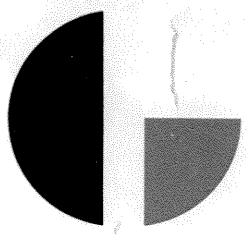

DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

LISTING #
092-000001-06
PAPER TAPE EDITOR
MODEL #
3013
SOURCE MODEL #
3009

0001 .MAIN

```
A 0002   .MAIN
01
02                 ;EDITOR
03
04                 ;DEFINE SOME BUFFER SIZES IN CHARACTERS
05          0012   .RDX 10
06          0012   TISZE=10
07          0170   TOSZE=120
08          1200   PSZE=640
09          0006   RSZE=6
10          0010   .RDX 8
11
12          0001   .LOC 1
13  0001    3152           INTRP            ;INTERRUPT PROCESSOR
14  0002    3000           JMP STRT         ;2=FULL RESTART
15  0003    3001           JMP RST          ;3=PARTIAL RESTART
16
17          0020   .LOC 20
18
19  0020    0000   XUA:    0                ;UP=COUNTING INDEX REGISTERS
20  0021    0000   XUB:    0
21  0022    0000   XUC:    0
22                 ;CONSTANTS
23  0023    0010   C10:    10
24  0024    0011   C11:    11
25  0025    0012   C12:    12
26  0026    0014   C14:    14
27  0027    0015   C15:    15
28  0030    0020   C20:    20
29  0031    0000   XDA:    0                ;DOWN=COUNTING INDEX REGISTERS
30  0032    0000   XDB:    0
31  0033    0033   C33:    33
32  0034    0040   C40:    40
33  0035    0042   C42:    42
34  0036    0044   C44:    44
35  0037    0052   C52:    52
36  0040    0053   C53:    53
37  0041    0055   C55:    55
38  0042    0060   C60:    60
39  0043    0072   C72:    72
40  0044    0100   C100:   100
41  0045    0104   C104:   104
42  0046    0115   C115:   115
43  0047    0127   C127:   127
44  0050    0134   C134:   134
45  0051    0136   C136:   136
46  0052    0177   C177:   177
47  0053    0200   C200:   200
48  0054    0377   C377:   377
49  0055    3777   C3777:  3777
50  0056    2000   C2000:  2000
51  0057   15433  ESC2:   015433
52  0060  177775  M3:     -3
53  0061  177774  M4:     -4
54  0062  177700  M100:   -100
55  0063  177400  M377:   177400
56
```

```
A 0003   .MAIN
01
02                      ;PAGE ZERO POINTERS
03 0064 2457 CRLF0:   CRLF            ;PRINT CR,LF
04 0065 2315 DECML:   DCML0           ;OUTPUT AC1 IN DECIMAL
05 0066 1347 DELT:    DELET           ;DO D COMMAND
06 0067 1612 EPROG:   ENDPR+ENDPR     ;END OF THE PROGRAM
07 0070 1261 ERROR:   ERR             ;STRING TYPER
08 0071 1777 FTEST:   FTST0           ;TEST FOR BUFFER FULL
09 0072 2371 GNPAR:   GNPR0           ;GENERATE PARITY
10 0073 1066 GTCHR:   GTCH0           ;GET A CHARACTER
11 0074 1071 GTCHI:   GTCH1           ;GET A CHARACTER&INCREMENT
12 0075 2711 INBF0:   INBUF           ;BUFFER INPUT
13 0076 2345 INIT:    INIT0           ;INITIALIZER ROUTINE
14 0077 3021 KEYRD:   KEYIN           ;INPUT FROM KEYBOARD
15 0100 0576 KINT0:   KINIT           ;KEYBOARD INITIALIZE
16 0101 1104 LNE:     LINE            ;DO L COMMAND
17 0102 2113 LNN:     LNN0            ;CALCULATE LINE # IN AC1
18 0103 1043 MVE:     MOVE            ;DO M COMMAND
19 0104 2445 PUNCH:   PUN             ;OUTPUT AC0 TO PUNCH
20 0105 2014 P0:      PNCH            ;DO P COMMAND
21 0106 3105 READ:    RDRIN           ;INPUT FROM READER
22 0107 3002 RESET:   RST+1           ;AC= DO RESET
23 0110 3001          RST             ;AT= DO TOTAL RESET
24 0111 1616 R0:      R               ;DO R COMMAND
25 0112 1211 SERCH:   SRCH            ;DO S COMMAND
26 0113 1212          SRCH+1          ;ENTRY FOR N
27 0114 1313 STCHR:   STCH0           ;STORE CHARACTER
28 0115 2403 TSPAR:   TSPR0           ;TEST PARITY
29 0116 2460 TYPE:    TYP             ;OUTPUT TO TELETYPE
30 0117 2172 YANK0:   YANK            ;DO Y COMMAND
```

```
A 0004   .MAIN
01
02                  ;PAGE ZERO VARIABLES
03
04 0120 000000 BEGIN:  0                ;POINTER TO BEGINNING
05                                       ;OF TEXT BUFFER
06 0121 000000 CHCNT:  0                ;CHARACTER COUNTER
07 0122 000000 COMPT:  0                ;COMMAND POINTER
08 0123 000000 CURPT:  0                ;CURRENT TEXT POINTER
09 0124 000000 XMODE:  0                ;0=NORMAL 1=EXPANDING MACRO
10 0125 000000 XDEF:   0                ;0=X UNDEFINED 1=X DEFINED
11 0126 000000 EOMEM:  0                ;ABSOLUTE END OF MEMORY
12 0127 000000 ARGX:   0                ;ARGUMENT FOR X MACRO
13 0130 000000 END:    0                ;POINTER TO END OF AVAILABLE MEMORY
14 0131 000000         0                ;END OF AVAILABLE MEMORY
15 0132 000000         0                ;POINTER TO L.H. OF END
16 0133 000000         0                ;TO L.H. OF END=1
17 0134 000000 LAST:   0                ;POINTER TO END
18                                       ;OF TEXT BUFFER
19 0135 000000 PNPAR:  0                ;0=GENERATE PUNCH PARITY
20 0136 000000         0                ;0=GENERATE CHECK CHARACTER
21 0137 000000 ROPAR:  0                ;0=TEST INPUT PARITY
22 0140 000000         0                ;0=TEST INPUT CHECK CHARACTER
23 0141 000000 SAC:    0                ;SAVE AC'S FOR
24 0142 000000         0                ;SUBROUTINES
25 0143 000000         0
26 0144 000000         0
27 0145 000000 SCRY:   0                ;SAVE CARRY+AC'S FOR
28 0146 000000         0                ;INTERRUPT
29 0147 000000         0
30 0150 000000         0
31 0151 000000         0
32 0152 000000 TABSM:  0                ;TAB SIMULATION SWITCH
33                                       ;0=DON'T SIMULATE
34 0153 000000 TEMA:   0                ;TEMPORARY STORAGE
35 0154 000000 TEMB:   0                ;TEMPORARY STORAGE
36 0155 000000 TEMC:   0                ;TEMPORARY STORAGE
37 0156 000000 TEMD:   0                ;TEMPORARY STORAGE
38 0157 000000 USTTI:  0                ;1=USE READER;0=USE TTI
39 0160 000000 USTTO:  0                ;1=USE PUNCH;0=USE TTO
40 0161 000000 YERR:   0          ;1=ERROR DURING YANK
```

```
A 0005   .MAIN
01
02                      ;READER DEVICE DATA BLOCK
03 000162 000000 RDCNT:  0                      ;CHARACTER COUNT
04 000163 000000         0                      ;OUTPUT POINTER
05 000164 000000         0                      ;INPUT POINTER
06 000165 001062         2*RDBUF+RSZE           ;END OF BUFFER
07 000166 001574         2*RDBUF                ;START OF BUFFER
08 000167 000006         RSZE                   ;SIZE OF BUFFER
09 000170 000000         0                      ;ACTIVE FLAG
10
11                      ;PUNCH DEVICE DATA BLOCK
12 000171 000000 PNCNT:  0
13 000172 000000         0
14 000173 000000         0
15 000174 001570         2*PNBUF+PSZE
16 000175 007370         2*PNBUF
17 000176 000120         PSZE
18 000177 000000         0
19
20                      ;TTO DEVICE DATA BLOCK
21 000200 000000 TOCNT:  0
22 000201 000000         0
23 000202 000000         0
24 000203 007364         2*TOBUF+TOSZE
25 000204 007174         2*TOBUF
26 000205 000170         TOSZE
27 000206 000000         0
28 000207 000000         0                      ;0=TYPING, 1=PUNCHING
29
30                      ;TTI DEVICE DATA BLOCK
31 000210 000000 TICNT:  0
32 000211 000000         0
33 000212 000000         0
34 000213 007170         2*TIBUF+TISZE
35 000214 007156         2*TIBUF
36 000215 000012         TISZE
37 000216 000000         0
38 000217 000000         0                      ;0=ECHO, OTHERWISE DON'T
39 000220 000000         0                      ;0=IN KEYBOARD MODE, OTHERWISE AS READER
```

```
A 0000   .MAIN
01
02          000300 .LOC 300
03                 ;FILE EDITOR INITIALIZER
04 000300 000342 STRT:   JMP STRT0                ;STARTING LOCATION
05 000301 004331 RST:    JSR RIO0          ;AT RESET TAPE AND DO AC
06 000302 004304         JSR RIO           ;PARTIAL RESET AC
07 000303 000533         JMP BGN-1
08 000304 102400 RIO:    SUB 0,0
09 000305 040124         STA 0,XMODE
10 000306 040171         STA 0,PNCNT
11 000307 040177         STA 0,PNCNT+6
12 000310 040200         STA 0,TOCNT
13 000311 040206         STA 0,TOCNT+6
14 000312 040207         STA 0,TOCNT+7
15 000313 040210         STA 0,TICNT
16 000314 040216         STA 0,TICNT+6
17 000315 040217         STA 0,TICNT+7
18 000316 040220         STA 0,TICNT+10
19 000317 020204         LDA 0,TOCNT+4
20 000320 040201         STA 0,TOCNT+1
21 000321 040202         STA 0,TOCNT+2
22 000322 020214         LDA 0,TICNT+4
23 000323 040211         STA 0,TICNT+1
24 000324 040212         STA 0,TICNT+2
25 000325 020175         LDA 0,PNCNT+4
26 000326 040172         STA 0,PNCNT+1
27 000327 040173         STA 0,PNCNT+2
28 000330 001400         JMP 0,3
29
30 000331 062677 RIO0:   IORST
31 000332 102400         SUB 0,0
32 000333 040162         STA 0,RDCNT
33 000334 040170         STA 0,RDCNT+6
34 000335 020166         LDA 0,RDCNT+4
35 000336 040163         STA 0,RDCNT+1
36 000337 040164         STA 0,RDCNT+2
37 000340 060177         INTEN
38 000341 001400         JMP 0,3
39 000342 004304 STRT0:  JSR RIO
40 000343 004331         JSR RIO0
41 000344 102400         SUB 0,0
42 000345 040125         STA 0,XDEF
43 000346 040157         STA 0,USTTI
44 000347 040160         STA 0,USTTO
45 000350 040135         STA 0,PNPAR
46 000351 040136         STA 0,PNPAR+1
47 000352 040137         STA 0,RDPAR
48 000353 040140         STA 0,RDPAR+1
49 000354 102000         ADC 0,0           ;SE TO DON'T
50 000355 040152         STA 0,TABSM               ;SIMULATE TABS
51 000356 034067         LDA 3,EPROG
52 000357 054120         STA 3,BEGIN               ;SET UP THE TEXT BUFFER
53 000360 054123         STA 3,CURPT
54 000361 054134         STA 3,LAST
55 000362 020057         LDA 0,ESC2
56 000363 175220         MOVZR 3,3
57 000364 041400         STA 0,0,3                 ;TERMINATE THE
58 000365 041401         STA 0,1,3                 ;BUFFER
59
```

0007 .MAIN

```
A 000000   .MAIN
01
02
03                      ;NOW, CALCULATE THE SIZE OF THE MEMORY
04 000360 003056                LDA 2,C2000
05 000361 155120                MOVZL 2,3
06 000362 041400                STA 0,0,3               ;STORE INTO END OF MEMORY
07 000363 025600                LDA 1,0,3               ;NOW, READ IT BACK
08 000364 122415                SUB# 1,0,SNR            ;AND COMPARE
09 000365 157000                ADD 2,3                 ;OK
10 000366 175112                MOVL# 3,3,SZC           ;TEST FOR 32K
11 000367 000403                JMP .+3         ;YES
12 000370 122415                SUB# 1,0,SNR
13 000371 000370                JMP .-7                 ;ADVANCE POINTER
14 000372 161220                MOVZR 3,0
15 000373 112512                SUBL# 0,2,SZC
16 000374 000403                JMP .+3                 ;GREATER THAN 1K
17 000375 063077                HALT                    ;ERROR
18 000376 000300                JMP STRT                ;START OVER
19 000377 003053                LDA 2,C20               ;COMPENSATE FOR LOADER, ETC
20 000400 150400                SUB 2,3
21 000401 054131                STA 3,END+1             ;STORE THE SIZE
22 000402 006004                JSR @CRLF0              ;TYPE CR,LF
23
24                      ;NOW, TYPE
25                      ;"TTO(1) OR PTP(2)?"
26 000403 006276                JSR @INIT
27 000404 007022                MESG1+MESG1
28 000405 000776                JMP .-2                 ;ILLEGAL INPUT
29 000406 010160                ISZ USTTO               ;(2)=USE PUNCH
```

```
A 0000   .MAIN
01                      ;TYPE
02                      ;"TTI(1) OR PTR(2)?"
03 000415 006076 STRT1: JSR @INIT
04 000416 007046        MESG2+MESG2
05 000417 000776        JMP .-2          ;ERROR
06 000420 010157        ISZ USTTI                ;(2)=USE THE READER
07
08                      ;"GENERATE PARITY ON OUTPUT (1) OR NOT (2)?"
09 000421 006076 STRT2: JSR @INIT
10 000422 007072        MESG3+MESG3
11 000423 000776        JMP .-2
12 000424 010135        ISZ PNPAR                ;(2)
13
14                      ;"TEST PARITY OF INPUT (1) OR NOT(2)?"
