```
                        ***     EDIT - HEATH HDOS TEXT EDITOR.
                        *
                        *       ADAPTED FROM 'HOS8 - WINTEK TEXT EDITOR'
                        *
                        *       J. G. L.,     12/12/77,  FOR *HEATH* CORPORATION
                        *
                        *       COPYRIGHT 12/1977, 79/05        BY *HEATH* CORPORATION.
                        *
                        *       G. C.,                  79/05   --.04.--
                        *                               79/12   --.05.--
                        *                               80/02
                        *       W. Z.,                  80/08   --.06.--
                        *


                        ***     EDIT PERFORMS SIMPLE LINE AND STRING EDITING FUNCTIONS.
                        *
                        *       SEE THE 'EDIT' USERS MANUAL FOR INSTRUCTIONS.
```

```
                        23  ****    ASSEMBLY CONSTANTS.
000.000                 24          XTEXT   ASCII

                        26X **      ASCII CHARACTER EQUIVALENCES.
                        27X
000.015                 28X CR      EQU     13              CARRIAGE RETURN
000.012                 29X LF      EQU     10              LINE FEED
000.200                 30X NULL    EQU     200Q            PAD CHARACTER
000.000                 31X NUL2    EQU     0
000.007                 32X BELL    EQU     7               BELL CHARACTER
000.177                 33X RUBOUT  EQU     177Q
000.010                 34X BKSP    EQU     10Q             CTL-H
000.026                 35X C.SYN   EQU     26Q             SYNC
000.002                 36X C.STX   EQU     2               STX
000.047                 37X QUOTE   EQU     47Q
000.011                 38X TAB     EQU     11Q
000.033                 39X ESC     EQU     33Q
000.012                 40X NL      EQU     12Q             NEW LINE (HDOS SYSTEMS)
000.212                 41X ENL     EQU     NL+200Q         NL + END-OF-LINE-FLAG
000.014                 42X FF      EQU     14Q             FORM FEED
000.001                 43X CTLA    EQU     01Q             CTL-A
000.002                 44X CTLB    EQU     02Q             CTL-B
000.003                 45X CTLC    EQU     03Q             CTL-C
000.004                 46X CTLD    EQU     04Q             CTL-D
000.017                 47X CTLO    EQU     17Q             CTL-O
000.020                 48X CTLP    EQU     20Q             CTL-P
000.021                 49X CTLQ    EQU     21Q             CTL-Q
000.023                 50X CTLS    EQU     23Q             CTL-S
000.032                 51X CTLZ    EQU     32Q             CTL-Z


                        53  **      COMMAND OPTIONS.
                        54
000.001                 55  OPT.A   EQU     1               PRINT LINE AFTER PROCESS
000.002                 56  OPT.B   EQU     2               PRINT LINE BEFORE PROCESSING


                        58  **      MACHINE INSTRUCTIONS.
                        59
000.072                 60  MI.LDA  EQU     072Q
000.000                 61  MI.NOP  EQU     000Q
000.311                 62  MI.RET  EQU     311Q
                        63
                        64  ****
```

```
000.000                 66          XTEXT   FBDEF


                        68X **      FILE BLOCK DEFINITIONS.
                        69X
000.000                 70X         ORG     0
000.000                 71X FB.CHA  DS      1           CHANNEL NUMBER
000.001                 72X FB.FLG  DS      1           FLAGS
000.002                 73X FB.FWA  DS      2           BUFFER FWA
000.004                 74X FB.PTR  DS      2           BUFFER POINTER
000.006                 75X FB.LIM  DS      2           LIMIT OF DATA IN BUFFER (READ OPERATIONS)
000.010                 76X FB.LWA  DS      2           LWA OF BUFFER
000.012                 77X FB.NAM  DS      4+8+4+1     NAME OF FILE
000.021                 78X FB.NAML EQU     *-FB.NAM
000.033                 79X FBENL   EQU     *           ENTRY LENGTH
000.033                 80          XTEXT   HOSDEF


                        82X **      HOSDEF - DEFINE HOS PARAMETER.
                        83X *
                        84X
                        85X
000.040                 86X VERS    EQU     2*16+0      VERSION 2.0
                        87X
000.377                 88X SYSCALL EQU     377Q        SYSCALL INSTRUCTION
                        89X
                        90X
000.000                 91X         ORG     0
                        92X
                        93X *       RESIDENT FUNCTIONS
                        94X
000.000                 95X .EXIT   DS      1           EXIT (MUST BE FIRST)
000.001                 96X .SCIN   DS      1           SCIN
000.002                 97X .SCOUT  DS      1           SCOUT
000.003                 98X .PRINT  DS      1           PRINT
000.004                 99X .READ   DS      1           READ
000.005                100X .WRITE  DS      1           WRITE
000.006                101X .CONSL  DS      1           SET/CLEAR CONSOLE OPTIONS
000.007                102X .CLRCO  DS      1           CLEAR CONSOLE BUFFER
000.010                103X .LOADO  DS      1           LOAD AN OVERLAY
000.011                104X .VERS   DS      1           RETURN HDOS VERSION NUMBER
000.012                105X .SYSRES DS      1           PRECEDING FUNCTIONS ARE RESIDENT
                       106X
                       107X
                       108X *       *HDOSOVLO.SYS* FUNCTIONS
                       109X
000.040                110X         ORG     40A
                       111X
000.040                112X .LINK   DS      1           LINK (MUST BE FIRST)
000.041                113X .CTLC   DS      1           CTL-C
000.042                114X .OPENR  DS      1           OPENR
000.043                115X .OPENW  DS      1           OPENW
000.044                116X .OPENU  DS      1           OPENU
```

```
000.045                117X .OPENC  DS      1               OPENC
000,046                118X .CLOSE  DS      1               CLOSE
000.047                119X .POSIT  DS      1               POSITION
000,050                120X .DELET  DS      1               DELETE
000.051                121X .RENAM  DS      1               RENAME
000,052                122X .SETTP  DS      1               SETTOP
000.053                123X .DECODE DS      1               NAME DECODE
000,054                124X .NAME   DS      1               GET FILE NAME FROM CHANNEL
000.055                125X .CLEAR  DS      1               CLEAR CHAN
000,056                126X .CLEARA DS      1               CLEAR ALL CHANS
000.057                127X .ERROR  DS      1               LOOKUP ERROR
000,060                128X .CHFLG  DS      1               CHANGE FLAGS
000.061                129X .DISMT  DS      1               FLAG SYSTEM DISK DISMOUNTED
000,062                130X .LOADD  DS      1               LOAD DEVICE DRIVER
000.063                131X .OPEN   DS      1               Parametrized Open
                       132X
                       133X
                       134X *       *HDOSOVL1.SYS*  FUNCTIONS
                       135X
000,200                136X         ORG     200Q
                       137X
000,200                138X .MOUNT  DS      1               MOUNT  (MUST BE FIRST)
000.201                139X .DMOUN  DS      1               DISMOUNT
000,202                140X .MONMS  DS      1               MOUNT/NO MESSAGE
000.203                141X .DMNMS  DS      1               DISMOUNT/NO MESSAGE
000,204                142X .RESET  DS      1               RESET = DISMOUNT/MOUNT OF UNIT
000.205                143X .CLEAN  DS      1               Clean device
000,206                144X .DAD    DS      1               Dismount All Disks          /80.08.sc/
000.207                145          XTEXT   HOSEQU


                       147X **      HDOS SYSTEM EQUIVALENCES.
                       148X *
                       149X
024.000                150X S.GRT0  EQU     24000A          SYSTEM AREA FOR  GRT0
025,000                151X S.GRT1  EQU     25000A          SYSTEM AREA FOR  GRT1
026.000                152X S.GRT2  EQU     26000A          SYSTEM AREA FOR  GRT2
                       153X
030.000                154X ROMBOOT EQU     30000A          ROM BOOT ENTRY
                       155X
040.100                156X         ORG     40100A          FREE SPACE FROM PAM-8
                       157X
040.100                158X         DS      8               JUMP TO SYSTEM EXIT
040,110                159X D.CON   DS      16              DISK CONSTANTS
040.130                160X SYDD    EQU     *               SYSTEM DISK ENTRY POINT
040,130                161X D.VEC   DS      24*3            SYSTEM ROM ENTRY VECTORS
040.240                162X D.RAM   DS      31              SYSTEM ROM WORK AREA
040,277                163X S.VAL   DS      36              SYSTEM VALUES
040.343                164X S.INT   DS      115             SYSTEM INTERNAL WORK AREAS
041,126                165X         DS      16
041.146                166X S.SOVR  DS      2               STACK OVERFLOW WARNING
041,150                167X         DS      42200A-*        SYSTEM STACK
001.032                168X STACKL  EQU     *-S.SOVR        STACK SIZE
                       169X
```

```
042.200                 170X STACK   EQU     *               LWA+1 SYSTEM STACK
042.200                 171X USERFWA EQU     *               USER FWA
042.200                 172          XTEXT   ESVAL


                        174X **      S.VAL - SYSTEM VALUE DEFINTIONS.
                        175X *
                        176X *       THESE VALUES ARE SET AND MAINTAINED BY THE SYSTEM.
                        177X *
                        178X *       THE DECK HOSEQU MUST BE MODIFIED WHEN THIS IS MODIFIED.
                        179X
                        180X
040.277                 181X         ORG     S.VAL
                        182X
040.277                 183X S.DATE  DS      9               SYSTEM DATE (IN ASCII)
040.310                 184X S.DATC  DS      2               CODED DATE
040.312                 185X S.TIME  DS      4               TIME FROM MIDNIGHT (IN TICS)
040.316                 186X S.HIMEM DS      2               HARDWARE HIGH MEMORY ADRESS+1
                        187X
040.320                 188X S.SYSM  DS      2               FWA RESIDENT SYSTEM
                        189X
040.322                 190X S.USRM  DS      2               LWA USER MEMORY
                        191X
040.324                 192X S.OMAX  DS      2               MAX OVERLAY SIZE FOR SYSTEM
                        193X
                        194X
                        195X **      THE FOLLOWING FIVE CELLS SHOULD BE MODIFIED/READ ONLY VIA THE .CONSL SYSCALL
                        196X
000.200                 197X CSL.ECH EQU     10000000B       SUPPRESS ECHO
000.004                 198X CSL.RAW EQU     00000100B       Raw Mode I/O                    /80.09.sc/
000.002                 199X CSL.WRP EQU     00000010B       WRAP LINES AT WIDTH
000.001                 200X CSL.CHR EQU     00000001B       OPERATE IN CHARACTER MODE
                        201X
000.000                 202X I.CSLMD EQU     0               S.CSLMD IS FIRST BYTE
040.326                 203X S.CSLMD DS      1               CONSOLE MODE
                        204X
000.200                 205X CTP.BKS EQU     10000000B       TERMINAL PROCESSES BACKSPACES
000.100                 206X CTP.FF  EQU     01000000B       Terminal Processes Form-Feed    /80.09.sc/
000.040                 207X CTP.MLI EQU     00100000B       MAP LOWER CASE TO UPPER ON INPUT
000.020                 208X CTP.MLO EQU     00010000B       MAP LOWER CASE TO UPPER ON OUTPUT
000.010                 209X CTP.2SB EQU     00001000B       TERMINAL NEEDS TWO STOP BITS
000.002                 210X CTP.BKM EQU     00000010B       MAP BKSP (UPON INPUT) TO RUBOUT
000.001                 211X CTP.TAB EQU     00000001B       TERMINAL SUPPORTS TAB CHARACTERS
                        212X
000.001                 213X I.CONTY EQU     1               S.CONTY IS 2ND BYTE
000.000                 214X         ERRNZ   *-S.CSLMD-I.CONTY
040.327                 215X S.CONTY DS      1               CONSOLE TYPE FLAGS
000.002                 216X I.CUSOR EQU     2               S.CUSOR IS 3RD BYTE
000.000                 217X         ERRNZ   *-S.CSLMD-I.CUSOR
040.330                 218X S.CUSOR DS      1               CURRENT CURSOR POSITION
000.003                 219X I.CONWI EQU     3               S.CONWI IS 4TH BYTE
000.000                 220X         ERRNZ   *-S.CSLMD-I.CONWI
040.331                 221X S.CONWI DS      1               CONSOLE WIDTH
                        222X
```

```
000.001                223X CO.FLG  EQU     00000001B       CTL-O FLAG
000.200                224X CS.FLG  EQU     10000000B       CTL-S FLAG
                       225X
000.004                226X I.CONFL EQU     4               S.CONFL IS 5TH BYTE
000.000                227X         ERRNZ   *-S.CSLMD-I.CONFL
040.332                228X S.CONFL DS      1               CONSOLE FLAGS
                       229X
040.333                230X S.CAADR DS      2               ADDRESS FOR ABORT PROCESSING (>256 IF VALID)
040.335                231X S.CCTAB DS      6               ADDR FOR CTL-A, CTL-B, CTL-C PROCESSING
040.343                232          XTEXT   ABSDEF



                       234X **      ABS FORMAT EQUIVALENCES.
                       235X
000.000                236X         ORG     0
                       237X
000.000                238X ABS.ID  DS      1               3770 = BINARY FILE FLAG
000.001                239X         DS      1               FILE TYPE (FT.ABS)
000.002                240X ABS.LDA DS      2               LOAD ADDRESS
000.004                241X ABS.LEN DS      2               LENGTH OF ENTIRE RECORD
000.006                242X ABS.ENT DS      2               ENTRY POINT
                       243X
000.010                244X ABS.COD DS      0               CODE STARTS HERE
000.010                245          XTEXT   ECDEF



                       247X **      ERROR CODE DEFINITIONS.
                       248X
000.000                249X         ORG     0
000.000                250X         DS      1               NO ERROR #0
000.001                251X EC.EOF  DS      1               END OF FILE
000.002                252X EC.EOM  DS      1               END OF MEDIA
000.003                253X EC.ILC  DS      1               ILLEGAL SYSCALL CODE
000.004                254X EC.CNA  DS      1               CHANNEL NOT AVAILABLE
000.005                255X EC.DNS  DS      1               DEVICE NOT SUITABLE
000.006                256X EC.IDN  DS      1               ILLEGAL DEVICE NAME
000.007                257X EC.IFN  DS      1               ILLEGAL FILE NAME
000.010                258X EC.NRD  DS      1               NO ROOM FOR DEVICE DRIVER
000.011                259X EC.FNO  DS      1               CHANNEL NOT OPEN
000.012                260X EC.ILR  DS      1               ILLEGAL REQUEST
000.013                261X EC.FUC  DS      1               FILE USAGE CONFLICT
000.014                262X EC.FNF  DS      1               FILE NAME NOT FOUND
000.015                263X EC.UND  DS      1               UNKNOWN DEVICE
000.016                264X EC.ICN  DS      1               ILLEGAL CHANNEL NUMBER
000.017                265X EC.DIF  DS      1               DIRECTORY FULL
000.020                266X EC.IFC  DS      1               ILLEGAL FILE CONTENTS
000.021                267X EC.NEM  DS      1               NOT ENOUGH MEMORY
000.022                268X EC.RF   DS      1               READ FAILURE
000.023                269X EC.WF   DS      1               WRITE FAILURE
000.024                270X EC.WPV  DS      1               WRITE PROTECTION VIOLATION
000.025                271X EC.WP   DS      1               DISK WRITE PROTECTED
000.026                272X EC.FAP  DS      1               FILE ALREADY PRESENT
000.027                273X EC.DDA  DS      1               DEVICE DRIVER ABORT
000.030                274X EC.FL   DS      1               FILE LOCKED
```

```
000.031                 275X EC.FAO  DS      1               FILE ALREADY OPEN
000.032                 276X EC.IS   DS      1               ILLEGAL SWITCH
000.033                 277X EC.UUN  DS      1               UNKNOWN UNIT NUMBER
000.034                 278X EC.FNR  DS      1               FILE NAME REQUIRED
000.035                 279X EC.DIW  DS      1               DEVICE IS NOT WRITABLE (OR WRITE LOCKED)
000.036                 280X EC.UNA  DS      1               UNIT NOT AVAILABLE
000.037                 281X EC.ILV  DS      1               ILLEGAL VALUE
000.040                 282X EC.ILO  DS      1               ILLEGAL OPTION
000.041                 283X EC.VPM  DS      1               VOLUME PRESENTLY MOUNTED ON DEVICE
000.042                 284X EC.NVM  DS      1               NO VOLUME PRESENTLY MOUNTED
000.043                 285X EC.FOD  DS      1               FILE OPEN ON DEVICE
000.044                 286X EC.NPM  DS      1               NO PROVISIONS MADE FOR REMOUNTING MORE DISKS
000.045                 287X EC.DNI  DS      1               DISK NOT INITIALIZED
000.046                 288X EC.DNR  DS      1               DISK IS NOT READABLE
000.047                 289X EC.DSC  DS      1               DISK STRUCTURE IS CORRUPT
000.050                 290X EC.NCV  DS      1               NOT CORRECT VERSION OF HDOS
000.051                 291X EC.NOS  DS      1               NO OPERATING SYSTEM MOUNTED
000.052                 292X EC.IOI  DS      1               ILLEGAL OVERLAY INDEX
000.053                 293X EC.OTL  DS      1               OVERLAY TO LARGE
000.054                 294          XTEXT   FILDEF

                        296X **      FILDEF - FILE TYPE DEFINITIONS.
                        297X *
                        298X *       DB      377Q,FT.XXX
                        299X
                        300X
000.000                 301X FT.ABS  EQU     0               ABSOLUTE BINARY
000.001                 302X FT.PIC  EQU     1               POSITION INDEPENDANT CODE
000.002                 303X FT.REL  EQU     2               RELOCATABLE CODE
000.003                 304X FT.BAC  EQU     3               COMPILED BASIC CODE
000.054                 305          XTEXT   DIRDEF

                        307X **      DIRECTORY ENTRY FORMAT.
                        308X
000.000                 309X         ORG     0
                        310X
                        311X
000.377                 312X DF.EMP  EQU     377Q            FLAGS ENTRY EMPTY
000.376                 313X DF.CLR  EQU     376Q            FLAGS ENTRY EMPTY, REST OF DIR ALSO CLEAR
                        314X
000.000                 315X DIR.NAM DS      8               NAME
000.010                 316X DIR.EXT DS      3               EXTENSION
000.013                 317X DIR.PRO DS      1               PROJECT
000.014                 318X DIR.VER DS      1               VERSION
000.015                 319X DIRIDL  EQU     *               FILE IDENTIFICATION LENGTH
                        320X
000.015                 321X DIR.CLU DS      1               CLUSTER FACTOR
000.016                 322X DIR.FLG DS      1               FLAGS
000.017                 323X         DS      1               RESERVED
000.020                 324X DIR.FGN DS      1               FIRST GROUP NUMBER
000.021                 325X DIR.LGN DS      1               LAST GROUP NUMBER
```

```
000.022                 326X DIR.LSI DS      1          LAST SECTOR INDEX (IN LAST GROUP)
000.023                 327X DIR.CRD DS      2          CREATION DATE
000.025                 328X DIR.ALD DS      2          LAST ALTERATION DATE
                        329X
000.027                 330X DIRELEN EQU     *          DIRECTORY ENTRY LENGTH
000.027                 331          XTEXT   OVLDEF


                        333X **      OVERLAY TABLE ENTRYS.
                        334X
000.000                 335X         ORG     0
                        336X
000.000                 337X OVL.COD DS      2          FIRST SECTOR OF OVERLAY CODE
000.002                 338X OVL.SIZ DS      2          OVERLAY SIZE
000.004                 339X OVL.ENT DS      2          OVERLAY ENTRY POINT
000.006                 340X OVL.FLB DS      1          OVERLAY FLAG BYTE
000.007                 341X         DS      1          DUMMY BYTE TO ROUND TABLE SIZE UP TO 8
000.010                 342X OVL.ENS EQU     *          OVERLAY ENTRY SIZE
                        343X
                        344X *       OVERLAY INDICES
                        345X
000.000                 346X         ORG     0
                        347X
000.000                 348X OVL0    DS      1
000.001                 349X OVL1    DS      1
000.002                 350          XTEXT   IOCDEF


                        352X **      I/O CHANNEL DEFINITIONS.
                        353X
000.000                 354X         ORG     0
                        355X
000.000                 356X IOC.LNK DS      2          ADDRESS OF NEXT CHANNEL, =0 IF LAST
000.002                 357X IOC.DDA DS      2          THREAD JUMP TO DEVICE DRIVER (VIA DEV TABLE)
                        358X
000.004                 359X IOC.FLG DS      1          FILE TYPE FLAGS
000.001                 360X FT.DD   EQU     00000001B  =1 IF DIRECTORY DEVICE
000.002                 361X FT.OR   EQU     00000010B  =1 IF OPEN FOR READ
000.004                 362X FT.OW   EQU     00000100B  =1 IF OPEN FOR WRITE
000.010                 363X FT.OU   EQU     00001000B  =1 IF OPEN FOR UPDATE
000.020                 364X FT.OC   EQU     00010000B  =1 IF OPEN FOR CHARACTER MODE ;/80.02,GC/
000.003                 365X IOC.SQL EQU     *-IOC.DDA  LENGTH OF INFO FOR SEQUENTIAL FILE (FROM IOC)
                        366X
000.005                 367X IOC.GRT DS      2          ADDRESS OF GROUP RESERVATION TABLE
000.007                 368X IOC.SPG DS      1          SECTORS PER GROUP, THIS DEVICE
000.010                 369X IOC.CGN DS      1          CURRENT GROUP NUMBER
000.011                 370X IOC.CSI DS      1          CURRENT SECTOR INDEX (IN CURRENT GROUP)
000.012                 371X IOC.LGN DS      1          LAST GROUP NUMBER
000.013                 372X IOC.LSI DS      1          LAST SECTOR INDEX (IN LAST GROUP)
000.010                 373X IOC.DRL EQU     *-IOC.FLG  LENGTH OF INFO NORMALLY COPIED BACK TO
                        374X *                          THE CHANNEL TABLE
000.014                 375X IOC.DTA DS      2          DEVICE TABLE ADDRESS FOR THIS DEVICE
000.016                 376X IOC.DES DS      2          SECTOR NUMBER OF DIRECTORY ENTRY
```

```
000.020                 377X IOC.DEV DS      2               DEVICE CODE
000.022                 378X IOC.UNI DS      1               UNIT NUMBER (0-9)
000.021                 379X IOC.DIL EQU     *-IOC.DDA       LENGTH OF INFO FOR DIRECTORY FILE (FROM IOC)
                        380X
000.023                 381X IOC.DIR DS      DIRELEN         DIRECTORY ENTRY
                        382X
000.052                 383X IOCELEN EQU     *               IOC ENTRY LENGTH
                        384X
000.001                 385X IOCCTD  EQU     1               INDEX OF USER CHANNEL #0 IN CHANTAB (FIRST = 0)


042.170                 387          ORG     USERFWA-ABS.COD
                        388
042.170  377 000       389          DB      377Q,FT.ABS     ABS HEADER
042.172  200 042       390          DW      USERFWA         ORG
042.174  246 017       391          DW      MEML-USERFWA    SIZE OF LOAD IMAGE
042.176  346 061       392          DW      ENTRY           ENTRY POINT
                        393
```

```
                                396
042,200                         397 START   EQU     *
042.200                         398 RESTART EQU     *               RESTART ADDRESS
                                399
                                400
                                401 *       ENTER HERE FOR RUBBOUT AND COMMANDS DONE.
                                402
042,200  076 201                403 EDIX    MVI     A,CSL,CHR+CSL,ECH       CHARACTER MODE, NO ECHO
042.202  062 326 040            404         STA     S.CSLMD         CLEAR TERMINAL CONTRL
042,205  373                    405         EI
042.206  315 153 052            406         CALL    CBE             CHECK FOR BUFFER EMPTY
042,211  315 333 053            407         CALL    MAM             SET MAXIMUM MEMORY
042.214  315 000 055            408         CALL    $CCO            CLEAR CTL-O
042,217  315 171 055            409         CALL    $GNL            GUARANTEE NEW LINE
042.222  315 136 031            410         CALL    $TYPTX
042,225  055 255                411         DB      '-','-'+200Q
042.227  257                    412         XRA     A
042,230  062 347 061            413         STA     LINE            NULL LINE
042.233  062 206 061            414         STA     CCFLG           CLEAR CTL-C DISABLE FLAG
042,236  062 207 061            415         STA     CCPEND          CLEAR PENDING CTL-C
042.241  315 044 053            416         CALL    ECC             ENABLE CTL-C
042,244  257                    417         XRA     A
042.245  062 214 061            418 EDTO    STA     PROCHA          CLEAR PROBATATION CHARACTER
                                419
                                420 *       RE-ENTER HERE FOR BACKSPACE AND ILLEGAL CHARACTERS
                                421
042.250  041 347 061            422 EDI1    LXI     H,LINE
042.253  042 212 061            423         SHLD    LINPTR
042.256  257                    424         XRA     A
042.257  062 156 053            425         STA     ENCA            CLEAR HELD CHARACTER
042.262  061 200 042            426         LXI     SP,STACK        RESTORE STACK
                                427
                                428 *       DECODE COMMAND
                                429
042.265  315 066 043            430         CALL    DCR             DECODE COMMAND RANGE
042.270  315 072 044            431         CALL    DCN             DECODE COMMAND NAME
042.273  315 310 044            432         CALL    DCQ             DECODE COMMAND QUALIFIER
042.276  315 326 044            433         CALL    DCO             DECODE COMMAND OPTION
042.301  052 174 061            434         LHLD    CRFPTR
042.304  042 200 061            435         SHLD    WRKPTR
                                436
                                437 *       PROCESS COMMAND
                                438
042.307  072 124 063            439         LDA     CMDGRP          SEE WHICH GROUP IS COMMAND
042.312  247                    440         ANA     A
042.313  072 123 063            441         LDA     PATCNT
042.316  302 323 042            442         JNZ     EDI1.5          IS IN FULL RANGE
042.321  306 006                443         ADI     CMDDSP          IS IN NO-DATA GROUP
042.323  041 200 042            444 EDI1.5  LXI     H,EDIX
042.326  345                    445         PUSH    H               SET 'RETURN ADDRESS'
042.327  315 061 031            446         CALL    $TJMP           JUMP TO PROCESSOR
                                447
                                448 *       THE FOLLOWING COMMANDS MAY BE USED ONLY IF DATA PRESENT.
                                449
042.332                         450 CMDADR  DS      0               START OF TABLE
042.332  111 045                451         DW      PRINT           PRINT
```

```
042.334  206 045        452          DW      DELETE          DELETE
042.336  116 046        453          DW      EDITC           EDIT
042.340  022 046        454          DW      REPLAC          REPLACE
042.342  311 051        455          DW      WRITE           WRITE
042.344  132 045        456          DW      XPRINT          XPRINT                                  /80.02.GC/
                        457
                        458  *       THE FOLLOWING COMMANDS MAY ALWAYS BE USED
                        459
000.006                 460  CMDDSP  EQU     *-CMDADR/2
042.346  371 044        461          DW      INSERT          INSERT
042.350  203 050        462          DW      READ            READ
042.352  052 046        463          DW      PURGE           PURGE
042.354  377 046        464          DW      FLUSH           FLUSH
042.356  146 050        465          DW      NEXT            NEXT
042.360  337 050        466          DW      SEARCH          SEARCH
042.362  041 047        467          DW      NEWIN           NEWIN
042.364  235 047        468          DW      NEWOUT          NEWOUT
042.366  001 050        469          DW      XOUT            XOUT                                    /80.02.GC/
042.370  112 051        470          DW      USE             USE
042.372  025 047        471          DW      BYE             BYE



                        473  **      CTL-C INTERRUPT RECEIVED.
                        474  *
                        475
042.374  315 136 031    476  INTRPT  CALL    $TYPTX
042.377  136 303        477          DB      '^','C'+200Q
043.001  072 206 061    478          LDA     CCFLG
043.004  247            479          ANA     A
043.005  302 023 043    480          JNZ     INT1                                                    /78.10.GC/
043.010  377 007        481          DB      SYSCALL,.CLRCO                                          /78.10.GC/
043.012  052 315 061    482          LHLD    XOUTFB+FB.FWA                                           /80.02.GC/
043.015  042 317 061    483          SHLD    XOUTFB+FB.PTR   ZERO THE  *XOUT*  BUFFER PTR.         /80.02.GC/
043.020  303 200 042    484          JMP     EDIX            CTL-C ALLOWED                           /78.10.GC/
                        485
043.023  076 001        486  INT1    MVI     A,1
043.025  062 207 061    487          STA     CCPEND          FLAG PENDING CTLC
043.030  311            488          RET                     DISCARD FOR NOW



                        490  **      REFUSE - REFUSE ENTERED CHARACTER.
                        491  *
                        492  *       REFUSE IS CALLED WHEN AN ILLEGAL ENTRY IS DETECTED.
                        493  *       IT TYPES A BELL, REMOVES THE LAST CHARACTER FROM THE INPUT
                        494  *       LINE, AND RE-PARSES THE COMMAND.
