     7745    7753    7798    7808    7852    7876    7920    7962    7984    8154    8193L
TAB     000011    382E     7307    9900
TAN     064243   3413     4576E
TAN1    064321   4593     4603L
TAN2    064347   4607     4615L
TANA    065016   4605     4630E
TBL1    112042   10083L   10089
TBL2    112060   10081    10093L
```

TBL3	112062	10086	10097L							
TCS	100200	2043	2608	7351E						
TCV	107260	8853	9024	9217	9252	9330E				
TDI	100206	1257	1886	6460	7365L					
TDI.	047264	1876	1886L	2816						
TLEN	000012	8088	8198E							
TSTTAB	112156	5861	6111	6346	10564L					
TXTF1	062363	4290L	4325							
TXTF2	062366	4296L								
TXTF3	063013	4304	4310L							
TXTFN	062340	3367	4277E							
TXTFNA	063041	4315	4322E							
TXTTAB	112120	1658	5239	5264	5266	7169	10393L			
UNFREZ	054041	1191	2834E							
UNFREZA	054057	1912	2837	2841L						
UNLOCK	051176	1192	2246L							
UNSAVE	054065	1193	2849E							
UNSAVE1	054103	2836	2850	2861L						
UO.CLK	000001	300E	1234	1235	1236					
UO.DDU	000002	299E	1235							
UO.HLT	000200	297E	1234	1235	1236					
UO.NFR	000100	298E	1234							
USERFWA	042200	665E	887	889	890	1916	1924	1931		
VAL	061270	3423	4013L							
VAR2	055117	3096	3102L							
VAR4	055136	3115L	3152							
VAR5	055220	3143	3156L							
VARIAB	055104	3092E	3340							
VARIAB.	055107	3094L	6535							
VERS	000040	561E	10642							
WEL	100225	2623	2673	7382L						
WELA	100241	7385	7389L							
WLF	100242	2204	7402L							
WLF.	100251	2614	2680	7353	7387	7414L				
WLF2	100301	7425	7435L							
WRKTAB	112144	1337	6942	6945	6948	6987	10527L			
XCY	100317	3825	3845	3851	4234	4249	4820	7451L		
XCY1	100332	7457L	7465							
XPOLYR	065142	4617	4650	4679L						
ZERO	115007	1415	1927	2215	2785	5417	7134	10696	10697	10767L
ZRO	100351	5483	7481L	7487						

13214 BYTES FREE