15 000425 006076        JSR @INIT
16 000426 007124        MESG5+MESG5
17 000427 000776        JMP .-2
18 000430 010137        ISZ RDPAR
19
20
21                      ;NOW=FINALIZE THE INITIALIZATION OF END
22 000431 034131        LDA 3,END+1
23 000432 030060        LDA 2,M3
24 000433 157000        ADD 2,3
25 000434 054131        STA 3,END+1
26 000435 054126        STA 3,EOMEM
27
```

```
A 0010   .MAIN
01
02                ;ROUTINE TO INPUT THE ENTIRE
03                ;COMMAND STRING AFTER TYPING A
04                ;CARRIAGE-RETURN, LINE-FEED AND AN ASTERISK
05                ;IF COMMAND OVERFLOWS THE AVAILABLE MEMORY,
06                ;THE MESSAGE:"
07                ;       BUFFER FULL DURING COMMAND INPUT
08                ;       COMMAND TERMINATED AND BEING EXECUTED"
09                ;
10                ;IS TYPED
11                ;AFTER COMMAND INPUT, A CARRIAGE-RETURN
12                ;LINE-FEED IS TYPED
13
14  0436 006064           JSR @CRLF0
15  0437 034131 BGN:      LDA 3,END+1             ;SET END OF MEMORY
16  0440 054122           STA 3,COMPT             ;INTO COMMAND POINTER
17  0441 171120           MOVZL 3,2               ;COMPLETE INITIALIZATION OF END
18  0442 050130           STA 2,END
19  0443 050132           STA 2,END+2
20  0444 050133           STA 2,END+3
21  0445 010132           ISZ END+2
22  0446 014133           DSZ END+3
23  0447 020037           LDA 0,C52               ;GET AN "*"
24  0450 006116           JSR @TYPE               ;AND TYPE IT
25  0451 102400           SUB 0,0                 ;INITIALIZE KEYBOARD
26  0452 004524           JSR KINIT               ;IN ECHO MODE
27  0453 102400           SUB 0,0                 ;CLEAR AC0
28
29  0454 176520 BGN0:     SUBZL 3,3               ;SET AC3=-1
30  0455 054520           STA 3,RHFLG             ;SET RIGHT-HALF FLAG
31  0456 040516 BGN1:     STA 0,PRVCH             ;STORE PREVIOUS CHARACTER
32  0457 006077           JSR @KEYRD              ;READ A CHARACTER
33  0460 101005           MOV 0,0,SNR             ;TEST FOR NULL-PARITY ERROR
34  0461 020050           LDA 0,C134              ;YES, SET TO \
35  0462 024052           LDA 1,C177              ;TEST FOR RUBOUT
36  0463 126405           SUB 0,1,SNR
37  0464 000453           JMP BGN5                ;YES
38  0465 001451           DSZ RHFLG               ;TEST FLAG
39  0466 000416           JMP BGN4                ;NO-USE THE LEFT HALF
40  0467 042131           STA 0,@END+1            ;STORE INTO RIGHT-HALF
41
```

```
A 0011   .MAIN
01
02                      ;TEST FOR END OF COMMAND, IE TWO ESC CHARACTERS
03 00470 024504 BGN2:   LDA 1,PRVCH
04 00471 030033         LDA 2,C33
05 00472 112415         SUB# 0,2,SNR
06 00473 132414         SUB# 1,2,SZR
07 00474 000762         JMP BGN1                    ;NO, GET NEXT
08                      ;END OF THE COMMAND
09 00475 102520 BGN3:   SUBZL 0,0                   ;INITIALIZE THE READER
10 00476 004500         JSR KINIT                   ;FOR NO ECHO
11 00477 006064         JSR @CRLF0                  ;TYPE CR,LF
12 00500 020122         LDA 0,COMPT                 ;TURN ADDRESS INTO A
13 00501 101120         MOVZL 0,0                   ;POINTER
14 00502 040122         STA 0,COMPT
15 00503 000506         JMP SCAN                    ;GO SCAN THE INPUT
16
17
18                      ;STORE THE CHARACTER INTO THE LEFT-HALF AND
19                      ;SLIDE THE MEMORY BACK
20 00504 006114 BGN4:   JSR @STCHR                  ;STORE THE CHARACTER
21 00505 000132         END+2                       ;POINTER TO THE LEFT-HALF
22 00506 014122         DSZ COMPT                   ;DECREMENT THE POINTER
23 00507 024122         LDA 1,COMPT                 ;GET THE POINTER
24 00510 044020         STA 1,XUA                   ;SOURCE LOCATION
25 00511 044021         STA 1,XUB                   ;DESTINATION LOCATION
26 00512 014021         DSZ XUB                     ;DESTINATION=SOURCE-1
27 00513 030131         LDA 2,END+1
28 00514 132400         SUB 1,2                     ;CALCULATE THE COUNT
29 00515 050153         STA 2,TEMA                  ;AND STORE IT
30 00516 036020         LDA 3,@XUA                  ;DO THE MOVE
31 00517 056021         STA 3,@XUB
32 00520 014153         DSZ TEMA                    ;TEST THE COUNT
33 00521 000775         JMP .-3
34 00522 176520         SUBZL 3,3                   ;SET FLAG TO RIGHT HALF
35 00523 054452         STA 3,RHFLG
36
```

```
A 0012   .MAIN
01
02                    ;NOW TEST TO SEE IF MEMORY HAS BECOME FULL
03 00524 030134            LDA 2,LAST
04 00525 151220            MOVZR 2,2
05 00526 034023            LDA 3,C10
06 00527 166400            SUB 3,1                    ;FORM COMPT=10
07 00530 132512            SUBL# 1,2,SZC              ;FORM (LAST)=(COMPT=10)
08 00531 000737            JMP BGN2                   ;OK, LAST<COMPT=10
09
10                    ;OVERFLOW=TERMINATE THE COMMAND
11 00532 020057            LDA 0,ESC2                 ;ESC INTO BOTH HALVES
12 00533 042131            STA 0,@END+1                        ;AND STORE IT
13 00534 006070            JSR @ERROR                 ;CALL STRING PRINTER
14 00535 006572            ERMS3+ERMS3
15 00536 000737            JMP BGN3                   ;GO, DO THE COMMAND
16
17                    ;PROCESS RUBOUT BY PRINTING THE PREVIOUS
18                    ;CHARACTER AND BACKING UP
19 00537 014436 BGN5:      DSZ RHFLG                  ;TEST FOR RIGHT HALF
20 00540 000423            JMP BGN6                   ;NO, LEFT HALF
21 00541 020122            LDA 0,COMPT                ;YES, MUST SLIDE THE DATA
22 00542 024131            LDA 1,END+1                ;UP IN MEMORY
23 00543 044031            STA 1,XDA
24 00544 044032            STA 1,XDB
25 00545 010032            ISZ XDB                    ;DESTINATION=SOURCE+1
26 00546 106405            SUB 0,1,SNR                ;FORM COUNT
27 00547 000667            JMP BGN-1                  ;NOTHING LEFT
28 00550 044153            STA 1,TEMA                 ;STORE THE COUNT
29 00551 026031            LDA 1,@XDA                 ;NOW, MOVE THE DATA
30 00552 046032            STA 1,@XDB
31 00553 014153            DSZ TEMA
32 00554 000775            JMP .-3                    ;NOT DONE YET
33 00555 010122            ISZ COMPT                  ;ALTER POINTER
34 00556 020416            LDA 0,PRVCH                ;GET THE PREVIOUS CHARACTER
35 00557 006116            JSR @TYPE                  ;AND TYPE IT
36 00560 006073            JSR @GTCHR                 ;GET A NEW PREVIOUS
37 00561 000130            END                        ;CHARACTER
38 00562 000674            JMP BGN1
39
40                    ;IT WAS SET TO LEFT=HALF
41 00563 020411 BGN6:      LDA 0,PRVCH                ;GET THE PREVIOUS
42 00564 006116            JSR @TYPE                  ;CHARACTER AND TYPE IT
43 00565 006073            JSR @GTCHR                 ;GET A NEW PREVIOUS
44 00566 000133            END+3                      ;CHARACTER
45 00567 024122            LDA 1,COMPT                ;TEST FOR ONLY ONE
46 00570 030131            LDA 2,END+1                ;CHARACTER IN THE BUFFER
47 00571 132405            SUB 1,2,SNR
48 00572 102400            SUB 0,0          ;YES, CLEAR PREVIOUS
49 00573 000661            JMP BGN0                   ;SET TO RIGHT HALF
50 00574 000000 PRVCH:     0                          ;PREVIOUS CHARACTER
51 00575 000000 RHFLG:     0                          ;RIGHT=HALF FLAG
```

```
A 0#13   .MAIN
#1
#2
#3                ;INITIALIZE KEYBOARD/READER
#4                ;AC0=ECHO MODE
#5
#6 00576 060277 KINIT:  INTDS                   ;DISABLE THE INTERRUPT
#7 00577 040217         STA 0,TICNT+7           ;SET THE ECHO MODE
#8 00600 102400         SUB 0,0
#9 00601 040210         STA 0,TICNT             ;RESET THE DEVICE DATA BLCOK
10 00602 040216         STA 0,TICNT+6           ;CLEAR ACTIVE
11 00603 040220         STA 0,TICNT+10          ;SET TO KEYBOARD
12 00604 020214         LDA 0,TICNT+4           ;SET THE POINTERS
13 00605 040211         STA 0,TICNT+1
14 00606 040212         STA 0,TICNT+2
15 00607 060177         INTEN                   ;RE-ENABLE INTERRUPT
16 00610 000400         JMP 0,3                 ;AND EXIT FROM THE ROUTINE
17
```

```
A 0014   .MAIN
01
02
03                ;ROUTINE TO SCAN THE COMMAND
04                ;WHEN A COMMAND IS FOUND, EXECUTE IT
05                ;IF COMMAND CANNOT BE FOUND, THE ENTIRE
06                ;UN-EXECUTED COMMAND STRING IS TYPED
07
08 00611 020122 SCAN:   LDA 0,COMPT              ;SAVE THE COMMAND POINTER
09 00612 040501         STA 0,SAVPT              ;IN CASE OF ERROR
10 00613 102400         SUB 0,0
11 00614 040505         STA 0,SCN6               ;CLEAR OUT NUMBER
12 00615 040154         STA 0,TEMB               ;AND SIGN
13 00616 040507         STA 0,SCN7               ;SET TO DECIMAL
14 00617 004523         JSR GET                  ;GET A CHARACTER
15 00620 024035         LDA 1,C42                ;TEST FOR "
16 00621 106414         SUB# 0,1,SZR
17 00622 000403         JMP .+3                  ;NO
18 00623 010502         ISZ SCN7                 ;YES, SET TO OCTAL
19 00624 004516         JSR GET                  ;AND GET ANOTHER
20 00625 024040         LDA 1,C53                ;TEST FOR "+"
21 00626 106415         SUB# 0,1,SNR
22 00627 000427         JMP SCN1                 ;YES
23 00630 024041         LDA 1,C55                ;NO, TEST FOR "-"
24 00631 106415         SUB# 0,1,SNR
25 00632 000426         JMP SCN1+2               ;YES
26
27 00633 024042 SCN0:   LDA 1,C60                ;NOW , TEST FOR A DIGIT
28 00634 030043         LDA 2,C72
29 00635 122513         SUBL# 1,0,SNC
30 00636 142513         SUBL# 2,0,SNC
31 00637 000423         JMP SCN2                 ;NOT A DIGIT
32 00640 122400         SUB 1,0                  ;FORM DIGIT
33 00641 024460         LDA 1,SCN6               ;GET NUMBER
34 00642 135120         MOVZL 1,3
35 00643 175120         MOVZL 3,3
36 00644 030461         LDA 2,SCN7               ;TEST FOR OCTAL
37 00645 151005         MOV 2,2,SNR
38 00646 000405         JMP .+5                  ;NO
39 00647 030023         LDA 2,C10                ;YES, TEST THE DIGIT
40 00650 142513         SUBL# 2,0,SNC
41 00651 000435         JMP SCN4                 ;8 OR 9, ERROR
42 00652 165121         MOVZL 3,1,SKP            ;OK
43 00653 167120         ADDZL 3,1                ;FORM NUMBER*10
44 00654 107000         ADD 0,1                  ;ADD IN THIS DIGIT
45 00655 044444         STA 1,SCN6               ;SAVE NUMBER
```

```
A 0015   .MAIN
01
02
03                      ;GET ANOTHER CHARACTER
04 000656 004464 SCN1:  JSR GET                 ;GET A CHARACTER
05 000657 000754        JMP SCN0                ;AND IGNORE CR
06
07                      ;REMEMBER MINUS SIGN
08 000660 010154        ISZ TEMB
09 000661 000775        JMP SCN1                ;GET NEXT CHARACTER
10
11                      ;NOT A DIGIT, +, OR -
12 000662 024437 SCN2:  LDA 1,SCN6              ;NEGATE NUMBER IF MINUS
13 000663 030055        LDA 2,C3777
14 000664 147400        AND 2,1
15 000665 014154        DSZ TEMB
16 000666 000403        JMP .+3                 ;NOT MINUS
17 000667 124400        NEG 1,1
18 000670 044431        STA 1,SCN6              ;STORE SIGNED NUMBER
19 000671 024033        LDA 1,C33               ;TEST FOR ESC
20 000672 106414        SUB# 0,1,SZR
21 000673 000406        JMP SCN3                ;NO, DO THE COMMAND
22 000674 004446        JSR GET                 ;YES, GET NEXT
23 000675 106415        SUB# 0,1,SNR            ;AND TEST IT
24 000676 000430        JMP SCN8                ;YES, CHECK MODE FOR EXIT
25 000677 014122        DSZ COMPT               ;NO- BACKUP AND GET NEXT
26 000700 000711        JMP SCAN                ;COMMAND
27
```

```
A 0016   .MAIN
01
02
03                      ;DO A COMMAND LOOKUP
04 000701 024423 SCN3:    LDA 1,CTBL            ;POINTER TO COMMAND TABLE
05 000702 044020          STA 1,XUA             ;STORE IN INDEX REGISTER
06 000703 026020          LDA 1,@XUA            ;GET WORD FROM TABLE
07 000704 125004          MOV 1,1,SZR           ;AND TEST FOR END
08 000705 000410          JMP SCN5              ;NO
09
10                      ;YES, COMMAND NOT FOUND
11 000706 102400 SCN4:    SUB 0,0
12 000707 040124          STA 0,XMODE           ;RESET TO NORMAL MODE
13 000710 006070          JSR @ERROR            ;PRINT ERROR MESSAGE
14 000711 006724          ERMS4+ERMS4
15 000712 006070          JSR @ERROR            ;PRINT ERRANT COMMAND
16 000713 000000 SAVPT:   0
17 000714 002420          JMP @NEXT             ;GET ANOTHER STRING
18
19 000715 036020 SCN5:    LDA 3,@XUA            ;GET FROM TABLE
20 000716 106414          SUB# 0,1,SZR          ;TEST FOR A MATCH
21 000717 000764          JMP SCN3+2            ;NO, GET THE NEXT
22 000720 004000          JSR 0,3               ;CALL ROUTINE
23 000721 000000 SCN6:    0                     ;ARGUMENT
24 000722 000764          JMP SCN4              ;ERROR
25 000723 000666          JMP SCAN              ;OK, DO NEXT
26 000724 000761 CTBL:    CTABL-1               ;COMMAND TABLE POINTER
27 000725 000000 SCN7:    0
28
29
30 000726 020124 SCN8:    LDA 0,XMODE           ;GET MODE WORD
31 000727 101005          MOV 0,0,SNR           ;EXPANSION MODE?