                        495
                        496
043.031  315 136 031    497  REFUSE  CALL    $TYPTX
043.034  207            498          DB      BELL+200Q
043.035  041 347 061    499          LXI     H,LINE
043.040  315 333 054    500          CALL    SNL             SCAN TO END
043.043  053            501          DCX     H               BACKSPACE TO LAST CHARACTER
```

```
043.044  053            502          DCX     H           HAVE ADVANCED PAST LAST CHARACTER
043.045  257            503          XRA     A
043.046  167            504          MOV     M,A
043.047  303 245 042     505          JMP     EDT0        CLEAR PROBATION (BAD) CHARACTER


                        507  **      EXIT - CTL-D STRUCK (END OF FILE ON CONSOLE)
                        508  *
                        509  *       SEE IF USER REALLY WANTS TO EXIT...
                        510
                        511
043.052  315 110 052    512  EXIT    CALL    AYS         ARE YOU SURE?
043.055  332 063 043    513          JC      EXIT1       CTL-D AGAIN
043.060  302 200 042    514          JNE     RESTART     NOT SURE
043.063  257            515  EXIT1   XRA     A
043.064  377 000        516          DB      SYSCALL,.EXIT   EXIT WITH EVERYTHING OPEN
```

```
                                    520  **      DCR - DECODE COMMAND RANGE.
                                    521  *
                                    522  *       DCR IS CALLED TO DETERMINE THE COMMAND RANGE.
                                    523  *
                                    524  *       CAN BE EITHER
                                    525  *
                                    526  *       =                       PREVIOUS RANGE
                                    527  *       ' '                     ALL TEXT
                                    528  *       EXPR                    LINE EXPRESSION
                                    529  *
                                    530  *       ENTRY   NONE
                                    531  *       EXIT    CRFPTR,CRLPTR,WRKPTR SETUP
                                    532  *       USES    ALL
                                    533
                                    534  ;-----------------------------------------------------------------------
        043.066                     535  DCR     EQU     *
        043.066  052 202 061        536          LHLD    PCFPTR
        043.071  042 174 061        537          SHLD    CRFPTR
        043.074  052 204 061        538          LHLD    PCLPTR
        043.077  042 176 061        539          SHLD    CRLPTR          SET DEFAULT RANGE TO RANGE OF PREVIOUS
        043.102  174               540          MOV     A,H
        043.103  265               541          ORA     L
        043.104  310               542          RZ                      IF NO DATA, DONT ALLOW RANGE
        043.105  052 170 061        543          LHLD    FILPTR
        043.110  315 064 053        544          CALL    ENC             EXAMINE NEXT CHARACTER
        043.113  376 040            545          CPI     ' '
        043.115  302 141 043        546          JNE     DCR1            NOT BLANK
                                    547
                                    548  *       IS BLANK, ENTIRE RANGE.
                                    549
        043.120  042 174 061        550          SHLD    CRFPTR
        043.123  052 172 061        551          LHLD    LALPTR
        043.126  174               552          MOV     A,H
        043.127  265               553          ORA     L
        043.130  304 322 054        554          CNZ     SLB             SCAN LINE BACKWARDS (IF ANY TEXT)
        043.133  042 176 061        555          SHLD    CRLPTR
        043.136  303 205 053        556          JMP     GNC             READ BLANK AND EXIT
                                    557
        043.141  376 075            558  DCR1    CPI     '='
        043.143  312 205 053        559          JE      GNC             IS OLD RANGE, READ = AND EXIT
        043.146  174               560          MOV     A,H
        043.147  265               561          ORA     L
        043.150  310               562          RZ                      NO TEXT, DONT ALLOW EXPRESSION
                                    563
                                    564  *       MUST BE EXPRESSION
                                    565
        043.151  315 235 043        566          CALL    DRE             DECODE RANGE EXPRESSION
        043.154  042 174 061        567          SHLD    CRFPTR          SET FIRST COMMAND
        043.157  042 176 061        568          SHLD    CRLPTR          ASSUME IS ONE LINE COMMAND
        043.162  315 064 053        569          CALL    ENC
        043.165  376 054            570          CPI     ','
        043.167  300               571          RNE                     NO 2ND EXPRESSION
        043.170  315 205 053        572          CALL    GNC             READ ,
        043.173  345               573          PUSH    H               SAVE BEGINNING OF RANGE
        043.174  315 235 043        574          CALL    DRE             DECODE RANGE EXPRESSION
        043.177  042 176 061        575          SHLD    CRLPTR          SET LAST
```

```
043.202  321              576           POP     D               (DE) = FIRST
                           577
                           578  *        MAKE SURE 1ST IS LESS THAN OR EQUAL TO LAST
                           579
043.203  175              580           MOV     A,L
043.204  223              581           SUB     E
043.205  174              582           MOV     A,H
043.206  232              583           SBB     D
043.207  320              584           RNC                     IS OK
043.210  315 136 031      585           CALL    $TYPTX
043.213  012 007 106      586           DB      NL,BELL,'First <= Las','t'+200Q
043.232  303 031 043      587           JMP     REFUSE
```

```
                                591  **      DRE - DECODE RANGE EXPRESSION.
                                592  *
                                593  *       DRE DECODES A COMMAND RANGE EXPRESSION.
                                594  *
                                595  *       TOKENS VALID AS 1ST TOKEN, ONLY
                                596  *
                                597  *       NULL    CURRENT 1ST LINE
                                598  *       $       LAST LINE IN BUFFER
                                599  *       ^       1ST LINE IN BUFFER
                                600  *
                                601  *       TOKENS VALID ANYWHERE
                                602  *
                                603  *       'STR' LINE CONTAINING STRING
                                604  *
                                605  *       TOKENS NOT VALID AT HEAD OF STRING
                                606  *
                                607  *       NNN     LINE COUNT
                                608  *
                                609  *       OPERATORS
                                610  *
                                611  *       +       SCAN FORWARD
                                612  *       -       SCAN BACKWARDS
                                613  *
                                614  *       ENTRY   NONE
                                615  *       EXIT    (HL) = RESULTANT LINE POINTER
                                616  *       USES    ALL
                                617
                                618
043.235                         619  DRE     EQU     *
043.235  076 377               620          MVI     A,-1
043.237  062 215 061           621          STA     SRCDIR          SET INITIAL DIRECTION FORWARD
                                622
                                623  *       DECODE INITIAL TOKEN.
                                624
043.242  315 064 053           625          CALL    ENC             PEEK AT CHARACTER
043.245  052 170 061           626          LHLD    FILPTR
043.250  376 136               627          CPI     '^'
043.252  312 302 043           628          JE      DRE1            START AT TOP
043.255  052 172 061           629          LHLD    LALPTR          ASSUME LAST
043.260  365                   630          PUSH    PSW             SAVE (A)
043.261  315 322 054           631          CALL    SLB             SCAN LINE BACKWARDS
043.264  361                   632          POP     PSW
043.265  376 044               633          CPI     '$'
043.267  312 302 043           634          JE      DRE1            NOT TO START AT BOTTOM
043.272  052 174 061           635          LHLD    CRFPTR
043.275  376 047               636          CPI     QUOTE
043.277  312 372 043           637          JE      DRE7            IS QUOTED STRING
043.302  314 205 053           638  DRE1    CZ      GNC             ACCEPT CHARACTER OF $ OR ARROW
                                639
043.305  042 200 061           640  DRE3    SHLD    WRKPTR          SET CURRENT LINE ADDRESS
                                641
                                642  *       DECODE OPERATOR
                                643
043.310  315 064 053           644  DRE4    CALL    ENC             EXAMINE NEXT CHARACTER
043.313  326 053               645          SUI     '+'
043.315  312 331 043           646          JZ      DRE5            IS FORWARD SEARCH
```

```
043,320  376 002        647          CPI     '-'-'+'
043,322  312 331 043    648          JE      DRE5            IS BACKWARD SEARCH
043,325  052 200 061    649          LHLD    WRKPTR          (HL) = LINE RANGE
043,330  311            650          RET                     EXIT WITH LINE POINTER
                        651
043,331  075            652  DRE5    DCR     A
043,332  062 215 061    653          STA     SRCDIR
043,335  315 205 053    654          CALL    GNC             READ + OR -
                        655
                        656  **      DECODE NEXT TOKEN.
                        657
043,340  315 064 053    658          CALL    ENC             EXAMINE CHARACTER
043,343  376 047        659          CPI     QUOTE
043,345  312 375 043    660          JE      DRE8            QUOTED STRING
                        661
                        662  *       HAVE NNN - STEP OVER LINES
                        663
043,350  315 265 052    664          CALL    DDN             MUST BE DECIMAL NUMBER
043,353  170            665  DRE6    MOV     A,B
043,354  261            666          ORA     C
043,355  312 305 043    667          JZ      DRE3            HAVE STEPPED ENOUGH LINES
043,360  013            668          DCX     B
043,361  315 020 044    669          CALL    MLP             MOVE LINE POINTER
043,364  042 200 061    670          SHLD    WRKPTR
043,367  303 353 043    671          JMP     DRE6
                        672
                        673  *       HAVE STRING VALUE.
                        674
043,372  042 200 061    675  DRE7    SHLD    WRKPTR
043,375  041 051 063    676  DRE8    LXI     H,QUALS         USE QUALS AREA FOR SCRATCH
044,000  314 073 054    677          CZ      RQS             READ QUOTED STRING
044,003  315 322 053    678          CALL    LQS             LOCATE QUOTED STRING
044,006  312 310 043    679          JE      DRE4            FOUND
044,011  315 020 044    680          CALL    MLP             MOVE LINE POINTER
044,014  264            681          ORA     H
044,015  303 372 043    682          JMP     DRE7            SEARCH AGAIN


                        684  **      MLP - MOVE LINE POINTER.
                        685  *
                        686  *       MLP MOVES THE LINE POINTER FORWARDS OR BACKWARDS ONE LINE,
                        687  *       DEPENDING UPON 'SRCDIR'.
                        688  *
                        689  *       IF SRCDIR <  0, FORWARDS
                        690  *       IF SRCDIR => 0, BACKWARDS
                        691  *
                        692  *       IF RUN OFF THE NED OF TEXT, EXIT TO 'REFUSE'
                        693  *
                        694  *       ENTRY   (HL) = LINE POINTER
                        695  *       EXIT    (HL) = NEW LINE POINTER
                        696  *       USES    A,F
                        697
                        698
044,020  325            699  MLP     PUSH    D
```

```
044.021  052 200 061   700          LHLD    WRKPTR
044.024  072 215 061   701          LDA     SRCDIR
044.027  247           702          ANA     A
044.030  362 053 044   703          JP      MLP1            BACKWARDS
044.033  315 333 054   704          CALL    SNL             SCAN TO NEXT LINE
044.036  353           705          XCHG
044.037  052 172 061   706          LHLD    LALPTR
044.042  353           707          XCHG
044.043  315 216 030   708          CALL    $CDEHL          COMPARE TO BOTTOM
044.046  321           709          POP     D
044.047  312 031 043   710          JE      REFUSE          IF ALREADY AT BOTTOM
044.052  311           711          RET
                       712
                       713  *       BACKWARDS
                       714
044.053  353           715  MLP1    XCHG
044.054  052 170 061   716          LHLD    FILPTR
044.057  353           717          XCHG
044.060  315 216 030   718          CALL    $CDEHL          SEE IF AT TOP
044.063  312 031 043   719          JE      REFUSE
044.066  321           720          POP     D
044.067  303 322 054   721          JMP     SLB             SCAN LINE BACKWARDS AND RETURN
```

```
                                    724  **      DCN - DECODE COMMAND NAME.
                                    725  *
                                    726  *       DCN DECODES AND COMPLETES THE COMMAND NAME.
                                    727  *
                                    728  *       ENTRY   NONE
                                    729  *       EXIT    (A) = COMMAND INDEX
                                    730
                                    731
044.072                             732  DCN     EQU     *
044.072  315 064 053               733          CALL    ENC             PRE-READ 1ST COMMAND CHARACTER
044.075  052 212 061               734          LHLD    LINPTR
044.100  053                       735          DCX     H
044.101  042 157 044               736          SHLD    DCNA            SET LINE POINTER
044.104  257                       737          XRA     A
044.105  062 156 053               738          STA     ENCA
044.110  303 116 044               739          JMP     CMD3
                                    740
                                    741  *       INPUT 1 CHARACTER
                                    742
044.113  315 205 053               743  CMD2    CALL    GNC             GET NEXT CHARACTER
                                    744
                                    745  *       CLEAR NXTCHA, PATCNT
                                    746
044.116  041 000 377               747  CMD3    LXI     H,377000A
044.121  042 122 063               748          SHLD    NXTCHA
                                    749
044.124  021 321 060               750          LXI     D,CMDTAB
044.127  052 174 061               751          LHLD    CREPTR
044.132  174                       752          MOV     A,H
044.133  265                       753          ORA     L               SEE IF ANY DATA
044.134  062 124 063               754          STA     CMDGRP          SET COMMAND GROUP
044.137  302 145 044               755          JNZ     CMD4            HAVE DATA
044.142  021 362 060               756          LXI     D,CMDTAB.       RESTRICT COMMAND RANGE
                                    757
                                    758  *       CHECK AGAINST NEXT COMMAND DESCRIPTION.
                                    759
044.145  041 123 063               760  CMD4    LXI     H,PATCNT
044.150  064                       761          INR     M
044.151  353                       762          XCHG
044.152  315 333 054               763          CALL    SNL             SCAN FOR NEW LINE
044.155  353                       764          XCHG
044.156  001 000 000               765          LXI     B,0             (BC) = COMMAND TEXT ADDRESS
044.157                             766  DCNA    EQU     *-2
044.161  032                       767          LDAX    D
044.162  247                       768          ANA     A
044.163  302 212 044               769          JNZ     CMD5            HAVE COMMAND ELEMENT
                                    770
                                    771  *       NO MORE COMMANDS, HAVE:
                                    772  *
                                    773  *       1) NO MATCHES, OR
                                    774  *       2) A UNIQUE NEXT CHARACTER
                                    775
044.166  072 122 063               776          LDA     NXTCHA
044.171  247                       777          ANA     A
044.172  312 031 043               778          JZ      REFUSE          NO MATCHES - ILLEGAL
044.175  052 212 061               779          LHLD    LINPTR
```

```
044.200  167            780          MOV     M,A
044.201  043            781          INX     H
044.202  066 000        782          MVI     M,0
044.204  062 214 061    783          STA     PROCHA
044.207  303 113 044    784          JMP     CMD2
                        785
                        786  *       CHECK NEXT TABLE ELEMENT FOR MATCH
                        787
044.212  012            788  CMD5    LDAX    B               (A) = NEXT LINE CHARACTER
044.213  247            789          ANA     A
044.214  302 265 044    790          JNZ     CMD7            IF SOME
                        791
                        792  *       NO MORE TEXT, SEE IF CAN ANTICIPATE NEXT CHARACTER
                        793
044.217  072 214 061    794          LDA     PROCHA
044.222  247            795          ANA     A
044.223  304 345 055    796          CNZ     $WCHAR
044.226  257            797  CMD6    XRA     A
044.227  062 214 061    798          STA     PROCHA          CLEAR PROBATION CHARACTER
044.232  140            799  CMD6.5  MOV     H,B
044.233  151            800          MOV     L,C             (HL) = NEW LINE POINTER
044.234  042 212 061    801          SHLD    LINPTR          SKIP OVER CHARACTERS ACCEPTED
044.237  032            802          LDAX    D               (A) = COMMAND ELEMENT
044.240  247            803          ANA     A
044.241  310            804          RZ                      EXIT IF ENTIRE COMMAND MATCHED
044.242  041 122 063    805          LXI     H,NXTCHA
                        806
                        807  *       SEE IF THIS IS THE FIRST COMPLETION CHARACTER,
                        808  *       OR IF IT IS THE SAME CHARACTER AS PREVIOUSLY FOUND
                        809
044.245  276            810          CMP     M
044.246  312 145 044    811          JE      CMD4            SAME AS PREVIOUS, CAN COMPLETE
044.251  325            812          PUSH    D
044.252  127            813          MOV     D,A
044.253  206            814          ADD     M
044.254  167            815          MOV     M,A
044.255  272            816          CMP     D               SEE IF NXTCHA WAS 0
044.256  321            817          POP     D
044.257  312 145 044    818          JE      CMD4            CAN COMPLETE
044.262  303 113 044    819          JMP     CMD2            CANNOT COMPLETE
                        820
                        821  *       HAVE PATTERN AND TEXT, SEE IF MATCH,
                        822
044.265  032            823  CMD7    LDAX    D
044.266  247            824          ANA     A
044.267  312 232 044    825          JZ      CMD6,5          TOTAL MATCH - PRETEND RAN OUT OF TEXT
044.272  147            826          MOV     H,A             (H) = NEXT REQUIRED CHARACTER
044.273  012            827          LDAX    B               (A) = NEXT TEXT ELEMENT
044.274  315 205 055    828          CALL    $MCU            MAP CHARACTER TO UPPER CASE
044.277  003            829          INX     B               ASSUME MATCH
044.300  274            830          CMP     H
044.301  302 145 044    831          JNE     CMD4            NO MATCH
044.304  023            832          INX     D
044.305  303 212 044    833          JMP     CMD5
```

```
                                    837  **      DCQ - DECODE COMMAND QUALIFIER.
                                    838  *
                                    839  *       DCQ READS AN OPTIONAL QUALIFICATION STRING FOLLOWING A
                                    840  *       COMMAND
                                    841  *
                                    842  *       COMMAND'STRING'
                                    843  *
                                    844  *       ENTRY   NONE
                                    845  *       EXIT    QUALS = STRING (NULL IF NONE)
                                    846
                                    847
044.310  041 051 063               848  DCQ     LXI     H,QUALS
044.313  066 000                   849          MVI     M,0             NULL IT
044.315  315 064 053               850          CALL    ENC             CHECK NEXT CHARACTER
044.320  376 047                   851          CPI     QUOTE
044.322  300                       852          RNE                     NO QUALIFIER
044.323  303 073 054               853          JMP     RQS             READ QUOTED STRING AND RETURN
```

```
                                857  **      DCO - DECODE COMMAND OPTIONS.
                                858  *
                                859  *       DCO DECODES THE COMMAND OPTION SPECIFICATION.
                                860  *
                                861  *       COMMANDOPTION
                                862  *
                                863  *       WHERE OPT = 'A' = PRINT LINE AFTER
                                864  *                         B - PRINT LINE BEFORE
                                865  *                         N - PRINT LINE NUMBERS
                                866
                                867
044.326  041 216 061           868  DCO     LXI     H,OPTS
044.331  066 000               869          MVI     M,0             CLEAR OPTIONS
044.333  315 064 053           870  DCO1    CALL    ENC             CHECK NEXT CHARACTER
044.336  315 205 055           871          CALL    $MCU            MAP CHARACTER TO UPPER CASE
044.341  376 101               872          CPI     'A'
044.343  312 351 044           873          JE      DCO2            IF 'A'
044.346  376 102               874          CPI     'B'
044.350  300                   875          RNE                     NOT OPTION
044.351  346 003               876  DCO2    ANI     OPT.A+OPT.B
044.353  107                   877          MOV     B,A             (B) = OPTION
044.354  246                   878          ANA     M
044.355  302 031 043           879          JNZ     REFUSE          ALREADY SET
044.360  170                   880          MOV     A,B
044.361  266                   881          ORA     M               SET IN FLAGS
044.362  167                   882          MOV     M,A
044.363  315 205 053           883          CALL    GNC             ACCEPT 'A' OR 'B'
044.366  303 333 044           884          JMP     DCO1
```

```
                                888  **      INSERT - INSERT TEXT INTO BUFFER.
                                889  *
                                890  *       ISNERT RECOGNIZES TWO SPECIAL CASES:
                                891  *
                                892  *       1) IF NO TEXT EXISTS, INITIALIZE STRUCTURE
                                893  *       2) IF THE LINE NUMBER IS ' ', INSERT BEFORD THE 1ST LINE
                                894
                                895
        044.371                 896  INSERT  EQU     *
        044.371  315 041 054   897          CALL    RCR             REQUIRE CARRIAGE RETURN
        044.374  052 200 061   898          LHLD    WRKPTR
        044.377  174           899          MOV     A,H
        045.000  265           900          ORA     L
        045.001  302 035 045   901          JNZ     INS1            HAVE PRE-EXISTING TEXT
                                902
                                903  *       READ 1ST LINE INTO EMPTY STRUCTURE
                                904
        045.004  315 072 052   905          CALL    ATL             READ TEXT
        045.007  315 255 052   906          CALL    DCC             DISABLE CTL-C
        045.012  353           907          XCHG                    (DE) = TEXT ADDRESS
        045.013  041 147 070   908          LXI     H,BUFFER
        045.016  315 070 046   909          CALL    SAP             SET ALL POINTERS
        045.021  345           910          PUSH    H
        045.022  117           911          MOV     C,A
        045.023  006 000       912          MVI     B,0             (BC) = LEN
        045.025  011           913          DAD     B
        045.026  042 172 061   914          SHLD    LALPTR
        045.031  341           915          POP     H
        045.032  303 077 045   916          JMP     INS3
                                917
        045.035  315 030 054   918  INS1    CALL    PLB             PRINT LINE BEFORE
        045.040  072 347 061   919          LDA     LINE
        045.043  376 040       920          CPI     ' '
        045.045  304 333 054   921  INS2    CNZ     SNL             (HL) = ADDRESS TO INSERT TEXT
        045.050  315 044 053   922          CALL    ECC             RE-ENABLE CTL-C
        045.053  315 171 052   923          CALL    CBO             CHECK FOR BUFFER OVERFLOW
                                924
                                925  *       INSERT A NEW LINE
                                926
        045.056  042 200 061   927          SHLD    WRKPTR
        045.061  353           928          XCHG
        045.062  315 072 052   929          CALL    ATL             ACCEPT TEXT LINE
        045.065  315 255 052   930          CALL    DCC             DISABLE CTL-C
        045.070  353           931          XCHG
        045.071  117           932          MOV     C,A
        045.072  315 244 053   933          CALL    ITBK            INSERT TEXT BLOCK       /80,02,BC/
        045.075  006 000       934          MVI     B,0
        045.077  315 007 056   935  INS3    CALL    $MOVL           MOVE TEXT IN            /WCZ080480/
        045.102  052 200 061   936          LHLD    WRKPTR
        045.105  264           937          ORA     H               CLEAR 'Z'
        045.106  303 045 045   938          JMP     INS2
```

```
                                    942  **      PRINT - PRINT TEXT LINES.
                                    943  *
                                    944
045.111  315 041 054               945  PRINT   CALL    RCR             REQUIRE CARRIAGE RETURN
045.114  315 231 054               946  PRI1    CALL    SEL             SCAN FOR ELIGIBLE LINE
045.117  310                       947          RZ                      IF NO MORE
045.120  315 345 054               948          CALL    TTX             TYPE SOURCE TEXT
045.121                            949  PRIA    EQU     *-2             PROCESSOR ADDRESS
045.123  315 045 052               950          CALL    ACL             ADVANCE COMMAND LINE
045.126  302 114 045               951          JNZ     PRI1
045.131  311                       952          RET                     DONE
```

```
                                956  **      XPRINT  -  PROCESS XPRINT COMMAND                    /80.02.GC/
                                957  *
                                958  *       XPRINT processes the XPRINT command which outputs
                                959  *       text to a specified alternate file.  The most
                                960  *       useful application of which, beins a listing to
                                961  *       an alternate printer.
                                962  *
                                963
045.132                         964  XPRINT  EQU     *
045.132  315 041 054           965          CALL    RCR
                                966
045.135  072 314 061           967          LDA     XOUTFB+FB.FLG
045.140  346 004               968          ANI     FT.OW
045.142  312 017 052           969          JZ      WRI4            REQUIRE AN OUTPUT FILE
                                970
045.145  315 231 054           971  XPR1    CALL    SEL
045.150  312 164 045           972          JZ      XPR2            NO MORE LINES
                                973
                                974  *       OUTPUT THE SPECIFIED LINE TO THE XPRINT DEVICE
                                975
045.153  315 173 045           976          CALL    XPR4            OUTPUT A LINE
                                977
045.156  315 045 052           978          CALL    ACL             ADVANCE ONE LINE
045.161  302 145 045           979          JNZ     XPR1
                                980
                                981  *       FLUSH THE OUPUT TO THE SPECIFIED DEVICE
                                982
045.164                         983  XPR2    EQU     *
                                984
045.164  041 313 061           985          LXI     H,XOUTFB        USE XOUT FILE BUFFER
045.167  315 356 057           986          CALL    $FWBRK          BREAKOUTPUT
                                987
045.172  311                   988          RET


                                990  **      OUTPUT A LINE
                                991
045.173                         992  XPR4    EQU     *
045.173  345                   993          PUSH    H
045.174  353                   994          XCHG                    DE = ADDRESS OF LINE
045.175  041 313 061           995          LXI     H,XOUTFB        HL = FILE BUFFER
045.200  315 127 057           996          CALL    $FWRIL          WRITE LINE
045.203  341                   997          POP     H               RESTORE LINE ADDRESS
045.204  311                   998          RET
                                999
045.205  000                  1000  XPRA    DB      0               FLUSH CHARACTER
000.001                      1001  XPRAL   EQU     *-XPRA          LENGTH ( SHOULD BE ONE TO LEAVE BUFFER EMPTY )
```

```
                          1005  **      DELETE - DELETE LINE RANGE.
                          1006
                          1007
045.206  072 347 061     1008  DELETE  LDA     LINE
045.211  376 040         1009          CPI     ' '
045.213  312 031 043     1010          JE      REFUSE          <BLANK>DELETE ILLEGAL
045.216  315 041 054     1011          CALL    RCR             REQUIRE CARRIAGE RETURN
                          1012
                          1013  *       ENTERED FROM *WRITE* HERE
                          1014
045.221  072 051 063     1015  DEL0    LDA     QUALS
045.224  247             1016          ANA     A
045.225  312 331 045     1017          JZ      DEL3            AM TO DELETE A BLOCK OF TEXT
045.230  315 044 053     1018  DEL1    CALL    ECC             ENABLE CTL-C
045.233  315 231 054     1019          CALL    SEL             SCAN FOR ELIGIBLE LINE          /10.04.77/
045.236  312 277 045     1020          JZ      DEL2            DONE                            /10.04.77/
045.241  345             1021          PUSH    H               SAVE ADDRESS                    /10.04.77/
045.242  052 200 061     1022          LHLD    WRKPTR
045.245  353             1023          XCHG
045.246  052 176 061     1024          LHLD    CRLPTR          SEE IF AT LAST TEXT LINE
045.251  173             1025          MOV     A,E
045.252  225             1026          SUB     L
045.253  172             1027          MOV     A,D
045.254  234             1028          SBB     H
045.255  341             1029          POP     H               (HL) = TEXT POINTER             /10.04.77/
045.256  365             1030          PUSH    PSW             SAVE RESULT FOR LATER TEST
045.257  315 030 054     1031          CALL    PLB             PRINT LINE BEFORE
045.262  315 255 052     1032          CALL    DCC             DISABLE CTL-C
045.265  315 361 054     1033          CALL    $CLL            COMPUTE LINE LENGTH
045.270  315 337 052     1034          CALL    DTBK            DELETE TEXT BLOCK               /80.02.GC/
045.273  361             1035          POP     PSW             RESTORE CONDITION AFTER TEST
045.274  332 230 045     1036          JC      DEL1            MORE TO GO
                          1037
                          1038  *       ALL DONE. CLEAR PREVIOUS COMMAND RANGE TO FORCE NEW RANGE
                          1039
045.277                   1040  DEL2    EQU     *                                       /80.02.GC/
045.277  052 172 061     1041          LHLD    LALPTR                                  /80.02.GC/
045.302  353             1042          XCHG                    DE = END OF LAST + 1    /80.02.GC/
045.303  052 174 061     1043          LHLD    CRFPTR          HL = CURRENT FIRST POINTER /80.02.GC/
045.306  315 216 055     1044          CALL    HLCPDE          COMPARE                 /80.02.GC/
045.311  332 322 045     1045          JC      DEL2.5          HL < DE                 /80.02.GC/
                          1046
045.314  052 172 061     1047          LHLD    LALPTR                                  /80.02.GC/
045.317  315 322 054     1048          CALL    SLB             SCAN BACK ONE LINE      /80.02.GC/
                          1049
045.322  042 202 061     1050  DEL2.5  SHLD    PCFPTR          SET PREVIOUS RANGE TO FIRST LINE
045.325  042 204 061     1051          SHLD    PCLPTR
045.330  311             1052          RET                     EXIT
                          1053
                          1054  *       NO QUALIFIER STRING, WILL THEREFORE DELETE AN ENTIRE BLOCK.
                          1055  *       LOCATE THAT BLOCK, AND DELETE ALL IN ONE SWOOP (RUNS A HECK OF A
                          1056  *       LOT FASTER!)
                          1057
045.331  315 255 052     1058  DEL3    CALL    DCC             DISABLE CTL-C
045.334  052 200 061     1059          LHLD    WRKPTR
045.337  042 020 046     1060          SHLD    DELA            SAVE FWA OF BLOCK
```

```
045.342  001 000 000  1061           LXI     B,0             (BC) = BYTES TO DELETE
                      1062
045.345  052 176 061  1063  DEL4     LHLD    CRLPTR          SEE IF THE LAST LINE IN THE RANGE
045.350  353          1064           XCHG
045.351  052 200 061  1065           LHLD    WRKPTR
045.354  175          1066           MOV     A,L
045.355  223          1067           SUB     E
045.356  174          1068           MOV     A,H
045.357  232          1069           SBB     D
045.360  365          1070           PUSH    PSW             SAVE RESULT
045.361  315 030 054  1071           CALL    PLB             PRINT LINE BEFORE
045.364  315 361 054  1072           CALL    $CLL            COMPUTE LINE LENTH
045.367  315 072 030  1073           CALL    $DADA           (HL) = LINE LWA+1
045.372  201          1074           ADD     C
045.373  117          1075           MOV     C,A
045.374  170          1076           MOV     A,B
045.375  316 000      1077           ACI     0
045.377  107          1078           MOV     B,A             ADD LENGTH TO (BC)
046.000  042 200 061  1079           SHLD    WRKPTR          ADVANCE POINTER
046.003  361          1080           POP     PSW             (PSW) = RESULTS OF WRKPTR-CRLPTR
046.004  332 345 045  1081           JC      DEL4            IF NOT ALL DONE
                      1082
                      1083  *        DELETE (BC) BYTES AT (DELA)
                      1084
046.007  052 020 046  1085           LHLD    DELA
046.012  315 350 052  1086           CALL    DTBK            DELETE A TEXT BLOCK          /80.02.GC/
046.015  303 277 045  1087           JMP     DEL2            FINISH UP
                      1088
046.020  000 000      1089  DELA     DW      0               FWA OF BLOCK TO DELETE
```

```
                                1093  **      REPLACE - PROCESS REPLACE COMMAND.
                                1094  *
                                1095
                                1096
046.022  315 041 054          1097  REPLAC  CALL    RCR             REQUIRE CARRIAGE RETURN
046.025  315 226 054          1098  REP1    CALL    SEL.            SCAN FOR ELIGIBLE LINE
046.030  310                  1099          RZ                      DONE
046.031  315 030 054          1100          CALL    PLB             PRINT LINE BEFORE
046.034  315 072 052          1101          CALL    ATL             ACCEPT TEXT LINE
046.037  117                  1102          MOV     C,A
046.040  315 144 054          1103          CALL    RSL             REPLACE SINGLE LINE
046.043  315 045 052          1104          CALL    ACL             ADVANCE COMMAND LINE
046.046  310                  1105          RZ
046.047  303 025 046          1106          JMP     REP1
```

```
                                1110  **      PURGE - PURGE TEXT BUFFER.
                                1111  *
                                1112  *       PURGE DELETES ALL TEXT, AND INITIALIZES THE DATA STRUCTURE.
                                1113  *
                                1114  *       THE NUMBER OF FREE BYTES REMAINING IS TYPED OUT.
                                1115
                                1116
046.052  315 041 054            1117  PURGE   CALL    RCR             REQUIRE CARRIAGE RETURN
046.055  315 110 052            1118          CALL    AYS             ARE YOU SURE
046.060  330                    1119          RC                      NOT SURE
046.061  300                    1120          RNE                     NOT SURE
                                1121
                                1122  **      PURGE. - PURGE WITHOUT WARNING.
                                1123  *
                                1124
046.062                         1125  PURGE.  EQU     *
046.062  041 000 000            1126          LXI     H,0
046.065  315 255 052            1127          CALL    DCC             DISABLE CTL-C


                                1129  **      SAP - SET ALL POINERS.
                                1130  *
                                1131  *       SAP SETS THE FOLLOWING POINTERS TO A SINGLE VALUE:
                                1132  *
                                1133  *       FILPTR  FIRST LINE POINTER
                                1134  *       LALPTR  LAST LINE POINTER
                                1135  *       CRFPTR  COMMAND FIRST LINE POINTER
                                1136  *       CRLPTR  COMMAND LAST LINE POINTER
                                1137  *       WRKPTR  WROKING POINTER
                                1138  *
                                1139  *       ENTRY   (HL) = VALUE
                                1140  *       EXIT    NONE
                                1141  *       USES    NONE
                                1142
                                1143
046.070  042 170 061            1144  SAP     SHLD    FILPTR
046.073  042 172 061            1145          SHLD    LALPTR
046.076  042 174 061            1146          SHLD    CRFPTR
046.101  042 176 061            1147          SHLD    CRLPTR
046.104  042 200 061            1148          SHLD    WRKPTR
046.107  042 202 061            1149          SHLD    PCFPTR
046.112  042 204 061            1150          SHLD    PCLPTR
046.115  311                    1151          RET
```

```
                                1155  **      EDITC - PROCESS EDIT COMMAND.
                                1156  *
                                1157  *       EDIT/FROM/TO/COUNT
                                1158
                                1159
046.116                         1160  EDITC   EQU     *
046.116  315 217 053            1161          CALL    GTC             GET DELIMITER
046.121  107                    1162          MOV     B,A             (B) = DELIMITER
                                1163
                                1164  *       READ /FROM/
                                1165
046.122  041 327 062            1166          LXI     H,EDIA
046.125  315 345 046            1167          CALL    RDS             READ DELIMITED STRING
046.130  171                    1168          MOV     A,C             (A) = LEN
046.131  247                    1169          ANA     A
046.132  312 031 043            1170          JZ      REFUSE          NULL IS ILLEGAL
                                1171
                                1172  *       READ /TO/ STRING
                                1173
046.135  041 000 063            1174          LXI     H,EDIB
046.140  121                    1175          MOV     D,C             (D) = LENGTH OF /FROM/
046.141  315 345 046            1176          CALL    RDS             READ DELIMITED STRING
046.144  102                    1177          MOV     B,D             (B) = LEN(FROM); (C) = LEN(TO)
046.145  305                    1178          PUSH    B               SAVE
046.146  001 000 000            1179          LXI     B,0
046.151  315 064 053            1180          CALL    ENC
046.154  376 052                1181          CPI     '*'
046.156  302 167 046            1182          JNE     EDI0            TO PROCESS ALL OF THEM
046.161  315 205 053            1183          CALL    GNC
046.164  303 175 046            1184          JMP     EDI2
                                1185
046.167  003                    1186  EDI0    INX     B               DEFAULT COUNT = 1
046.170  376 012                1187          CPI     NL
046.172  304 265 052            1188          CNE     DDN             DECODE IF DECIMAL
046.175  315 041 054            1189  EDI2    CALL    RCR             REQUIRE CARRIAGE RETURN
                                1190
                                1191  *       GET NEXT LINE
                                1192
046.200  315 226 054            1193  EDI3    CALL    SEL             SCAN FOR ELIGIBLE LIN
046.203  312 335 046            1194          JZ      EDI5            ALL DONE
046.206  052 200 061            1195          LHLD    WRKPTR
046.211  315 361 054            1196          CALL    $CLL            COMPUTE LINE LENGTH
046.214  305                    1197          PUSH    B               SAVE REPEAT COUNT
046.215  117                    1198          MOV     C,A
046.216  006 000                1199          MVI     B,0             (BC) = LINE LENGTH
046.220  353                    1200          XCHG                    (DE) = FROM
046.221  041 137 062            1201          LXI     H,WRKSTR
046.224  345                    1202          PUSH    H               SAVE DEST ADDRESS
046.225  315 007 056            1203          CALL    $MOVL           MOVE INTO WRKSTR                /WCZ080480/
046.230  341                    1204          POP     H               (HL) = #WRKSTR
046.231  301                    1205          POP     B               (BC) = REPEAT COUNT
046.232  021 327 062            1206          LXI     D,EDIA
046.235  315 264 054            1207          CALL    SFS             SEE IF SOURCE STRING IS PRESENT
046.240  302 335 046            1208          JNZ     EDI5            NOT FOUND
046.243  353                    1209          XCHG                    SAVE (HL) IN (DE)
046.244  315 030 054            1210          CALL    PLB             PRINT LINE BEFORE
```

```
046.247  353            1211            XCHG                    RESTORE (HL)
                        1212
                        1213  *         REPLACE STRING
                        1214
046.250  321            1215            POP     D               (D) = LEN(FROM), (E) = LEN(TO)
046.251  305            1216            PUSH    B               SAVE REPLACEMENT COUNTS
046.252  325            1217            PUSH    D               SAVE LENGTHS
046.253  345            1218            PUSH    H               SAVE ADDRESS OF MATCH
                        1219
                        1220  *         SOURCE LINE IS HEAD MATCH TAIL
                        1221  *
                        1222  *         MOVE TAIL TO ITS NEW POSITION TO MAKE ROOM FOR /TO/
                        1223
046.254  112            1224            MOV     C,D             (BC) = LEN(FROM)
046.255  006 000        1225            MVI     B,0
046.257  120            1226            MOV     D,B             (DE) = LEN(TO)
046.260  031            1227            DAD     D               (HL) = NEW TIAL ADDRESS
046.261  353            1228            XCHG
046.262  341            1229            POP     H
046.263  345            1230            PUSH    H
046.264  011            1231            DAD     B               (HL) = CURRENT TAIL ADDRESS
046.265  315 361 054     1232            CALL    $CLL            COMPUTE LINE LENGTH
046.270  006 000        1233            MVI     B,0
046.272  117            1234            MOV     C,A             (BC) = LENGTH OF TAIL
046.273  353            1235            XCHG
046.274  315 007 056     1236            CALL    $MOVL           MOVE TAIL                       /WCZ080480/
046.277  341            1237            POP     H               (HL) = MATCH ADDRES
046.300  301            1238            POP     B               (BC) = LENGTHS
046.301  305            1239            PUSH    B
046.302  006 000        1240            MVI     B,0
046.304  021 000 063     1241            LXI     D,EDIB
046.307  315 007 056     1242            CALL    $MOVL           COPY INTO PLACE                 /WCZ080480/
                        1243
                        1244  *         COMPRESS STRING AND PUT BACK IN BUFFER
                        1245
046.312  041 137 062     1246            LXI     H,WRKSTR
046.315  315 361 054     1247            CALL    $CLL            COMPUTE LINE LENGTH
046.320  117            1248            MOV     C,A             (C) = LENGTH
046.321  315 144 054     1249            CALL    RSL             REPLACE SINGLE LINE
                        1250
                        1251  *         DECREMENT REQUEST COUNT
                        1252
046.324  321            1253            POP     D
046.325  301            1254            POP     B
046.326  325            1255            PUSH    D
046.327  013            1256            DCX     B
                        1257
                        1258  *         SEE IF MORE TO GO
                        1259
046.330  170            1260            MOV     A,B
046.331  261            1261            ORA     C
046.332  312 343 046     1262            JZ      EDI6            NO MORE LINES TO CONSIDER
                        1263
046.335  315 045 052     1264  EDI5      CALL    ACL             ADVANCE COMMAND LINE
046.340  302 200 046     1265            JNZ     EDI3            MORE TO GO
046.343  301            1266  EDI6      POP     B
```

```
046.344  311              1267            RET

                          1269  **        RDS - READ DELIMITED STRING.
                          1270  *
                          1271  *         ENTRY   (B) = DELIMITER
                          1272  *                 (HL) = ADDRESS FOR STRING
                          1273  *         EXIT    (HL) UNCHANGED
                          1274  *                 (C) = LENGTH OF STRING
                          1275  *         USES    A,F,C
                          1276
                          1277
046.345  016 377         1278  RDS       MVI     C,377Q
046.347  345              1279            PUSH    H
046.350  325              1280            PUSH    D
046.351  026 050         1281            MVI     D,40            (D) = MAX COUNT
046.353  025              1282  RDS1      DCR     D
046.354  312 031 043      1283            JZ      REFUSE          TOO MANY
046.357  315 217 053      1284            CALL    GTC             GET TEXT CHARACTER
046.362  167              1285            MOV     M,A
046.363  043              1286            INX     H
046.364  014              1287            INR     C
046.365  270              1288            CMP     B
046.366  302 353 046      1289            JNE     RDS1            NOT DELIMITER
                          1290
                          1291  *         OUT OF STRING
                          1292
046.371  053              1293            DCX     H
046.372  066 000         1294            MVI     M,0             END IT
046.374  321              1295            POP     D               RESTORE (DE)
046.375  341              1296            POP     H
046.376  311              1297            RET
```

```
                                1301  **      FLUSH - PROCESS FLUSH COMMAND.
                                1302  *
                                1303
                                1304
     046.377                    1305  FLUSH   EQU     *               ENTRY POINT
     046.377  315 041 054       1306          CALL    RCR             REQUIRE CARRIAGE RETURN
     047.002  072 226 061       1307  FLUSH1  LDA     INFB+FB.FLG
     047.005  365               1308          PUSH    PSW             SAVE FLAG
     047.006  315 151 050       1309          CALL    NEXT.           MOVE DATA THROUGH
     047.011  361               1310          POP     PSW
     047.012  346 002           1311          ANI     FT.OR
     047.014  302 002 047       1312          JNZ     FLUSH1          NOT AT EOF YET
                                1313
                                1314  *       HAVE READ EOF, WRITE ALL.
                                1315
     047.017  041 260 061       1316          LXI     H,OUTFB
     047.022  303 217 056       1317          JMP     $FCLO           CLOSE AND EXIT
```

```
                                1320  ***     BYE - EXIT EDITOR.
                                1321  *
                                1322  *       BYE (CR)
                                1323  *
                                1324  *       BYE FLUSHES OUT THE EXISTING FILES, AND EXITS.
                                1325
                                1326
     047.025  315 377 046       1327  BYE     CALL    FLUSH
     047.030  041 313 061       1328          LXI     H,XOUTFB        CLOSE  *XOUT*  FILE                /80.02.GC/
     047.033  315 217 056       1329          CALL    $FCLO                                              /80.02.GC/
     047.036  257               1330          XRA     A
     047.037  377 000           1331          DB      SYSCALL,.EXIT           EXIT
```

```
                                1335  **      NEWIN - PROCESS NEWIN COMMAND.
                                1336  *
                                1337
                                1338
        047.041                 1339  NEWIN   EQU     *
                                1340
                                1341  *       SET NEW 'IN' FILE
                                1342
        047.041 315 217 053     1343          CALL    GTC             GET DELIMITER
        047.044 376 012         1344          CPI     NL
        047.046 312 031 043     1345          JE      REFUSE          NO NAME
        047.051 107             1346          MOV     B,A
        047.052 041 327 062     1347          LXI     H,EDIA
        047.055 315 345 046     1348          CALL    RDS             READ DELIMITED STRING
        047.060 315 151 055     1349          CALL    $MLU            MAP LINE TO UPPER CASE
        047.063 315 041 054     1350          CALL    RCR             REQUIRE CARRIAGE RETURN
        047.066 315 350 053     1351          CALL    MIM             REQUEST MINIMUM MEMORY
        047.071 076 021         1352          MVI     A,FB.NAML
        047.073 271             1353          CMP     C               SEE IF TOO LONG A NAME GIVEN
        047.074 332 204 047     1354          JC      NEWIN4          TOO LONG
        047.077 072 226 061     1355          LDA     INFB+FB.FLG
        047.102 346 002         1356          ANI     FT.OR
        047.104 312 155 047     1357          JZ      NEWIN1          NOT ALREADY OPEN
        047.107 315 136 031     1358          CALL    $TYPTX
        047.112 012 117 154     1359          DB      NL,'Old Input File Not Finished.',' '+200Q
        047.150 315 110 052     1360          CALL    AYS             ARE YOU SURE?
        047.153 330             1361          RC                      NOT SURE
        047.154 300             1362          RNE                     NOT SURE
        047.155 041 225 061     1363  NEWIN1  LXI     H,INFB
        047.160 315 217 056     1364          CALL    $FCLO           CLOSE OLD ONE
        047.163 345             1365          PUSH    H
        047.164 315 271 055     1366          CALL    $MOVLL                                  /WCZ080480/
        047.167 021 000         1367          DW      FB.NAML
        047.171 327 062         1368          DW      EDIA
        047.173 237 061         1369          DW      FB.NAM+INFB     SET NAME
        047.175 341             1370          POP     H
        047.176 021 217 061     1371          LXI     D,DEFALT
        047.201 303 057 056     1372          JMP     $FOPER          OPEN FOR READ AND EXIT
                                1373
                                1374  *       ILLEGAL FILE NAME GIVEN
                                1375
        047.204 315 136 031     1376  NEWIN4  CALL    $TYPTX
        047.207 007 111 154     1377          DB      BELL,'Illegal File Name',' '+200Q
        047.232 303 200 042     1378          JMP     EDIX
```

```
                                1382  **      NEWOUT,'NAME'
                                1383  *
                                1384
                                1385
047.235                         1386  NEWOUT  EQU     *
                                1387
                                1388  *       SET NEW 'OUT' FILE
                                1389
047.235  315 217 053           1390          CALL    GTC             GET DELIMITER
047.240  107                   1391          MOV     B,A             (B) = DELIMITER
047.241  376 012               1392          CPI     NL
047.243  312 031 043           1393          JE      REFUSE          NO NEW FILE
047.246  041 327 062           1394          LXI     H,EDIA
047.251  315 345 046           1395          CALL    RDS             READ DELIMITED STRING
047.254  315 151 055           1396          CALL    $MLU            MAP LINE TO UPPER CASE
047.257  315 041 054           1397          CALL    RCR             REQUIRE CARRIAGE RETURN
047.262  315 350 053           1398          CALL    MIM             REQUEST MINIMUM MEMROY
047.265  076 021               1399          MVI     A,FB.NAML
047.267  271                   1400          CMP     C
047.270  332 204 047           1401          JC      NEWIN4          TOO MANY CHARACTERS FOR FILE NAME
047.273  072 261 061           1402          LDA     OUTFB+FB.FLG
047.276  346 004               1403          ANI     FT.OW
047.300  312 352 047           1404          JZ      NEWO1           OUTPUT CLOSED
047.303  315 136 031           1405          CALL    $TYPTX
047.306  012 117 154           1406          DB      NL,'Old Output File Not Finished.',' '+200Q
047.345  315 110 052           1407          CALL    AYS             SURE?
047.350  330                   1408          RC                      NOT SURE
047.351  300                   1409          RNE                     NOT SURE
047.352  041 260 061           1410  NEWO1   LXI     H,OUTFB
047.355  315 217 056           1411          CALL    $FCLO           CLOSE OLD STUFF
047.360  345                   1412          PUSH    H
047.361  315 271 055           1413          CALL    $MOVLL                                  /WCZ080480/
047.364  021 000               1414          DW      FB.NAML
047.366  327 062               1415          DW      EDIA
047.370  272 061               1416          DW      OUTFB+FB.NAM
047.372  341                   1417          POP     H               (HL) = FB ADDRESS
047.373  021 217 061           1418          LXI     D,DEFALT
047.376  303 066 056           1419          JMP     $FOPEW          OPEN FOR WRITE AND EXIT
```

```
                                1423  **      XOUT    -  PROCESS XOUT COMMAND                        /80.02.GC/
                                1424  *
                                1425  *       XOUT closes any currently specified XPRINT channel,
                                1426  *       and opens the newly specified one.
                                1427  *
                                1428
        050.001                 1429  XOUT    EQU     *
                                1430
                                1431  *       SET NEW 'OUT' FILE
                                1432
        050.001  315 217 053    1433          CALL    GTC
        050.004  107            1434          MOV     B,A
        050.005  376 012        1435          CPI     NL
        050.007  312 031 043    1436          JE      REFUSE          NO NEW FILE
                                1437
        050.012  041 327 062    1438          LXI     H,EDIA
        050.015  315 345 046    1439          CALL    RDS             READ DELIMITED STRING
        050.020  315 151 055    1440          CALL    $MLU            MAP TO UPPER CASE
        050.023  315 041 054    1441          CALL    RCR             GET NEWLINE
        050.026  315 350 053    1442          CALL    MIM             MINIMUM MEMORY
                                1443
        050.031  076 021        1444          MVI     A,FB.NAML
        050.033  271            1445          CMP     C
        050.034  332 204 047    1446          JC      NEWIN4          TOO MANY CHARACTERS
                                1447
        050.037  072 314 061    1448          LDA     XOUTFB+FB.FLG
        050.042  346 004        1449          ANI     FT.OW
        050.044  312 117 050    1450          JZ      XOUT1           OUTPUT CLOSED
                                1451
        050.047  315 136 031    1452          CALL    $TYPTX
        050.052  012 117 154    1453          DB      NL,'Old XOUT File is not finished.',' '+200Q
        050.112  315 110 052    1454          CALL    AYS             SURE?
        050.115  330            1455          RC                      NOT SURE
        050.116  300            1456          RNE                     NOT SURE
                                1457
        050.117  041 313 061    1458  XOUT1   LXI     H,XOUTFB
        050.122  315 217 056    1459          CALL    $FCLO           CLOSE THE OLD ONES
        050.125  345            1460          PUSH    H
        050.126  315 271 055    1461          CALL    $MOVLL                                  /WCZ080480/
        050.131  021 000        1462          DW      FB.NAML
        050.133  327 062        1463          DW      EDIA
        050.135  325 061        1464          DW      XOUTFB+FB.NAM
        050.137  341            1465          POP     H
        050.140  021 217 061    1466          LXI     D,DEFALT
        050.143  303 066 056    1467          JMP     $FOPEW          OPEN FOR WRITE AND EXIT
```

```
                                1471  **      NEXT - PROCESS 'NEXT' COMMAND.
                                1472  *
                                1473
                                1474
        050.146                 1475  NEXT    EQU     *
        050.146  315 041 054    1476          CALL    RCR             REQUIRE CARRIAGE RETURN
        050.151                 1477  NEXT.   EQU     *
        050.151  052 172 061    1478          LHLD    LALPTR
        050.154  174            1479          MOV     A,H
        050.155  265            1480          ORA     L
        050.156  312 215 050    1481          JZ      READ.           NOTHING TO WRITE
        050.161  315 322 054    1482          CALL    SLB             SCAN LINE BACKWARDS
        050.164  042 176 061    1483          SHLD    CRLPTR
        @50.167  042 200 061    1484          SHLD    WRKPTR
        050.172  042 174 061    1485          SHLD    CRPPTR
        050.175  315 314 051    1486          CALL    WRITE.          WRITE ALL
        050.200  303 215 050    1487          JMP     READ.           LOAD BACK UP
```

```
                                1491  **      READ - READ LINES FROM FILE.
                                1492  *
                                1493
050.203                         1494  READ    EQU     *
050.203  315 041 054            1495          CALL    RCR             REQUIRE CARRIAGE RETURN
050.206  315 215 050            1496          CALL    READ.
050.211  332 215 052            1497          JC      CBO1            NO ROOM
050.214  311                    1498          RET
                                1499
050.215  072 226 061            1500  READ.   LDA     INFB+FB.FLG
050.220  346 002                1501          ANI     FT.OR
050.222  312 307 050            1502          JZ      READ2           AT EOF
050.225  052 172 061            1503  READO   LHLD    LALPTR          (HL) = LAST LINE POINTER
050.230  174                    1504          MOV     A,H
050.231  265                    1505          ORA     L
050.232  302 243 050            1506          JNZ     READ1           NOT EMPTY
050.235  041 147 070            1507          LXI     H,BUFFER
050.240  315 070 046            1508          CALL    SAP             SET ALL POINTERS IF NOT TEXT YET
050.243  021 000 002            1509  READ1   LXI     D,512           (DE) = ROOM TO LEAVE IN BUFFER
050.246  031                    1510          DAD     D
050.247  353                    1511          XCHG                    (DE) = PROPOSED NEW LALPTR
050.250  052 210 061            1512          LHLD    BUFMAX
050.253  175                    1513          MOV     A,L             SEE IF WOULD EXCEDE MEMORY
050.254  223                    1514          SUB     E
050.255  174                    1515          MOV     A,H
050.256  232                    1516          SBB     D
050.257  330                    1517          RC                      CBO1 => NO ROOM
                                1518
                                1519  *       HAVE ROOM. READ A LINE.
                                1520
050.260  052 172 061            1521          LHLD    LALPTR
050.263  353                    1522          XCHG
050.264  001 200 000            1523          LXI     B,128
050.267  041 225 061            1524          LXI     H,INFB
050.272  315 324 056            1525          CALL    $FREAL          READ LINE
050.275  332 307 050            1526          JC      READ2           EOF
050.300  353                    1527          XCHG                    (HL) = NEW LWA+1
050.301  042 172 061            1528          SHLD    LALPTR          UPDATE POINTER
050.304  303 225 050            1529          JMP     READO           READ SOME MORE
                                1530
                                1531  *       AT EOF
                                1532
050.307  315 136 031            1533  READ2   CALL    $TYPTX
050.312  012 105 156            1534          DB      NL,'End of Fil','e'+200Q
050.326  041 225 061            1535          LXI     H,INFB
050.331  315 217 056            1536          CALL    $FCLO           CLOSE BUFFER: AM DONE
050.334  067                    1537          STC
050.335  077                    1538          CMC                     CLEAR CARRY
050.336  311                    1539          RET
```

```
                                1543  **      SEARCH - PROCESS SEARCH COMMAND.
                                1544  *
                                1545
                                1546
        050.337                 1547  SEARCH  EQU     *
                                1548
                                1549  *       DECODE SEARCH STRING
                                1550
        050.337  315 217 053   1551          CALL    GTC             GET DELIMITER
        050.342  107           1552          MOV     B,A             (B) = DELIMITER
        050.343  041 327 062   1553          LXI     H,EDIA
        050.346  315 345 046   1554          CALL    RDS             READ DELIMITER STRING
        050.351  171           1555          MOV     A,C
        050.352  247           1556          ANA     A
        050.353  312 031 043   1557          JZ      REFUSE          NULL STRING IS ILLEGAL
        050.356  315 041 054   1558          CALL    RCR             REQUIRE CR
                                1559
                                1560  *       TRY TO FIND LINE.
                                1561
        050.361  052 172 061   1562  SEA0    LHLD    LALPTR
        050.364  174           1563          MOV     A,H
        050.365  265           1564          ORA     L
        050.366  312 027 051   1565          JZ      SEA2            NO DATA IN BUFFER
        050.371  315 322 054   1566          CALL    SLB             SCAN LINE BACKWARDS
        050.374  042 176 061   1567          SHLD    CRLPTR          SET COMMAND LIMIT
        050.377  315 226 054   1568  SEA1    CALL    SEL.            SCAN FOR ELIGIBLE LINE
        051.002  312 027 051   1569          JZ      SEA2            NONE IN BUFFER
        051.005  052 200 061   1570          LHLD    WRKPTR          (HL) = ADDRESS OF TEXT LINE
        051.010  021 327 062   1571          LXI     D,EDIA
        051.013  315 264 054   1572          CALL    SFS             SEE IF FOUND
        051.016  312 056 051   1573          JZ      SEA3            FOUND IT
        051.021  315 045 052   1574          CALL    ACL             ADVANCE LINE
        051.024  302 377 050   1575          JNZ     SEA1            MORE GO TO
                                1576
                                1577  *       NOT FOUND IN THIS BUFFER.
                                1578
        051.027  072 226 061   1579  SEA2    LDA     INFB+FB.FLG
        051.032  346 002       1580          ANI     FT.OR
        051.034  312 074 051   1581          JZ      SEA4            AT END OF FILE
        051.037  315 151 050   1582          CALL    NEXT.           ADVANCE TEXT
        051.042  052 170 061   1583          LHLD    FILPTR
        051.045  042 174 061   1584          SHLD    CRFPTR
        051.050  042 200 061   1585          SHLD    WRKPTR
        051.053  303 361 050   1586          JMP     SEA0
                                1587
                                1588  *       FOUND IT
                                1589
        051.056  363           1590  SEA3    DI                      LOCK OUT CTL-C
        051.057  052 200 061   1591          LHLD    WRKPTR
        051.062  042 202 061   1592          SHLD    PCFPTR
        051.065  042 204 061   1593          SHLD    PCLPTR          SET BOUNDS TO FOUND LINE
        051.070  373           1594          EI                      RE-ALLOW CTL-C
        051.071  303 020 054   1595          JMP     PLA             PRINT LINE AFTER
                                1596
                                1597  *       NOT FOUND ANYWHERE.
                                1598
```

```
051.074  315 136 031  1599  SEA4    CALL    $TYPTX
051.077  012 116 157  1600          DB      NL,'Not Foun','d'+200Q
051.111  311          1601          RET
```

```
                                1605  **      USE - TYPE MEMORY STATISTICS.