32 000730 002404          JMP @NEXT             ;NO..GET ANOTHER STRING
33 000731 014127          DSZ ARGX              ;YES..DECREMENT ARGUMENT
34 000732 000405          JMP SCN9              ;NOT DONE, RESET FOR MACRO
35 000733 014124          DSZ XMODE             ;DONE, RESET MODE
36
37                      ;N.B. SKIP ALWAYS OCCURS
38
39 000734 000437 NEXT:    BGN
40
41 000735 020155          LDA 0,TEMC            ;PICKUP SAVED COMMAND POINTER
42 000736 000402          JMP .+2
43
44 000737 020156 SCN9:    LDA 0,TEMD            ;PICKUP MACRO COMMAND POINTER
45 000740 040122          STA 0,COMPT           ;COMPT RESET FOR SCANNER
46 000741 000650          JMP SCAN              ;RETURN
47
```

```
A 0017   .MAIN
01
02
03                ;SUBROUTINE TO GET A CHARACTER AND IGNORE A
04                ;CARRIAGE RETURN
05 00742 054153 GET:    STA 3,TEMA              ;SAVE RETURN
06 00743 006074         JSR @GTCHI              ;GET AND INCREMENT
07 00744 000122         COMPT                   ;COMMAND POINTER
08 00745 030027         LDA 2,C15               ;TEST FOR A CARRIAGE RETURN
09 00746 112405         SUB 0,2,SNR
10 00747 000774         JMP GET+1               ;YES, GET THE NEXT CHARACTER
11 00750 002153         JMP @TEMA               ;NO, EXIT
12
13                ;TRANSLATOR TABLE FOR TYPING THE INVISIBLE
14                ;CHARACTERS
15                ;TABLE CONSISTS OF PAIRS OF ENTRIES TERMINATED BY A ZERO
16                ;ELEMENT
17                ;IF FIRST.LE.CHARACTER.LE.SECOND, THEN TYPE AS
18                ;       A FOLLOWED BY CHARACTER+100
19
20 00751 000001 TABL:   01
21 00752 000006         06                      ;01-06
22
23 00753 000010         10
24 00754 000010         10                      ;10
25
26 00755 000013         13
27 00756 000014         14                      ;13-14
28
29 00757 000016         16
30 00760 000037         37                      ;16-37
31
32 00761 000000         0                       ;END OF TABLE
33
34                .EOT
```

```
A 0019   .MAIN
01
02
03                ;TAPE 2
04                ;CTABL= COMMAND TABLE
05                ;       ENTRY PAIRS CONTAIN:
06                ;       CHARACTER CODE
07                ;       ADDRESS OF ROUTINE
08                ;
09                ;TABLE IS TERMINATED BY A ZERO
10
11 00762 000101 CTABL:  "A                          ;A=APPEND
12 00763 002174         APPND
13
14 00764 000102         "B                          ;B=BEGINNING
15 00765 001204         B
16
17 00766 000103         "C                          ;C=CHANGE
18 00767 001407         CHNG
19
20 00770 000104         "D                          ;D=DELETE
21 00771 001347         DELET
22
23 00772 000105         "E                  ;E=END TAPE
24 00773 001644         ENDTP
25
26 00774 000106         "F                          ;F=FEED
27 00775 002134         F
28
29 00776 000111         "I                          ;I=INSERT
30 00777 001447         INSRT
31
32 01000 000112         "J                          ;J=JUMP
33 01001 001167         JUMP
34
35 01002 000113         "K                          ;K=KILL
36 01003 001335         KILL
37
38 01004 000114         "L                          ;L=LINE
39 01005 001104         LINE
40
41 01006 000115         "M                          ;M=MOVE
42 01007 001043         MOVE
43
44 01010 000116         "N                          ;N=SEARCH
45 01011 001564         N
46
47 01012 000120         "P                          ;P=PUNCH
48 01013 002014         PNCH
49
50 01014 000121         "Q                  ;Q=SEARCH BUT DON'T PUNCH
51 01015 001563         Q
```

```
A 0020   .MAIN
01
02 01016 000122          "R                    ;R=PUNCH AND READ
03 01017 001616          R
04
05 01020 000123          "S                    ;S=SEARCH
06 01021 001211          SRCH
07
08 01022 000124          "T                    ;T=TYPE
09 01023 002011          TPE
10
11 01024 000130          "X                    ;X=MACRO
12 01025 001665          XMAC
13
14 01026 000131          131                   ;Y=YANK
15 01027 002172          YANK
16
17 01030 000132          "Z                    ;Z=END
18 01031 001207          Z
19
20 01032 000072          ":                    ;:=TYPE NUMBER OF LINES
21 01033 001146          COLON
22
23 01034 000075          "=                    ;==TYPE NUMBER OF CHARACTERS
24 01035 001155          EQUAL
25
26 01036 000056          ".          ;.=TYPE CURRENT LINE NUMBER
27 01037 001162          POINT
28
29 01040 000011          11                    ;TAB=INSERT TAB AND TEXT
30 01041 001443          TAB
31
32 01042 000000          0
33
34
35
```

```
▲ 0021   .MAIN
01
02                ;MOVE-SUBROUTINE TO MOVE THE POINTER
03                ;       ARG POSITIONS WHERE ARG
04                ;       FOLLOWS THE CALL
05                ;
06                ;       JSR MOVE
07                ;       ARG
08                ;       (UNUSED)
09                ;       NORMAL RETURN
10                ;
11                ;CHARACTER TO THE RIGHT OF THE POINTER IS
12                ;RETURNED IN AC0
13
14
15 01043 054422 MOVE:   STA 3,MRTN              ;SAVE RETURN
16 01044 025000         LDA 1,0,3               ;GET ARG
17 01045 020123         LDA 0,CURPT
18 01046 123000         ADD 1,0          ;AND ADD TO CURPT
19 01047 024120         LDA 1,BEGIN
20 01050 122512         SUBL# 1,0,SZC           ;COMPARE WITH BEGIN+1
21 01051 000411         JMP MOVE2               ;END OF THE BUFFER
22 01052 024134         LDA 1,LAST
23 01053 122513         SUBL# 1,0,SNC           ;COMPARE WITH LAST
24 01054 000406         JMP MOVE2               ;END OF THE BUFFER
25 01055 040123         STA 0,CURPT
26 01056 006073         JSR @GTCHR              ;GET THE NEXT CHARACTER
27 01057 000123         CURPT
28 01060 034005 MOVE1:  LDA 3,MRTN              ;GET RETURN
29 01061 001402         JMP 2,3          ;AND EXIT
30
31 01062 044123 MOVE2:  STA 1,CURPT             ;END OF THE BUFFER
32 01063 020033         LDA 0,C33               ;GET ESC CODE
33 01064 000774         JMP MOVE1               ;AND EXIT
34 01065 000000 MRTN:   0
35
```

```
A 0022   .MAIN
01
02                ;GTCH0  SUBROUTINE TO GET A CHARACTER
03                ;       FROM THE BUFFER AND PLACE IT
04                ;       INTO AC0
05                ;
06                ;CALLING SEQUENCE:
07                ;       JSR @GTCHR (OR GTCHI)
08                ;       ADDRESS OF POINTER
09                ;       RETURN (AC1,AC2 ARE PRESERVED)
10                ;
11                ;AN AUXILLIARY ENTRY "GTCH1" IS IDENTICAL
12                ;TO "GTCHR" BUT, ADDITIONALLY, IT INCREMENTS THE POINTER
13
14  01066 050143 GTCH0:  STA 2,SAC+2              ;SAVE AC2
15  01067 033400         LDA 2,@0,3               ;GET THE POINTER INTO AC2
16  01070 000404         JMP .+4                  ;AND PROCEED
17
18  01071 050143 GTCH1:  STA 2,SAC+2              ;SAVE AC2
19  01072 033400         LDA 2,@0,3               ;GET THE POINTER INTO AC2
20  01073 034000         ISZ @0,3                 ;INCREMENT THE POINTER
21  01074 151220         MOVZR 2,2                ;TURN POINTER INTO ADDRESS
22  01075 021000         LDA 0,0,2                ;AND GET WORD
23  01076 030054         LDA 2,C377               ;GET AN 8 BIT MASK
24  01077 101002         MOV 0,0,SZC              ;TEST RH OR LF
25  01100 101300         MOVS 0,0                 ;LEFT HALF, SWAP
26  01101 143400         AND 2,0                  ;8 BIT CHARACTER INTO AC0
27  01102 030143         LDA 2,SAC+2              ;RESTORE AC2
28  01103 001401         JMP 1,3                  ;AND RETURN
29
```

```
A 0023   .MAIN
01
02
03                ;LINE    SUBROUTINE TO MOVE THE POINTER OVER
04                ;         ARG LINES WHERE ARG FOLLOWS
05                ;         THE CALL
06                ;
07                ;         JSR LINE
08                ;         ARG
09                ;         (UNUSED)
10                ;         NORMAL RETURN
11                ;
12                ;ALL AC'S ARE LOST
13
14 01104 054440 LINE:    STA 3,LNRT                ;SAVE THE RETURN
15 01105 031400           LDA 2,0,3                 ;GET THE ARG
16 01106 150400           NEG 2,2                   ;AND NEGATE IT
17 01107 151113           MOVL# 2,2,SNC             ;TEST FOR NEGATIVE
18 01110 000404           JMP .+4                   ;NO
19 01111 126520           SUBZL 1,1                 ;YES, SET AC1=+1(FORWARD)
20 01112 150400           NEG 2,2
21 01113 000403           JMP .+3                   ;GO STORE AC1,AC2
22
23 01114 151400           INC 2,2                   ;INCREMENT
24 01115 126000           ADC 1,1                   ;SET AC1=-1(REVERSE)
25 01116 050427           STA 2,NLINE               ;STORE ARG
26 01117 044406           STA 1,LINE0+1             ;STORE DIRECTION
27 01120 006073           JSR @GTCHR                ;GET A CHARACTER
28 01121 000123           CURPT
29 01122 125113           MOVL# 1,1,SNC             ;TEST FOR REVERSE
30 01123 000404           JMP .+4          ;NO, FORWARD
31
```

```
A 0024   .MAIN
01                    ;ENTER HERE ON EACH CHARACTER
02  01124 006103 LINE0:  JSR @MVE                    ;MOVE POINTER
03  01125 000000          0                           ;EITHER -1 OR +1
04  01126 063077          HALT                        ;???????
05  01127 030033          LDA 2,C33                   ;TEST FOR ESC
06  01130 112415          SUB# 0,2,SNR
07  01131 000411          JMP LINE1                   ;YES, END OF BUFFER
08  01132 024027          LDA 1,C15                   ;TEST CHARACTER (IN AC0)
09  01133 106404          SUB 0,1,SZR                 ;FOR A CARRIAGE RETURN
10  01134 000770          JMP LINE0                   ;NO
11  01135 014410          DSZ NLINE                   ;YES, TEST COUNTER
12  01136 000766          JMP LINE0                   ;CONTINUE
13  01137 006103          JSR @MVE                    ;YES, MOVE THE POINTER
14  01140 000001          1
15  01141 063077          HALT                        ;????????
16  01142 034402 LINE1:   LDA 3,LNRT                  ;AND RETURN
17  01143 001402          JMP 2,3                     ;EXIT
18
19  01144 000000 LNRT:    0                           ;SAVED RETURN
20  01145 000000 NLINE:   0
21
22                    ;:      SUBROUTINE TO PRINT TOTAL
23                    ;       NUMBER OF LINES IN THE BUFFER
24                    ;
25                    ;       JSR COLON
26                    ;       (UNUSED)
27                    ;       (UNUSED)
28                    ;       RETURN
29                    ;
30
31  01146 054776 COLON:   STA 3,LNRT                  ;SAVE THE RETURN
32  01147 006102          JSR @LNN                    ;GET NUMBER OF LINES
33  01150 000134          LAST
34  01151 006065 COLN1:   JSR @DECML                  ;AND TYPE IT
35  01152 006064          JSR @CRLF0                  ;TYPE CR,LF
36  01153 030771          LDA 2,LNRT                  ;AND RETURN
37  01154 001002          JMP 2,2
38
39
40
41                    ;=      SUBROUTINE TO PRINT
42                    ;       NUMBER OF CHARS IN THE BUFFER
43
44  01155 054767 EQUAL:   STA 3,LNRT                  ;SAVE RETURN
45  01156 024134          LDA 1,LAST
46  01157 030120          LDA 2,BEGIN
47  01160 146400          SUB 2,1
48  01161 000770          JMP COLN1                   ;PRINT THE NUMBER
49
50                    ;.      PRINT CURRENT LINE NUMBER
51  01162 054762 POINT:   STA 3,LNRT      ;SAVE RETURN
52  01163 006102          JSR @LNN
53  01164 000123          CURPT
54  01165 125400          INC 1,1
55  01166 000763          JMP COLN1       ;GO PRINT AND RETURN
56
```

```
A 0025   .MAIN
01
02
03                ;JUMP   JUMP TO LINE ARG WHERE ARG
04                ;       FOLLOWS THE CALL
05                ;
06                ;       JSR JUMP
07                ;       ARG
08                ;       (UNUSED)
09                ;       RETURN
10
11 01167 054414 JUMP:   STA 3,JRTN              ;SAVE RETURN
12 01170 020120         LDA 0,BEGIN             ;SET CURPT TO BEGINNING
13 01171 040123         STA 0,CURPT
14 01172 021400         LDA 0,0,3               ;GET ARGUMENT
15 01173 101400         NEG 0,0                 ;AND SUBTRACT 1 FROM IT
16 01174 100000         COM 0,0
17 01175 040402         STA 0,.+2
18 01176 006101         JSR @LNE                ;MOVE ARG=1 LINES
19 01177 000000         0
20 01200 063077         HALT                    ;?????
21 01201 034402         LDA 3,JRTN              ;GET RETURN
22 01202 001402         JMP 2,3                 ;AND EXIT
23 01203 000000 JRTN:   0
24
25
26                ;B,Z JUMP TO THE BEGINNING (END) OF THE BUFFER
27                ;
28                ;CALLING SEQUENCE:
29                ;       JSR B (OR Z)
30                ;       ARG
31                ;       UNUSED
32                ;       RETURN
33 01204 020120 B:      LDA 0,BEGIN
34 01205 040123         STA 0,CURPT
35 01206 001402         JMP 2,3
36
37 01207 020134 Z:      LDA 0,LAST
38 01210 000775         JMP B+1
39
```

```
A 0026   .MAIN
01                ;SRCH    DO A STRING SEARCH
02                ;        TRY TO FIND THE STRING LOCATED
03                ;        BY COMPT WITHIN THE STRING LOCATED
04                ;        BY CURPT
05                ;
06                ;        JSR SRCH
07                ;        (UNUSED)
08                ;        ERROR=BUFFER END REACHED WITHOUT MATCH
09                ;        NORMAL=CURPT POINTS TO CHARACTER AFTER MATCH
10
11 01211 102001 SRCH:    ADC 0,0,SKP              ;SET ENTRY MODE
12 01212 102400          SUB 0,0          ;ENTRY FOR N SEARCH
13 01213 040770          STA 0,JRTN       ;SAVE MODE
14 01214 054444          STA 3,SRTN               ;SAVE RETURN
15 01215 010443          ISZ SRTN                 ;AND INCREMENT IT
16 01216 020123          LDA 0,CURPT              ;MOVE POINTERS INTO
17 01217 040153          STA 0,TEMA               ;TEMPS
18 01220 020122          LDA 0,COMPT
19 01221 040154          STA 0,TEMB
20 01222 024033          LDA 1,C33
21
22 01223 006073 SRCH0:   JSR @GTCHR               ;GET A CHARACTER AND INCREMENT
23 01224 000154          TEMB
24 01225 106415          SUB# 0,1,SNR
25 01226 000414          JMP SRCH1                ;YES, OK
26 01227 111000          MOV 0,2                  ;NO, SAVE CHARACTER
27 01230 006074          JSR @GTCHI               ;GET CHARACTER
28 01231 000153          TEMA
29 01232 106415          SUB# 0,1,SNR                      ;TEST FOR ESC
30 01233 000415          JMP SRCH2                ;YES, FAIL
31 01234 112405          SUB 0,2,SNR              ;NO, COMPARE CHARACTERS
32 01235 000403          JMP .+3                  ;YES, KEEP GOING
33 01236 010123          ISZ CURPT                ;NOT EQUAL, ADVANCE CURPT AND
34 01237 000757          JMP SRCH+5               ;START OVER AGAIN
35
36 01240 010154          ISZ TEMB                 ;ADVANCE TEMB AND TRY
37 01241 000762          JMP SRCH0                ;AGAIN
38
39 01242 010416 SRCH1:   ISZ SRTN                 ;INCREMENT RETURN
40 01243 020154          LDA 0,TEMB
41 01244 040122          STA 0,COMPT              ;SET NOW COMPT
42 01245 020153          LDA 0,TEMA
43 01246 040123          STA 0,CURPT
44 01247 002411          JMP @SRTN                ;AND RETURN
45
46                ;SEARCH FAILED
47 01250 014122 SRCH2:   DSZ COMPT                ;DECREMENT COMMAND POINTER
48 01251 020120          LDA 0,BEGIN
49 01252 040123          STA 0,CURPT
50 01253 010730          ISZ JRTN         ;TEST MODE
51 01254 002404          JMP @SRTN                ;N ENTRY MODE
52 01255 006070          JSR @ERROR
53 01256 006766          ERMS6*2          ;TYPE "STR NOT FOUND"
54 01257 002401          JMP @SRTN                ;AND RETURN (ERROR)
55 01260 000000 SRTN:    0                        ;SAVED RETURN
56
```

```
A 0027   .MAIN
01
02
03                ;ERR    SUBROUTINE TO TYPE ERROR MESSAGES
04                ;
05                ;CALLING SEQUENCE:
06                ;       JSR @ERROR
07                ;       POINTER TO MESSAGE
08                ;       RETURN
09                ;MESSAGE IS TERMINATED BY ESC,ESC
10                ;OR BY A NULL
11
12 01261 054431 ERR:    STA 3,ERRTN             ;SAVE RETURN
13 01262 102520         SUBZL 0,0
14 01263 040217         STA 0,TICNT+7           ;SET TO NO ECHO
15 01264 006064         JSR @CRLF0              ;PRINT CR,LF
16 01265 022425         LDA 0,@ERRTN            ;GET POINTER
17 01266 040153         STA 0,TEMA              ;AND SAVE IT
18 01267 102400         SUB 0,0
19 01270 010422         ISZ ERRTN
20
21 01271 040154 ERR0:   STA 0,TEMB              ;RESET PREVIOUS
22 01272 006073         JSR @GTCHR              ;GET A CHARACTER
23 01273 000153         TEMA
24 01274 101005         MOV 0,0,SNR
25 01275 002415         JMP @ERRTN              ;EXIT ON NULL
26 01276 006116         JSR @TYPE               ;AND TYPE IT