                                1606  *
                                1607
                                1608
051.112                         1609  USE     EQU     *
051.112  315 041 054           1610          CALL    RCR             REQUIRE CARRIAGE RETURN
051.115  001 000 000           1611          LXI     B,0             (BC) = LINE COUNT
                                1612
051.120  315 231 054           1613  USE1    CALL    SEL             SCAN FOR ELIGIBLE LINE
051.123  312 143 051           1614          JZ      USE2            NO MORE
051.126  003                   1615          INX     B               COUNT LINE
051.127  315 030 054           1616          CALL    PLB             PRINT LINE BEFORE
051.132  315 020 054           1617          CALL    PLA             PRINT LINE AFTER
051.135  315 045 052           1618          CALL    ACL             ADVANCE COMMAND LINE
051.140  302 120 051           1619          JNZ     USE1            LOOP IF MORE IN RANGE
                                1620
                                1621  *       (BC) = COUNT OF LINES WITHIN RANGE
                                1622
051.143  076 005               1623  USE2    MVI     A,5
051.145  041 246 051           1624          LXI     H,USEB
051.150  315 157 031           1625          CALL    $UDD
051.153  052 170 061           1626          LHLD    FILPTR
051.156  353                   1627          XCHG                    (DE) = FIRST TEXT BYTE ADDRESS
051.157  052 172 061           1628          LHLD    LALPTR          (HL) = LAST TEXT BYTE ADDRESS
051.162  345                   1629          PUSH    H               SAVE
051.163  175                   1630          MOV     A,L
051.164  223                   1631          SUB     E
051.165  117                   1632          MOV     C,A
051.166  174                   1633          MOV     A,H
051.167  232                   1634          SBB     D
051.170  107                   1635          MOV     B,A             (BC) = BYTES USED
051.171  076 005               1636          MVI     A,5
051.173  041 264 051           1637          LXI     H,USEC
051.176  315 157 031           1638          CALL    $UDD
051.201  321                   1639          POP     D               (DE) = LAST
051.202  172                   1640          MOV     A,D
051.203  263                   1641          ORA     E
051.204  302 212 051           1642          JNZ     USE3            NON-ZERO
051.207  021 147 070           1643          LXI     D,BUFFER
051.212                         1644  USE3    EQU     *
051.212  052 210 061           1645          LHLD    BUFMAX          (HL) = MAX BUFFER SIZE
051.215  175                   1646          MOV     A,L
051.216  223                   1647          SUB     E
051.217  117                   1648          MOV     C,A
051.220  174                   1649          MOV     A,H
051.221  232                   1650          SBB     D
051.222  107                   1651          MOV     B,A             (BC) = AMOUNT UNUSED
051.223  076 005               1652          MVI     A,5
051.225  041 302 051           1653          LXI     H,USED
051.230  315 157 031           1654          CALL    $UDD            UNPACK COUNT
051.233  315 136 031           1655          CALL    $TYPTX
051.236  114 151 156           1656          DB      'Lines = '
051.246  130 130 130           1657  USEB    DB      'XXXXX',NL,'Used  = '
051.264  130 130 130           1658  USEC    DB      'XXXXX',NL,'Free  = '
051.302  130 130 130           1659  USED    DB      'XXXXX',ENL
051.310  311                   1660          RET
```

```
                        1664  **      WRITE - WRITE LINES TO OUTPUT FILE.
                        1665  *
                        1666  *       WRITE TEXT BLOCKS FROM THE TOP OF THE BUFFER UNTIL THE CURRENT
                        1667  *       LINE
                        1668
                        1669
      051.311           1670  WRITE   EQU     *
      051.311 315 041 054 1671          CALL    RCR
      051.314 076 000   1672  WRITE.  MVI     A,MI.NOP        DELETE TEXT AFTER WRITE
      051.316 062 374 051 1673  WRI..   STA     WRIA            SET FLAG
      051.321 052 170 061 1674          LHLD    FILPTR
      051.324 042 200 061 1675          SHLD    WRKPTR          START AT TOP OF TEXT
      051.327 072 261 061 1676          LDA     OUTFB+FB.FLG
      051.332 346 004   1677          ANI     FT.DW
      051.334 312 017 052 1678          JZ      WRI4            REQUIRE NEWOUT
                        1679
                        1680  *       SEE IF MORE TEXT TO WRITE.
                        1681
      051.337 052 174 061 1682          LHLD    CRFPTR
      051.342 174       1683          MOV     A,H
      051.343 265       1684          ORA     L
      051.344 312 374 051 1685          JZ      WRI3            NO DATA
                        1686
                        1687  *       WRITE ANOTHER LINE
                        1688
      051.347 052 200 061 1689          LHLD    WRKPTR
      051.352 353       1690          XCHG                    (DE) = CURRENT LINE
      051.353 052 174 061 1691  WRI1    LHLD    CRFPTR          (HL) = LIMIT
      051.356 315 216 030 1692          CALL    $CDEHL          COMPARE
      051.361 365       1693          PUSH    PSW             SAVE RESULTS
      051.362 041 260 061 1694          LXI     H,OUTFB
      051.365 315 127 057 1695          CALL    $FWRIL          WRITE LINE
      051.370 361       1696          POP     PSW             (A) = RESULTS OF TEST
      051.371 302 353 051 1697          JNE     WRI1            MORE TO TO
                        1698
                        1699  *       END OF WRITTING, DELETE LINES WRITTEN.
                        1700
      051.374           1701  WRI3    EQU     *
      051.374 000       1702  WRIA    NOP                     SET TO *RET* FOR SAVE
      051.375 052 174 061 1703          LHLD    CRFPTR
      052.000 042 176 061 1704          SHLD    CRLPTR          SET LINES WRITTEN AS COMMAND RANGE
      052.003 052 170 061 1705          LHLD    FILPTR
      052.006 042 174 061 1706          SHLD    CRFPTR
      052.011 042 200 061 1707          SHLD    WRKPTR
      052.014 303 221 045 1708          JMP     DEL0            DELETE
                        1709
                        1710  *       REQUIRE NEWOUT
                        1711
      052.017 315 136 031 1712  WRI4    CALL    $TYPTX
      052.022 012 007 116 1713          DB      NL,BELL,'No Output Fil','e'+200Q
      052.042 303 200 042 1714          JMP     EDIX
```

```
                                1718  **      ACL - ADVANCE COMMAND LINE.
                                1719  *
                                1720  *       ACL ADVANCES WRKPTR TO THE NEXT COMMAND LINE.
                                1721  *
                                1722  *       EXIT    (WRKPTR) UPDATED
                                1723  *               (HL) = (WRKPTR)
                                1724  *               'Z' SET IF AT END OF RANGE
                                1725  *       USES    A,F,H,L
                                1726
                                1727
052,045  325                   1728  ACL     PUSH    D
052,046  052 176 061           1729          LHLD    CRLPTR
052,051  353                   1730          XCHG
052,052  052 200 061           1731          LHLD    WRKPTR
052,055  315 216 030           1732          CALL    $CDEHL          COMPARE
052,060  321                   1733          POP     D
052,061  310                   1734          RZ                      IF AT END
052,062  315 333 054           1735          CALL    SNL             SCAN TO NEXT LINE
052,065  042 200 061           1736          SHLD    WRKPTR
052,070  264                   1737          ORA     H               CLEAR 'Z'
052,071  311                   1738          RET


                                1740  **      ATL - ACCEPT TEXT LINE
                                1741  *
                                1742  *       ATL READS A LINE OF TEXT FROM THE CONSOLE INTO *LINE*.
                                1743  *
                                1744  *       THE LINE IS TERMINATED BY A 00 BYTE
                                1745  *
                                1746  *       ENTRY   NONE
                                1747  *       EXIT    (HL) = *LINE
                                1748  *               (A) = BYTE COUNT
                                1749  *       USES    A;F;H;L
                                1750
                                1751
052,072  041 347 061           1752  ATL     LXI     H,LINE
052,075  257                   1753          XRA     A
052,076  062 326 040           1754          STA     S.CSLMD         SET LINE-MODE INPUT
052,101  315 233 055           1755          CALL    $RTL            READ LINE
052,104  320                   1756          RNC                     NOT CTL-D
052,105  303 052 043           1757          JMP     EXIT            CTL-D STROCK


                                1759  **      AYS - ASK ARE YOU SURE?
                                1760  *
                                1761  *       AYS PROMPTS THE USER, 'SURE?'
                                1762  *       AND GETS HIS REPLY.
                                1763  *
                                1764  *       ENTRY   NONE
                                1765  *       EXIT    'C' SET IF CTL-D
                                1766  *               'C' CLEAR IF NOT CTL-D
                                1767  *                'Z' SET IF SURE
```

```
                                1768  *       USES    ALL
                                1769
                                1770
052.110  315 136 031           1771  AYS     CALL    $TYPTX
052.113  007 101 162           1772          DB      BELL,'Are You Sure?',' '+200Q
052.132  315 337 055           1773          CALL    $RCHAR
052.135  315 345 055           1774          CALL    $WCHAR          ECHO
052.140  315 205 055           1775          CALL    $MCU            MAP TO UPPER
052.143  376 004               1776          CPI     CTLD
052.145  067                   1777          STC                     ASSUME CTL-D
052.146  310                   1778          RE                      CTL-D
052.147  326 131               1779          SUI     'Y'             SEE IF 'Y'
052.151  247                   1780          ANA     A               CLEAR CARRY
052.152  311                   1781          RET                     RETURN WITH CODES SET


                                1783  **      CBE - CHECK FOR BUFFER EMPTY.
                                1784  *
                                1785  *       IF FILPTR=LALPTR, ZERO POINTERS.
                                1786
052.153  052 170 061           1787  CBE     LHLD    FILPTR
052.156  353                   1788          XCHG
052.157  052 172 061           1789          LHLD    LALPTR
052.162  315 216 030           1790          CALL    $CDEHL
052.165  300                   1791          RNE                     NOT EMPTY
052.166  303 062 046           1792          JMP     PURGE.          HAVE DELETED ALL.


                                1794  **      CBO - CHECK BUFFER OVERFLOW
                                1795  *
                                1796  *       CBO IS CALLED BY COMMANDS WHICH MAY INCREASE THE SIZE
                                1797  *       OF THE BUFFER TEXT. IF THERE IS NOT ROOM ENOUGH FOR
                                1798  *       THE MAXIMUM SIZE INCREASE (120 CHARACTERS), AN OVERFLOW
                                1799  *       IS DECLARED.
                                1800  *
                                1801  *       ENTRY   NONE
                                1802  *       EXIT    TO (RET) IF OK
                                1803  *       USES    A,F
                                1804
052.171  345                   1805  CBO     PUSH    H
052.172  325                   1806          PUSH    D
052.173  052 172 061           1807          LHLD    LALPTR
052.176  021 170 000           1808          LXI     D,120
052.201  031                   1809          DAD     D
052.202  353                   1810          XCHG                    (DE) = NEW LIMIT
052.203  052 210 061           1811          LHLD    BUFMAX
052.206  175                   1812          MOV     A,L
052.207  223                   1813          SUB     E
052.210  174                   1814          MOV     A,H
052.211  232                   1815          SBB     D
052.212  321                   1816          POP     D
052.213  341                   1817          POP     H
052.214  320                   1818          RNC                     IS OK
```

```
052.215  315 136 031  1819  CBO1    CALL    $TYPTX
052.220  012 007 116  1820          DB      NL,BELL,'Not Enough RA','M'+200Q
052.240  303 200 042  1821          JMP     EDIX            ABORT COMMAND


                      1823  **      CDV - CHECK DECIMAL VALIDITY.
                      1824  *
                      1825  *       CDV EXAMINES THE NEXT CHARACTER TO SEE IF IT IS A DECIMAL
                      1826  *       DIGIT.
                      1827  *
                      1828  *       ENTRY   NONE
                      1829  *       EXIT    NEXT CHARACTER NOT READ
                      1830  *               'C' SET IF OK
                      1831  *                (A) = DIGIT VALUE (0=9)
                      1832  *               'C' SET IF NOT DECIMAL DIGIT
                      1833
                      1834
052.243  315 064 053  1835  CDV     CALL    ENC             EXAMINE NEXT CHARACTER
052.246  326 060      1836          SUI     '0'
052.250  330          1837          RC
052.251  376 012      1838          CPI     9+1
052.253  077          1839          CMC
052.254  311          1840          RET


                      1842  **      DCC - DISABLE CTL-C PROCESSING.
                      1843  *
                      1844  *       DCC IS CALLED WHEN A PROCESSOR IS ABOUT TO ENTER SENSITIVE CODE.
                      1845  *       CTL-C'S WILL BE HELD UNTIL A COMPANION CALL TO 'ECC' IS MADE.
                      1846  *
                      1847  *       ENTRY   NONE
                      1848  *       EXIT    NONE
                      1849  *       USES    NONE
                      1850
052.255  365          1851  DCC     PUSH    PSW
052.256  076 001      1852          MVI     A,1
052.260  062 206 061  1853          STA     CCFLG           FLAG DISABLED
052.263  361          1854          POP     PSW
052.264  311          1855          RET


                      1857  **      DDN - DECODE DECIMAL NUMBER.
                      1858  *
                      1859  *       ENTRY   NONE
                      1860  *       EXIT    (BC) = VALUE (IF NON-NULL)
                      1861  *               TO 'REFUSE' IF NULL
                      1862  *       USES    A,B,C,F
                      1863
                      1864
052.265  345          1865  DDN     PUSH    H
```

```
052.266  325          1866          PUSH   D
052.267  315 243 052  1867          CALL   CDV            CHECK DECIMAL VALUE
052.272  332 031 043  1868          JC     REFUSE         NOT DECIMAL DIGIT
052.275  021 000 000  1869          LXI    D,0            (DE) = ACCUMULATOR
052.300  315 243 052  1870  DDN1    CALL   CDV            CHECK DECIMAL VALUE
052.303  332 332 052  1871          JC     DDN2           NO MORE DIGITS
052.306  315 324 030  1872          CALL   $MU10          (HL) = (DE)*10
052.311  332 031 043  1873          JC     REFUSE         OVERFLOW
052.314  137          1874          MOV    E,A
052.315  026 000      1875          MVI    D,0            (DE) = DIGIT VALUE
052.317  031          1876          DAD    D
052.320  332 031 043  1877          JC     REFUSE         NO GOOD
052.323  353          1878          XCHG                  (DE) = VALUE
052.324  315 205 053  1879          CALL   GNC            READ DECIMAL VALUE
052.327  303 300 052  1880          JMP    DDN1           ACCEPT ANOTHER
                      1881
                      1882  *       NUMBER ACCUMULATED, RETURN.
                      1883
052.332  102          1884  DDN2    MOV    B,D
052.333  113          1885          MOV    C,E
052.334  321          1886          POP    D
052.335  341          1887          POP    H
052.336  311          1888          RET


                      1890  **      DTBK - DELETE TEXT BLOCK                    /80,02,GC/
                      1891  *
                      1892  *       DTBK DELETES THE SPECIFIED TEXT BLOCK FROM THE TABLE
                      1893  *
                      1894  *
                      1895  *       ENTRY:  A      = COUNT
                      1896  *               HL     = ADDRESS IN BLOCK
                      1897  *
                      1898  *       EXIT:   NONE
                      1899  *
                      1900  *       USES:   PSW
                      1901  *
                      1902
052.337  305          1903  DTBK    PUSH   B
052.340  117          1904          MOV    C,A
052.341  006 000      1905          MVI    B,0            BC = FULL WORD COUNT
052.343  315 350 052  1906          CALL   DTBK.
052.346  301          1907          POP    B
052.347  311          1908          RET
```

```
                                1910  **      BC      = FULL WORD COUNT
                                1911  *
                                1912
052.350  345                   1913  DTBK.   PUSH    H
052.351  325                   1914          PUSH    D
052.352  353                   1915          XCHG                    DE = BUFFER ADDRESS
                                1916
                                1917  *       FIX POINTERS THAT WILL MOVE
                                1918
052.353  052 176 061           1919          LHLD    CRLPTR          HL = CURRENT RANGE LAST POINTER
052.356  315 216 055           1920          CALL    HLCPDE
052.361  332 375 052           1921          JC      DTBK1           DELETION IS NOT IN RANGE
052.364  312 375 052           1922          JZ      DTBK1           DELETION IS NOT IN RANGE
                                1923
052.367  315 026 053           1924          CALL    DTBK3           HL = HL - BC
052.372  042 176 061           1925          SHLD    CRLPTR
052.375                        1926  DTBK1   EQU     *
                                1927
052.375  052 172 061           1928          LHLD    LALPTR
053.000  345                   1929          PUSH    H
053.001  315 026 053           1930          CALL    DTBK3           HL = HL - BC
053.004  042 172 061           1931          SHLD    LALPTR
053.007  341                   1932          POP     H
                                1933
053.010  353                   1934          XCHG                    HL = ADDRESS IN BUFFER
053.011  345                   1935          PUSH    H               SAVE DESTINATION
053.012  011                   1936          DAD     B
053.013  353                   1937          XCHG                    DE = SOURCE ADDRESS
053.014  315 035 053           1938          CALL    DTBK4           BC = HL - DE
053.017  341                   1939          POP     H               HL = DESTINATION ADDRESS
                                1940
053.020  315 007 056           1941  DTBK2   CALL    $MOVL                                   /WCZ080480/
                                1942
053.023  321                   1943          POP     D
053.024  341                   1944          POP     H
053.025  311                   1945          RET


053.026  175                   1947  DTBK3   MOV     A,L
053.027  221                   1948          SUB     C
053.030  157                   1949          MOV     L,A
053.031  174                   1950          MOV     A,H
053.032  230                   1951          SBB     B
053.033  147                   1952          MOV     H,A
053.034  311                   1953          RET
```

```
053.035  175            1955  DTBK4   MOV     A,L
053.036  223            1956          SUB     E
053.037  117            1957          MOV     C,A
053.040  174            1958          MOV     A,H
053.041  232            1959          SBB     D
053.042  107            1960          MOV     B,A
053.043  311            1961          RET


                        1963  **      ECC - ENABLE CTL-C.
                        1964  *
                        1965  *       ECC IS CALLED TO RESTORE CTL-C PROCESSING AFTER
                        1966  *       A CALL TO *DCC*
                        1967  *
                        1968  *       IF A CTL-C WAS HIT IN THE INTERIM, IT WILL BE PROCESSED NOW.
                        1969  *
                        1970  *       ENTRY   NONE
                        1971  *       EXIT    TO CTL-C PROCESSOR IF ONE WAS STRUCK
                        1972  *       USES    NONE
                        1973
                        1974
053.044  365            1975  ECC     PUSH    PSW
053.045  363            1976          DI                      INTERLOCK
053.046  257            1977          XRA     A
053.047  062 206 061    1978          STA     CCFLG           CLEAR FLAG
053.052  072 207 061    1979          LDA     CCPEND
053.055  373            1980          EI
053.056  247            1981          ANA     A
053.057  302 374 042    1982          JNZ     INTRPT          PROCESS THAT NOW
053.062  361            1983          POP     PSW
053.063  311            1984          RET


                        1986  **      ENC - EXAMINE NEXT CHARACTER.
                        1987  *
                        1988  *       ENC RETURNS A PREVIEW OF THE NEXT INPUT CHARACTER. THE CHARACTER
                        1989  *       'POINTER' IS NOT UPDATED.
                        1990  *
                        1991  *       ENTRY   NONE
                        1992  *       EXIT    (A) = CHARACTER
                        1993  *       USES    A,F
                        1994
                        1995
053.064  072 156 053    1996  ENC     LDA     ENCA
053.067  247            1997          ANA     A
053.070  300            1998          RNZ                     HAVE CHARACTER
                        1999
                        2000  *       MUST READ ANOTHER CHARACTER FROM LINE OR TERMINAL.
                        2001
053.071  345            2002          PUSH    H
053.072  052 212 061    2003          LHLD    LINPTR
053.075  175            2004          MOV     A,L
```

```
053.076  074          2005          INR     A
053.077  365          2006          PUSH    PSW             SAVE FOR LATER COMPARE
053.100  176          2007          MOV     A,M             (A) = CHARACTER
053.101  043          2008          INX     H
053.102  247          2009          ANA     A
053.103  302 137 053  2010          JNZ     ENC1            GOT CHARACTER IN LINE
                      2011
                      2012  *       MUST READ ANOTHER CHARACTER FROM TERMINAL
                      2013
053.106  072 214 061  2014          LDA     PROCHA
053.111  247          2015          ANA     A
053.112  304 345 055  2016          CNZ     $WCHAR          ECHO PROBATION CHARACTER
053.115  315 015 055  2017          CALL    $INCHA          READ ANOTHER CHARACTER
053.120  376 004      2018          CPI     CTLD
053.122  312 052 043  2019          JE      EXIT            IS CTL-D
053.125  052 212 061  2020          LHLD    LINPTR
053.130  167          2021          MOV     M,A             STORE IN LINE
053.131  062 214 061  2022          STA     PROCHA          PUT ON 'PROBATION'
053.134  043          2023          INX     H
053.135  066 000      2024          MVI     M,0
053.137  042 212 061  2025  ENC1    SHLD    LINPTR          UPDATE LINE POINTER
053.142  062 156 053  2026          STA     ENCA            SET PRE-READ CHARACTER
053.145  147          2027          MOV     H,A             SAVE CHARACTER
053.146  361          2028          POP     PSW             (A) = PREVIOUS *L* VALUE+1
053.147  275          2029          CMP     L
053.150  302 250 042  2030          JNE     EDI1            BACKSPACE OR RUBBOUT
053.153  174          2031          MOV     A,H             (A) = SAVED CHARACTER
053.154  341          2032          POP     H               RESTORE (HL)
053.155  311          2033          RET
                      2034
053.156  000          2035  ENCA    DB      0               HELD CHARACTER


                      2037  **      ERROR - PROCESS ERROR MESSAGES.
                      2038  *
                      2039  *       ERROR IS CALLED WHEN A FILE ERROR OCCURS.
                      2040  *       IT EXITS TO *RESTART*, WHICH CLEANS THE STACK.
                      2041  *
                      2042  *       ENTRY   (A) = ERROR CODE
                      2043  *       EXIT    TO RESTART
                      2044  *       USES    ALL
                      2045
                      2046
053.157  365          2047  ERROR   PUSH    PSW             SAVE CODE
053.160  315 136 031  2048          CALL    $TYPTX
053.163  012 007 105  2049          DB      NL,BELL,'Error -',' '+200Q
053.175  361          2050          POP     PSW
053.176  046 012      2051          MVI     H,NL
053.200  377 057      2052          DB      SYSCALL,.ERROR
053.202  303 200 042  2053          JMP     RESTART
```

```
                        2055  **      GNC - GET NEXT CHARACTER.
                        2056  *
                        2057  *       GNC READS THE NEXT CHARACTER, AND ADVANCES THE POINTER.
                        2058  *
                        2059  *       ENTRY   NONE
                        2060  *       EXIT    (A) = CHARACTER
                        2061  *       USES    A,F
                        2062
                        2063
053.205  315 064 053    2064  GNC     CALL    ENC             EXAMINE NEXT
053.210  365            2065          PUSH    PSW             SAVE CHARACTER
053.211  257            2066          XRA     A
053.212  062 156 053    2067          STA     ENCA            CLEAR HELD CHARACTER
053.215  361            2068          POP     PSW
053.216  311            2069          RET


                        2071  **      GTC - GET TEXT CHARACTER.
                        2072  *
                        2073  *       GTC GETS A CHARACTER FROM THE INPUT STREAM, AND REQUIRES IT TO B
                        2074  *       PRINTABLE CHARACTER.
                        2075  *
                        2076  *       ENTRY   NONE
                        2077  *       EXIT    (A) = CHARACTER
                        2078  *       USES    A,F
                        2079
                        2080
053.217  315 064 053    2081  GTC     CALL    ENC
053.222  376 011        2082          CPI     TAB
053.224  312 205 053    2083          JE      GNC             ALLOW TABS
053.227  376 014        2084          CPI     FF
053.231  312 205 053    2085          JE      GNC             ALLOW FORM FEEDS
053.234  376 040        2086          CPI     20H
053.236  332 031 043    2087          JC      REFUSE          BAD
053.241  303 205 053    2088          JMP     GNC             GET IT AND RETURN


                        2090  **      ITBK - INSERT TEXT BLOCK                /80.02.GC/
                        2091  *
                        2092  *       ITBK INSERTS THE SPECIFIED NUMBER OF BYTES INTO
                        2093  *       THE SPECIFIED TEXT BLOCK AT THE SPECIFIED ADDRESS.
                        2094  *
                        2095  *
                        2096  *       ENTRY:  A       = COUNT
                        2097  *               HL      = ADDRESS IN BUFFER
                        2098  *
                        2099  *       EXIT:   NONE
                        2100  *
                        2101  *       USES:   PSW
                        2102  *
                        2103
053.244  305            2104  ITBK    PUSH    B
```

```
053.245  117          2105           MOV     C,A
053.246  006 000      2106           MVI     B,0             BC = FULL WORD COUNT
053.250  315 255 053  2107           CALL    ITBK.
053.253  301          2108           POP     B
053.254  311          2109           RET


                      2111   **      BC      = FULL WORD COUNT
                      2112   *
                      2113
053.255  345          2114   ITBK.   PUSH    H
053.256  325          2115           PUSH    D
053.257  353          2116           XCHG                    DE = ADDRESS IN BUFFER
                      2117
                      2118   *       FIX MOVING POINTERS
                      2119
053.260  052 176 061  2120           LHLD    CRLPTR
053.263  315 216 055  2121           CALL    HLCPDE
053.266  332 300 053  2122           JC      ITBK1           DELETION IS NOT IN RANGE
053.271  312 300 053  2123           JZ      ITBK1           DELETION IS NOT IN RANGE
                      2124
053.274  011          2125           DAD     B
053.275  042 176 061  2126           SHLD    CRLPTR          UPDATE CURRENT RANGE LAST POINTER
053.300               2127   ITBK1   EQU     *
                      2128
053.300  052 172 061  2129           LHLD    LALPTR
053.303  345          2130           PUSH    H
053.304  011          2131           DAD     B
053.305  042 172 061  2132           SHLD    LALPTR
053.310  341          2133           POP     H
                      2134
053.311  305          2135           PUSH    B               SAVE COUNT
053.312  315 035 053  2136           CALL    DTBK4           BC = HL - DE
053.315  341          2137           POP     H               HL = COUNT
053.316  031          2138           DAD     D               HL = HL + DE = DESTINATION
                      2139
053.317  303 020 053  2140           JMP     DTBK2           MOVE IT OUT


                      2142   **      LQS - LOCATE QUOTED STRING.
                      2143   *
                      2144   *       LQS FINDS A QUOTED STRING IN A TEXT LINE,
                      2145   *
                      2146   *       THE LINE IS EXPANDED INTO WRKSTR, AND THE SEARCH IS MADE.
                      2147   *
                      2148   *       ENTRY   (HL) = ADDRSS OF STRING
                      2149   *       EXIT    'Z' SET IF FOUND
                      2150   *               (DE) = ADDRESS IN LINWRK, IF FOUND
                      2151   *               (HL) UNCHANGED
                      2152   *       USES    A,F,D,E
                      2153
                      2154
```

```
053.322  353              2155  LOS     XCHG
053.323  052 200 061      2156          LHLD    WRKPTR          POINT TO TEXT
053.326  315 264 054      2157          CALL    SFS             SEARCH FOR STRING
053.331  353              2158          XCHG
053.332  311              2159          RET


                          2161  **      MAM - REQUEST MAXIMUM MEMORY ALLOCATION.
                          2162  *
                          2163  *       MAM REQUESTS THE MAXIMUM MEMORY AVAILABLE SO THAT THE HDOS OVERLAY
                          2164  *       CAN REMAIN RESIDENT.
                          2165  *
                          2166  *       THE SPACE IS GIVEN TO *BUFFER*.
                          2167  *
                          2168  *       * * NOTE * * - SOME OF THE MOVE AND MANAGEMENT ROUTINES
                          2169  *       USED BY *EDIT* CANNOT HANDLE TRANSFERS OF >32768; THEREFORE
                          2170  *       MAM REFUSES TO ALLOCATE MORE THAN 32000 TO THE BUFFER.
                          2171  *       DONT CHANGE THIS WITHOUT CAREFULLY CHECKING THINGS.
                          2172  *
                          2173  *       * * NOTE * * - THIS HOPEFULLY HAS BEEN FIXED AS OF  /80.02.GC/
                          2174  *
                          2175  *       ENTRY   NONE
                          2176  *       EXIT    NONE
                          2177  *       USES    NONE
                          2178  *
                          2179
053.333  315 054 031      2180  MAM     CALL    $SAVALL
053.336  052 320 040      2181          LHLD    S.SYSM
053.341  021 366 377      2182          LXI     D,-10                                   /79.05.sc/
053.344  031              2183          DAD     D                                       /79.05.sc/
053.345  303 372 053      2184          JMP     MIM1            REQUEST AND STORE       /80.02.GC/


                          2186  **      MIM - REQUEST MINIMUM MEMORY.
                          2187  *
                          2188  *       MIM SETS THE CURRENT PROGRAM SIZE TO THE MINIMUM POSSIBLE
                          2189  *       (IMMEDIATELY ABOVE THE LAST TEXT IN MEMORY)
                          2190  *
                          2191  *       ENTRY   NONE
                          2192  *       EXIT    NONE
                          2193  *       USES    NONE
                          2194
                          2195
053.350  315 054 031      2196  MIM     CALL    $SAVALL
053.353  052 172 061      2197          LHLD    LALPTR
053.356  174              2198          MOV     A,H
053.357  265              2199          ORA     L
053.360  302 366 053      2200          JNZ     MIM0            HAVE TEXT
                          2201
                          2202  *       NO TEXT, JUST LOOK AT BUFFER SIZE
                          2203
053.363  041 147 070      2204          LXI     H,BUFFER
```

```
                                2205
053.366  021 040 000  2206  MIM0    LXI     D,32
053.371  031          2207          DAD     D               ADD SOME SLOP
053.372  042 210 061  2208  MIM1    SHLD    BUFMAX
053.375  353          2209          XCHG                    (DE) = NEW LIMIT
053.376  052 322 040  2210          LHLD    S.USRM
054.001  315 216 030  2211          CALL    $CDEHL          SEE IF ALREADY HAVE THAT AMOUNT
054.004  312 047 031  2212          JE      $RSTALL         DONT ASK, WE HAVE IT!
054.007  353          2213          XCHG                    (HL) = AMOUNT TO ASK FOR
054.010  377 052      2214          DB      SYSCALL,.SETTP
054.012  322 047 031  2215          JNC     $RSTALL         IF OK, RESTORE AND EXIT
054.015  303 157 053  2216          JMP     ERROR


                                2218  **      PLA - PRINT LINE AFTER.
                                2219  *
                                2220  *       PLA PRINTES THE LINE IF THE *A* OPTION HAS BEEN SPECIFIED.
                                2221  *
                                2222  *       ENTRY   (WRKPTR) = LINE POINTER
                                2223  *       EXIT    NONE
                                2224  *       USES    A,F
                                2225
                                2226
054.020  072 216 061  2227  PLA     LDA     OPTS
000.000                 2228          ERRNZ   OPT.A-1
054.023  037          2229          RAR
054.024  320          2230          RNC                     NOT SET
054.025  303 342 054  2231          JMP     TTX.            TYPE TEXT


                                2233  **      PLB - PRINT LINE BEFORE.
                                2234  *
                                2235  *       PLB PRINTS THE WORKING LINE IF TGE *BEFORE* OPTION IS
                                2236  *       SELECTED.
                                2237  *
                                2238  *       ENTRY   (WRKPTR) = NEXT LINE TO CONSIDER
                                2239  *       EXIT    (HL) = (WRKPTR)
                                2240  *       USES    A,F,H,L
                                2241
                                2242
054.030  072 216 061  2243  PLB     LDA     OPTS
054.033  346 002      2244          ANI     OPT.B
054.035  310          2245          RZ                      NOT SET
054.036  303 342 054  2246          JMP     TTX.            TYPE TEXT
```

```
                                2248  **      RCR - REQUIRE CARRIAGE RETURN.
                                2249  *
                                2250  *       RCR IS CALLED BY THOSE COMMANDS WHICH END WITH A CARRIAGE
                                2251  *       RETURN.  TOO MAKE SURE THAT CARRIAGE RETURN WAS ENTERED.
                                2252  *
                                2253  *       ENTRY   NONE
                                2254  *       EXIT    NONE
                                2255  *       USES    A,F
                                2256
                                2257
054.041  315 205 053            2258  RCR     CALL    GNC
054.044  376 012                2259          CPI     NL
054.046  302 031 043            2260          JNE     REFUSE          NO GOOD
054.051  315 001 056            2261          CALL    $CRLF           ECHO CRLF
054.054  345                    2262          PUSH    H               SAVE (HL)
054.055  052 174 061            2263          LHLD    CRFPTR
054.060  042 202 061            2264          SHLD    PCFPTR          SAVE PREVIOUS COMMAND BOUNDS
054.063  052 176 061            2265          LHLD    CRLPTR
054.066  042 204 061            2266          SHLD    PCLPTR
054.071  341                    2267          POP     H
054.072  311                    2268          RET


                                2270  **      RQS - READ QUOTED STRING.
                                2271  *
                                2272  *       RQS READS A QUOTED STRING FROM THE INPUT LINE, AND PLACES
                                2273  *       IT IN MEMORY.
                                2274  *
                                2275  *       ENTRY   (HL) = ADDRESS FOR STRING
                                2276  *       EXIT    (HL) = UNCHANGED
                                2277  *               STRING IN MEMORY
                                2278  *       USES    A,F
                                2279
                                2280
054.073  345                    2281  RQS     PUSH    H
054.074  325                    2282          PUSH    D               SAVE (DE)
054.075  315 205 053            2283          CALL    GNC             READ INITIAL QUOTE
054.100  026 050                2284          MVI     D,40
                                2285
                                2286  *       READ ANOTHER CHARACTER
                                2287
054.102  025                    2288  RQS1    DCR     D
054.103  312 031 043            2289          JZ      REFUSE          TOO MANY CHARACTERS
054.106  315 217 053            2290          CALL    GTC             GET TEXT CHARACTER
054.111  376 047                2291          CPI     QUOTE
054.113  167                    2292          MOV     M,A             STORE IN MEMORY
054.114  043                    2293          INX     H
054.115  302 102 054            2294          JNE     RQS1            NOT QUOTE
                                2295
                                2296  *       HAVE QUOTE
                                2297
054.120  315 064 053            2298          CALL    ENC             EXAMINE NEXT
054.123  376 047                2299          CPI     QUOTE
054.125  302 136 054            2300          JNE     RQS2            SONGLE QUOTE - EXIT
```

```
                                2301
                                2302  *       HAVE DOUBLE QUOTE
                                2303
054.130  315 217 053            2304          CALL    GTC             READ '
054.133  303 162 054            2305          JMP     RQS1
                                2306
                                2307  *       END OF STRING
                                2308
054.136  053                    2309  RQS2    DCX     H
054.137  066 000                2310          MVI     M,0             END STRING
054.141  321                    2311          POP     D
054.142  341                    2312          POP     H
054.143  311                    2313          RET


                                2315  **      RSL - REPLACE SINGLE LINE.
                                2316  *
                                2317  *       RSL REPLACES A SINGLE LINE IN THE TEXT BLOCK WITH A LINE
                                2318  *       IN MEMORY.
                                2319  *
                                2320  *       ENTRY   (HL) = REPLACEMENT LINE ADDRESS
                                2321  *               (C) = LENGTH
                                2322  *               (WRKPTR) = ADDRESS IN BLOCK OF LINE TO REPLACE
                                2323  *       EXIT    LINE REPLACED
                                2324  *       USES
                                2325
                                2326
054.144  315 255 052            2327  RSL     CALL    DCC             DISABLE CTL-C
054.147  353                    2328          XCHG
054.150  052 200 061            2329          LHLD    WRKPTR
054.153  315 361 054            2330          CALL    $CLL            CHECK OLD LINE LENGTH
054.156  221                    2331          SUB     C               OLD - NEW                       /80.02.GC/
054.157  332 170 054            2332          JC      RSL1            OLD < NEW                       /80.02.GC/
                                2333
                                2334  *       OLD >= NEW, DELETE EXTRA BYTES                          /80.02.GC/
                                2335
054.162  315 337 052            2336          CALL    DTBK            DELETE BLOCK                    /80.02.GC/
054.165  303 175 054            2337          JMP     RSL2                                            /80.02.GC/
                                2338
                                2339  *       OLD <  NEW, INSERT EXTRA BYTES                          /80.02.GC/
                                2340
054.170  057                    2341  RSL1    CMA                                                     /80.02.GC/
054.171  074                    2342          INR     A                                               /80.02.GC/
054.172  315 244 053            2343          CALL    ITBK            INSERT BLOCK                    /80.02.GC/
000.000                         2344          ERRNZ   *-RSL2                                          /80.02.GC/
                                2345
                                2346  *       MOVE THE TEXT ACTUALLY IN
                                2347
054.175                         2348  RSL2    EQU     *                                               /80.02.GC/
054.175  006 000                2349          MVI     B,0
054.177  315 007 056            2350          CALL    $MOVL           INSERT LINE                             /WCZ080480/
054.202  315 044 053            2351          CALL    ECC             RESTORE CTL-C PROCESSING
054.205  303 020 054            2352          JMP     PLA             PRINT LINE AFTER AND RETURN
```

```
                                2354  **      R8N - READ 8 BIT NUMBER.
                                2355  *
                                2356  *       R8N READS AN 8 BIT NUMBER FROM THE COMMAND STREAM.
                                2357  *
                                2358  *       ENTRY   NONE
                                2359  *       EXIT    (A) = VALUE
                                2360  *               TO 'REFUSE' IF BAD
                                2361  *       USES    A,B,C,F
                                2362
                                2363
054.210  315 265 052            2364  R8N     CALL    D0N             DECODE NUMBER
054.213  170                    2365          MOV     A,B
054.214  247                    2366          ANA     A
054.215  302 031 043            2367          JNZ     REFUSE          TOO LARGE
054.220  171                    2368          MOV     A,C             (A) = VALUE
054.221  311                    2369          RET


                                2371  **      SEL - SCAN FOR ELIGIBLE LINE.
                                2372  *
                                2373  *       SEL SCANS TO FIND THE NEXT LINE MEETING THE QUALIFIER STRING.
                                2374  *
                                2375  *       * * NOTE * *   'DELETE' ASSUMES THAT SEL ONLY CHECKS FOR
                                2376  *               QUALIFIER STRINGS IN Q'QUALS', AND SKIPS
                                2377  *               CALLING SEL IF 'QUALS' IS 00. THIS MUST BE MODIFIED IF MORE
                                2378  *               QUALIFIACTION SPECIFICATIONS ARE ALLOWED IN THE FUTURE.
                                2379  *
                                2380  *       ENTRY   (WRKPTR) = NEXT LINE TO CONSIDER
                                2381  *       EXIT    (WRKPTR) = NEXT LINE TO PROCESS
                                2382  *               (HL) = (WRKPTR)
                                2383  *               'Z' SET IF NO MORE LINES
                                2384  *       USES    A,F,H,L
                                2385
                                2386
054.222  315 045 052            2387  SEL1    CALL    ACL             ADVANCE COMMAND LINE
054.225  310                    2388          RZ                      DONE
                                2389
054.226  315 171 052            2390  SEL.    CALL    CBO             CHECK FOR BUFFER OVERFLOW
054.231  052 200 061            2391  SEL     LHLD    WRKPTR
054.234  174                    2392          MOV     A,H
054.235  265                    2393          ORA     L
054.236  310                    2394          RZ                      NO TEXT ESISTS
054.237  041 051 063            2395          LXI     H,QUALS
054.242  176                    2396          MOV     A,M
054.243  247                    2397          ANA     A
054.244  312 257 054            2398          JZ      SEL2            NO QUAL STRING
                                2399
                                2400  *       SEE IF MEET QUALIFIER STRING
                                2401
054.247  325                    2402          PUSH    D
054.250  315 322 053            2403          CALL    LQS             LOCATE QUOTED STRING
054.253  321                    2404          POP     D
054.254  302 222 054            2405          JNZ     SEL1            DONT HAVE IT
                                2406
```

```
                                2407  *       HAVE QUALIFIED LINE.
                                2408
054.257  052 200 061            2409  SEL2    LHLD    WRKPTR
054.262  264                    2410          ORA     H               CLEAR 'Z'
054.263  311                    2411          RET

                                2413  **      SFS - SEARCH FOR STRING.
                                2414  *
                                2415  *       SFS SCANS AN EXPANDED CHARACTER STRING FOR A MATCH FOR
                                2416  *       SOME PATTERN STRING
                                2417  *
                                2418  *       ENTRY   (DE) = STRING ADDRESS
                                2419  *               (HL) = LINE ADDRESS
                                2420  *       EXIT    (DE) UNCHANGED
                                2421  *               (HL) = ADDRESS OF 1ST MATCH CHARACTER
                                2422  *       USES    A;F;H;L
                                2423
                                2424
054.264  325                    2425  SFS     PUSH    D               SAVE STRING ADDRESS
054.265  345                    2426          PUSH    H
054.266  176                    2427          MOV     A,M
054.267  247                    2428          ANA     A
054.270  076 001                2429          MVI     A,1
054.272  312 316 054            2430          JZ      SFS2            NOT FOUND - NO MORE TEXT
                                2431
                                2432  *       COMPARE STRINGS
                                2433
054.275  032                    2434  SFS1    LDAX    D
054.276  247                    2435          ANA     A
054.277  312 316 054            2436          JZ      SFS2            A MATCH
054.302  276                    2437          CMP     M
054.303  023                    2438          INX     D
054.304  043                    2439          INX     H
054.305  312 275 054            2440          JE      SFS1            KEEP TRYING
                                2441
                                2442  *       A FAILURE
                                2443
054.310  341                    2444          POP     H
054.311  321                    2445          POP     D
054.312  043                    2446          INX     H
054.313  303 264 054            2447          JMP     SFS
                                2448
054.316  341                    2449  SFS2    POP     H
054.317  321                    2450          POP     D
054.320  247                    2451          ANA     A               SET 'Z' IF FOUND
054.321  311                    2452          RET
```

```
                        2454  **      SLB - SCAN LINE BACKWARDS.
                        2455  *
                        2456  *       SLB SCANS BACKWARDS OVER THE PREVIOUS LINE.
                        2457  *
                        2458  *       ENTRY   (HL) = 1ST BYTE OF CURRENT LINE
                        2459  *       EXIT    (HL) = FIRST BYTE OF PREVIOUS LINE
                        2460  *       USES    A,F,H,L
                        2461
                        2462
054,322  053           2463  SLB     DCX     H
054,323  053           2464  SLB1    DCX     H
054,324  176           2465          MOV     A,M
054,325  247           2466          ANA     A
054,326  302 323 054    2467          JNZ     SLB1
054,331  043           2468          INX     H
054,332  311           2469          RET


                        2471  **      SNL - SCAN TO NEXT LINE.
                        2472  *
                        2473  *       SNL SCANS THE TEXT BLOCK FOR THE NEXT LINE.
                        2474  *
                        2475  *       ENTRY   (HL) = START OF CURRENT LINE
                        2476  *       EXIT    (HL) = START OF NEXT LINE
                        2477  *       USES    A,F,H
                        2478
                        2479
054,333  176           2480  SNL     MOV     A,M
054,334  043           2481          INX     H
054,335  247           2482          ANA     A
054,336  302 333 054    2483          JNZ     SNL
054,341  311           2484          RET


                        2486  **      TTX - TYPE TEXT LINE.
                        2487  *
                        2488  *       TTX TYPES THE TEXT FOR A LINE.
                        2489  *
                        2490  *       ENTRY   (HL) = FIRST BYTE
                        2491  *       EXIT    (HL) UNCHANGED
                        2492  *       USES    A,F
                        2493
                        2494
054,342  052 200 061    2495  TTX.    LHLD    WRKPTR
054,345  315 361 054    2496  TTX     CALL    $CLL            COMPUTE LENGTH
054,350  345           2497          PUSH    H               SAVE ADDRESS
054,351  075           2498          DCR     A               REMOVE COUNT OF '00'
054,352  315 314 055    2499          CALL    $TYPCC          TYPE IT
054,355  341           2500          POP     H
054,356  303 001 056    2501          JMP     $CRLF
```

```
     054.361                2504          XTEXT   CLL


                            2506X **      CLL - COMPUTE LINE LENGTH.
                            2507X *
                            2508X *       CLL COUNTS THE NUMBER OF CHARACTERS IN A SOURCE LINE.
                            2509X *       THE LINE IS TERMINATED BY A 00 BYTE; THE 00 BYTE IS ENCLUDED
                            2510X *       IN THE COUNT.
                            2511X *
                            2512X *       ENTRY   (HL) = FWA OF LINE
                            2513X *       EXIT    (HL) UNCHANGED
                            2514X *               (A) = LENGTH OF LINE
                            2515X *       USES    A,F
                            2516X
                            2517X
     054.361  345           2518X $CLL    PUSH    H               SAVE STARTING ADDRESS
     054.362  325           2519X         PUSH    D
     054.363  026 000       2520X         MVI     D,0
                            2521X
     054.365  176           2522X CLL1    MOV     A,M
     054.366  024           2523X         INR     D
     054.367  247           2524X         ANA     A
     054.370  043           2525X         INX     H
     054.371  302 365 054   2526X         JNZ     CLL1            SCAN FOR END
     054.374  172           2527X         MOV     A,D
     054.375  321           2528X         POP     D
     054.376  341           2529X         POP     H
     054.377  311           2530X         RET
     055.000                2531          XTEXT   CCO


                            2533X **      $CCO - CLEAR CONTROL-O
                            2534X *
                            2535X *       $CCO IS CALLED TO CLEAR THE EFFECT OF THE CTL-O CHARACTER.
                            2536X *
                            2537X *       ENTRY   NONE
                            2538X *       EXIT    NONE
                            2539X *       USES    NONE
                            2540X
                            2541X
     055.000  315 054 031   2542X $CCO    CALL    $SAVALL         SAVE REGISTERS
     055.003  076 004       2543X         MVI     A,I.CONFL
     055.005  001 001 000   2544X         LXI     B,CO.FLG        CLEAR CO.FLG
     055.010  377 006       2545X         DB      SYSCALL,.CONSL
     055.012  303 047 031   2546X         JMP     $RSTALL         RESTORE REGISTERS AND RETURN
     055.015                2547          XTEXT   INCHA
```

```
                                2549X **      $INCHA - READ ONE CHARACTER.
                                2550X *
                                2551X *       $INCHA READS ONE CHARACTER FROM THE TERMINAL.
                                2552X *
                                2553X *       CHAR = CTL-U: ERASE LINE
                                2554X *            = BKSP: BACKSPACE CHARACTER
                                2555X *            = RUBOUT: BACKSPACE CHARACTER
                                2556X
                                2557X ********************************************************************************8
                                2558X **
P   000.001                     2559X         ERRNZ   1               THIS ROUTINE IS OBSOLETE
                                2560X
                                2561X ********************************************************************************
                                2562X
                                2563X
    055.015  315 337 055        2564X $INCHA  CALL    $RCHAR          READ A CHARACTER
    055.020  376 010            2565X         CPI     BKSP
    055.022  312 063 055        2566X         JE      INC0            IS BKSP
    055.025  376 177            2567X         CPI     RUBOUT
    055.027  312 063 055        2568X         JE      INC0            IS RUBOUT
    055.032  365                2569X         PUSH    PSW             SAVE CODE
    055.033  072 150 055        2570X         LDA     $INCHAA         (A) = RUBOUT FLAG
    055.036  247                2571X         ANA     A
    055.037  304 345 055        2572X         CNZ     $WCHAR          ECHO RUBOUT CHAR, IF ANY
    055.042  257                2573X         XRA     A
    055.043  062 150 055        2574X         STA     $INCHAA         CLEAR FLAG
    055.046  361                2575X         POP     PSW
    055.047  376 025            2576X         CPI     'U'-'@'
    055.051  300                2577X         RNE                     NOT CTL-U, RETURN
                                2578X
                                2579X *       IS CTL-U
                                2580X
    055.052  041 347 061        2581X         LXI     H,LINE
    055.055  315 001 056        2582X         CALL    $CRLF
    055.060  303 112 055        2583X         JMP     INC1            CLEAR LINE AND SET LINPTR
                                2584X
                                2585X *       IS BKSP
                                2586X
    055.063  052 212 061        2587X INC0    LHLD    LINPTR
    055.066  076 347            2588X         MVI     A,#LINE
    055.070  275                2589X         CMP     L
    055.071  312 015 055        2590X         JE      $INCHA          IF ALREADY AT FRONT
    055.074  053                2591X         DCX     H
    055.075  072 327 040        2592X         LDA     S.CONTY         SEE IF BACKSPACING
    055.100  247                2593X         ANA     A
    055.101  362 122 055        2594X         JP      INC3            IS NON-CRT
    055.104  315 136 031        2595X         CALL    $TYPTX
    055.107  010 040 210        2596X         DB      BKSP,' ',BKSP+200Q      BACKSPACE FOR CRT
    055.112  042 212 061        2597X INC1    SHLD    LINPTR
    055.115  066 000            2598X         MVI     M,0             CLEAR ENTRY
    055.117  303 015 055        2599X         JMP     $INCHA          AGAIN
                                2600X
                                2601X *       BACKSPACE FOR NON-CRT
                                2602X
    055.122  072 150 055        2603X INC3    LDA     $INCHAA         (A) = FLAG
    055.125  247                2604X         ANA     A
```

```
055.126  302 141 055  2605X          JNZ     INC4            AM STILL BACKSPACING
055.131  076 057      2606X          MVI     A,'/'
055.133  062 150 055  2607X          STA     $INCHAA         SET FLAG
055.136  315 345 055  2608X          CALL    $WCHAR          TYPE
055.141  176          2609X INC4     MOV     A,M
055.142  315 345 055  2610X          CALL    $WCHAR          SHOW CHARACTER BEING REMOVED
055.145  303 112 055  2611X          JMP     INC1            CLEAR IT
                      2612X
055.150  000          2613X $INCHAA  DB      0               RUBOUT FLAG
055.151               2614           XTEXT   UDD
```

```
                      2616X **       $UDD - UNPACK DECIMAL DIGITS.
                      2617X *
                      2618X *        UDD CONVERTS A 16 BIT VALUE INTO A SPECIFIED NUMBER OF
                      2619X *        DECIMAL DIGITS. THE RESULT IS ZERO FILLED.
                      2620X *
                      2621X *        ENTRY   (B,C) = ADDRESS VALUE
                      2622X *                (A) = DIGIT COUNT
                      2623X *                (H,L) = MEMORY ADDRESS
                      2624X *        EXIT    (HL) = (HL) + (A)
                      2625X *        USES    ALL
                      2626X
                      2627X
031.157               2628X $UDD     EQU     31157A          IN H17 ROM
055.151               2629           XTEXT   MLU
```

```
                      2631X **       MLU - MAP LOWER CASE LINE TO UPPER CASE.
                      2632X *
                      2633X *        MLU MAPS THE LOWER CASE ALPHABETICS IN A LINE TO UPPER CASE.
                      2634X *
                      2635X *        ENTRY   (HL) = LINE FWA
                      2636X *        EXIT    NONE
                      2637X *        USES    NONE
                      2638X
                      2639X
055.151  365          2640X $MLU     PUSH    PSW             SAVE (PSW)
055.152  345          2641X          PUSH    H               SAVE FWA
055.153  053          2642X          DCX     H               ANTICIPATE INX H
055.154  043          2643X $MLU1    INX     H
055.155  176          2644X          MOV     A,M             (A)= CHARACTER
055.156  315 205 055  2645X          CALL    $MCU            MAP CHAR TO UPPER
055.161  167          2646X          MOV     M,A
055.162  247          2647X          ANA     A
055.163  302 154 055  2648X          JNZ     $MLU1           MORE TO GO
055.166  341          2649X          POP     H               RESTORE (HL)
055.167  361          2650X          POP     PSW             RESTORE (PSW)
055.170  311          2651X          RET
055.171               2652           XTEXT   GNL
```

```
                                2654X **      $GNL - GUARANTEE NEW LINE.
                                2655X *
                                2656X *       $GNL GUARANTEES THE START OF A NEW LINE BY ISSUING A CRLF
                                2657X *       IF THE CURSOR IS NOT AT COLUMN 1.,
                                2658X *
                                2659X *       ENTRY   NONE
                                2660X *       EXIT    NONE
                                2661X *       USES    ALL
                                2662X
                                2663X
055.171  076 002               2664X $GNL    MVI     A,I.CUSOR
055.173  001 000 000           2665X         LXI     B,0
055.176  377 006               2666X         DB      SYSCALL,.CONSL          READ CURSOR
055.200  075                   2667X         DCR     A
055.201  310                   2668X         RZ                      AT COLUMN 1
055.202  303 001 056           2669X         JMP     $CRLF           NEW LINE
055.205                        2670          XTEXT   MCU


                                2672X **      MCU - MAP LOWER CASE TO UPPER CASE.
                                2673X *
                                2674X *       MCU MAPS A LOWER CASE ALPHABETIC TO UPPER
                                2675X *       CASE.
                                2676X *
                                2677X *       ENTRY   (A) = CHARACTER
                                2678X *       EXIT    (A) = CHARACTER RESULT
                                2679X *       USES    A,F
                                2680X
                                2681X
055,205  376 141               2682X $MCU    CPI     'a'
055.207  330                   2683X         RC                      NOT LOWER CASE
055,210  376 173               2684X         CPI     'z'+1
055.212  320                   2685X         RNC                     NOT LOWER CASE
055,213  326 040               2686X         SUI     'a'-'A'
055.215  311                   2687X         RET
055,216                        2688          XTEXT   CHL


                                2690X **      $CHL - COMPLEMENT (HL).
                                2691X *
                                2692X *       (HL) = -(HL)            TWO'S COMPLEMENT
                                2693X *
                                2694X *       ENTRY   NONE
                                2695X *       EXIT    NONE
                                2696X *       USES    A,F,H,L
                                2697X
                                2698X
030,224                         2699X $CHL    EQU     30224A          IN H17 ROM
055.216                         2700          XTEXT   HLCPDE                          /80.02.GC/
                                2701X **      HLCPDE - (HL) COMPARED TO (DE)
                                2702X *
                                2703X *       THIS ROUTINE IS DOUBLE WORD COMPARE OF REGISTER PAIRS (DE) AND (HL).
```

```
                        2704X *
                        2705X *       ENTRY:  (HL)&(DE) SET UP
                        2706X *
                        2707X *       EXIT:   (PSW)   =
                        2708X *                       'Z' SET   IF (HL) =  (DE)
                        2709X *                       'C' SET   IF (HL) <  (DE)
                        2710X *                       'C' CLEAR IF (HL) >= (DE)
                        2711X *
                        2712X *
                        2713X *       USES:   (PSW)
                        2714X *
                        2715X
055.216  174           2716X HLCPDE  MOV     A,H
055.217  272           2717X         CMP     D       'C' SET => (A) < (D)
055.220  300           2718X         RNZ
055.221  175           2719X         MOV     A,L
055.222  273           2720X         CMP     E       'C' SET => (L) < (E)
055.223  311           2721X         RET
055.224                2722          XTEXT   SAVALL


                        2724X **      $RSTALL - RESTORE ALL REGISTERS.
                        2725X *
                        2726X *       $RSTALL RESTORES ALL THE REGISTERS OFF THE STACK, AND
                        2727X *       RETURNS TO THE PREVIOUS CALLER.
                        2728X *
                        2729X *       ENTRY   (SP) = PSW
                        2730X *               (SP+2) = BC
                        2731X *               (SP+4) = DE
                        2732X *               (SP+6) = HL
                        2733X *               (SP+8) = RET
                        2734X *       EXIT    TO *RET*, REGISTERS RESTORED
                        2735X *       USES    ALL
                        2736X
                        2737X
031.047                 2738X $RSTALL EQU     31047A          IN H17 ROM


                        2740X **      $SAVALL - SAVE ALL REGISTERS ON STACK.
                        2741X *
                        2742X *       $SAVALL SAVES ALL THE REGISTERS ON THE STACK.
                        2743X *
                        2744X *       ENTRY   NONE
                        2745X *       EXIT    (SP) = PSW
                        2746X *               (SP+2) = BC
                        2747X *               (SP+4) = DE
                        2748X *               (SP+6) = HL
                        2749X *       USES    H,L
                        2750X
                        2751X
031.054                 2752X $SAVALL EQU     31054A          IN H17 ROM
055.224                 2753          XTEXT   RTL
```

```
                                2755X **      $RTL - READ TEXT LINE.
                                2756X *
                                2757X *       $RTL READS A LINE FROM THE TERMINAL.
                                2758X *
                                2759X *       CHARACTER ARE ACCEPTED FROM THE TERMINAL, RUBOUT AND BACKSPACE
                                2760X *       CHARACTERS ARE PROCESSED. WHEN A CARRIAGE RETURN IS ENTERED,
                                2761X *       $RTL RETURNS.
                                2762X *
                                2763X *       ENTRY   (HL) = BUFFER FWA
                                2764X *       EXIT    'C' CLEAR IF OK
                                2765X *                DATA IN BUFFER
                                2766X *                (A) = TEXT LENGTH
                                2767X *                'C' SET IF CTL-D STRUCK
                                2768X *       USES    A,F
                                2769X
                                2770X
055.224  315 233 055           2771X $RTL.   CALL    $RTL            $RTL IN UPPER CASE
055.227  330                   2772X         RC                      CTL-D
055.230  303 151 055           2773X         JMP     $MLU            MAP LINE TO UPPER CASE
                                2774X
055.233                        2775X $RTL    EQU     *
055.233  345                   2776X         PUSH    H               SAVE FWA
055.234  315 337 055           2777X $RTL1   CALL    $RCHAR
055.237  376 004               2778X         CPI     CTLD
055.241  312 266 055           2779X         JE      $RTL2           CTL-D STRUCK
055.244  167                   2780X         MOV     M,A
055.245  043                   2781X         INX     H
055.246  376 012               2782X         CPI     NL
055.250  302 234 055           2783X         JNE     $RTL1
055.253  053                   2784X         DCX     H
055.254  066 000               2785X         MVI     M,0
055.256  043                   2786X         INX     H
                                2787X
                                2788X *       ALL DONE. COMPUTE LENGTH
                                2789X
055.257  353                   2790X         XCHG                    (DE) = LWA+1
055.260  343                   2791X         XTHL                    (HL) = FWA
055.261  173                   2792X         MOV     A,E
055.262  225                   2793X         SUB     L               (A) = LENGTH
055.263  247                   2794X         ANA     A               CLEAR CARRY
055.264  321                   2795X         POP     D               RESTORE (DE)
055.265  311                   2796X         RET
                                2797X
                                2798X *       CTL-D STRUCK
                                2799X
055.266  341                   2800X $RTL2   POP     H               (HL) = FWA
055.267  067                   2801X         STC
055.270  311                   2802X         RET
055.271                        2803          XTEXT   MOVLL                                   /WCZ080480/
```

```
                                2805X **      $MOVLL - MOVE DATA
                                2806X *
                                2807X *       $MOVLL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                                2808X *
                                2809X *       WHEN THE MOVE IS ACTUALLY DONE, THE ROUTINE $MOVL IS CALLED
                                2810X *       TO DO THE WORK.  $MOVL HAS THE CAPABILITY TO MOVE
                                2811X *       0 TO 65535 BYTES.