27 01277 006074         JSR @GTCHI              ;AND IT AGAIN AND INCREMENT
28 01300 000153         TEMA
29 01301 024154         LDA 1,TEMB              ;GET PREVIOUS
30 01302 030033         LDA 2,C33               ;AND GET AN ESC
31 01303 112415         SUB# 0,2,SNR
32 01304 132414         SUB# 1,2,SZR
33 01305 000764         JMP ERR0                ;OK, NOT AT END
34 01306 020206         LDA 0,TOCNT+6           ;WAIT FOR OUTPUT TO FINISH
35 01307 101004         MOV 0,0,SZR
36 01310 000776         JMP .-2
37 01311 002401         JMP @ERRTN              ;EXIT
38 01312 000000 ERRTN:  0                       ;SAVED RETURN
39
```

```
A 0028   .MAIN
01
02
03                ;STCH0  STORE CHARACTER IN AC0 INTO
04                ;       BUFFER
05                ;
06                ;CALLING SEQUENCE:
07                ;       JSR @STCHR
08                ;       ADDRESS OF POINTER
09                ;       RETURN (AC0,AC1,AC2 ARE PRESERVED)
10
11  0000 054421 STCH0:  STA 3,STRTN             ;SAVE RETURN
12  0001 044142         STA 1,SAC+1             ;SAVE AC1
13  0002 050143         STA 2,SAC+2             ;SAVE AC2
14  0003 033400         LDA 2,@0,3              ;GET PINTER
15  0004 151220         MOVZR 2,2               ;AND FORM AN ADDRESS
16  0005 035000         LDA 3,0,2               ;GET WORD FROM STORAGE
17  0006 024063         LDA 1,M377              ;GET 8 BIT MASK INTO LH
18  0007 100102         MOV 0,0,SZC             ;TEST WHICH HALF
19  0010 175300         MOVS 3,3                ;INTO LEFT HALF
20  0011 137402         AND 1,3,SZC             ;MASK AND TEST
21  0012 117301         ADDS 0,3,SKP
22  0013 117000         ADD 0,3
23  0014 055000         STA 3,0,2               ;RETURN UPDATED WORD
24  0015 024142         LDA 1,SAC+1             ;RESTORE THE AC'S
25  0016 030143         LDA 2,SAC+2
26  0017 034402         LDA 3,STRTN             ;GET RETURN ADDRESS
27  0020 001401         JMP 1,3                 ;AND EXIT
28  0021 000000 STRTN:  0                       ;SAVED RETURN
29
```

```
A 0029  .MAIN
01
02
03                ;KILL   SUBROUTINE TO KILL ARG LINES
04                ;
05                ;CALLING SEQUENCE:
06                ;       JSR KILL
07                ;       ARG
08                ;       (UNUSED)
09                ;       RETURN
10                ;
11                ;DELET  SUBROUTINE TO DELETE ARG
12                ;       CHARACTERS
13                ;
14                ;CALLING SEQUENCE:
15                ;       JSR DELET
16                ;       ARG
17                ;       (UNUSED)
18                ;       RETURN
19                ;
20
21 01335 054451 KILL:   STA 3,DLRTN             ;SAVE RETURN
22 01336 060277         INTDS                   ;DISABLE INTERRUPT
23 01337 020123         LDA 0,CURPT             ;SAVE CURPT
24 01340 040154         STA 0,TEMB
25 01341 021400         LDA 0,0,3               ;GET THE ARG
26 01342 040402         STA 0,.+2               ;AND STORE
27 01343 006101         JSR  @LNE               ;MOVE OVER LINES
28 01344 000000         0                       ;ARG
29 01345 063077         HALT                    ;IMPOSSIBLE
30 01346 000412         JMP DLT0                ;GO, AHEAD
31
32
```

```
A 0030   .MAIN
01
02
03 01347 054437 DELET:  STA 3,DLRTN              ;SAVE RETURN
04 01350 060277         INTDS                    ;DISABLE INTERRUPT
05 01351 020123         LDA 0,CURPT              ;SAVE CURRENT POINTER
06 01352 040154         STA 0,TEMB
07 01353 021400         LDA 0,0,3                ;GET ARGUMENT
08 01354 040402         STA 0,.+2
09 01355 006103         JSR @MVE                 ;MOVE POINTER
10 01356 000000         0
11 01357 063077         HALT                     ;IMPOSSIBLE
12
13                ;COMMON CODE FOR DELETE AND KILL
14 01360 020154 DLT0:   LDA 0,TEMB
15 01361 030123         LDA 2,CURPT
16 01362 142513         SUBL# 2,0,SNC
17 01363 000404         JMP .+4
18 01364 105000         MOV 0,1                  ;REVERSE THE ARGS
19 01365 141000         MOV 2,0
20 01366 131000         MOV 1,2
21 01367 050123         STA 2,CURPT              ;SET CURRENT POINTER
22 01370 050134         STA 2,LAST               ;STORAGE POINTER
23 01371 040153         STA 0,TEMA               ;PICKUP LOCATION
24 01372 024033         LDA 1,C33                ;SET AC1=ESCAPE CHAR
25
26
27                ;MAIN LOOP
28                ;PICK-UP AND STORE BACK INTO BUFFER
29
30 01373 006074 KILL0:  JSR @GTCHI               ;GET CHARCTER AND INCREMENT
31 01374 000153         TEMA
32 01375 006114         JSR @STCHR               ;AND STORE INTO BUFFER
33 01376 000134         LAST
34 01377 106415         SUB# 0,1,SNR             ;TEST FOR ESC
35 01400 000403         JMP .+3                  ;YES
36 01401 010134         ISZ LAST                 ;NO, INCREMENT AND KEEP GOING
37 01402 000771         JMP KILL0
38
39 01403 034403         LDA 3,DLRTN              ;GET RETURN
40 01404 060177         INTEN                    ;RE-ENABLE INTERRUPT
41 01405 001402         JMP 2,3                  ;AND EXIT
42
43 01406 000000 DLRTN:  0                        ;SAVED RETURN
```

```
A 0031   .MAIN
01
02
03                ;CHNG   CHANGE TEXT
04                ;       DO SEARCH, DELETE AND INSERT
05                ;
06                ;CALLING SEQUENCE:
07                ;       JSR CHNG
08                ;       (UNUSED)
09                ;       ERROR
10                ;       NORMAL RETURN
11
12 01407 054432 CHNG:   STA 3,CHRTN             ;SAVE RETURN
13 01410 010431         ISZ CHRTN               ;AND INCREMENT IT
14 01411 020122         LDA 0,COMPT             ;GET COMMAND POINTER
15 01412 040430         STA 0,CHNG0             ;AND SAVE IT
16 01413 006112         JSR @SERCH              ;DO A SEARCH
17 01414 000000         0
18 01415 002424         JMP @CHRTN              ;ERROR
19 01416 020122         LDA 0,COMPT
20 01417 024423         LDA 1,CHNG0
21 01420 106400         SUB 0,1          ;GET CHARACTER COUNT
22 01421 044402         STA 1,.+2               ;AND SET UP FOR A DELETE
23 01422 006066         JSR @DELT
24 01423 000000         0
25 01424 063077         HALT                    ;???
26 01425 010122         ISZ COMPT               ;INCREMENT COMMAND POINTER
27 01426 004421         JSR INSRT               ;AND DO THE INSERT
28 01427 000000         0
29 01430 063077         HALT                    ;????
30 01431 014122         DSZ COMPT
31 01432 006073         JSR @GTCHR
32 01433 000122         COMPT
33 01434 024033         LDA 1,C33
34 01435 106404         SUB 0,1,SZR
35 01436 010122         ISZ COMPT
36 01437 010402         ISZ CHRTN               ;INDEX RETURN
37 01440 002401         JMP @CHRTN              ;AND EXIT
38 01441 000000 CHRTN:  0
39 01442 000000 CHNG0:  0
```

```
A 0032  .MAIN
01                ;INSRT  ROUTINE TO INSERT CHARACTERS
02                ;       FROM THE COMMAND BUFFER
03                ;       INTO THE TEXT BUFFER
04                ;TAB ALSO INSERTS THE TAB
05                ;CALLING SEQUENCE:
06                ;       JSR INSRT
07                ;       ARG
08                ;       UNUSED (ERROR FOR TAB)
09                ;       NORMAL RETURN
10
11
12  01443 021400  TAB:    LDA 0,0,3       ;GET THE ARG
13  01444 101004          MOV 0,0,SZR     ;IT SHOULD BE 0
14  01445 001401          JMP 1,3         ;NO, ERROR
15  01446 014122          DSZ COMPT       ;BACKUP OVER THE TAB
16
17  01447 054512  INSRT:  STA 3,INRTN              ;SAVE RETURN
18  01450 020134          LDA 0,LAST              ;GET LAST
19  01451 101220          MOVZR 0,0
20  01452 101400          INC 0,0                 ;FORM (LAST/2)+1
21  01453 040031          STA 0,XDA               ;AND STORE INTO XR
22  01454 024122          LDA 1,COMPT             ;GET COMMAND POINTER
23  01455 030124          LDA 2,XMODE
24  01456 151004          MOV 2,2,SZR
25  01457 024155          LDA 1,TEMC
26  01460 125220          MOVZR 1,1               ;FORM AN ADDRESS
27  01461 044032          STA 1,XDB
28  01462 014032          DSZ XDB
29  01463 024032          LDA 1,XDB
30  01464 106400          SUB 0,1                 ;AC1=NUMBER OF FREE WORDS
31  01465 125235          MOVZR# 1,1,SNR          ;TEST MAGNITUEDE
32  01466 000467          JMP INSR2               ;LESS THAN 4 CHARACTER SPACES
33  01467 006277          INTDS                   ;DISABLE INTERRUPT
34  01470 125120          MOVZL 1,1               ;AC1=2*N=NUMBER OF CHARACTERS
35  01471 030134          LDA 2,LAST
36  01472 133001          ADD 1,2                 ;UPDATE LAST IN CASE OF RECURS
37  01473 050134          STA 2,LAST
38  01474 030123          LDA 2,CURPT             ;GET CURRENT POINTER
39  01475 155220          MOVZR 2,3               ;AC3=(CURPT)/2
40  01476 044464          STA 1,NINSR             ;NINSR=NUMBER OF CHARACTERS
41  01477 133000          ADD 1,2                 ;TO BE MOVED
42  01500 050153          STA 2,TEMA              ;END OF INSERTION AREA
43  01501 116405          SUB 0,3,SNR             ;AC3=NUMBER OF MOVES
44  01502 000405          JMP .+5                 ;NONE
45  01503 022031          LDA 0,@XDA              ;MOVE WORDS INTO UPPER MEMORY
46  01504 042032          STA 0,@XDB              ;OPENING UP A HOLE
47  01505 175404          INC 3,3,SZR             ;TEST FOR COMPLETION
48  01506 000775          JMP .-3                 ;NO, DO ANOTHER
49  01507 024033          LDA 1,C33               ;GET ESC CHARACTER
```

```
A 0033   .MAIN
01
02 01510 034451          LDA 3,INRTN
03 01511 021400          LDA 0,0,3                ;GET ARG
04 01512 030052          LDA 2,C177               ;MAKE 7 BITS
05 01513 143405          AND 2,0,SNR
06 01514 000407          JMP INSR0                ;NO ARG
07 01515 106415          SUB# 0,1,SNR             ;YES,
08 01516 000420          JMP INSR1+1              ;DON'T INSERT ESC
09 01517 006114          JSR @STCHR
10 01520 000123          CURPT
11 01521 010123          ISZ CURPT
12 01522 000414          JMP INSR1+1              ;OK, ALL DONE
13
14
15                       ;NOW, DO THE ACTUAL INSERTION
16                       ;MOVE CHARACTERS FROM THE COMMAND BUFFER INTO THE
17                       ;TEXT AREA
18
19 01523 006074 INSR0:   JSR @GTCHI               ;GET A CHARACTER AND
20 01524 000122          COMPT                    ;INCREMENT POINTER
21 01525 106415          SUB# 0,1,SNR             ;TEST IT FOR ESC
22 01526 000407          JMP INSR1                ;YES
23 01527 006114          JSR @STCHR               ;NO, STORE IT
24 01530 000123          CURPT                    ;INTO TEXT AREA
25 01531 010123          ISZ CURPT                ;INCREMENT POINTER
26 01532 014430          DSZ NINSR                ;DECREMENT COUNTER
27 01533 000770          JMP INSR0                ;STILL ROOM LEFT
28 01534 000714          JMP INSRT+1              ;NO ROOM, SHUFFLE AGAIN
29
30                       ;ALL TEXT FROM COMMAND BUFFER HAS BEEN
31                       ;ABSORBED- MOVE REST OF TEXT DOWN
32 01535 014122 INSR1:   DSZ COMPT                ;DECREMENT COMMAND POINTER
33 01536 020123          LDA 0,CURPT              ;MAKE ANOTHER COPY OF THE
34 01537 040154          STA 0,TEMB               ;CURRENT POINTER
35 01540 006074          JSR @GTCHI               ;GET CHARACTER FROM UPPER
36 01541 000153          TEMA                     ;MEMORY
37 01542 006114          JSR @STCHR               ;AND STORE INTO THE
38 01543 000154          TEMB                     ;TEXT AREA
39 01544 106415          SUB# 0,1,SNR             ;TEST FOR ESC
40 01545 000403          JMP .+3          ;YES
41 01546 010154          ISZ TEMB                 ;NO, INCREMENT AND CONTINUE
42 01547 000771          JMP INSR1+3
43
44                       ;ALL DONE, UPDATE LAST AND RETURN
45 01550 020154          LDA 0,TEMB
46 01551 040134          STA 0,LAST
47 01552 060177          INTEN                    ;RE-ENABLE INTERRUPT
48 01553 034406          LDA 3,INRTN
49 01554 001402          JMP 2,3
50                       ;PRINT "BUFFER CAPACITY EXCEEDED DURING INSERT"
51 01555 006070 INSR2:   JSR @ERROR               ;STRING PRINTER
52 01556 006410          ERMS0+ERMS0
53 01557 010402          ISZ INRTN
54 01560 002401          JMP @INRTN               ;ERROR RETURN
55 01561 000000 INRTN:   0                        ;SAVED RETURN
56 01562 000000 NINSR:   0                        ;TEMPORARY STORAGE
57
```

```
A 0034   .MAIN
01
02
03                ;N,Q     SUBROUTINE TO SEARCH
04                ;        IF FOUND, RETURN, ELSE PUNCH
05                ;        OUT THE BUFFER(IF N), YANK AND SEARCH AGAIN
06                ;        IF END-OF-FILE IS ENCOUNTERED, FAIL
07                ;
08                ;CALLING SEQUENCE:
09                ;        JSR N
10                ;        (UNUSED)
11                ;        FAIL
12                ;        NORMAL RETURN-POINTER TO CHARACTER
13                ;        IMMEDIATELY AFTER THE MATCH
14
15  01563 102521  Q:       SUBZL 0,0,SKP    ;SET MODE
16  01564 102400  N:       SUB 0,0
17  01565 040430           STA 0,NMODE      ;SAVE THE MODE
18  01566 054426           STA 3,NRTN                ;SAVE RETURN
19  01567 021400           LDA 0,0,3                 ;TEST FOR NO ARG
20  01570 101004           MOV 0,0,SZR
21  01571 001401           JMP 1,3                   ;ERROR
22  01572 101422           ISZ NRTN                  ;INCREMENT RETURN
23  01573 006113  N1:      JSR @SERCH+1              ;GO DO A SEARCH
24  01574 000000           0
25  01575 000403           JMP .+3                   ;FAILED
26  01576 101416           ISZ NRTN                  ;OK, NORMAL RETURN
27  01577 002415           JMP @NRTN
28
29                ;SEARCH FAILED
30  01600 020415           LDA 0,NMODE      ;TEST THE MODE
31  01601 101004           MOV 0,0,SZR
32  01602 000406           JMP QMDE         ;Q MODE
33  01603 006111           JSR @R0
34  01604 000001           1                         ;DO P,Y
35  01605 002407           JMP @NRTN                 ;END-OF-FILE
36  01606 101122  N2:      ISZ COMPT                 ;ADVANCE COMMAND POINTER
37  01607 000764           JMP N1                    ;AND TRY AGAIN
38  01610 006117  QMDE:    JSR @YANK0       ;DO A Y
39  01611 000001           1
40  01612 002402           JMP @NRTN        ;ERROR
41  01613 000773           JMP N2           ;PROCEED
42  01614 000000  NRTN:    0                         ;SAVED RETURN
43  01615 000000  NMODE:   0        ;SAVED MODE
```

```
A 0035  .MAIN
01
02                      ;R      PUNCH AND YANK
03                      ;
04                      ;CALLING SEQUENCE:
05                      ;       JSR R
06                      ;       ARG
07                      ;       FAIL--END-OF-FILE,PARITY ERROR, OR BUFFER FULL
08                      ;       NORMAL RETURN, CURPT=UNCHANGED
09
10 01616 054424 R:      STA 3,RRTN              ;SAVE RETURN
11 01617 021600         LDA 0,0,3               ;GET ARG
12 01620 001422         ISZ RRTN                ;INCREMENT RETURN
13 01621 101112         MOVL# 0,0,SZC           ;TEST ARG
14 01622 000141         JMP 1,3         ;FAIL, <0
15 01623 101005         MOV 0,0,SNR
16 01624 102520         SUBZL 0,0               ;0, MAKE A 1
17 01625 040416         STA 0,NR
18 01626 014122         DSZ COMPT
19 01627 000615         JSR @P0                 ;PUNCH ENTIRE BUFFER
20 01630 000000         0
21 01631 063077         HALT
22 01632 010122         ISZ COMPT
23 01633 000617         JSR @YANK0              ;GO YANK
24 01634 000000         0
25 01635 002405         JMP @RRTN               ;FAIL
26 01636 014405         DSZ NR                  ;TEST COUNT
27 01637 000767         JMP .-11                        ;KEEP GOING
28 01640 010402         ISZ RRTN                ;ALL DONE, INCREMENT RETURN
29 01641 002401         JMP @RRTN               ;AND EXIT
30 01642 000000 RRTN:   0
31 01643 000000 NR:     0
32
```

```
▲ 0036   .MAIN
01
02                ;E        END THE TAPE
03                ;         CALLING SEQUENCE
04                ;         JSR ENDTP
05                ;         UNUSED
06                ;         ERROR RETURN=PARITY ERROR OR BUFFER FULL
07                ;         NORMAL RETURN
08
09  001644 054750 ENDTP:    STA 3,NRTN        ;SAVE RETURN
10  001645 010747            ISZ NRTN          ;UPDATE IT
11  001646 020134            LDA 0,LAST        ;SEE IF ANYTHING ON THE PAGE
12  001647 024120            LDA 1,BEGIN
13  001650 100404            SUB 0,1,SZR
14  001651 000404            JMP ENDT0         ;YES, DO R
15  001652 006117            JSR @YANK0        ;NO, DO A YANK
16  001653 000000            0
17  001654 000405            JMP ENDT1         ;ERROR, TEST IT
18  001655 004741 ENDT0:     JSR R             ;CALL R
19  001656 077777            77777             ;WITH LARGE ARGUMENT
20  001657 000402            JMP ENDT1
21  001660 000775            JMP ENDT0         ;KEEP GOING???
22
23  001661 020161 ENDT1:     LDA 0,YERR        ;TEST THE Y ERROR FLAG
24  001662 101005            MOV 0,0,SNR
25  001663 010731            ISZ NRTN          ;OK, NO ERRORS
26  001664 002730            JMP @NRTN         ;RETURN
27
28
29
```

```
A 0037   .MAIN
01
02
03                      ;X      MACRO EXPANSION/DEFINITION
04                      ;
05                      ;CALLING SEQUENCE:
06                      ;       JSR XMAC
07                      ;       ARG
08                      ;       ERROR
09                      ;       NORMAL RETURN
10
11 001665 054511 XMAC:  STA 3,XRTN              ;SAVE RETURN
12 001666 020124        LDA 0,XMODE             ;CHECK FOR EXPANSION MODE
13 001667 101004        MOV 0,0,SZR
14 001670 000475        JMP XMAC4               ;YES..REQUEST ILLEGAL
15 001671 031400        LDA  2,0,3              ;NO..PICKUP ARGUMENT
16 001672 151132        MOVZL# 2,2,SZC          ;CHECK ITS SIGN
17 001673 001401        JMP 1,3                 ;-VE ARGUMENT => ERROR
18 001674 151005        MOV 2,2,SNR             ;CHECK FOR ZERO ARGUMENT
19 001675 152520        SUBZL 2,2               ;YES..MAKE IT ONE
20 001676 050127        STA 2,ARGX              ;STORE IT AWAY
21
22 001677 000073        JSR @GTCHR              ;PICKUP NEXT CHARACTER
23 001700 000122        COMPT
24 001701 024046        LDA 1,C115              ;CHECK FOR 'M'
25 001702 106415        SUB# 0,1,SNR
26 001703 000416        JMP XMAC1               ;YES => MACRO DEFINITION
27 001704 024045        LDA 1,C104              ;CHECK FOR 'D'
28 001705 106415        SUB# 0,1,SNR
29 001706 000454        JMP XMAC3               ;YES => DELETE DEFINITION
30
31                      ;MACRO CALL
32
33 001707 020125        LDA 0,XDEF              ;CHECK IF MACRO DEFINED
34 001710 101005        MOV 0,0,SNR
35 001711 000454        JMP XMAC4               ;NOT DEFINED => ERROR
36 001712 020122        LDA 0,COMPT
37 001713 040155        STA 0,TEMC              ;SAVE COMMAND POINTER
38 001714 020156        LDA 0,TEMD              ;PICKUP MACRO POINTER
39 001715 040122        STA 0,COMPT             ;'COMPT' RESET FOR SCAN
40 001716 010124        ISZ XMODE               ;INDICATE EXPANSION MODE
41 001717 034457        LDA 3,XRTN              ;EXIT
42 001720 001402        JMP 2,3
43
44                      ;MACRO DEFINITION
45
46 001721 010122 XMAC1: ISZ COMPT               ;STEP PAST 'M'
47 001722 020122        LDA 0,COMPT             ;PICKUP COMMAND POINTER
48 001723 111220        MOVZR 0,2               ;CONVERT TO ADDRESS
49 001724 024131        LDA 1,END+1             ;PICKUP END ADDRESS OF COMMAND
50 001725 146400        SUB 2,1                 ;CALCULATE LENGTH OF MACRO
51 001726 044153        STA 1,TEMA              ;SAVE AS COUNT
52 001727 010153        ISZ TEMA          ;NEED ONE MORE
53 001730 030126        LDA 2,EOMEM             ;CALCULATE POINTER FOR MACRO
54 001731 132400        SUB 1,2
55 001732 151120        MOVZL 2,2
56 001733 050156        STA 2,TEMD
57 001734 101232        MOVZR# 0,0,SZC          ;SET BYTE BIT
58 001735 010156        ISZ TEMD
59 001736 024126        LDA 1,EOMEM
```

```
  0038   .MAIN
01 01737 125400          INC 1,1                    ;SETUP AUTO INDEX
02 01740 044032          STA 1,XDB
03 01741 020131          LDA 0,END+1
04 01742 101400          INC 0,0
05 01743 040031          STA 0,XDA
06 01744 106405          SUB 0,1,SNR
07 01745 000405          JMP .+5                    ;NO SHIFTING REQUIRED
08
09                ;SHIFT MACRO BACK
10
11 01746 022031          LDA 0,@XDA
12 01747 042032          STA 0,@XDB
13 01750 014153          DSZ TEMA
14 01751 177775          JMP .-3
15
16                ;READJUST END+1
17 01752 102000          ADC 0,0
18 01753 024156          LDA 1,TEMD
19 01754 125220          MOVZR 1,1
20 01755 123000          ADD 1,0
21 01756 041125          ISZ XDEF
22
23 01757 040131 XMAC2:   STA 0,END+1
24 01760 000401          JMP @.+1
25 01761 000437          BGN
26
27 01762 011122 XMAC3:   ISZ COMPT                  ;STEP PAST 'D'
28 01763 001413          ISZ XRTN                   ;SET FOR NORMAL RETURN
29 01764 000403          JMP XMAC4+2                ;EXIT
30
31 01765 006070 XMAC4:   JSR @ERROR
32 01766 000006          ERMS7+ERMS7
33 01767 002126          LDA 0,EOMEM
34 01770 040131          STA 0,END+1
35 01771 003405          LDA 3,XRTN
36 01772 102400          SUB 0,0
37 01773 040125          STA 0,XDEF
38 01774 040124          STA 0,XMODE
39 01775 001401          JMP 1,3
40
41 01776 000000 XRTN:    0
42
43                .EOT
```

```
 0039  .MAIN
01
02                ;TAPE 3
03
04
05                ;FTST0  TEST FOR BUFFER FULL
06                ;
07                ;CALLING SEQUENCE:
08                ;       JSR FTEST
09                ;       NORMAL RETURN
10                ;       ERROR RETURN=BUFFER IS FULL, IE.
11                ;       LAST.GE.(COMPT=200)  (OCTAL)
12
13 01777 02@134 FTST0:  LDA 0,LAST
14 02@@@ 024122         LDA 1,COMPT
15 02@@1 03@124         LDA 2,XMODE
16 02@@2 151@@4         MOV 2,2,SZR
17 02@@3 024155         LDA 1,TEMC
18 02@@4 03@@53         LDA 2,C200
19 02@@5 1464@@         SUB 2,1
20 02@@6 122432         SUBZ# 1,0,SZC
21 02@@7 @@14@1         JMP 1,3          ;TEST FAILED
22 02@1@ @@14@@         JMP 0,3          ;OK, EXIT
23
24
25
26                ;TPE    HANDLE TYPEWRITER OUTPUT
27                ;PNCH   HANDLE PUNCH OUTPUT
28                ;
29                ;CALLING SEQUENCE:
30                ;       JSR (TPE OR PNCH)
31                ;       ARG
32                ;       FAIL  (ARG<0)
33                ;       NORMAL RETURN
34
35 02@11 02@116 TPE:    LDA 0,TYPE               ;GET SUBROUTINE ADDRESS
36 02@12 126520         SUBZL 1,1                ;SET AC1=-1
37 02@13 @@@4@3         JMP .+3
38
39 02@14 02@1@4 PNCH:   LDA 0,PUNCH              ;GET SUBR ADDRESS
40 02@15 1264@@         SUB 1,1          ;SET AC1=0
41 02@16 04@473         STA 0,TFLAG              ;SAVE SUBROUTINE ADDRESS
42 02@17 @44473         STA 1,TFLAG+1            ;SAVE FLAG
43 02@2@ @54473         STA 3,TRTN               ;SAVE RETURN
44 02@21 @214@@         LDA 0,0,3                ;GET ARG
45 02@22 1@1112         MOVL# 0,0,SZC
46 02@23 @@14@1         JMP 1,3                  ;<0, ERROR
47 02@24 024123         LDA 1,CURPT              ;GET CURRENT POINTER
48 02@25 1@1@@5         MOV 0,0,SNR
49 02@26 @24120         LDA 1,BEGIN              ;ARG=0, START AT BEGIN
50 02@27 @44153         STA 1,TEMA               ;SAVE OUTPUT POINTER
51 02@3@ @4@457         STA 0,NTYPE              ;SAVE ARG
52
```

```
A 0040   .MAIN
01
02
03                      ;MAIN OUTPUT LOOP
04 02031 006074 TPE1:   JSR @GTCHI              ;GET A CHARACTER AND INCREMEN
05 02032 000153         TEMA                    ;THE POINTER
06 02033 024033         LDA 1,C33               ;IS IT ESC?
07 02034 106415         SUB# 0,1,SNR
08 02035 000416         JMP TPE3                ;YES, ALL DONE
09 02036 024024         LDA 1,C11               ;NO, IS IT A TAB?
10 02037 106415         SUB# 0,1,SNR
11 02040 000432         JMP TPE4                ;YES
12 02041 024027         LDA 1,C15               ;TEST FOR FORM-FEED
13 02042 106015         ADC# 0,1,SNR
14 02043 000435         JMP TPE5                ;YES
15 02044 106415         SUB# 0,1,SNR            ;NO, TEST FOR CR
16 02045 000403         JMP .+3                 ;YES
17 02046 006443 TPE2:   JSR @TFLAG              ;OUTPUT CHARACTER
18 02047 000762         JMP TPE1                ;AND GET THE NEXT
19 02050 006441         JSR @TFLAG              ;OUTPUT
20 02051 014436         DSZ NTYPE               ;DECREMENT COUNTER
21 02052 000757         JMP TPE1                ;AND DO THE NEXT
22
23                      ;ALL DONE
24 02053 024437 TPE3:   LDA 1,TFLAG+1           ;TEST FOR PUNCH
25 02054 125004         MOV 1,1,SZR
26 02055 000413         JMP .+13
27 02056 006073         JSR @GTCHR              ;YES, TEST NEXT COMMAND
28 02057 000122         COMPT
29 02060 024047         LDA 1,C127              ;TEST FOR A W
30 02061 106414         SUB# 0,1,SZR
31 02062 000403         JMP .+3
32 02063 010122         ISZ COMPT
33 02064 000404         JMP .+4          ;YES
34 02065 004447         JSR F                   ;NO, PUNCH FORM FEED
35 02066 000000         0
36 02067 063077         HALT
37 02070 034420         LDA 3,TRTN              ;GET RETURN
38 02071 001402         JMP 2,3          ;AND EXIT
```

```
A 0041   .MAIN
01
02                      ;HANDLE A TAB
03 002072 006417 TPE4:  JSR @TFLAG
04 002073 024417        LDA 1,TFLAG+1           ;TEST FOR PUNCH
05 002074 002052        LDA 0,C177              ;GET A RUBOUT
06 002075 125005        MOV 1,1,SNR             ;IF PUNCH, FOLLOW
07 002076 006413        JSR @TFLAG              ;WITH THE RUBOUT
08 002077 000732        JMP TPE1                ;CONTINUE
09
10
11                      ;HANDLE A FORM-FEED
12 002100 024412 TPE5:  LDA 1,TFLAG+1           ;TEST FOR PUNCH
13 002101 125004        MOV 1,1,SZR
14 002102 000744        JMP TPE2                ;NO, TYPE-WRITER
15 002103 004431        JSR F
16 002104 000000        0
17 002105 063077        HALT
18 002106 000723        JMP TPE1
19 002107 000000 NTYPE: 0                       ;LINE COUNTER
20 002110 000000 TRTN:  0                       ;SAVED RETURN
21 002111 000000 TFLAG: 0
22 002112 000000        0
23
24
25                      ;SUBROUTINE TO CALCULATE NUMBER OF
26                      ;LINES .