                                2812X *
                                2813X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                                2814X *       FIRST TO LAST.
                                2815X *
                                2816X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                                2817X *       LAST TO FIRST.
                                2818X *
                                2819X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                                2820X *
                                2821X *       CALL    $MOVLL
                                2822X *       DW      COUNT
                                2823X *       DW      FROM
                                2824X *       DW      TO
                                2825X *
                                2826X *       ENTRY   ((SP)) = RET
                                2827X *               (RET+0) = COUNT (WORD VALUE)
                                2828X *               (RET+2) = FROM
                                2829X *               (RET+4) = TO
                                2830X *       EXIT    TO (RET+6)
                                2831X *               (DE) = ADDRESS OF NEXT FROM BYTE
                                2832X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                                2833X *               'C' CLEAR
                                2834X *       USES    ALL
                                2835X
                                2836X
055.271  341                    2837X $MOVLL  POP     H               (HL) = RET
055.272  116                    2838X         MOV     C,M
055.273  043                    2839X         INX     H
055.274  106                    2840X         MOV     B,M             (BC) = COUNT
055.275  043                    2841X         INX     H
055.276  136                    2842X         MOV     E,M
055.277  043                    2843X         INX     H
055.300  126                    2844X         MOV     D,M             (DE) = FROM
055.301  043                    2845X         INX     H
055.302  325                    2846X         PUSH    D               ((SP)) = FROM
055.303  136                    2847X         MOV     E,M
055.304  043                    2848X         INX     H
055.305  126                    2849X         MOV     D,M             (DE) = TO
055.306  043                    2850X         INX     H
055.307  343                    2851X         XTHL                    ((SP)) = RET, (HL) = FROM
055.310  353                    2852X         XCHG                    (DE) = FROM , (HL) = TO
055.311  303 007 056            2853X         JMP     $MOVL           MOVE IT
055.314                         2854          XTEXT   TYPCC
```

```
                        2856X **      $TYPCC - TYPE A CHARACTER STRING BY COUNT.
                        2857X *
                        2858X *       $TYPCC TYPES A STRING OF CHARACTERS. THE CALLER SUPPLIES
                        2859X *       THE CHARACTER ADDRESS AND COUNT.
                        2860X *
                        2861X *       ENTRY   (HL) = ADDRESS
                        2862X *               (A) = COUNT
                        2863X *       EXIT    (HL) = LAST CHARACTER ADDRESS+1
                        2864X *       USES    A,F,H,L
                        2865X
                        2866X
  055,314               2867X $TYPCC  EQU     *
  055,314  247          2868X         ANA     A
  055,315  310          2869X         RZ                      NOTHING TO TYPE
  055,316  365          2870X         PUSH    PSW             SAVE COUNT
  055,317  176          2871X         MOV     A,M             (A) = CHARACTER
  055,320  043          2872X         INX     H
  055,321  377 002       2873X         DB      SYSCALL,,SCOUT
  055,323  361          2874X         POP     PSW
  055,324  075          2875X         DCR     A
  055,325  303 314 055  2876X         JMP     $TYPCC
  055,330               2877          XTEXT   TYPCH

                        2879X **      $TYPCH - TYPE SINGLE CHARACTER.
                        2880X *
                        2881X *       ENTRY   (RET) = CHARACTER
                        2882X *       EXIT    TO (RET)+1
                        2883X *               (A) = CHARACTER TYPED
                        2884X
                        2885X
  055,330  343          2886X $TYPCH  XTHL                    (HL) = RETURN ADDRESS
  055,331  176          2887X         MOV     A,M             (A) = CHARACTER
  055,332  043          2888X         INX     H
  055,333  343          2889X         XTHL                    RESTORE ADVANCED EXIT ADDRESS
                        2890X
                        2891X **      $TYPC, - TYPE SINGLE CHARACTER.
                        2892X *
                        2893X *       ENTRY   (A) = CHARACTER
                        2894X *       EXIT    TO (RET)
                        2895X
  055,334  377 002      2896X $TYPC,  DB      SYSCALL,,SCOUT
  055,336  311          2897X         RET
  055,337               2898          XTEXT   RCHAR
```

```
                                2900X **        $RCHAR - READ SINGLE CHARACTER FROM CONSOLE.
                                2901X *
                                2902X *         ENTRY   NONE
                                2903X *         EXIT    (A) = CHARACTER
                                2904X *         USES    A,F
                                2905X
                                2906X
     055.337  377 001           2907X $RCHAR    DB      SYSCALL,.SCIN
     055.341  332 337 055       2908X           JC      $RCHAR          NOT READY
     055.344  311               2909X           RET
                                2910X
     055.345  377 002           2911X $WCHAR    DB      SYSCALL,.SCOUT
     055.347  311               2912X           RET
     055.350                    2913            XTEXT   INDL


                                2915X **        $INDL - INDEXED LOAD.
                                2916X *
                                2917X *         $INDL LOADS DE WITH THE TWO BYTES AT (HL)+DISPLACMENT
                                2918X *
                                2919X *         THIS ACTS AS AN INDEXED FULL WORD LOAD.
                                2920X *
                                2921X *         (DE) = ( (HL) + DSPLACEMENT )
                                2922X *
                                2923X *         ENTRY   ((RET)) = DISPLACMENT (FULL WORD)
                                2924X *                 (HL) = TABLE ADDRESS
                                2925X *         EXIT    TO (RET+2)
                                2926X *         USES    A,F,D,E
                                2927X
                                2928X
     030.234                    2929X $INDL     EQU     30234A          IN H17 ROM
     055.350                    2930            XTEXT   TBLS


                                2932X **        $TBLS - TABLE SEARCH
                                2933X *
                                2934X *         TABLE FORMAT
                                2935X *
                                2936X *         DB      KEY1,VAL1,
                                2937X *         .       .
                                2938X *         .       .
                                2939X *         DB      KEYN,VALN
                                2940X *         DB      0
                                2941X *
                                2942X *         ENTRY   (A) = PATTERN
                                2943X *                 (H,L) = TABLE FWA
                                2944X *         EXIT    (A) = PATTERN IF FOUND
                                2945X *                 'Z' SET IF FOUND
                                2946X *                 'Z' CLEAR IF NOT FOUND OR PATTERN=0              /78.10.GC/
                                2947X *         USES    A,F,H,L
                                2948X
                                2949X
```

```
055,350  305          2950X $TBLS  PUSH   B
055,351  376 000      2951X        CPI    0                                    /78.10.GC/
055,353  312 375 055  2952X        JZ     TBL2                                 /78.10.GC/
055,356  107          2953X        MOV    B,A
055,357  176          2954X TBL1   MOV    A,M          (A) = CHARACTER
055,360  043          2955X        INX    H
055,361  270          2956X        CMP    B
055,362  312 377 055  2957X        JZ     TBL3         IF MATCH
055,365  247          2958X        ANA    A
055,366  043          2959X        INX    H            SKIP PAST
055,367  302 357 055  2960X        JNZ    TBL1         IF NOT END OF TABLE
055,372  053          2961X        DCX    H
055,373  053          2962X        DCX    H
055,374  257          2963X        XRA    A            SET TO ZERO FOR OLD USERS    /78.10.GC/
055,375  376 001      2964X TBL2   CPI    1            CLEAR ZERO                   /78.10.GC/
                      2965X
                      2966X *      DONE
                      2967X
055,377  301          2968X TBL3   POP    B
056,000  311          2969X        RET
056,001               2970         XTEXT  CDEHL


                      2972X **     $CDEHL - COMPARE (DE) TO (HL)
                      2973X *
                      2974X *      $CDEHL COMPARES (DE) TO (HL) FOR EQUALITY.
                      2975X *
                      2976X *      ENTRY  NONE
                      2977X *      EXIT   'Z' SET IF (DE) = (HL)
                      2978X *      USES   A,F
                      2979X
                      2980X
030,216               2981X $CDEHL EQU    30216A       IN H17 ROM
056,001               2982         XTEXT  CRLF


                      2984X **     $CRLF - TYPE CARRIAGE RETURN/ LINE FEED
                      2985X *
                      2986X *      $CRLF IS USED TO GENERATE PADDED CRLF'S.
                      2987X *
                      2988X *      ENTRY  NONE
                      2989X *      EXIT   (A) = 0
                      2990X *      USES   A,F
                      2991X
                      2992X
056,001  076 012      2993X $CRLF  MVI    A,NL
056,003  377 002      2994X        DB     SYSCALL,,SCOUT
056,005  257          2995X        XRA    A
056,006  311          2996X        RET
056,007               2997         XTEXT  DADA
```

```
                        2999X **      $DADA - PERFORM (H,L) = (H,L) + (0,A)
                        3000X *
                        3001X *       ENTRY   (H,L) = BEFORE VALUE
                        3002X *               (A) = BEFORE VALUE
                        3003X *       EXIT    (H,L) = (H,L) + (0,A)
                        3004X *               'C' SET IF OVERFLOW
                        3005X *       USES    F,H,L
                        3006X
                        3007X
      030.072           3008X $DADA   EQU     30072A          IN H17 ROM
      056.007           3009          XTEXT   MOVL                                    /WCZ080480/



                        3011X **      $MOVL - MOVE DATA
                        3012X *
                        3013X *       $MOVL MOVES A BLOCK OF BYTES TO A NEW MEMORY ADDRESS.
                        3014X *
                        3015X *       THIS MOVE ROUTINE WILL MOVE 0 TO 65535 BYTES.  IT SHOULD BE
                        3016X *       USED IN PLACE OF THE H17 ROM ROUTINE WHICH ONLY MOVES
                        3017X *       0 TO 32767 BYTES.
                        3018X *
                        3019X *       IF THE MOVE IS TO A LOWER ADDRESS, THE BYTES ARE MOVED FROM
                        3020X *       FIRST TO LAST.
                        3021X *
                        3022X *       IF THE MOVE IS TO A HIGHER ADDRESS, THE BYTES ARE MOVED FROM
                        3023X *       LAST TO FIRST.
                        3024X *
                        3025X *       THIS IS DONE SO THAT AN OVERLAPED MOVE WILL NOT 'RIPPLE'.
                        3026X *
                        3027X *       ENTRY   (BC) = COUNT
                        3028X *               (DE) = FROM
                        3029X *               (HL) = TO
                        3030X *       EXIT    MOVED
                        3031X *               (DE) = ADDRESS OF NEXT FROM BYTE
                        3032X *               (HL) = ADDRESS OF NEXT *TO* BYTE
                        3033X *               'C' CLEAR
                        3034X *       USES    ALL
                        3035X
                        3036X
      056.007           3037X $MOVL   EQU     *
                        3038X
                        3039X *       IF COUNT IS EQUAL TO 0, THEN RETURN IMMEDIATELY.
                        3040X
      056.007  170      3041X         MOV     A,B
      056.010  261      3042X         ORA     C
      056.011  310      3043X         RZ
                        3044X
                        3045X *       DETERMINE IF WE ARE MOVING TO LOWER OR HIGHER ADDRESS.
                        3046X
      056.012  175      3047X         MOV     A,L
      056.013  223      3048X         SUB     E
      056.014  174      3049X         MOV     A,H
      056.015  232      3050X         SBB     D
      056.016  332 044 056  3051X     JC      $MOVL2          MOVE IS TO A LOWER ADDRESS
```

```
                        3052X
                        3053X *     MOVE IS TO HIGHER ADDRESS, THEREFORE MOVE LAST TO FIRST.
                        3054X
056.021  353            3055X       XCHG                    ADJUST *FROM*
056.022  011            3056X       DAD     B                 ADDRESS TO
056.023  353            3057X       XCHG                        LAST BYTE + 1
056.024  325            3058X       PUSH    D               SAVE IT
                        3059X
056.025  011            3060X       DAD     B               ADJUST *TO* ADDRESS TO LAST BYTE + 1
056.026  345            3061X       PUSH    H               SAVE IT
                        3062X
056.027                 3063X $MOVL1 EQU     *
056.027  033            3064X       DCX     D               DECREMENT POINTERS
056.030  053            3065X       DCX     H
056.031  032            3066X       LDAX    D               MOVE BYTE
056.032  167            3067X       MOV     M,A
056.033  013            3068X       DCX     B               DECREMENT COUNT
056.034  170            3069X       MOV     A,B             Q. HAS COUNT
056.035  261            3070X       ORA     C                   GONE TO ZERO
056.036  302 027 056     3071X       JNZ     $MOVL1              BR IF NOT
                        3072X
056.041  341            3073X       POP     H               RETRIEVE *TO* LWA+1
056.042  321            3074X       POP     D               RETRIEVE *FROM* LWA+1
056.043  311            3075X       RET
                        3076X
                        3077X *     MOVE IS TO A LOWER ADDRESS, THEREFORE MOVE FIRST TO LAST.
                        3078X
056.044                 3079X $MOVL2 EQU     *
056.044  032            3080X       LDAX    D               MOVE BYTE
056.045  167            3081X       MOV     M,A
056.046  023            3082X       INX     D               INCREMENT POINTERS
056.047  043            3083X       INX     H
056.050  013            3084X       DCX     B               DECREMENT COUNT
056.051  170            3085X       MOV     A,B             Q. HAS COUNT
056.052  261            3086X       ORA     C                   GONE TO ZERO
056.053  302 044 056     3087X       JNZ     $MOVL2              BR IF NOT
                        3088X
056.056  311            3089X       RET
056.057                 3090        XTEXT   MU10


                        3092X **    $MU10 - MULTIPLY UNSIGNED 16 BIT QUANTITY BY 10.
                        3093X *
                        3094X *     (HL) = (DE)*10
                        3095X *
                        3096X *     ENTRY   (DE) = MULTIPLIER
                        3097X *     EXIT    'C' CLEAR IF OK
                        3098X *               (HL) = PRODUCT
                        3099X *               'C' SET IF ERROR
                        3100X *     USES    D,E,H,L,F
                        3101X
                        3102X
030.324                 3103X $MU10  EQU     30324A          IN H17 ROM
056.057                 3104        XTEXT   IBRA
```

```
                                3106X **      $TBRA - BRANCH RELATIVE THOUGH TABLE.
                                3107X *
                                3108X *       $TBRA USES THE SUPPLIED INDEX TO SELECT A BYTE FROM THE
                                3109X *       JUMP TABLE. THE CONTENTS OF THIS BYTE ARE ADDED TO THE
                                3110X *       ADDRESS OF THE BYTE, YEILDING THE PROCESSOR ADDRESS.
                                3111X *
                                3112X *       CALL    $TBRA
                                3113X *       DB      LAB1-*          INDEX = 0 FOR LAB1
                                3114X *       DB      LAB2-*          INDEX = 1 FOR LAB2
                                3115X *       DB      LABN-*          INDEX = N-1 FOR LABN
                                3116X *
                                3117X *       ENTRY   (A) = INDEX
                                3118X *               (RET) = TABLE FWA
                                3119X *       EXIT    TO COMPUTED ADDRESS
                                3120X *       USES    F,H,L
                                3121X
                                3122X
        031.076                 3123X $TBRA   EQU     31076A          IN H17 ROM
        056.057                 3124          XTEXT   FOPE


                                3126X **      $FOPEx - OPEN FILE BLOCK FOR I/O
                                3127X *
                                3128X *       $FOPEx IS CALLED BEFORE ANY I/O IS DONE VIA A
                                3129X *       FILE BLOCK. $FOPEx SETS UP THE FILE BLOCK, AND OPENS
                                3130X *       THE FILE VIA *HDOS*.
                                3131X *
                                3132X *       ENTRY   (DE) = ADDRESS OF DEFAULT BLOCK
                                3133X *               (HL) = ADDRESS OF FILE BLOCK
                                3134X *       EXIT    TO $FERROR IF ERROR
                                3135X *               TO CALLER IF OK
                                3136X *       USES    A,F,B,C,D,E
                                3137X
                                3138X
        056.057  315 104 056    3139X $FOPER  CALL    $FOPER.
        056.062  320            3140X         RNC
        056.063  303 237 060    3141X         JMP     $FERROR         IN ERROR
                                3142X
        056.066  315 107 056    3143X $FOPEW  CALL    $FOPEW.
        056.071  320            3144X         RNC
        056.072  303 237 060    3145X         JMP     $FERROR         IN ERROR
                                3146X
        056.075  315 112 056    3147X $FOPEU  CALL    $FOPEU.
        056.100  320            3148X         RNC
        056.101  303 237 060    3149X         JMP     $FERROR         IN ERROR
                                3150X
                                3151X
        056.104  076 002        3152X $FOPER. MVI     A,FT.OR         FILE TYPE OF OPEN FOR READ
        056.106  001            3153X         DB      001Q            LXI,B TO SKIP NEXT MVI
        056.107  076 004        3154X $FOPEW. MVI     A,FT.OW         OPEN FOR WRITE
        056.111  001            3155X         DB      001Q            LXI,B TO SKIP NEXT MIV
        056.112  076 006        3156X $FOPEU. MVI     A,FT.OR+FT.OW
                                3157X
                                3158X *       (A) = FILE FLAGS
```

```
                        3159X
056,114  345          3160X          PUSH    H               SAVE FILE BLOCK ADDRESS
056,115  365          3161X          PUSH    PSW             SAVE NEW FLAGS
000,000               3162X          ERRNZ   FB.CHA
056,116  106          3163X          MOV     B,M             (B) = CHANNEL NUMBER
056,117  305          3164X          PUSH    B               SAVE HANNEL NUMBER
000.000               3165X          ERRNZ   FB.FLG-FB.CHA-1
056,120  043          3166X          INX     H
056,121  117          3167X          MOV     C,A             (C) = NEW FILE FLAGS
056,122  176          3168X          MOV     A,M             (A) = CURRENT TYPE
056.123  247          3169X          ANA     A
056,124  171          3170X          MOV     A,C             (A) = NEW FLAGS TO BE SET
056.125  312 137 056  3171X          JZ      $FOPE1          NOT ALREADY OPEN
                      3172X
                      3173X *        ALREADY OPEN, SQUACK
                      3174X
056.130  301          3175X          POP     B               RESTORE (BC)
056,131  361          3176X          POP     PSW             DISCARD NEW FLAGS
056.132  341          3177X          POP     H               (HL) = FB ADDRESS
056,133  076 031      3178X          MVI     A,EC,FAO        FILE ALREADY OPEN
056.135  067          3179X          STC
056,136  311          3180X          RET
                      3181X
000.000               3182X          ERRNZ   FB.FWA-FB.FLG-1
056.137  043          3183X $FOPE1   INX     H               (HL) = $FB.FWA
056,140  116          3184X          MOV     C,M
056.141  043          3185X          INX     H
056,142  106          3186X          MOV     B,M             (BC) = FB.FWA
056.143  043          3187X          INX     H
000.000               3188X          ERRNZ   FB.PTR-FB.FWA-2
056.144  161          3189X          MOV     M,C             SET FB.PTR = FB.FWA
056,145  043          3190X          INX     H
056.146  160          3191X          MOV     M,B
056,147  043          3192X          INX     H
000.000               3193X          ERRNZ   FB.LIM-FB.PTR-2
056,150  161          3194X          MOV     M,C             SET FB.LIM = FB.FWA
056.151  043          3195X          INX     H
056,152  160          3196X          MOV     M,B
056.153  043          3197X          INX     H
000,000               3198X          ERRNZ   FB.NAM-FB.LIM-4
056.154  043          3199X          INX     H
056,155  043          3200X          INX     H               (HL) = $FB.NAM
                      3201X
                      3202X *        FILE BLOCK POINTERS SETUP, OPEN FILE
                      3203X
056,156  345          3204X          PUSH    H               SAVE NEW ADDRESS FOR NAME
056.157  041 210 056  3205X          LXI     H,$FOPEB
056,162  247          3206X          ANA     A                                       /78.10.GC/
056.163  312 172 056  3207X          JZ      $FOPE2
000,000               3208X          ERRNZ   .EXIT
056.166  315 350 055  3209X          CALL    $TBLS           FIND CODE
056,171  176          3210X          MOV     A,M
056.172  062 200 056  3211X $FOPE2   STA     $FOPEA          SET SYSCALL CODE
056.175  341          3212X          POP     H               (HL) = $FB.NAM
056.176  361          3213X          POP     PSW             (A) = CHANNEL NUMBER
056,177  377 000      3214X          DB      SYSCALL+.EXIT
```

```
056.200                 3215X $FOPEA  EQU     *-1             SYSCALL CODE
056.201  321           3216X         POP     D               (D) = NEW FLAG
056.202  341           3217X         POP     H               (HL) = FILE BLOCK ADDRESS
056.203  330           3218X         RC                      EXIT IF ERROR
056.204  043           3219X         INX     H
000.000                 3220X         ERRNZ   FB.FLG-1
056.205  162           3221X         MOV     M,D             SET NEW FLAGS
056.206  053           3222X         DCX     H               RESTORE (HL)
056.207  311           3223X         RET
                        3224X
056.210  002 042       3225X $FOPEB  DB      FT.OR,.OPENR    TABLE OF SYSCALL CODES
056.212  004 043       3226X         DB      FT.OW,.OPENW
056.214  006 044       3227X         DB      FT.OR+FT.OW,.OPENU
056.216  000           3228X         DB      0               SHOULD NOT OCCUR
056.217                 3229          XTEXT   FCLO


                        3231X **      $FCLO - CLOSE FILE BLOCK.
                        3232X *
                        3233X *       $FCLO IS CALLED TO TERMINATE PROCESSING THROUGH A FILE
                        3234X *       BLOCK.
                        3235X *
                        3236X *       ENTRY   (HL) = FILE BLOCK ADDRESS
                        3237X *       EXIT    TO $FERROR IF ERROR
                        3238X *                TO CALLER IF OK
                        3239X *       USES    A,F,B,C,D,E
                        3240X
                        3241X
056.217  315 226 056    3242X $FCLO   CALL    $FCLO.
056.222  320           3243X         RNC                     NO ERROR
056.223  303 237 060    3244X         JMP     $FERROR
                        3245X
056.226  345           3246X $FCLO.  PUSH    H               SAVE FILE BLOCK ADDRESS
000.000                 3247X         ERRNZ   FB.FLG-1
056.227  043           3248X         INX     H               (HL) = $FB.FLG
056.230  176           3249X         MOV     A,M
056.231  066 000       3250X         MVI     M,0             CLEAR FLAG
056.233  247           3251X         ANA     A
056.234  312 322 056    3252X         JZ      $FCLO4          FILE NOT OPEN
056.237  346 004       3253X         ANI     FT.OW
056.241  312 314 056    3254X         JZ      $FCLO3          NO WRITING, NO FLUSHING NEEDED
                        3255X
                        3256X *       WAS OPEN FOR WRITE, SEE IF NEED FLUSH THE LAST SECTOR
                        3257X
056.244  315 234 030    3258X         CALL    $INDL
056.247  003 000       3259X         DW      FB.PTR-FB.FLG
056.251  325           3260X         PUSH    D               SAVE (FB.PTR)
056.252  315 234 030    3261X         CALL    $INDL           (DE) = (FB.FWA)
056.255  001 000       3262X         DW      FB.FWA-FB.FLG
056.257  341           3263X         POP     H               (HL) = (FB.PTR)
056.260  175           3264X         MOV     A,L
056.261  223           3265X         SUB     E
056.262  117           3266X         MOV     C,A
056.263  174           3267X         MOV     A,H
```

```
056.264  232            3268X          SBB     D
056.265  107            3269X          MOV     B,A             (BC) = AMOUNT IN BLOCK
056.266  261            3270X          ORA     C
056.267  312 314 056    3271X          JZ      $FCLO3          NONE TO FLUSH
                        3272X
                        3273X *        NEED TO FLUSH BUFFER
                        3274X *
                        3275X *        (BC) = DATA AMOUNT
                        3276X *        (DE) = FWA
                        3277X *        (HL) = LWA+1
                        3278X
056.272  171            3279X          MOV     A,C
056.273  247            3280X          ANA     A
056.274  312 307 056    3281X          JZ      $FCLO2          DONT HAVE PARTIAL SECTOR
                        3282X
                        3283X *        ZERO FILL PARTIAL SECTOR
                        3284X
056.277  066 000        3285X $FCLO1   MVI     M,0
056.301  043            3286X          INX     H
056.302  014            3287X          INR     C
056.303  302 277 056    3288X          JNZ     $FCLO1
056.306  004            3289X          INR     B               COUNT ANOTHER FULL SECTOR
056.307  341            3290X $FCLO2   POP     H               (HL) = FB FWA
056.310  176            3291X          MOV     A,M             (A) = CHANNEL NUMBER
000.000                 3292X          ERRNZ   FB.CHA
056.311  345            3293X          PUSH    H
056.312  377 005        3294X          DB      SYSCALL,.WRITE          FLUSH
                        3295X
                        3296X *        READY TO CLOSE FILE.
                        3297X *
                        3298X *        'C' SET IF ERROR
                        3299X *        (A) = ERROR CODE
                        3300X
056.314  341            3301X $FCLO3   POP     H               (HL) = FILE BLOCK ADDRESS
056.315  330            3302X          RC                      ERROR
000.000                 3303X          ERRNZ   FB.CHA
056.316  176            3304X          MOV     A,M             (A) = CHANNEL NUMBER
056.317  345            3305X          PUSH    H
056.320  377 046        3306X          DB      SYSCALL,.CLOSE  CLOSE CHANNEL
056.322  341            3307X $FCLO4   POP     H               (HL) = FILE BLOCK ADDRESS
056.323  311            3308X          RET
056.324                 3309           XTEXT   FREAL


                        3311X **       $FREAL - READ BYTES FROM FILE BUFFER.
                        3312X *
                        3313X *        $FREAL IS CALLED TO READ A NUMBER OF BYTES FROM A FILE BUFFER.
                        3314X *
                        3315X *        ENTRY   (BC) = BYTE COUNT
                        3316X *                (DE) = FWA FOR BYTES
                        3317X *                (HL) = ADDRESS OF FILE BUFFER
                        3318X *        EXIT    TO *ERROR* IF ERROR
                        3319X *                TO CALLER IF OK
                        3320X *                (BC) = UNREAD BYTE COUNT (ONLY IF EOF)
```

```
                                3321X *                 (DE) = ADDRESS OF FIRST UNUSED BYTE
                                3322X *                 'C' SET IF EOF DURING READ
                                3323X *         USES    A,F,B,C,D,E
                                3324X
                                3325X
056,324 315 337 056             3326X $FREAL    CALL    $FREAL,
056.327 320                     3327X           RNC                     RETURN IF OK
056,330 376 001                 3328X           CPI     EC.EOF
056.332 302 237 060             3329X           JNE     $FERROR         ERROR IS NOT EOF
056,335 067                     3330X           STC
056.336 311                     3331X           RET                     ERROR IS SIMPLY EOF
                                3332X
                                3333X
056,337                         3334X $FREAL.   EQU     *
056.337 013                     3335X           DCX     B               (BC) = COUNT NOT ENCLUDING 00 BYTE
056,340 257                     3336X           XRA     A
056.341 062 236 060             3337X           STA     EOFFLG          CLEAR EOF FLAG
056,344 345                     3338X           PUSH    H
056.345 315 062 060             3339X           CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                                3340X
                                3341X *         COPY DATA FROM BUFFER TO TARGET
                                3342X
056.350 325                     3343X $REAL2    PUSH    D               SAVE TARGET ADDRESS
056,351 072 225 060             3344X           LDA     T.FLG
056.354 346 002                 3345X           ANI     FT.OR
056,356 076 011                 3346X           MVI     A,EC.FNO
056.360 067                     3347X           STC                     ASSUME FILE NOT OPEN
056,361 312 115 057             3348X           JZ      $REAL8          ERROR
056.364 170                     3349X           MOV     A,B
056,365 261                     3350X           ORA     C
056.366 312 115 057             3351X           JZ      $REAL8          ALL DONE
                                3352X
                                3353X *         COMPUTE MIN( DATA IN BUFFER, DATA REQUESTED)
                                3354X
056.371 052 230 060             3355X $REAL3    LHLD    T.PTR
056,374 353                     3356X           XCHG                    (DE) = (FB.PTR) = ADDRESS OF DATA
056.375 052 232 060             3357X           LHLD    T.LIM           (HL) = LIMIT ADDRESS
057,000 175                     3358X           MOV     A,L
057.001 223                     3359X           SUB     E
057,002 157                     3360X           MOV     L,A
057.003 174                     3361X           MOV     A,H
057,004 232                     3362X           SBB     D
057.005 147                     3363X           MOV     H,A             (HL) = NUMBER OF BYTES IN BUFFER
057,006 171                     3364X           MOV     A,C
057.007 225                     3365X           SUB     L               COMPARE TO REQUESTED COUNT
057,010 170                     3366X           MOV     A,B
057.011 234                     3367X           SBB     H
057,012 322 017 057             3368X           JNC     $REAL4          LESS THAN REQUESTED COUNT
057.015 140                     3369X           MOV     H,B
057.016 151                     3370X           MOV     L,C             DONT TRANSFER MORE THAN LIMIT
057.017 174                     3371X $REAL4    MOV     A,H
057,020 265                     3372X           ORA     L
057.021 302 035 057             3373X           JNZ     $REAL6          SOME IN BUFFER
                                3374X
                                3375X *         BUFFER IS EMPTY. RE-FILL IT
                                3376X
```

```
057.024  315 142 060  3377X          CALL    $FFB            FILL FILE BUFFER
057.027  332 115 057  3378X          JC      $REAL8          ERROR CONDITION
057.032  303 371 056  3379X          JMP     $REAL3          COUNT THE DATA
                      3380X
                      3381X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                      3382X *
                      3383X *        (BC) = LIMIT COUNT
                      3384X *        (DE) = FROM
                      3385X *        (HL) = COUNT
                      3386X *        ((SP)) = TO
                      3387X
057.035  171          3388X $REAL6  MOV     A,C
057.036  225          3389X         SUB     L
057.037  117          3390X         MOV     C,A
057.040  170          3391X         MOV     A,B
057.041  234          3392X         SBB     H
057.042  107          3393X         MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
057.043  305          3394X         PUSH    B
057.044  343          3395X         XTHL                    (HL) = REMAINING REQUEST COUNT
057.045  301          3396X         POP     B               (BC) = COUNT FOR THIS COPY
057.046  343          3397X         XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
057.047  032          3398X $REAL7  LDAX    D
057.050  023          3399X         INX     D
057.051  167          3400X         MOV     M,A
057.052  043          3401X         INX     H
057.053  247          3402X         ANA     A               SEE IF 00 BYTE
057.054  302 063 057  3403X         JNZ     $REL7.3         NOT 00
                      3404X
                      3405X *        IS 00 BYTE. IGNORE IT
                      3406X
057.057  343          3407X         XTHL
057.060  043          3408X         INX     H               ADD ONE TO UNREQUITED COUNT
057.061  343          3409X         XTHL
057.062  053          3410X         DCX     H               BACKSPACE OVER CHARACTER
057.063  013          3411X $REL7.3 DCX     B
057.064  376 012      3412X         CPI     NL
057.066  312 106 057  3413X         JE      $REL7.5         IS END OF LINE
057.071  170          3414X         MOV     A,B
057.072  261          3415X         ORA     C
057.073  302 047 057  3416X         JNZ     $REAL7          MORE TO GO
057.076  353          3417X         XCHG
057.077  042 230 060  3418X         SHLD    T.PTR           UPDATE POINTER
057.102  301          3419X         POP     B               (BC) = REMAINING COUNT
057.103  303 350 056  3420X         JMP     $REAL2          SEE IF MORE IN BUFFER
                      3421X
                      3422X *        END OF CODED LINE
                      3423X
057.106  353          3424X $REL7.5 XCHG
057.107  033          3425X         DCX     D               BACK OVER NL CHARACTER
057.110  042 230 060  3426X         SHLD    T.PTR           UPDATE POINTER
057.113  301          3427X         POP     B               (BC) = REMAINING COUNT
057.114  325          3428X         PUSH    D               SAVE TARGET LWA
                      3429X
                      3430X *        READ COMPLETE.
                      3431X *
                      3432X *        (PSW) = COMPLETION FLAGS
```

```
                        3433X
057.115  321           3434X $REAL8  POP     D               RESTORE TARGET ADDRESS
057.116  365           3435X         PUSH    PSW             SAVE RETURN CODE
057.117  257           3436X         XRA     A
057.120  022           3437X         STAX    D               FLAG END OF LINE
057.121  361           3438X         POP     PSW             RESTORE RESULT FLAGS
057.122  023           3439X         INX     D               POINT TO NEXT FREE
057.123  341           3440X $REAL9  POP     H
057.124  303 110 060   3441X         JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK; EXIT
057.127               3442          XTEXT   FWRIL


                        3444X **      $FWRIL - WRITE LINE FROM FILE BUFFER.
                        3445X *
                        3446X *       $FWRIL IS CALLED TO WRITE A LINE TO A FILE BUFFER.
                        3447X *
                        3448X *       ENTRY   (DE) = FWA FOR BYTES
                        3449X *               (HL) = ADDRESS OF FILE BUFFER
                        3450X *       EXIT    TO *FERROR* IF ERROR
                        3451X *               TO CALLER IF OK
                        3452X *                       (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                        3453X *       USES    A,F,B,C,D,E
                        3454X
                        3455X
057.127  315 136 057   3456X $FWRIL  CALL    $FWRIL;
057.132  320           3457X         RNC                     RETURN IF OK
057.133  303 237 060   3458X         JMP     $FERROR         ERROR
                        3459X
                        3460X *       SCAN FOR END OF LINE
                        3461X
057.136  325           3462X $FWRIL; PUSH    D               SAVE LINE POINTER
057.137  001 377 377   3463X         LXI     B,-1            (BC) = COUNT
057.142  032           3464X $FWRIL1 LDAX    D
057.143  023           3465X         INX     D
057.144  003           3466X         INX     B
057.145  247           3467X         ANA     A
057.146  302 142 057   3468X         JNZ     $FWRIL1         MORE TO GO
057.151  321           3469X         POP     D
057.152  315 174 057   3470X         CALL    $FWRIB          WRITE BYTES
057.155  330           3471X         RC                      ERROR
                        3472X
                        3473X *       WRITE 'NL' CHARACTER
                        3474X
057.156  023           3475X         INX     D
057.157  325           3476X         PUSH    D
057.160  001 001 000   3477X         LXI     B,1
057.163  021 173 057   3478X         LXI     D,$FWRILA
057.166  315 174 057   3479X         CALL    $FWRIB
057.171  321           3480X         POP     D
057.172  311           3481X         RET
                        3482X
057.173  012           3483X $FWRILA DB      NL
057.174               3484          XTEXT   FWRIB
```

```
                                3486X **      $FWRIB - WRITE BYTES FROM FILE BUFFER.
                                3487X *
                                3488X *       $FWRIB IS CALLED TO WRITE A NUMBER OF BYTES FROM A FILE BUFFER.
                                3489X *
                                3490X *       ENTRY   (BC) = BYTE COUNT
                                3491X *               (DE) = FWA FOR BYTES
                                3492X *               (HL) = ADDRESS OF FILE BUFFER
                                3493X *       EXIT    TO *FERROR* IF ERROR
                                3494X *               TO CALLER IF OK
                                3495X *                (DE) = ADDRESS OF FIRST UNWRITTEN BYTE
                                3496X *       USES    A,F,B,C,D,E
                                3497X
                                3498X
057.174  315 203 057           3499X $FWRIB  CALL    $FWRIB.
057.177  320                   3500X         RNC                     RETURN IF OK
057.200  303 237 060           3501X         JMP     $FERROR         ERROR
                                3502X
                                3503X
057.203                         3504X $FWRIB. EQU     *
057.203  345                   3505X         PUSH    H
057.204  315 062 060           3506X         CALL    CBT             COPY BUFFER POINTERS TO TEMP CELLS
                                3507X
                                3508X *       COPY DATA FROM USER AREA TO BUFFER
                                3509X
057.207  325                   3510X $WRIB2  PUSH    D               SAVE AREA ADDRESS
057.210  072 225 060           3511X         LDA     T.FLG
057.213  346 004               3512X         ANI     FT.OW           SEE IF OPEN FOR WRITE
057.215  312 351 057           3513X         JZ      $WRIB8          FILE NOT OPEN FOR WRITE
057.220  170                   3514X         MOV     A,B
057.221  261                   3515X         ORA     C
057.222  312 351 057           3516X         JZ      $WRIB8          ALL DONE
                                3517X
                                3518X *       COMPUTE MIN( ROOM IN BUFFER, WRITE COUNT REQUESTED)
                                3519X
057.225  052 230 060           3520X $WRIB3  LHLD    T.PTR
057.230  353                   3521X         XCHG                    (DE) = (FB,PTR) = ADDRESS OF ROOM
057.231  052 234 060           3522X         LHLD    T.LWA           (HL) = LIMIT ADDRESS
057.234  175                   3523X         MOV     A,L
057.235  223                   3524X         SUB     E
057.236  157                   3525X         MOV     L,A
057.237  174                   3526X         MOV     A,H
057.240  232                   3527X         SBB     D
057.241  147                   3528X         MOV     H,A             (HL) = BYTES OF ROOM IN BUFFER
057.242  171                   3529X         MOV     A,C             COMPARE REQUESTED COUNT TO BUFFER ROOM
057.243  225                   3530X         SUB     L
057.244  170                   3531X         MOV     A,B
057.245  234                   3532X         SBB     H
057.246  322 253 057           3533X         JNC     $WRIB4          MORE REQUESTED THEN ROOM
057.251  140                   3534X         MOV     H,B
057.252  151                   3535X         MOV     L,C             USE REQUESTED COUNT
057.253  174                   3536X $WRIB4  MOV     A,H
057.254  265                   3537X         ORA     L
057.255  302 315 057           3538X         JNZ     $WRIB6          SOME ROOM IN BUFFER
                                3539X
                                3540X *       BUFFER IS FULL. EMPTY IT
                                3541X
```

```
057.260  305            3542X          PUSH    B               SAVE COUNT
057.261  052 226 060    3543X          LHLD    T.FWA
057.264  042 230 060    3544X          SHLD    T.PTR           CLEAR REMOVAL POINTER
057.267  353            3545X          XCHG
057.270  052 234 060    3546X          LHLD    T.LWA
057.273  175            3547X          MOV     A,L
057.274  223            3548X          SUB     E
057.275  117            3549X          MOV     C,A
057.276  174            3550X          MOV     A,H
057.277  232            3551X          SBB     D
057.300  107            3552X          MOV     B,A             (BC) = DATA IN BUFFER
057.301  072 224 060    3553X          LDA     T.CHA
057.304  377 005        3554X          DB      SYSCALL,.WRITE  WRITE BUFFER
057.306  301            3555X          POP     B               (BC) = DESIRED COUNT
057.307  322 225 057    3556X          JNC     $WRIB3          GOT THE DATA
                        3557X
                        3558X *        ERROR ON WRITE.
                        3559X
057.312  303 351 057    3560X          JMP     $WRIB8          HAVE ERROR
                        3561X
                        3562X *        GOT THE DATA. MOVE IT FROM BUFFER TO TARGET
                        3563X *
                        3564X *        (BC) = REQUEST COUNT
                        3565X *        (DE) = TO
                        3566X *        (HL) = COUNT
                        3567X *        ((SP)) = FROM
                        3568X
057.315  171            3569X $WRIB6   MOV     A,C
057.316  225            3570X          SUB     L
057.317  117            3571X          MOV     C,A
057.320  170            3572X          MOV     A,B
057.321  234            3573X          SBB     H
057.322  107            3574X          MOV     B,A             REMOVE BYTES ABOUT TO BE MOVED FROM REQUEST COUNT
057.323  305            3575X          PUSH    B
057.324  343            3576X          XTHL                    (HL) = REMAINING REQUEST COUNT
057.325  301            3577X          POP     B               (BC) = COUNT FOR THIS COPY
057.326  343            3578X          XTHL                    (HL) = TARGET ADDR, ((SP)) = REMAINING REQ. COUNT
057.327  176            3579X $WRIB7   MOV     A,M
057.330  022            3580X          STAX    D
057.331  023            3581X          INX     D
057.332  043            3582X          INX     H
057.333  013            3583X          DCX     B
057.334  170            3584X          MOV     A,B
057.335  261            3585X          ORA     C
057.336  302 327 057    3586X          JNZ     $WRIB7          MORE TO GO
057.341  353            3587X          XCHG
057.342  042 230 060    3588X          SHLD    T.PTR           UPDATE POINTER
057.345  301            3589X          POP     B               (BC) = REMAINING COUNT
057.346  303 207 057    3590X          JMP     $WRIB2          SEE IF MORE IN BUFFER
                        3591X
                        3592X *        WRITE COMPLETE.
                        3593X *
                        3594X *        (PSW) = COMPLETION FLAGS
                        3595X
057.351  321            3596X $WRIB8   POP     D               RESTORE TARGET ADDRESS
057.352  341            3597X          POP     H
```

```
057.353  303 110 060  3598X          JMP     CTB             COPY TEMP POINTERS BACK TO BLOCK, EXIT


                      3600X **       $FWBRK  -  BREAKOUTPUT                                  /80.02.GC/
                      3601X *
                      3602X *        $FWBRK empties the specified buffer by filling it with NULLs
                      3603X *        and then writing it.  Note this is used to insure that block
                      3604X *        mode I/O is output if it is not really a serial device (eg.
                      3605X *        writing to AT: from  *EDIT*.
                      3606X *
                      3607X *
                      3608X *        ENTRY:  HL      =  FILE BLOCK POINTER
                      3609X *
                      3610X *        EXIT:   HL      =  FILE BLOCK POINTER
                      3611X *                TO  $FERROR IF ERROR
                      3612X *
                      3613X *        USES:   PSW,BC,DE
                      3614X *
                      3615X
057.356  315 365 057  3616X $FWBRK   CALL    $FWBRK,
057.361  320          3617X          RNC                     NO ERROR
                      3618X
057.362  303 237 060  3619X          JMP     $FERROR
                      3620X
057.365  345          3621X $FWBRK.  PUSH    H
057.366  315 062 060  3622X          CALL    CBT             COPY BUFFER TO TEMPORARY
057.371  315 001 060  3623X          CALL    $FWBRK1
057.374  341          3624X          POP     H
057.375  315 110 060  3625X          CALL    CTB             COPY TEMPORARY TO BUFFER
060.000  311          3626X          RET
                      3627X
060.001  052 234 060  3628X $FWBRK1  LHLD    T.LWA
060.004  353          3629X          XCHG                    DE = BUFFER LWA
060.005  052 230 060  3630X          LHLD    T.PTR           HL = BUFFER PTR
060.010  173          3631X          MOV     A,E
060.011  225          3632X          SUB     L
060.012  117          3633X          MOV     C,A
060.013  172          3634X          MOV     A,D
060.014  234          3635X          SBB     H
060.015  107          3636X          MOV     B,A             BC = DE - HL
060.016  261          3637X          ORA     C
060.017  310          3638X          RZ                      THE BUFFER IS ALREADY FLUSHED
                      3639X
                      3640X *        FILL THE BUFFER WITH NULLS
                      3641X
060.020  170          3642X FWBRK2   MOV     A,B
060.021  261          3643X          ORA     C
060.022  312 034 060  3644X          JZ      FWBRK3          NO MORE LEFT TO FILL
                      3645X
060.025  066 000      3646X          MVI     M,0
060.027  043          3647X          INX     H
060.030  013          3648X          DCX     B
060.031  303 020 060  3649X          JMP     FWBRK2
                      3650X
```

```
060.034  052 226 060  3651X FWBRK3  LHLD    T.FWA
060.037  042 230 060  3652X         SHLD    T.PTR
060.042  353          3653X         XCHG                DE = BUFFER FWA
060.043  052 234 060  3654X         LHLD    T.LWA       HL = BUFFER LWA
060.046  175          3655X         MOV     A,L
060.047  223          3656X         SUB     E
060.050  117          3657X         MOV     C,A
060.051  174          3658X         MOV     A,H
060.052  232          3659X         SBB     D
060.053  107          3660X         MOV     B,A         BC = HL - DE  ( BC = COUNT )
060.054  072 224 060  3661X         LDA     T.CHA
060.057  377 005      3662X         DB      SYSCALL+.WRITE
060.061  311          3663X         RET
060.062               3664          XTEXT   FUTIL


                      3666X **      $FUTIL - UTILITY ROUTINES FOR FILE BLOCK ROUTINES.
                      3667X
                      3668X **      CBT - COPY BLOCK POINTERS TO TEMP CELLS.
                      3669X *
                      3670X *       ENTRY   (HL) = FILE BLOK FWA
                      3671X *       EXIT    NONE
                      3672X *       USES    A,F,H,L
                      3673X
060.062  325          3674X CBT     PUSH    D
060.063  305          3675X         PUSH    B           SAVE REGISTERS
000.000               3676X         ERRNZ   TLEN-10     ASSUME 10 BYTES TO MOVE
060.064  021 224 060  3677X         LXI     D,T.CHA     (DE) = TARGET FOR MOVE
060.067  006 005      3678X         MVI     B,10/2
060.071  176          3679X CBT1    MOV     A,M         COPY FILE BUFFER INTO WORK AREA
060.072  022          3680X         STAX    D
060.073  043          3681X         INX     H
060.074  023          3682X         INX     D
060.075  176          3683X         MOV     A,M
060.076  022          3684X         STAX    D
060.077  043          3685X         INX     H
060.100  023          3686X         INX     D
060.101  005          3687X         DCR     B
060.102  302 071 060  3688X         JNZ     CBT1        MORE TO GO
060.105  301          3689X         POP     B
060.106  321          3690X         POP     D           (DE) = DATA TARGET ADDRESS
060.107  311          3691X         RET
                      3692X
                      3693X
                      3694X **      CTB - COPY TEMP CELLS BACK TO FILE BLOCK.
                      3695X *
                      3696X *       ENTRY   (HL) = FILE BLOCK ADDRESS
                      3697X *       EXIT    NONE
                      3698X *       USES    NONE
                      3699X
060.110  365          3700X CTB     PUSH    PSW
060.111  325          3701X         PUSH    D
060.112  305          3702X         PUSH    B
060.113  345          3703X         PUSH    H           SAVE REGISTERS
```

```
060.114  006 004      3704X          MVI     B,8/2
060.116  021 224 060  3705X          LXI     D,T.CHA
060.121  032          3706X CTB1     LDAX    D
060.122  167          3707X          MOV     M,A
060.123  023          3708X          INX     D
060.124  043          3709X          INX     H
060.125  032          3710X          LDAX    D
060.126  167          3711X          MOV     M,A
060.127  023          3712X          INX     D
060.130  043          3713X          INX     H
060.131  005          3714X          DCR     B
060.132  302 121 060  3715X          JNZ     CTB1            RESTORE FILE BUFFER VALUES
060.135  341          3716X          POP     H
060.136  301          3717X          POP     B
060.137  321          3718X          POP     D
060.140  361          3719X          POP     PSW
060.141  311          3720X          RET


                      3722X **       $FFB - FILE FILE BUFFER,
                      3723X *
                      3724X *        $FFB FILLS THE FILE BUFFER BY READING FROM THE FILE,
                      3725X *
                      3726X *        ENTRY   NONE
                      3727X *        EXIT    'C' SET IF READ INCOMPLETE
                      3728X *                (A) = ERROR CODE
                      3729X *                'C' CLEAR IF READ COMPLETEE
                      3730X *                DATA IN BUFFER
                      3731X *        USES    A,F,D,E,H,L
                      3732X
                      3733X
060.142  072 236 060  3734X $FFB     LDA     EOFFLG
060.145  037          3735X          RAR
060.146  330          3736X          RC                      EOF
                      3737X
                      3738X *        CAN READ MORE, DO SO
                      3739X
060.147  305          3740X          PUSH    B               SAVE COUNT
060.150  052 226 060  3741X          LHLD    T.FWA
060.153  042 230 060  3742X          SHLD    T.PTR           CLEAR REMOVAL POINTER
060.156  353          3743X          XCHG
060.157  052 234 060  3744X          LHLD    T.LWA
060.162  042 232 060  3745X          SHLD    T.LIM           SET DATA LIMIT
060.165  175          3746X          MOV     A,L
060.166  223          3747X          SUB     E
060.167  117          3748X          MOV     C,A
060.170  174          3749X          MOV     A,H
060.171  232          3750X          SBB     D
060.172  107          3751X          MOV     B,A             (BC) = ROOM IN BUFFER
060.173  072 224 060  3752X          LDA     T.CHA
060.176  377 004      3753X          DB      SYSCALL,.READ   READ BUFFER
060.200  120          3754X          MOV     D,B             (D) = SECTORS UNREAD
060.201  301          3755X          POP     B               (BC) = DESIRED COUNT
060.202  320          3756X          RNC                     GOT THE DATA
```

```
                                3757X
                                3758X *       ERROR ON READ, SEE IF EOF
                                3759X
060.203  027                    3760X         RAL
060.204  062 236 060           3761X         STA     EOFFLG          SET EOF, WE HOPE
060.207  376 003               3762X         CPI     EC.EOF*2+1
060.211  037                   3763X         RAR
060.212  300                   3764X         RNE                     IS NOT EOF, RETURN NOW!
060.213  072 233 060           3765X         LDA     T.LIM+1
060.216  222                   3766X         SUB     D
060.217  062 233 060           3767X         STA     T.LIM+1         SET AMOUNT OF DATA WE DID GET
060.222  247                   3768X         ANA     A
060.223  311                   3769X         RET                     EXIT WITH DATA
                                3770X
                                3771X
                                3772X **      TEMP CELLS TO HOLD FILE BLOCK POINTERS DURING I/O
                                3773X
000.000                         3774X         ERRNZ   FB.CHA
060.224  000                   3775X T.CHA   DB      0               CHANNEL NUMBER
000.000                         3776X         ERRNZ   *-T.CHA-FB.FLG
060.225  000                   3777X T.FLG   DB      0               FLAG BYTE
000.000                         3778X         ERRNZ   *-T.CHA-FB.FWA
060.226  000 000               3779X T.FWA   DW      0
000.000                         3780X         ERRNZ   *-T.CHA-FB.PTR
060.230  000 000               3781X T.PTR   DW      0
000.000                         3782X         ERRNZ   *-T.CHA-FB.LIM
060.232  000 000               3783X T.LIM   DW      0
000.000                         3784X         ERRNZ   *-T.CHA-FB.LWA
060.234  000 000               3785X T.LWA   DW      0
000.012                         3786X TLEN    EQU     *-T.CHA         LENGTH OF TEMP CELLS
                                3787X
060.236  000                   3788X EOFFLG  DB      0
060.237                         3789          XTEXT   FERROR



                                3791X **      $FERROR - PROCESS FILE ERRORS.
                                3792X *
                                3793X *       $FERROR IS CALLED TO COMPLAIN ABOUT AN ERROR ENCOUNTERED
                                3794X *       WHEN PROCESSING FILES.
                                3795X *
                                3796X *       ENTRY   (A) = ERROR CODE
                                3797X *               (HL) = ADDRESS OF FILE NAME - FB.NAM
                                3798X *       EXIT    TO RESTART
                                3799X *       USES    ALL
                                3800X
                                3801X
060.237  365                   3802X $FERROR PUSH    PSW             SAVE CODE
060.240  315 136 031           3803X         CALL    $TYPTX
060.243  012 007 105           3804X         DB      NL,BELL,'ERROR ON FILE',' '+200Q
060.263  021 012 000           3805X         LXI     D,FB.NAM
060.266  031                   3806X         DAD     D
                                3807X
                                3808X *       PRINT FILE NAME
                                3809X
```

```
060.267  176            3810X $FERR1  MOV     A,M
060.270  043            3811X         INX     H               ADVANCE MESSAGE
060.271  247            3812X         ANA     A
060.272  312 303 060    3813X         JZ      $FERR2
060.275  315 345 055    3814X         CALL    $WCHAR
060.300  303 267 060    3815X         JMP     $FERR1
                        3816X
                        3817X *       TYPE ERROR MESSAGE
                        3818X
060.303  315 136 031    3819X $FERR2  CALL    $TYPTX
060.306  040 055 240    3820X         DB      ' -',' '+200Q
060.311  046 012        3821X         MVI     H,NL
060.313  361            3822X         POP     PSW             (A) = CODE
060.314  377 057        3823X         DB      SYSCALL,.ERROR
060.316  303 200 042    3824X         JMP     RESTART         EXIT
060.321                 3825          XTEXT   TJMP


                        3827X **      $TJMP - TABLE JUMP.
                        3828X *
                        3829X *       USAGE
                        3830X *
                        3831X *       CALL      $TJMP           (A) = INDEX
                        3832X *       DW        ADDR1
                        3833X *       .           .
                        3834X *       .           .
                        3835X *       .           .
                        3836X *       DW        ADDRN
                        3837X *
                        3838X *       ENTRY     (A) = INDEX
                        3839X *       EXIT      TO PROCESSOR
                        3840X *                 (A) = INDEX*2
                        3841X *       USES      NONE.
                        3842X
                        3843X
031.061                 3844X $TJMP   EQU     31061A          IN H17 ROM, (A) = INDEX*2
                        3845X
031.062                 3846X $TJMP.  EQU     31062A          IN H17 ROM
060.321                 3847          XTEXT   TYPTX


                        3849X **      $TYPTX - TYPE TEXT.
                        3850X *
                        3851X *       $TYPTX IS CALLED TO TYPE A BLOCK OF TEXT ON THE SYSTEM CONSOLE.
                        3852X *
                        3853X *       IMBEDDED ZERO BYTES INDICATE A CARRIAGE RETURN LINE FEED,
                        3854X *       A BYTE WITH THE 200Q BIT SET IS THE LAST BYTE IN THE MESSAGE.
                        3855X *
                        3856X *       ENTRY   (RET) = TEXT
                        3857X *       EXIT    TO (RET+LENGTH)
                        3858X *       USES    A,F
                        3859X
```

```
                    3860X
      031.136       3861X $TYPTX  EQU     31136A          IN H17 ROM
                    3862X
      031.144       3863X $TYPTX. EQU     31144A          IN H17 ROM
```

```
                                3866  **      CMDTAB - COMMAND TABLE.
                                3867  *
                                3868
        060.321                 3869  CMDTAB  EQU     *
        060.321  000           3870          DB      0               DUMY FIRST COMMAND
        060.322  120 122 111   3871          DB      'PRINT',0
        060.330  104 105 114   3872          DB      'DELETE',0
        060.337  105 104 111   3873          DB      'EDIT',0
        060.344  122 105 120   3874          DB      'REPLACE',0
        060.354  127 122 111   3875          DB      'WRITE',0
        060.362                 3876  CMDTAB. EQU     *               THESE COMMANDS ALLOWED WITH NO TEXT
        060.362  130 120 122   3877          DB      'XPRINT',0      IS DUMY COMMAND FOR 2ND GROUP, REAL COMMAND FOR 1ST
        060.371  111 116 123   3878          DB      'INSERT',0
        061.000  122 105 101   3879          DB      'READ',0
        061.005  102 114 111   3880          DB      'BLITZ',0
        061.013  106 114 125   3881          DB      'FLUSH',0
        061.021  116 105 130   3882          DB      'NEXT',0
        061.026  123 105 101   3883          DB      'SEARCH',0
        061.035  116 105 127   3884          DB      'NEWIN',0
        061.043  116 105 127   3885          DB      'NEWOUT',0
        061.052  130 117 125   3886          DB      'XOUT',0
        061.057  125 123 105   3887          DB      'USE',0
        061.063  102 131 105   3888          DB      'BYE',0
        061.067  000           3889          DB      0
```

```
061.070                 3892  PATCH   DS      64
```

```
                                3896  **      LINE POINTERS INTO TEXT PAGE.