27                      ;RETURN WITH RESULT IN AC1
28                      ;       CALL+1=ADDRESS OF STOPPING POINTER
29 002113 054422 LNN0:  STA 3,LNN2              ;SAVE LINE NUMBER
30 002114 010417        ISZ LNN2
31 002115 002120        LDA 0,BEGIN
32 002116 040153        STA 0,TEMA
33 002117 126400        SUB 1,1                 ;CLEAR COUNT
34 002120 002153 LNN1:  LDA 0,TEMA
35 002121 034412        LDA 3,LNN2
36 002122 033777        LDA 2,0-1,3
37 002123 112405        SUB 0,2,SNR
38 002124 002407        JMP @LNN2               ;DONE, RETURN
39 002125 006074        JSR @GTCHI              ;GET A CHARACTER
40 002126 000153        TEMA
41 002127 030027        LDA 2,C15               ;AND TEST FOR CR
42 002130 112405        SUB 0,2,SNR
43 002131 125401        INC 1,1                 ;YES, INDEX LINE COUNT
44 002132 000766        JMP LNN1                ;AND PROCEED
45 002133 000000 LNN2:  0
46
```

```
A 0042   .MAIN
01
02                      ;F      IF ARG<0 FAIL
03                      ;       IF ARG=0 PUNCH FORM FEED
04                      ;       IF ARG>0 PUNCH ARG INCHES OF TAPE
05                      ;
06                      ;CALLING SEQUENCE
07                      ;       JSR F
08                      ;       ARG
09                      ;       FAILURE RETURN
10                      ;       NORMAL RETURN
11
12 0 2134 0 2140 0 F:       LDA 0,0,3               ;GET ARG
13 0 2135 1 0 1112          MOVL# 0,0,SZC           ;TEST FOR ARG<0
14 0 2136 0 0 140 1         JMP 1,3                 ;YES, ERROR
15 0 2137 0 5443 0          STA 3,FRTN              ;SAVE RETURN
16 0 2140 1 0 5125          MOVZL 0,1,SNR           ;TEST FOR ZERO
17 0 2141 0 0 0 4 0 6       JMP F1          ;YES, PUNCH FORM-FEED
18 0 2142 125120            MOVZL 1,1               ;NO, FORM 10*ARG
19 0 2143 123120            ADDZL 1,0               ;IN AC0
20
21 0 2144 0 0 4411 F0:      JSR PFEED               ;PUNCH TAPE FEED
22 0 2145 0 34422           LDA 3,FRTN              ;GET RETURN ADDRESS
23 0 2146 0 0 140 2         JMP 2,3                 ;AND EXIT
24
25                      ;PUNCH FORM FEED
26 0 2147 0 2000 23 F1:     LDA 0,C10               ;PRECEED BY .8" OF TAPE
27 0 2150 0 0 4405          JSR PFEED
28 0 2151 0 2000 26         LDA 0,C14               ;GET FEED
29 0 2152 0 0 5104          JSR @PUNCH              ;AND PUNCH IT
30 0 2153 0 2000 30         LDA 0,C20               ;AND FOLLOW WITH
31 0 2154 0 0 0 7700        JMP F0                  ;1.6" OF FEED
32
33                      ;AC0 CONTAINS NUMBER OF TAPE FEEDS TO BE PUNCHED
34 0 2155 0 2456 4 PFEED:   LDA 1,CNVTB+1
35 0 2156 1 0 6433          SUBZ# 0,1,SNC           ;SEE IF >1000
36 0 2157 121000            MOV 1,0                 ;YES
37 0 2160 0 4 0 410         STA 0,NFEED             ;SAVE THE COUNT
38 0 2161 0 5441 0          STA 3,NFEED+1           ;SAVE RETURN
39 0 2162 1 0 2400          SUB 0,0         ;CLEAR AC0
40 0 2163 0 0 5104          JSR @PUNCH              ;AND PUNCH THE BLANK
41 0 2164 0 1 0 0 4         DSZ NFEED               ;TEST THE COUNT
42 0 2165 0 0 0 775         JMP .-3                 ;NOT DONE
43 0 2166 0 0 2403          JMP @NFEED+1            ;ALL DONE, EXIT
44 0 2167 0 0 0 0 0 FRTN:   0
45 0 2170 0 0 0 0 0 NFEED:  0
46 0 2171 0 0 0 0 0         0
47
```

```
A 0043   .MAIN
01
02
03                ;YANK   KILL BUFFER AND READ INTO IT
04                ;
05                ;APPND  APPEND TEXT FROM READER INTO BUFFER
06                ;
07                ;CALLING SEQUENCE:
08                ;       JSR (Y OR APPND)
09                ;       (IGNORED)
10                ;       RETURN IF COMMAND CANNOT BE DONE
11                ;                (OR IF NO INPUT TAPE)
12                ;       NORMAL RETURN
13                ;
14                ;INPUT FROM READER STOPS WHEN FORM-FEED IS
15                ;ENCOUNTERED OR IF BUFFER BECOMES FULL IN WHICH
16                ;CASE A MESSAGE IS PRINTED
17
18 02172 020120 YANK:   LDA 0,BEGIN             ;INPUT INTO BEGINNING
19 02173 000402         JMP .+2
20
21 02174 020134 APPND:  LDA 0,LAST              ;INPUT INTO LAST
22 02175 040134         STA 0,LAST              ;SET LAST
23 02176 040123         STA 0,CURPT             ;SET CURRENT POINTER
24 02177 054473         STA 3,YRTN              ;SAVE SUBROUTINE RETURN
25 02200 010472         ISZ YRTN                ;INCREMENT RETURN
26 02201 102400         SUB 0,0
27 02202 040161         STA 0,YERR      ;CLEAR YANK ERROR FLAG
28 02203 000771         JSR @FTEST              ;TEST FOR BUFFER FULL
29 02204 000404         JMP .+4                 ;NO
30 02205 000770         JSR @ERROR              ;YES, PRINT "BUFFER
31 02206 000462         ERMS1+ERMS1             ;IS FULL CANNOT DO A"
32 02207 002463         JMP @YRTN               ;ERROR RETURN
33                ;NOW, CALCULATE CURRENT LINE NUMBER
34 02210 000172         JSR @LNN
35 02211 000134         LAST
36 02212 044461         STA 1,LCNT
37 02213 102400         SUB 0,0
38 02214 040462         STA 0,LCNT+3    ;CLEAR ERROR FLAG
39 02215 102400 A0:     SUB 0,0
40 02216 040456         STA 0,LCNT+1            ;CLEAR ERROR COUNT
41 02217 020050         LDA 0,C134
42 02220 040455         STA 0,LCNT+2            ;SET MAX LINE SIZE
```

```
A 0044   .MAIN
01
02
03                ;MAIN SUBROUTINE LOOP TO GET A CHARACTER
04                ;FROM THE READER, STORE IT INTO THE TEXT
05                ;BUFFER AND TEST FOR TERMINATION.
06                ;BAD PARITY CHARACTERS ARE PLACED INTO THE
07                ;BUFFER AS 0.
08
09  02221  006106  A1:    JSR @READ                ;GET A CHAR INTO AC0
10  02222  000434          JMP A4                   ;NO MORE CHARS
11  02223  024027          LDA 1,C15                ;TEST FOR FORM-FEED
12  02224  106015          ADC# 0,1,SNR
13  02225  000435          JMP A3                   ;YES, TERMINATE AND EXIT
14  02226  030033          LDA 2,C33
15  02227  112415          SUB# 0,2,SNR             ;TEST FOR ESC
16  02230  000403          JMP .+3          ;YES, TREAT AS AN ERROR
17  02231  101004          MOV 0,0,SZR              ;TEST FOR AN ERROR
18  02232  000403          JMP .+3
19  02233  010441          ISZ LCNT+1               ;YES, SET FLAG
20  02234  020060          LDA 0,C134               ;GET "0"
21  02235  006114          JSR @STCHR               ;STORE THE CHARACTER
22  02236  000134          LAST
23  02237  010134          ISZ LAST
24  02240  106415          SUB# 0,1,SNR             ;TEST FOR CARRIAGE RETURN
25  02241  000403          JMP .+3
26  02242  014433          DSZ LCNT+2               ;TEST FOR BIG LINE
27  02243  000756          JMP A1                   ;NO, GET THE NEXT
28
```

```
A 0045   .MAIN
01
02                      ;END-OF-LINE, TEST FOR ERRORS WITHIN THE LINE
03 02244 004433                 JSR ERTST
04
05
06                      ;NOW, INCREMENT THE LINE COUNT AND TEST FOR
07                      ;BUFFER FULL
08 02245 020430                 LDA 0,LCNT+2            ;TEST FOR AN
09 02246 101004                 MOV 0,0,SZR             ;OVERFLOW
10 02247 021424                 ISZ LCNT
11 02250 006071                 JSR @FTEST
12 02251 000744                 JMP A0          ;OK, GET THE NEXT LINE
13 02252 011161                 ISZ YERR        ;SET ERROR FLAG
14 02253 006070                 JSR @ERROR              ;BUFFER IS FULL
15 02254 006520                 ERMS2+ERMS2             ;PRINT MESSAGE
16 02255 000404                 JMP .+4
17 02256 020123 A4:             LDA 0,CURPT     ;SEE IF ANYTHING
18 02257 024134                 LDA 1,LAST      ;WAS READ
19 02260 100645                 SUB 0,1,SNR
20 02261 010415                 ISZ LCNT+3              ;SET ERROR FLAG
21 02262 020033 A3:             LDA 0,C33
22 02263 006114                 JSR @STCHR              ;TERMINATE THE BUFFER
23 02264 000134                 LAST
24 02265 004412                 JSR ERTST               ;TEST FOR ERRORS
25 02266 020410                 LDA 0,LCNT+3    ;TEST THE ERROR FLAG
26 02267 100105                 MOV 0,0,SNR
27 02270 011402                 ISZ YRTN        ;NO ERRORS, INCREMENT RETURN
28 02271 000241                 JMP @YRTN               ;EXIT
29 02272 000000 YRTN:   0               ;SAVED RETURN
30 02273 000000 LCNT:   0
31 02274 000000         0
32 02275 000000         0
33 02276 000000         0
34
35
36
```

```
A 0046   .MAIN
01
02                      ;TEST FOR PARITY ERRORS
03 02277 054415 ERTST:  STA 3,ERTS0             ;SAVE RETURN
04 02300 02v774         LDA 0,LCNT+1
05 02301 101005         MOV 0,0,SNR
06 02302 001400         JMP 0,3                 ;NO, RETURN
07 02303 006070         JSR @ERROR              ;YES, PRINT MESSAGE
08 02304 006737         ERMS5+ERMS5
09 02305 024766         LDA 1,LCNT
10 02306 125400         INC 1,1
11 02307 006065         JSR @DECML              ;PRINT LINE NUMBER
12 02310 006064         JSR @CRLF0
13 02311 017765         ISZ LCNT+3      ;SET ERROR FLAG
14 02312 011761         ISZ YERR
15 02313 002411         JMP @ERTS0
16 02314 000000 ERTS0:  0
17
18                      ;TYPE C(AC1) IN DECIMAL
19 02315 054422 DCML0:  STA 3,DCML2             ;SAVE RETURN
20 02316 030061         LDA 2,M4                ;GET COUNTER
21 02317 050153         STA 2,TEMA              ;AND SAVE IT
22 02320 030420         LDA 2,CNVTB             ;GET POINTER TO CONVERSION
23 02321 050020         STA 2,XUA               ;TABLE
24 02322 032020 DCML1:  LDA 2,@XUA              ;GET POWER OF 10
25 02323 020042         LDA 0,C60               ;GET "0"
26 02324 146512         SUBL# 2,1,SZC           ;IS NUMBER ,GE. POWER OF 10?
27 02325 000404         JMP .+4                 ;NO
28 02326 146400         SUB 2,1                 ;YES, SUBTRACT
29 02327 101400         INC 0,0                 ;AND INCREMENT DIGIT
30 02330 000774         JMP .-4                 ;AND PROCEED
31 02331 044154         STA 1,TEMB              ;SAVE PARTIAL DIGIT
32 02332 006116         JSR @TYPE               ;AND TYPE THE DIGIT
33 02333 024154         LDA 1,TEMB              ;GET DIGIT BACK
34 02334 011753         ISZ TEMA                ;AND TEST THE COUNT
35 02335 000765         JMP DCML1               ;DO ANOTHER
36 02336 002411         JMP @DCML2              ;ALL DONE
37 02337 000000 DCML2:  0                       ;SAVED RETURN
38
39                      ;CONVERSION TABLE
40 02340 002340 CNVTB:  .
41        000012        .RDX 10
42 02341 001750         1000
43 02342 000144         100
44 02343 000012         10
45 02344 000001         1
46
47        000010        .RDX 8
48
```

```
A 00047  .MAIN
01
02
03                ;INITIALIZING SUBROUTINE
04                ;
05                ;CALLING SEQUENCE:
06                ;        JSR @INIT
07                ;        POINTER TO MESSAGE
08                ;        RETURN IF ERROR
09                ;        RETURN IF (2)
10                ;        RETURN IF (1)
11
12 02345 054423 INIT0:  STA 3,INTRT          ;SAVE RETURN
13 02346 021400          LDA 0,0,3           ;GET ARG
14 02347 040402          STA 0,.+2
15 02350 000070          JSR @ERROR          ;CALL THE STRING PRINTER
16 02351 000000          0
17                ;NOW, DO AN INPUT
18 02352 102400          SUB 0,0
19 02353 006100          JSR @KINT0          ;INITIALIZE KEYBOARD
20 02354 006077          JSR @KEYRD
21 02355 040153          STA 0,TEMA          ;SAVE CHARACTER
22 02356 006064          JSR @CRLF0          ;TYPE CR,LF
23 02357 020153          LDA 0,TEMA          ;GET CHARACTER
24 02360 034401          LDA 3,INTRT         ;GET RETURN
25 02361 024042          LDA 1,C60           ;GET AN ASCII 0
26 02362 122405          ADC 1,0,SNR
27 02363 006143          JMP 3,3             ;A ONE
28 02364 100400          NEG 0,0
29 02365 102404          COM 0,0,SZR
30 02366 006141          JMP 1,3             ;ERROR
31 02367 006142          JMP 2,3             ;A TWO
32 02370 000000 INTRT:  0                    ;SAVED RETURN
```

```
▲ 0048   .MAIN
01
02
03                ;GNPR0  GENERATE EVEN PARITY OF
04                ;       THE 7 BIT CHARACTER IN AC0
05                ;       AC1,AC2 ARE LOST
06
07 02371 152620 GNPR0:  SUBZR 2,2                ;GET 10000000 00000000
08 02372 105221         MOVZR 0,1,SKP
09 02373 125200         MOVR 1,1
10 02374 125112         MOVL# 1,1,SZC
11 02375 147000         ADD 2,1
12 02376 125004         MOV 1,1,SZR
13 02377 000774         JMP .-4
14 02400 151302         MOVS 2,2,SZC             ;PUT INTO PROPER POSITION
15 02401 143000         ADD 2,0                  ;IT WAS ODD, MAKE EVEN
16 02402 001400         JMP 0,3                  ;RETURN
17
18                ;TSPR0  TEST PARITY OF AC0
19                ;
20                ;CALLING SEQUENCE:
21                ;       JSR TSPT0
22                ;       NORMAL : AC0=7 BIT CHARACTER
23                ;       (AC0 IS CLEARED IF PARITY IS ODD)
24                ;
25                ;       AC1,AC2 ARE LOST
26
27 02403 152620 TSPR0:  SUBZR 2,2                ;GET CONSTANT
28 02404 105221         MOVZR 0,1,SKP
29 02405 125200         MOVR 1,1
30 02406 125112         MOVL# 1,1,SZC
31 02407 147000         ADD 2,1
32 02410 125004         MOV 1,1,SZR
33 02411 000774         JMP .-4
34 02412 024052         LDA 1,C177
35 02413 123402         AND 1,0,SZC
36 02414 102400         SUB 0,0                  ;IT WAS ODD, CLEAR IT
37 02415 001400         JMP 0,3                  ;RETURN
38
```

```
A 00049  .MAIN
01
02
03                      ;PUN1   PUNCH AC0
04                      ;
05                      ;CALLING SEQUENCE:
06                      ;       JSR PUN
07                      ;       RETURN
08
09 02416 054426 PUN1:   STA 3,PNRTN             ;SAVE RETURN
10 02417 024135         LDA 1,PNPAR             ;GET PUNCH PARITY MODE
11 02420 125005         MOV 1,1,SNR             ;TEST MODE
12 02421 006072         JSR @GNPAR              ;GENERATE
13 02422 024160         LDA 1,USTTO             ;TEST FOR TELETYPE
14 02423 125005         MOV 1,1,SNR
15 02424 000546         JMP TYPE2               ;YES, USE TTO
16 02425 006277         INTDS                   ;DISABLE INTERRUPT
17 02426 024177         LDA 1,PNCNT+6           ;IS PUNCH ACTIVE?
18 02427 125004         MOV 1,1,SZR
19 02430 000405         JMP .+5                 ;YES
20 02431 061113         DOAS 0,PTP              ;NO, ACTIVATE PUNCH
21 02432 101177         ISZ PNCNT+6             ;SET PUNCH RUN
22 02433 000407         JMP PUN0                ;AND EXIT
23
24                      ;PUNCH IS ACTIVE, TRY TO PLACE CHARACTER
25                      ;INTO THE BUFFER
26 02434 006177         INTEN                   ;ENABLE INTERRUPT
27 02435 006075         JSR @INBF0
28 02436 000171         PNCNT                   ;POINTER TO DEVICE-DATA-BLOCK
29 02437 000775         JMP .-3                 ;BUFFER FULL, TRY AGAIN
30 02440 000402         JMP PUN0                ;NOT FULL
31 02441 101177         ISZ PNCNT+6             ;RESET ACTIVE
32 02442 006177 PUN0:   INTEN                   ;ENABLE INTERRUPT
33 02443 002401         JMP @PNRTN              ;EXIT
34 02444 000000 PNRTN:  0                       ;SAVED RETURN
35
36                      ;PUN    PUNCH AC0
37                      ;       PROCESS CARRIAGE RETURNS
38 02445 054411 PUN:    STA 3,PUN2              ;SAVE RETURN
39 02446 024027         LDA 1,C15               ;TEST FOR CARRIAGE RETURN
40 02447 100405         SUB 0,1,SNR
41 02450 000403         JMP .+3
42 02451 004745         JSR PUN1
43 02452 002404         JMP @PUN2               ;EXIT
44 02453 004743         JSR PUN1
45 02454 020025         LDA 0,C12               ;YES, FOLLOW WITH LINE-FEED
46 02455 000774         JMP .-4
47 02456 000000 PUN2:   0
```

```
▲ 0050   .MAIN
0001
0002
0003                    ;TYPE   TYPE CHARACTER IN AC0
0004                    ;       AND PRINT ALL INVISIBLE CHARACTERS
0005
0006 002457 020027 CRLF:   LDA 0,C15                ;TYPE CR,LF
0007 002460 054454 TYP:    STA 3,TYRTN              ;SAVE RETURN
0008 002461 040454         STA 0,TYRTN+1            ;SAVE CHARACTER
0009 002462 024007         LDA 1,TOCNT+7            ;GET TTY MODE
0010 002463 125005         MOV 1,1,SNR              ;TEST FOR PUNCHING
0011 002464 000407         JMP .+7                  ;NO, OK
0012 002465 024006         LDA 1,TOCNT+6            ;YES, WAIT UNTIL DONE
0013 002466 125004         MOV 1,1,SZR
0014 002467 000776         JMP .-2
0015 002470 044007         STA 1,TOCNT+7            ;SET NEW MODE
0016 002471 101020         MOVZ 0,0                 ;CLEAR CARRY
0017 002472 063077         HALT                     ;TURN PUNCH OFF
0018 002473 024052         LDA 1,C177
0019 002474 123400         AND 1,0                  ;MAKE 7 BIT CHARACTER
0020 002475 024033         LDA 1,C33                ;TEST AC0 FOR ESC
0021 002476 106415         SUB 0,1,SNR
0022 002477 020036         LDA 0,C44                ;YES, MAKE A "$"
0023 002500 024027         LDA 1,C15
0024 002501 106415         SUB# 0,1,SNR             ;TEST FOR CARRIAGE RETURN
0025 002502 000435         JMP TP0          ;YES
0026 002503 024024         LDA 1,C11                ;NO, TEST FOR A TAB
0027 002504 106415         SUB# 0,1,SNR
0028 002505 000437         JMP TP1                  ;YES
0029
0030 002506 024430         LDA 1,TBL                ;NOW, LOOK FOR INVISIBLES
0031 002507 044022         STA 1,XUC
0032 002510 026022         LDA 1,@XUC               ;GET LOWER LIMIT
0033 002511 032022         LDA 2,@XUC               ;GET UPPER LIMIT
0034 002512 125005         MOV 1,1,SNR              ;TEST FOR END
0035 002513 000412         JMP TYPE3-1              ;YES
0036 002514 122513         SUBL# 1,0,SNC            ;NO, TEST FOR CHAR.GE.LOWER
0037 002515 112512         SUBL# 0,2,SZC            ;YES, TEST FOR CHAR.LE.UPPER
0038 002516 000772         JMP .-6                  ;NO, TEST NEXT PAIR
0039 002517 020051         LDA 0,C136               ;GET ^
0040 002520 010121         ISZ CHCNT
0041 002521 004454         JSR TYPE0                ;AND OUTPUT IT
0042 002522 020413         LDA 0,TYRTN+1            ;GET CHARACTER
0043 002523 024044         LDA 1,C100
0044 002524 123000         ADD 1,0
0045 002525 010121         ISZ CHCNT                ;INDEX COUNT
0046 002526 004447 TYPE3:  JSR TYPE0                ;OUTPUT CHARACTER
0047 002527 024406         LDA 1,TYRTN+1
0048 002530 020027         LDA 0,C15                ;TEST FOR FORM-FEED
0049 002531 122415         ADC# 1,0,SNR
0050 002532 000727         JMP TYP+1                ;YES
0051 002533 024401         JMP @TYRTN               ;EXIT
0052 002534 000000 TYRTN:  0                ;SAVED RETURN
0053 002535 000000         0                        ;SAVED CHARACTER
0054 002536 000750 TBL:    TABL-1           ;POINTER TO TRANSLATION TABLE
```

```
▲ 0051   .MAIN
01
02
03                      ;A CARRIAGE-RETURN, RESET CHARACTER COUNTER
04 0002537 0004436 TP0:     JSR TYPE0
05 0002540 0002025          LDA 0,C12                   ;FOLLOW WITH A LINE-FEED
06 0002541 1264000          SUB 1,1
07 0002542 0044121          STA 1,CHCNT
08 0002543 0000763          JMP TYPE3                   ;AND TYPE CHARACTER
09
10                      ;HANDLE A TAB
11 0002544 0024152 TP1:     LDA 1,TABSM                 ;TEST FOR SIMULATION
12 0002545 1250004          MOV 1,1,SZR
13 0002546 0000404          JMP .+4                     ;YES
14 0002547 0004426          JSR TYPE0                   ;DON'T SIMULATE TAB
15 0002550 0002052          LDA 0,C177                  ;FOLLOW WITH A RUBOUT
16 0002551 0000770          JMP TP0+2
17
18 0002552 0020121          LDA 0,CHCNT                 ;GET COUNT
19 0002553 0024023          LDA 1,C10
20 0002554 1224422          SUBZ 1,0,SZC
21 0002555 0000777          JMP .-1                     ;REDUCE TO MODULO 8
22 0002556 0040121          STA 0,CHCNT
23 0002557 0002034          LDA 0,C40                   ;GET A SPACE
24 0002560 0004415          JSR TYPE0
25 0002561 0010121          ISZ CHCNT                       ;TEST COUNT
26 0002562 0000775          JMP .-3                     ;CONTINUE
27 0002563 0002751          JMP @TYRTN                  ;ALL DONE, EXIT
28
29
30
31                  .EOT
```

```
                    ;TAPE 4


                    ;TYPE0  TYPE C(AC0)
                    ;
                    ;CALLING SEQUENCE:
                    ;       JSR TYPE0 (CHAR IN AC0)
                    ;       RETURN= CHARACTER PLACED INTO BUFFER
                    ;       AC1,AC2 ARE LOST

02564 034600 TYPE2:     LDA 3,PNRTN             ;CALLED BY PUN
02565 024207            LDA 1,TOCNT+7           ;TEST FOR TYPING MODE
02566 125004            MOV 1,1,SZR
02567 000406            JMP .+6                 ;NO
02570 024206            LDA 1,TOCNT+6           ;YES, WAIT FOR IO
02571 125044            MOVO 1,1,SZR            ;TO FINISH
02572 000776            JMP .-2
02573 010207            ISZ TOCNT+7             ;SET THE NEW MODE
02574 063077            HALT                    ;HALT TO TURN ON PUNCH

02575 054420 TYPE0:     STA 3,TORTN             ;SAVE RETURN
02576 060277            INTDS                   ;DISABLE INTERRUPT
02577 024206            LDA 1,TOCNT+6           ;TEST FOR TTO ACTIVE
02600 125004            MOV 1,1,SZR
02601 000405            JMP .+5                 ;IT IS ACTIVE
02602 061111            DOAS 0,TTO              ;NO, OUTPUT THE CHARACTER
02603 010206            ISZ TOCNT+6             ;AND SET ACTIVE
02604 000407            JMP TYPE1               ;AND EXIT

                    ;OK, THE DEVICE IS ACTIVE, PLACE CHARACTER INTO THE BUFFER
02605 060177            INTEN
02606 004503            JSR INBUF
02607 000200            TOCNT                   ;POINTER TO DEVICE DATA BLOCK
02610 000775            JMP .-3
02611 000402            JMP TYPE1               ;NOT FULL
02612 010206            ISZ TOCNT+6             ;RESET ACTIVE
02613 060177 TYPE1:     INTEN               ;ENABLE INTERRUPT
02614 002401            JMP @TORTN              ;AND RETURN
02615 000000 TORTN:     0
```

```
A 0053   .MAIN
01
02
03
04
05                ;HANDLE TTI INTERRUPTS
06                ;ENTRY HERE ON INTERRUPT AFTER CPU STATE
07                ;HAS BEEN SAVED
08
09 002616 006010 TIINT:  DIAC 0,TTI                        ;READ AND CLEAR
10 002617 034052          LDA 3,C177
11 002620 117405          AND 0,3,SNR              ;TEST FOR NULLS
12 002621 000436          JMP TINT1                ;IGNORE NULLS
13 002622 030222          LDA 2,TICNT+10
14 002623 151004          MOV 2,2,SZR              ;TEST FOR READER
15 002624 000416          JMP TINT2                   ;YES
16 002625 024006          LDA 1,M3          ;NO, KEYBOARD
17 002626 167005          ADD 3,1,SNR       ;TEST FOR AC
18 002627 002107          JMP @RESET        ;YES, DO A RESET
19 002630 024032          LDA 1,M24         ;TEST FOR AT
20 002631 167005          ADD 3,1,SNR
21 002632 002110          JMP @RESET+1      ;YES
22 002633 024052          LDA 1,C177        ;TEST FOR 175,176
23 002634 166245          ADCOR 3,1,SNR
24 002635 020033          LDA 0,C33         ;YES, SET TO ESC(33)
25 002636 030217          LDA 2,TICNT+7     ;TEST FOR ECHO
26 002637 151004          MOV 2,2,SZR
27 002640 000543          JMP DISIN                ;KEYBOARD=NO ECHO
28 002641 000404          JMP TINT3         ;KEYBOARD=DON'T TEST PARITY
29 002642 024137 TINT2:   LDA 1,RDPAR
30 002643 125005          MOV 1,1,SNR              ;TEST FOR READER PARITY
31 002644 006115          JSR @TSPAR               ;YES, TEST PARITY
32 002645 024052 TINT3:   LDA 1,C177
33 002646 123400          AND 1,0                  ;MAKE 7 BITS
34 002647 030025          LDA 2,C12                ;TEST FOR LINE-FEED
35 002650 112403          SUB 0,2,SNR
36 002651 000404          JMP TINT1                ;YES, IGNORE LINE-FEEDS
37
38
39 002652 004437          JSR INBUF                ;PLACE CHAR IN BUFFER
40 002653 000210          TICNT                    ;AND PLACE INTO BUFFER
41 002654 063077          HALT                     ;BUFFER FULL=ERROR
42 002655 000402          JMP TINT1                ;NOT FULL, CLEAR ACTIVE
43 002656 000525          JMP DISIN                ;FULL, DISMIS INTERRUPT
44
45
46
47
48                ;CLEAR ACTIVE
49 002657 102400 TINT1:   SUB 0,0
50 002660 040216          STA 0,TICNT+6            ;CLEAR ACTIVE FLAG
51 002661 000522          JMP DISIN                ;DISMISS THE INTERRUPT
52 002662 177754 M24:     -24
53
54
```

```
A 0054   .MAIN
01
02
03                ;HANDLE READER INTERRUPT
04                ;ENTRY IS MADE HERE ON INTERRUPT AFTER CPU STATE
05                ;HAS BEEN SAVED
06
07 002663 006612 RDINT:  DIAC 0,PTR              ;READ AND CLEAR
08 002664 101005         MOV 0,0,SNR             ;TEST FOR NULL
09 002665 002054         LDA 0,C377      ;MAKE RUBOUT
10 002666 024162         LDA 1,RDCNT             ;GET COUNT
11 002667 030167         LDA 2,RDCNT+5           ;GET SIZE
12 002670 132004         ADC 1,2,SZR             ;SEE IF TIME TO RESTART PTR
13 002671 006112         NIOS PTR                ;YES
14 002672 024052         LDA 1,C177
15 002673 030137         LDA 2,RDPAR             ;GET READER PARITY FLAG
16 002674 151004         MOV 2,2,SZR
17 002675 123401         AND 1,0,SKP             ;DON'T TEST IT
18 002676 006115         JSR @TSPAR              ;TEST PARITY
19 002677 030025         LDA 2,C12               ;IGNORE LINE-FEED
20 002700 112405         SUB 0,2,SNR             ;AND RUBOUT
21 002701 000406         JMP RDNT0               ;GO FETCH ANOTHER
22 002702 004407         JSR INBUF               ;NEITHER, STORE INTO BUFFER
23 002703 000162         RDCNT
24 002704 063077         HALT                    ;IMPOSSIBLE
25 002705 000401         JMP .+1
26 002706 000475         JMP DISIN               ;DISMIS INTERRUPT
27
28 002707 006112 RDNT0:  NIOS PTR
29 003710 000473         JMP DISIN
30
```

```
A 0055   .MAIN
01
02
03                ;INBUF  PLACE A CHARACTER FROM
04                ;       AC0 INTO BUFFER
05                ;       (AND DISABLE THE INTERRUPT)
06                ;
07                ;CALLING SEQUENCE:
08                ;       JSR INBUF
09                ;       POINTER TO DEVICE-DATA-BLOCK
10                ;       RETURN-BUFFER FULL, NO STORAGE
11                ;       RETURN-BUFFER NOT FULL
12                ;       RETURN-BUFFER BECAME FULL (ACTIVE CLEARED)
13
14 02711 060277 INBUF:  INTDS            ;DISABLE INTERRUPT
15 02712 054436          STA 3,INBRT     ;SAVE RETURN AND
16 02713 010435          ISZ INBRT       ;INCREMENT IT
17 02714 031400          LDA 2,0,3               ;GET COUNTER
18 02715 025000          LDA 1,0,2
19 02716 035005          LDA 3,5,2               ;GET MAX VALUE
20 02717 136415          SUB# 1,3,SNR            ;TEST FOR FULL
21 02720 002436          JMP @INBRT              ;YES, EXIT
22 02721 125400          INC 1,1                 ;NO, INCREMENT COUNTER
23 02722 045000          STA 1,0,2               ;AND STORE IT
24 02723 136425          SUB 1,3,SNR             ;TEST IF NOW FULL
25 02724 010424          ISZ INBRT               ;YES
26 02725 055006          STA 3,6,2               ;IF YES, CLEAR ACTIVE
27 02726 010422          ISZ INBRT               ;INCREMENT RETURN AGAIN
28 02727 011002          ISZ 2,2                 ;INCREMENT INPUT POINTER
29 02730 035002          LDA 3,2,2               ;GET INPUT POINTER
30 02731 025003          LDA 1,3,2               ;GET MAXIMUM VALUE
31 02732 166404          SUB 3,1,SZR             ;TEST FOR MAX
32 02733 000403          JMP .+3                 ;NO
33 02734 035004          LDA 3,4,2               ;YES, RESET TO BEGINNING
34 02735 055002          STA 3,2,2
35 02736 171220          MOVZR 3,2               ;NOW, STORE CHARACTER
36 02737 035000          LDA 3,0,2               ;GET STORAGE WORD
37 02740 024063          LDA 1,M377
38 02741 101002          MOV 0,0,SZC
39 02742 175300          MOVS 3,3                        ;INTO LEFT HALF
40 02743 137402          AND 1,3,SZC
41 02744 117301          ADDS 0,3,SKP            ;INTO LEFT HALF
42 02745 117000          ADD 0,3
43 02746 055000          STA 3,0,2               ;STORE BACK
44 02747 002401          JMP @INBRT              ;AND EXIT
45 02750 000000 INBRT:   0                       ;SAVED RETURN
```

```
A 0056   .MAIN
01
02
03                ;OUTBF   TAKE A CHARACTER OUT OF THE
04                ;        BUFFER  AND PLACE IT INTO AC0
05                ;
06                ;CALLING SEQUENCE:
07                ;        JSR OUTBF
08                ;        POINTER TO DEVICE-DATA-BLOCK
09                ;        RETURN IF BUFFER NOT EMPTY (CHAR IN AC0)
10                ;        RETURN IF BUFFER WAS EMPTY (ACTIVE CLEARED)
11                ;
12                ;DEVICE DATA BLOCK (DDB) FORMAT:
13                ;        0.  CHARACTER COUNT
14                ;        1.  OUTPUT PONTER
15                ;        2.  INPUT PONTER
16                ;        3.  END OF BUFFER
17                ;        4.  START OF BUFFER
18                ;        5.  SIZE OF BUFFER
19                ;        6.  DEVICE ACTIVE FLAG
20
21  02751 03140  OUTBF:  LDA 2,0,3                ;GET POINTER TO DDB
22  02752 02100          LDA 0,0,2                ;TEST THE COUNTER
23  02753 10105          MOV 0,0,SNR
24  02754 04106          STA 0,6,2                ;AND CLEAR ACTIVE IF ZERO
25  02755 10103          MOV 0,0,SNR
26  02756 03402          JMP 2,3          ;BUFFER WAS EMPTY, EXIT
27  02757 01500          DSZ 0,2                  ;OK, DECREMENT COUNTER
28  02760 00401          JMP .+1
29  02761 01101          ISZ 1,2                  ;INCREMENT OUTPUT POINTER
30  02762 02101          LDA 0,1,2                ;GET NEW VALUE
31  02763 02503          LDA 1,3,2                ;GET MAX VALUE
32  02764 10644          SUB 0,1,SZR              ;TEST FOR MAX
33  02765 00403          JMP .+3                  ;NO
34  02766 02104          LDA 0,4,2                ;YES, RESET TO BEGINNING
35  02767 04101          STA 0,1,2
36  02770 11122          MOVZR 0,2                ;FORM ADDRESS
37  02771 02454          LDA 1,C377
38  02772 02100          LDA 0,0,2                ;GET WORD
39  02773 10102          MOV 0,0,SZC              ;TEST THE CARRY
40  02774 10130          MOVS 0,0                 ;IN LEFT HALF
41  02775 12340          AND 1,0          ;CHAR INTO AC0
42  02776 00141          JMP 1,3                  ;EXIT
43
```

```
A 00057   .MAIN
01
02
03                ;HANDLE PUNCH INTERRUPTS
04                ;ENTRY HERE ON INTERRUPT AFTER THE
05                ;CPU STATE HAS BEEN SAVED
06
07 002777 060213 PNINT:  NIOC PTP                    ;CLEAR DONE
08 003000 004751         JSR OUTBF                   ;GET A CHARACTER FROM
09 003001 000171         PNCNT                       ;THE BUFFER
10 003002 061113         DOAS 0,PTP                  ;NOT EMPTY
11
12                ;DISIN  DISMIS INTERRUPT
13                ;       AND RESTORE CPU STATE
14
15 003003 002145 DISIN:  LDA 0,SCRY
16 003004 120000         MOVR 0,0                    ;RESTORE CRY
17 003005 002146         LDA 0,SCRY+1                ;AND THE AC'S
18 003006 024147         LDA 1,SCRY+2
19 003007 030150         LDA 2,SCRY+3
20 003010 034151         LDA 3,SCRY+4
21 003011 061177         INTEN                       ;ENABLE INTERRUPT
22 003012 002000         JMP @0                      ;AND EXIT
23
24                ;HANDLE TTO INTERRUPT
25                ;ENTER ON INTERRUT AFTER THE CPU STATE
26                ;HAS BEEN SAVED
27
28 003013 060211 TOINT:  NIOC TTO                    ;CLEAR DONE
29 003014 004735         JSR OUTBF                   ;GET A CHARACTER FROM THE BUFFEI
30 003015 000200         TOCNT
31 003016 061111         DOAS 0,TTO                  ;NOT EMPTY
32 003017 000764         JMP DISIN                   ;DISMIS
33
```

```
A 00058  .MAIN
01
02
03                ;KEYIN  READ FROM KEYBOARD
04                ;RDTTI  READ FROM KEYBOARD READER
05                ;
06                ;CALLING SEQUENCE:
07                ;       JSR KEYIN
08                ;       RETURN, CHAR, IN AC0
09                ;       (RDTTI CALLED WITH A JMP FROM RDRIN)
10                ;
11
12 003020 102521 RDTTI:  SUBZL 0,0,SKP
13
14 003021 102400 KEYIN:  SUB 0,0                    ;SET TO KEYBOARD MODE
15 003022 040022          STA 0,TICNT+10            ;SET DEVICE=DATA=BLOCK
16 003023 054461          STA 3,KEYRT               ;SAVE RETURN
17 003024 024220          LDA 0,TICNT+10            ;GET KEYBOARD MODE
18 003025 176400          SUB 3,3                   ;CLEAR COUNTER
19 003026 024210 KEYI0:   LDA 1,TICNT               ;GET COUNT
20 003027 125004          MOV 1,1,SZR
21 003030 000421          JMP KEYI3                 ;NON=ZERO, OK
22 003031 101005          MOV 0,0,SNR               ;ZERO, TEST MODE
23 003032 000774          JMP KEYI0                 ;0=KEYBOARD MODE
24 003033 175405          INC 3,3,SNR               ;READER, TEST COUNT
25 003034 000407          JMP KEYI2                 ;GONE TO 0, ERROR EXIT
26 003035 024216          LDA 1,TICNT+6             ;TEST ACTIVE
27 003036 125004          MOV 1,1,SZR
28 003037 000767          JMP KEYI0                 ;IT IS ACTIVE
29 003040 101216          ISZ TICNT+6               ;MAKE IT ACTIVE
30 003041 060110          NIOS TTI                  ;AND START THE READER
31 003042 000763          JMP KEYI0-1
32
33 003043 060210 KEYI2:   NIOC TTI                  ;CLEAR READER
34 003044 054216          STA 3,TICNT+6             ;CLEAR ACTIVE
35 003045 054210          STA 3,TICNT       ;CLEAR COUNT
36 003046 034211          LDA 3,TICNT+1     ;SET IN=OUT
37 003047 054212          STA 3,TICNT+2
38 003050 002434          JMP @KEYRT                ;AND EXIT
39
40 003051 060277 KEYI3:   INTDS
41 003052 004677          JSR OUTBF                 ;GET FROM BUFFER
42 003053 000210          TICNT
43 003054 060177          INTEN                     ;TURN INTERRUPT ON
44
```

```
A 0059   .MAIN
01
02 0055 034427          LDA 3,KEYRT             ;GET RETURN
03 0056 024220          LDA 1,TICNT+10          ;GET MODE
04 0057 030052          LDA 2,C177              ;GET RUBOUT
05 0060 112414          SUB# 0,2,SZR            ;TEST FOR RUBOUT
06 0061 000404          JMP .+4           ;NO
07 0062 125004          MOV 1,1,SZR             ;YES, TEST FOR READER
08 0063 000741          JMP KEYI0-2             ;YES, IGNORE RO
09 0064 001400          JMP 0,3           ;KEYBOARD, EXIT
10 0065 125004          MOV 1,1,SZR             ;TEST FOR READER
11 0066 001401          JMP 1,3           ;YES, EXIT
12
13                ;KEYBOARD-ECHO THE CHARACTER
14 0067 040414          STA 0,TICHR             ;SAVE CHARACTER
15 0070 101105          MOV 0,0,SNR             ;TEST FOR BAD PARITY
16 0071 020050          LDA 0,C134              ;YES, TYPE ^
17 0072 000116          JSR @TYPE
18 0073 020414          LDA 0,TICHR
19 0074 024030          LDA 1,C20               ;OK, NOW TEST FOR ^P
20 0075 030152          LDA 2,TABSM
21 0076 150000          COM 2,2           ;IF YES, COMPLEMENT THE TAB SWITCH
22 0077 106414          SUB# 0,1,SZR
23 0100 002404          JMP @KEYRT              ;EXIT
24 0101 050152          STA 2,TABSM             ;COMPLEMENT THE TAB SWITCH
25 0102 000722          JMP KEYI0-2             ;AND GET NEXT CHARACTER
26 0103 000000  TICHR:  0
27 0104 000000  KEYRT:  0                       ;SAVED RETURN
```

```
A 0000   .MAIN
01
02
03                ;RDRIN  FETCH A READER CHARACTER
04                ;
05                ;CALLING SEQUENCE:
06                ;       JSR RDRIN
07                ;       ERROR= END OF FILE
08                ;       NORMAL= CHARACTER IN AC0
09
10  03105 02157 RDRIN:  LDA 0,USTTI             ;GET USE TTI FLAG
11  03106 101005        MOV 0,0,SNR             ;AND TEST IT
12  03107 000711        JMP RDTTI               ;GO USE TTI
13  03110 054437        STA 3,RDRTN             ;SAVE RETURN
14  03111 034440        LDA 3,RBFCN+1   ;GET TIME-OUT CONSTANT
15  03112 030036        LDA 2,RBFCN             ;GET RESTART CONSTANT
16
17  03113 024162 RDR0:  LDA 1,RDCNT             ;GET CHARACTER COUNT
18  03114 020170        LDA 0,RDCNT+6           ;GET READER ACTIVE
19  03115 101005        MOV 0,0,SNR             ;TEST ACTIVE
20  03116 146513        SUBL# 2,1,SNC           ;TEST COUNT<RESTART CONSTANT
21  03117 000403        JMP .+3                 ;RUNNING OR ENOUGH CHARACTERS
22  03120 060112        NIOS PTR                ;STOPPED AND ALMOST EMPTY
23  03121 101170        ISZ RDCNT+6             ;SET ACTIVE
24  03122 175404        INC 3,3,SZR
25  03123 000467        JMP RDR1
26  03124 060212        NIOC PTR                ;CLEAR THE READER
27  03125 054170        STA 3,RDCNT+6           ;CLEAR ACTIVE
28  03126 054162        STA 3,RDCNT     ;CLEAR COUNT
29  03127 034163        LDA 3,RDCNT+1   ;SET IN=OUT
30  03130 054164        STA 3,RDCNT+2
31  03131 002416        JMP @RDRTN              ;EXPIRED= END=OF=FILE
32  03132 020000 RDR1:  LDA 0,M3
33  03133 107112        ADDL# 0,1,SZC   ;TEST CHARACTER COUNT
34  03134 000757        JMP RDR0                ;<3,  TRY AGAIN
35
36                ;OK, THERE ARE AT LEAST FOUR CHARACTER IN THE BUFFER
37                ;GET IT
38
39  03135 060277        INTDS                   ;DISABLE INTERRUPT
40  03136 004613        JSR OUTBF               ;GET FROM BUFFER
41  03137 000162        RDCNT                   ;POINTER TO DEVICE DATA BLOCK
42  03140 000401        JMP .+1         ;NOT EMPTY
43  03141 060177        INTEN                   ;ENABLE INTERRUPT
44  03142 024052        LDA 1,C177
45  03143 106415        SUB# 0,1,SNR    ;TEST FOR RUBOUT
46  03144 000745        JMP RDRIN+4     ;YES, IGNORE IT
47  03145 010402        ISZ RDRTN       ;INCREMENT RETURN
48  03146 002401        JMP @RDRTN              ;AND EXIT WITH CHAR IN AC0
49  03147 000000 RDRTN: 0                               ;SAVED RETURN
50  03150 000003 RBFCN: RSZE/2                  ;RESTART CONSTANT
51
52        000012 .RDX 10
53  03151 171124        -3500
54        000018 .RDX 8
55
```

```
                        ;SAVE AC0
                        ;SAVE CRY

                        ;SAVE OTHER AC'S
D,                                  W,  TEST INTERRUPT SOURCE
;RE                                     IN DEVICE CODE
                        ;TEST READER
                        ;YES- GO PROCESS
                        ;NO, TEST THE PUNCH
                        ;YES, GO PROCESS
                        ;TEST TTO
                        ;YES, GO PROCESS
                        ;TEST TTI
                        ;YES, GO PROCESS
;GE                                 A  NIOC INSTRUCTION
;AN                                 EXECUTE IT

                        ;AND DISMIS
```

```
A 0062  .MAIN
0001
0002
0003
0004                    ERMS0:  .TXT /BUFFER CAPACITY EXCEEDED DURING INSERT<15><12>/
    0032 0525 02
    0033 0431 06
    0034 0511 05
    0035 0414 40
    0036 0501 01
    0037 0415 01
    0038 0521 11
    0039 0201 31
    0040 0541 05
    0041 0425 03
    0042 0421 05
    0043 0421 05
    0044 0420 40
    0045 0511 25
    0046 0471 11
    0047 0201 67
    0050 0471 11
    0051 0425 23
    0052 0521 22
    0053 0050 15
    0054 0000 00
0055
0056
0057
0058
0059
0060
0061
0062                    ERMS1:  .TXT /BUFFER IS FULL=CANNOT DO A<15><12>/
    0063 0525 02
    0064 0431 06
    0065 0511 05
    0066 0444 40
    0067 0201 23
    0070 0525 06
    0071 0461 14
    0072 0414 55
    0073 0471 01
    0074 0475 16
    0075 0201 24
    0076 0475 04
    0077 0404 44
    0100 0050 15
    0101 0000 00
```

```
A 0063   .MAIN
01
02
03
04
05                  ERMS2:   .TXT /BUFFER IS FULL=Y OR A INPUT TERMINATED<15><12>/
    0325 0525 2
    0326 0431 6
    0327 0511 5
    0330 0444 0
    0331 0201 3
    0332 0525 6
    0333 0461 4
    0334 0544 5
    0335 0474 0
    0336 0201 2
    0337 0201 1
    0340 0471 1
    0341 0525 0
    0342 0201 4
    0343 0425 4
    0344 0465 2
    0345 0471 1
    0346 0521 1
    0347 0421 5
    0350 0015 5
    0351 0000 0
06
```

```
A 0064  .MAIN

                        ERMS3:  .TXT /BUFFER CAPACITY EXCEEDED DURING COMMAND INPUT<15>
    003275 052502
    003276 043106
    003277 051105
    003300 041440
    003301 050101
    003302 041501
    003303 052111
    003304 020131
    003305 054105
    003306 042503
    003307 042105
    003310 042105
    003311 042040
    003312 051125
    003313 047111
    003314 020107
    003315 047503
    003316 046515
    003317 047101
    003320 020104
    003321 047111
    003322 052520
    003323 006524
    003324 047503 COMMAND IS TERMINATED AND BEING EXECUTED<15><12>/
    003325 046515
    003326 047101
    003327 020104
    003330 051511
    003331 052040
    003332 051105
    003333 044515
    003334 040516
    003335 042524
    003336 020104
    003337 047101
    003340 020104
    003341 042502
    003342 047111
    003343 020107
    003344 054105
    003345 041505
    003346 052125
    003347 042105
    003350 005015
    003351 000000
```

```
A 0065   .MAIN
01
02                  ERMS4:   .TXT /??/
    003352 037477
    003353 000000
03
04
05
06
07                  ERMS5:   .TXT /PARITY ERROR IN LINE NUMBER /
    003354 040520
    003355 044522
    003356 054524
    003357 042440
    003360 051122
    003361 051117
    003362 044440
    003363 020116
    003364 044514
    003365 042516
    003366 047040
    003367 046525
    003370 042502
    003371 020122
    003372 000000
08
09
10                  ERMS6:   .TXT /STR NOT FOUND<15><12>/
    003373 052123
    003374 020122
    003375 047516
    003376 020124
    003377 047506
    003400 047125
    003401 000654
    003402 000012
11
12
13                  ERMS7:   .TXT /MACRO ERROR/
    003403 040515
    003404 051103
    003405 020117
    003406 051105
    003407 047522
    003410 000122
```

```
A 0000   .MAIN
01
02
03                  MESG1:   .TXT /TTO(1) OR PTP(2)? /
    0341 052124
    0342 024117
    0343 024461
    0344 047440
    0345 020122
    0346 052120
    0347 024120
    0350 024462
    0351 020077
    0352 000000
04
05
06                  MESG2:   .TXT /TTI(1) OR PTR(2)? /
    0353 052124
    0354 024111
    0355 024461
    0356 047440
    0357 020122
    0360 052120
    0361 024122
    0362 024462
    0363 020077
    0364 000000
```

```
 0067   .MAIN
1
2                   MESG3:   .TXT /PARITY OUT(1) OR NOT(2)? /
    3435 0452A
    3436 0452A
    3437 0452A
    3440 0474A
    3441 0521A
    3442 0345A
    3443 0251A
    3444 0511A
    3445 0470A
    3446 0521A
    3447 0315A
    3450 0375A
    3451 0000A
03
04
05                  MESG5:   .TXT /PARITY IN(1) OR NOT(2)? /
    3452 0452A
    3453 0452A
    3454 0452A
    3455 0444A
    3456 0241A
    3457 0244A
    3460 0474A
    3461 0251A
    3462 0475A
    3463 0241A
    3464 0244A
    3465 0251A
    3466 0000A
06
07
08
09
10
11                  ;DEFINE IO BUFFERS
12
13                  TIBUF:
14          0007   .BLK TISZE/2+2
15
16                  TOBUF:
17          0076   .BLK TOSZE/2+2
18
19                  PNBUF:
20          0052   .BLK PSZE/2+2
21
22                  RDBUF:
23          0005   .BLK RSZE/2+2
24
25
26                  ;END OF THE PROGRAM, FILLED WITH ESC'S
27  4303 15433              15433
28  4304 15433              15433
29  4305 15433 ENDPR:       15433
30
31          0030            .END STRT
```