                                3897
061.170  000 000                3898  FILPTR  DW      0               ADDRESS OF 1ST LINE IN BUFFER
061.172  000 000                3899  LALPTR  DW      0               ADDRESS OF END OFLAST LINE IN BUFFER +1
061.174  000 000                3900  CRFPTR  DW      0               COMMAND RANGE 1ST LINE POINTER
061.176  000 000                3901  CRLPTR  DW      0               COMMAND RANGE LAST LINE POINTER
061.200  000 000                3902  WRKPTR  DW      0               COMMAND RANGE WORKING POINTER
061.202  000 000                3903  PCFPTR  DW      0               PREVIOUS COMMAND 'FIRST' POINTER
061.204  000 000                3904  PCLPTR  DW      0               PREVIOUS COMMAND 'LAST' POINTER
                                3905
061.206  000                    3906  CCFLG   DB      0               <>0 IF CTL-C DISABLED
061.207  000                    3907  CCPEND  DB      0               <>0 IF CTL-C HIT DURING DISABLED PERIOD
                                3908
061.210  000 000                3909  BUFMAX  DW      0               MAX ADDRESS FOR *BUFFER*


                                3911  **      CELLS AND POINTERS
                                3912
061.212  000 000                3913  LINPTR  DW      0               LINE POINTER
061.214  000                    3914  PROCHA  DB      0               PROBATION CHARACTER
061.215  000                    3915  SRCDIR  DB      0               SEARCH DIRECTION
061.216  000                    3916  OPTS    DB      0               OPTION FLAGS
                                3917
                                3918  *       FILE BUFFERS
                                3919
061.217  123 131 060            3920  DEFALT  DB      'SY0',0,0,0     DEFAULT DEVICE AND EXTENSION
                                3921
061.225                         3922  INFB    DS      0               INPUT FILE BUFFER
061.225  001                    3923          DB      1               CHANNEL NUMBER
061.226  000                    3924          DB      0               FLAGS
061.227  146 063                3925          DW      INBUF
061.231  146 063                3926          DW      INBUF
061.233  146 063                3927          DW      INBUF
061.235  146 065                3928          DW      INBUFE
061.237                         3929          DS      FB.NAML         NAME
                                3930
061.260                         3931  OUTFB   DS      0               OUTPUT FILE BUFFER
061.260  000                    3932          DB      0
061.261  000                    3933          DB      0               FLAGS
061.262  146 065                3934          DW      OUTBUF
061.264  146 065                3935          DW      OUTBUF
061.266  146 065                3936          DW      OUTBUF
061.270  146 067                3937          DW      OUTBUFE
061.272                         3938          DS      FB.NAML         NAME
                                3939
061.313                         3940  XOUTFB  DS      0               XOUT FILE BUFFER
061.313  002                    3941          DB      2
061.314  000                    3942          DB      0               FLAGS
061.315  146 067                3943          DW      XOTBUF
061.317  146 067                3944          DW      XOTBUF
061.321  146 067                3945          DW      XOTBUF
061.323  146 070                3946          DW      XOTBUFE
061.325                         3947          DS      FB.NAML
```

```
                        3951  **      PRS - PERFORM PRESET PROCESSING.
                        3952  *
                        3953  *       THIS CODE IS ONLY USED UPON ENTRY, AND THEN IS OVERLAID BY BUFFERS.
                        3954  *
                        3955  *       IT 1) TYPES THE BANNER MESSAGE
                        3956  *             2) DETERMINES THE MEMORY SIZE
                        3957  *             3) PRESETS THE TEXT PAGE TO NULL
                        3958  *
                        3959  *       ENTRY   NONE
                        3960  *       EXIT    DATA STRUCTURE INITIALIZED
                        3961
                        3962
061,346                 3963  ENTRY   EQU     *
061,346  257            3964          XRA     A
061,347  062 146 070     3965          STA     BUFFER-1        SET DUMY END-OF-LINE FOR BUFFER
061,352  062 346 061     3966          STA     LINE-1          SETUP 00 BYTE REQUIRED BEFORE *LINE*
                        3967
                        3968  *       CHECK VERSIONS AND LOAD *HDOSOVL0.SYS*
                        3969
061,355  377 011        3970          DB      SYSCALL,,VERS
061,357  332 035 062     3971          JC      PRSERR1         PROBABLY NO ,VERS SYSTEM CALL
061,362  376 040        3972          CPI     VERS
061,364  302 035 062     3973          JNZ     PRSERR1         NOT THE CORRECT VERSION
                        3974
                        3975  *       SETUP HIGH MEMORY
                        3976
061,367  315 333 053     3977          CALL    MAM             SET MAXIMUM MEMORY SIZE
                        3978
                        3979  *       SETUP CTL-C PROCESSING
                        3980
061,372  041 374 042     3981          LXI     H,INTRPT
061,375  076 003        3982          MVI     A,CTLC
061,377  377 041        3983          DB      SYSCALL,,CTLC
062,001  315 136 031     3984          CALL    $TYPTX
062,004  105 104 111     3985          DB      'EDIT Issue #103.06.00',ENL                          /WCZ080480/
062,032  303 200 042     3986          JMP     START           STARTUP
                        3987
062,035  076 050        3988  PRSERR1 MVI     A,EC.NCV        NOT THE CORRECT VERSION
                        3989
062,037  046 012        3990  ENTEXT  MVI     H,NL
062,041  377 057        3991          DB      SYSCALL,,ERROR  THERE WAS AN ERROR UPON ENTRY
062,043  257            3992          XRA     A
062,044  377 000        3993          DB      SYSCALL,,EXIT
                        3994
                        3995  **      BUFFERS OVERLAYING PRS
                        3996
062,046                 3997  MEML    EQU     *               END OF LOAD IMAGE
061,346                 3998          ORG     ENTRY
```

```
                        4000  **      STRING AND TEXT STORE AREAS
                        4001
061.346                 4002          DS      1               REQUIRED 0 BEFORE 'LINE'
                        4003
061.347                 4004  LINE    DS      120             LINE BUFFER
061.347                 4005  FNRA    EQU     LINE            $FNR WORK AREA
062.137                 4006  WRKSTR  DS      120             EXPANDED STRING WORK AREA
062.327                 4007  EDIA    DS      41              EDIT WORK AREA
063.000                 4008  EDIB    DS      41              EDIT WORK AREA
063.051                 4009  QUALS   DS      41              QUALIFIER STRING
063.122                 4010  NXTCHA  DS      1               NEXT COMMAND CHARACTER
063.123                 4011  PATCNT  DS      1               INDEX OF CURRENT PATTERN
063.124                 4012  CMDGRP  DS      1               ZERO IF RESTRICTED COMMAND GROUP
                        4013
063.125                 4014  $FOPWRK DS      FB.NAML         USED BY $FOPEX
                        4015
063.146                 4016  INBUF   DS      512
065.146                 4017  INBUFE  EQU     *
                        4018
065.146                 4019  OUTBUF  DS      512
067.146                 4020  OUTBUFE EQU     *
                        4021
067.146                 4022  XOTBUF  DS      256
070.146                 4023  XOTBUFE EQU     *



                        4025  **      TEXT BUFFER.
                        4026
070.146                 4027          DS      1               0 BYTE NEEDED FOR BACKWARDS SCAN OF 1ST LINE
070.147                 4028  BUFFER  DS      0


070.147                 4030          END
ASSEMBLY COMPLETE
4030 STATEMENTS
    1 ERRORS DETECTED
11468 BYTES FREE
```

$CCO	055000	408	2542L							
$CDEHL	030216	708	718	1692	1732	1790	2211	2981E		
$CHL	030224	2699E								
$CLL	054361	1033	1072	1196	1232	1247	2330	2496	2518L	
$CRLF	056001	2261	2501	2582	2669	2993L				
$DADA	030072	1073	3008E							
$FCLO	056217	1317	1329	1364	1411	1459	1536	3242L		
$FCLO.	056226	3242	3246L							
$FCLO1	056277	3285L	3288							
$FCLO2	056307	3281	3290L							
$FCLO3	056314	3254	3271	3301L						
$FCLO4	056322	3252	3307L							
$FERR1	060267	3810L	3815							
$FERR2	060303	3813	3819L							
$FERROR	060237	3141	3145	3149	3244	3329	3458	3501	3619	3802L
$FFB	060142	3377	3734L							
$FOPE1	056137	3171	3183L							
$FOPE2	056172	3207	3211L							
$FOPEA	056200	3211	3215E							
$FOPEB	056210	3205	3225L							
$FOPER	056057	1372	3139L							
$FOPER.	056104	3139	3152L							
$FOPEU	056075	3147L								
$FOPEU.	056112	3147	3156L							
$FOPEW	056066	1419	1467	3143L						
$FOPEW.	056107	3143	3154L							
$FOPWRK	063125	4014L								
$FREAL	056324	1525	3326L							
$FREAL.	056337	3326	3334E							
$FWBRK	057356	986	3616L							
$FWBRK.	057365	3616	3621L							
$FWBRK1	060001	3623	3628L							
$FWRIB	057174	3470	3479	3499L						
$FWRIB.	057203	3499	3504E							
$FWRIL	057127	996	1695	3456L						
$FWRIL.	057136	3456	3462L							
$FWRIL1	057142	3464L	3468							
$FWRILA	057173	3478	3483L							
$GNL	055171	409	2664L							
$INCHA	055015	2017	2564L	2590	2599					
$INCHAA	055150	2570	2574	2603	2607	2613L				
$INDL	030234	2929E	3258	3261						
$MCU	055205	828	871	1775	2645	2682L				
$MLU	055151	1349	1396	1440	2640L	2773				
$MLU1	055154	2643L	2648							
$MOVL	056007	935	1203	1236	1242	1941	2350	2853	3037E	
$MOVL1	056027	3063E	3071							
$MOVL2	056044	3051	3079E	3087						
$MOVLL	055271	1366	1413	1461	2837L					
$MU10	030324	1872	3103E							
$RCHAR	055337	1773	2564	2777	2907L	2908				
$REAL2	056350	3343L	3420							
$REAL3	056371	3355L	3379							
$REAL4	057017	3368	3371L							
$REAL6	057035	3373	3388L							
$REAL7	057047	3398L	3416							
$REAL8	057115	3348	3351	3378	3434L					
$REAL9	057123	3440L								

```
$REL7.3 057063  3403    3411L
$REL7.5 057106  3413    3424L
$RSTALL 031047  2212    2215    2546    2738E
$RTL    055233  1755    2771    2775E
$RTL.   055224  2771L
$RTL1   055234  2777L   2783
$RTL2   055266  2779    2800L
$SAVALL 031054  2180    2196    2542    2752E
$TBLS   055350  2950L   3209
$TBRA   031076  3123E
$TJMP   031061   446    3844E
$TJMP.  031062  3846E
$TYPC.  055334  2896L
$TYPCC  055314  2499    2867E   2876
$TYPCH  055330  2886L
$TYPTX  031136   410     476     497     585    1358    1376    1405    1452    1533    1599    1655    1712
                1771    1819    2048    2595    3803    3819    3861E   3984
$TYPTX. 031144  3863E
$UDD    031157  1625    1638    1654    2628E
$WCHAR  055345   796    1774    2016    2572    2608    2610    2911L   3814
$WRIB2  057207  3510L   3590
$WRIB3  057225  3520L   3556
$WRIB4  057253  3533    3536L
$WRIB6  057315  3538    3569L
$WRIB7  057327  3579L   3586
$WRIB8  057351  3513    3516    3560    3596L
.CHFLG  000060   128L
.CLEAN  000205   143L
.CLEAR  000055   125L
.CLEARA 000056   126L
.CLOSE  000046   118L   3306
.CLRCO  000007   102L    481
.CONSL  000006   101L   2545    2666
.CTLC   000041   113L   3983
.DAD    000206   144L
.DECODE 000053   123L
.DELET  000050   120L
.DISMT  000061   129L
.DMNMS  000203   141L
.DMOUN  000201   139L
.ERROR  000057   127L   2052    3823    3991
.EXIT   000000    95L    516    1331    3208    3214    3993
.LINK   000040   112L
.LOADD  000062   130L
.LOADO  000010   103L
.MONMS  000202   140L
.MOUNT  000200   138L
.NAME   000054   124L
.OPEN   000063   131L
.OPENC  000045   117L
.OPENR  000042   114L   3225
.OPENU  000044   116L   3227
.OPENW  000043   115L   3226
.POSIT  000047   119L
.PRINT  000003    98L
.READ   000004    99L   3753
.RENAM  000051   121L
.RESET  000204   142L
```

```
.SCIN    000001    96L   2907
.SCOUT   000002    97L   2873  2896  2911  2994
.SETTP   000052   122L   2214
.SYSRES  000012   105L
.VERS    000011   104L   3970
.WRITE   000005   100L   3294  3554  3662
ABS.COD  000010   244L    387
ABS.ENT  000006   242L
ABS.ID   000000   238L
ABS.LDA  000002   240L
ABS.LEN  000004   241L
ACL      052045   950    978  1104  1264  1574  1618  1728L 2387
ATL      052072   905    929  1101  1752L
AYS      052110   512   1118  1360  1407  1454  1771L
BELL     000007    32E    498   586  1377  1713  1772  1820  2049  3804
BKSP     000010    34E   2565  2596  2596
BUFFER   070147   908    1507  1643  2204  3965  4028L
BUFMAX   061210  1512    1645  1811  2208  3909L
BYE      047025   471    1327L
C.STX    000002    36E
C.SYN    000026    35E
CBE      052153   406    1787L
CBO      052171   923    1805L 2390
CBO1     052215  1497    1819L
CBT      060062  3339    3506  3622  3674L
CBT1     060071  3679L   3688
CCFLG    061206   414     478  1853  1978  3906L
CCPEND   061207   415     487  1979  3907L
CDV      052243  1835L   1867  1870
CLL1     054365  2522L   2526
CMD2     044113   743L    784   819
CMD3     044116   739     747L
CMD4     044145   755     760L   811   818   831
CMD5     044212   769     788L   833
CMD6     044226   797L
CMD6.5   044232   799L    825
CMD7     044265   790     823L
CMDADR   042332   450L    460
CMDDSP   000006   443     460E
CMDGRP   063124   439     754  4012L
CMDTAB   060321   750    3869E
CMDTAB.  060362   756    3876E
CO.FLG   000001   223E   2544
CR       000015    28E
CRFPTR   061174   434     537   550   567   635   751  1043  1146  1485  1584  1682  1691
                 1703    1706  2263  3900L
CRLPTR   061176   539     555   568   575  1024  1063  1147  1483  1567  1704  1729  1919
                 1925    2120  2126  2265  3901L
CS.FLG   000200   224E
CSL.CHR  000001   200E    403
CSL.ECH  000200   197E    403
CSL.RAW  000004   198E
CSL.WRP  000002   199E
CTB      060110  3441    3598  3625  3700L
CTB1     060121  3706L   3715
CTLA     000001    43E
CTLB     000002    44E
CTLC     000003    45E   3982
```

```
CTLD    000004    46E    1776    2018    2778
CTLO    000017    47E
CTLP    000020    48E
CTLQ    000021    49E
CTLS    000023    50E
CTLZ    000032    51E
CTP.2SB 000010   209E
CTP.BKM 000002   210E
CTP.BKS 000200   205E
CTP.FF  000100   206E
CTP.MLI 000040   207E
CTP.MLO 000020   208E
CTP.TAB 000001   211E
D.CON   040110   159L
D.RAM   040240   162L
D.VEC   040130   161L
DCC     052255   906    930    1032    1058    1127    1851L    2327
DCN     044072   431    732E
DCNA    044157   736    766E
DCO     044326   433    868L
DCO1    044333   870L   884
DCO2    044351   873    876L
DCQ     044310   432    848L
DCR     043066   430    535E
DCR1    043141   546    558L
DDN     052265   664    1188    1865L    2364
DDN1    052300  1870L   1880
DDN2    052332  1871    1884L
DEFALT  061217  1371    1418    1466    3920L
DELO    045221  1015L   1708
DEL1    045230  1018L   1036
DEL2    045277  1020    1040E    1087
DEL2.5  045322  1045    1050L
DEL3    045331  1017    1058L
DEL4    045345  1063L   1081
DELA    046020  1060    1085    1089L
DELETE  045206   452    1008L
DF.CLR  000376   313E
DF.EMP  000377   312E
DIR.ALD 000025   328L
DIR.CLU 000015   321L
DIR.CRD 000023   327L
DIR.EXT 000010   316L
DIR.FGN 000020   324L
DIR.FLG 000016   322L
DIR.LGN 000021   325L
DIR.LSI 000022   326L
DIR.NAM 000000   315L
DIR.PRO 000013   317L
DIR.VER 000014   318L
DIRELEN 000027   330E    381
DIRIDL  000015   319E
DRE     043235   566    574    619E
DRE1    043302   628    634    638L
DRE3    043305   640L   667
DRE4    043310   644L   679
DRE5    043331   646    648    652L
DRE6    043353   665L   671
```

```
DRE7    043372    637      675L    682
DRE8    043375    660      676L
DTBK    052337   1034     1903L   2336
DTBK.   052350   1086     1906    1913L
DTBK1   052375   1921     1922    1926E
DTBK2   053020   1941L    2140
DTBK3   053026   1924     1930    1947L
DTBK4   053035   1938     1955L   2136
EC.CNA  000004    254L
EC.DDA  000027    273L
EC.DIF  000017    265L
EC.DIW  000035    279L
EC.DNI  000045    287L
EC.DNR  000046    288L
EC.DNS  000005    255L
EC.DSC  000047    289L
EC.EOF  000001    251L    3328    3762
EC.EOM  000002    252L
EC.FAO  000031    275L    3178
EC.FAP  000026    272L
EC.FL   000030    274L
EC.FNF  000014    262L
EC.FNO  000011    259L    3346
EC.FNR  000034    278L
EC.FOD  000043    285L
EC.FUC  000013    261L
EC.ICN  000016    264L
EC.IDN  000006    256L
EC.IFC  000020    266L
EC.IFN  000007    257L
EC.ILC  000003    253L
EC.ILO  000040    282L
EC.ILR  000012    260L
EC.ILV  000037    281L
EC.IOI  000052    292L
EC.IS   000032    276L
EC.NCV  000050    290L    3988
EC.NEM  000021    267L
EC.NOS  000051    291L
EC.NPM  000044    286L
EC.NRD  000010    258L
EC.NVM  000042    284L
EC.OTL  000053    293L
EC.RF   000022    268L
EC.UNA  000036    280L
EC.UND  000015    263L
EC.UUN  000033    277L
EC.VPM  000041    283L
EC.WF   000023    269L
EC.WP   000025    271L
EC.WPV  000024    270L
ECC     053044     416      922    1018    1975L   2351
EDIO    046167    1182     1186L
EDI1    042250     422L    2030
EDI1.5  042323     442      444L
EDI2    046175    1184     1189L
EDI3    046200    1193L    1265
EDI5    046335    1194     1208    1264L
```

```
EDIA    046343  1262    1266L
EDIA    062327  1166    1206    1347    1368    1394    1415    1438    1463    1553    1571    4007L
EDIB    063000  1174    1241    4008L
EDITC   046116   453    1160E
EDIX    042200   403L    444     484    1378    1714    1821
EDTO    042245   418L    505
ENC     053064   544     569     625     644     658     733     850     870    1180    1835    1996L   2064
                2081    2298
ENC1    053137  2010    2025L
ENCA    053156   425     738    1996    2026    2035L   2067
ENL     000212    41E    1659    3985
ENTEXT  062037  3990L
ENTRY   061346   392    3963E   3998
EOFFLG  060236  3337    3734    3761    3788L
ERROR   053157  2047L   2216
ESC     000033    39E
EXIT    043052   512L   1757    2019
EXIT1   043063   513     515L
FB.CHA  000000   71L    3162    3165    3292    3303    3774
FB.FLG  000001    72L     967    1307    1355    1402    1448    1500    1579    1676    3165    3182    3220
                3247    3259    3262    3776
FB.FWA  000002    73L     482    3182    3188    3262    3778
FB.LIM  000006    75L    3193    3198    3782
FB.LWA  000010    76L    3784
FB.NAM  000012    77L      78    1369    1416    1464    3198    3805
FB.NAML 000021    78E    1352    1367    1399    1414    1444    1462    3929    3938    3947    4014
FB.PTR  000004    74L     483    3188    3193    3259    3780
FBENL   000033    79E
FF      000014    42E    2084
FILPTR  061170   543     626     716    1144    1583    1626    1674    1705    1787    3898L
FLUSH   046377   46A    1305E   1327
FLUSH1  047002  1307L   1312
FNRA    061347  4005E
FT.ABS  000000   301E    389
FT.BAC  000003   304E
FT.DD   000001   360E
FT.DC   000020   364E
FT.OR   000002   361E    1311    1356    1501    1580    3152    3156    3225    3227    3345
FT.OV   000010   363E
FT.OW   000004   362E     968    1403    1449    1677    3154    3156    3226    3227    3253    3512
FT.PIC  000001   302E
FT.REL  000002   303E
FWBRK2  060020  3642L   3649
FWBRK3  060034  3644    3651L
GNC     053205   556     559     572     638     654     743     883    1183    1879    2064L   2083    2085
                2088    2258    2283
GTC     053217  1161    1284    1343    1390    1433    1551    2081L   2290    2304
HLCPDE  055216  1044    1920    2121    2716L
I.CONFL 000004   226E    227     2543
I.CONTY 000001   213E    214
I.CONWI 000003   219E    220
I.CSLMD 000000   202E
I.CUSOR 000002   216E    217     2664
INBUF   063146  3925    3926    3927    4016L
INBUFE  065146  3928    4017E
INC0    055063  2566    2568    2587L
INC1    055112  2583    2597L   2611
INC3    055122  2594    2603L
```

```
INC4    055141  2605    2609L
INFB    061225  1307    1355    1363    1369    1500    1524    1535    1579    3922L
INS1    045035   901     918L
INS2    045045   921L    938
INS3    045077   916     935L
INSERT  044371   461     896E
INT1    043023   480     486L
INTRPT  042374   476L    1982    3981
IOC.CGN 000010   369L
IOC.CSI 000011   370L
IOC.DDA 000002   357L    365     379
IOC.DES 000016   376L
IOC.DEV 000020   377L
IOC.DIL 000021   379E
IOC.DIR 000023   381L
IOC.DRL 000010   373E
IOC.DTA 000014   375L
IOC.FLG 000004   359L    373
IOC.GRT 000005   367L
IOC.LGN 000012   371L
IOC.LNK 000000   356L
IOC.LSI 000013   372L
IOC.SPG 000007   368L
IOC.SQL 000003   365E
IOC.UNI 000022   378L
IOCCTD  000001   385E
IOCELEN 000052   383E
ITBK    053244   933     2104L   2343
ITBK.   053255  2107     2114L
ITBK1   053300  2122     2123    2127E
LALPTR  061172   551     629     706     914     1041    1047    1145    1478    1503    1521    1528    1562
                1628    1789    1807    1928    1931    2129    2132    2197    3899L
LF      000012    29E
LINE    061347   413     422     499     919     1008    1752    2581    2588    3966    4004L   4005
LINPTR  061212   423     734     779     801     2003    2020    2025    2587    2597    3913L
LQS     053322   678    2155L   2403
MAM     053333   407    2180L   3977
MEML    062046   391    3997E
MI.LDA  000072    60E
MI.NOP  000000    61E    1672
MI.RET  000311    62E
MIM     053350  1351    1398    1442    2196L
MIM0    053366  2200    2206L
MIM1    053372  2184    2208L
MLP     044020   669     680     699L
MLP1    044053   703     715L
NEWIN   047041   467    1339E
NEWIN1  047155  1357    1363L
NEWIN4  047204  1354    1376L   1401    1446
NEWOI   047352  1404    1410L
NEWOUT  047235   468    1386E
NEXT    050146   465    1475E
NEXT.   050151  1309    1477E   1582
NL      000012    40E      41     586    1187    1344    1359    1392    1406    1435    1453    1534    1600
                1657    1658    1713    1820    2049    2051    2259    2782    2993    3412    3483    3804    3821
                3990
NUL2    000000    31E
NULL    000200    30E
```

```
NXTCHA  063122   748   776   805  4010L
OPT.A   000001    55E  876  2228
OPT.B   000002    56E  876  2244
OPTS    061216   868  2227  2243  3916L
OUTBUF  065146  3934  3935  3936  4019L
OUTBUFE 067146  3937  4020E
OUTFR   061260  1316  1402  1410  1416  1676  1694  3931L
OVL.COD 000000   337L
OVL.ENS 000010   342E
OVL.ENT 000004   339L
OVL.FLB 000006   340L
OVL.SIZ 000002   338L
OVL0    000000   348L
OVL1    000001   349L
PATCH   061070  3892L
PATCNT  063123   441   760  4011L
PCFPTR  061202   536  1050  1149  1592  2264  3903L
PCLPTR  061204   538  1051  1150  1593  2266  3904L
PLA     054020  1595  1617  2227L 2352
PLB     054030   918  1031  1071  1100  1210  1616  2243L
PRI1    045114   946L  951
PRIA    045121   949E
PRINT   045111   451   945L
PROCHA  061214   418   783   794   798  2014  2022  3914L
PRSERR1 062035  3971  3973  3988L
PURGE   046052   463  1117L
PURGE.  046062  1125E  1792
QUALS   063051   676   848  1015  2395  4009L
QUOTE   000047    37E  636   659   851  2291  2299
R8N     054210  2364L
RCR     054041   897   945   965  1011  1097  1117  1189  1306  1350  1397  1441  1476
                1495  1558  1610  1671  2258L
RDS     046345  1167  1176  1278L 1348  1395  1439  1554
RDS1    046353  1282L 1289
READ    050203   462  1494E
READ.   050215  1481  1487  1496  1500L
READQ   050225  1503L 1529
READ1   050243  1506  1509L
READ2   050307  1502  1526  1533L
REFUSE  043031   497L  587   710   719   778   879  1010  1170  1283  1345  1393  1436
                1557  1868  1873  1877  2087  2260  2289  2367
REP1    046025  1098L 1106
REPLAC  046022   454  1097L
RESTART 042200   398E  514  2053  3824
ROMBOOT 030000   154E
RQS     054073   677   853  2281L
RQS1    054102  2288L 2294  2305
RQS2    054136  2300  2309L
RSL     054144  1103  1249  2327L
RSL1    054170  2332  2341L
RSL2    054175  2337  2344  2348E
RUBOUT  000177    33E  2567
S.CAADR 040333   230L
S.CCTAB 040335   231L
S.CONFL 040332   228L
S.CONTY 040327   215L  2592
S.CONWI 040331   221L
S.CSLMD 040326   203L   214   217   220   227   404  1754
```

```
S.CUSOR 040330   218L
S.DATC  040310   184L
S.DATE  040277   183L
S.GRT0  024000   150E
S.GRT1  025000   151E
S.GRT2  026000   152E
S.HIMEM 040316   186L
S.INT   040343   164L
S.OMAX  040324   192L
S.SOVR  041146   166L   168
S.SYSM  040320   188L  2181
S.TIME  040312   185L
S.USRM  040322   190L  2210
S.VAL   040277   163L   181
SAP     046070   909   1144L  1508
SEA0    050361  1562L  1586
SEA1    050377  1568L  1575
SEA2    051027  1585   1569   1579L
SEA3    051056  1573   1590L
SEA4    051074  1581   1599L
SEARCH  050337   466   1547E
SEL     054231   946    971   1019   1613   2391L
SEL.    054226  1098   1193   1568   2390L
SEL1    054222  2387L  2405
SEL2    054257  2398   2409L
SFS     054264  1207   1572   2157   2425L  2447
SFS1    054275  2434L  2440
SFS2    054316  2430   2436   2449L
SLB     054322   554    631    721   1048   1482   1566   2463L
SLB1    054323  2464L  2467
SNL     054333   500    704    763    921   1735   2480L  2483
SRCDIR  061215   621    653    701   3915L
STACK   042200   170E   426
STACKL  001032   168E
START   042200   397E  3986
SYDD    040130   160E
SYSCALL 000377    88E   481    516   1331   2052   2214   2545   2666   2873   2896   2907   2911
                 2994  3214   3294  3306   3554   3662   3753   3823   3970   3983   3991   3993
T.CHA   060224  3553   3661   3677  3705   3752   3775L  3776   3778   3780   3782   3784   3786
T.FLG   060225  3344   3511   3777L
T.FWA   060226  3543   3651   3741  3779L
T.LIM   060232  3357   3745   3765  3767   3783L
T.LWA   060234  3522   3546   3628  3654   3744   3785L
T.PTR   060230  3355   3418   3426  3520   3544   3588   3630   3652   3742   3781L
TAB     000011    38E  2082
TBL1    055357  2954L  2960
TBL2    055375  2952   2964L
TBL3    055377  2957   2968L
TLEN    000012  3676   3786E
TTX     054345   948   2496L
TTX.    054342  2231   2246   2495L
USE     051112   470   1609E
USE1    051120  1613L  1619
USE2    051143  1614   1623L
USE3    051212  1642   1644E
USEB    051246  1624   1657L
USEC    051264  1637   1658L
USED    051302  1653   1659L
```

```
USERFWA 042200    171E    387    390    391
VERS    000040     86E   3972
WRI..   051316   1673L
WRI1    051353   1691L   1697
WRI3    051374   1685    1701E
WRI4    052017    969    1678   1712L
WRIA    051374   1673    1702L
WRITE   051311    455    1670E
WRITE.  051314   1486    1672L
WRKPTR  061200    435     640    649    670    675    700    898    927    936   1022   1059   1065
                 1079    1148   1195   1484   1570   1585   1591   1675   1689   1707   1731   1736   2156
                 2329    2391   2409   2495   3902L
WRKSTR  062137   1201    1246   4006L
XOTBUF  067146   3943    3944   3945   4022L
XOTBUFE 070146   3946    4023E
XOUT    050001    469    1429E
XOUT1   050117   1450    1458L
XOUTFB  061313    482     483    967    985    995   1328   1448   1458   1464   3940L
XPR1    045145   971L     979
XPR2    045164    972     983E
XPR4    045173    976     992E
XPRA    045205   1000L   1001
XPRAL   000001   1001E
XPRINT  045132    456     964E
```

18044 BYTES FREE