```
0001  INT10
01                ;                                        RCSL:    43-GL2556
02                ;                                        AUTHOR:  DHA
03                ;                                        EDITED:  76.04.01
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31                ;                  INT100
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49                ; KEYWORD:         MUS, MUSIL, INTERPRETER, LISTING.
50                ; ABSTRACT:        INTERPRETER FOR MUSIL CODE,USED IN BASIC SYSTEM.
51                ;                  ASCII PPER TAPE RCSL: 43-GL2557
52                ;                  INT100 REL.BIN. PAPER TAPE RCSL: 43-GL2558.
53                ;                  INTT5 (TESTOUTPUT) REL.BIN. PAPER TAPE RCSL:
54                ;                  43-GL1390.
55
56
57
58
```

```
↑ 0002  INT10
01                ; TESTOUTPUT ON LPT.
02                ; THE ASSEMBLY CONSTANT TESTOUT SHOULD BE DEFINED ON A
03                ; SEPERATE TAPE
04                ; TESTOUT=0, MEANS NO TESTOUTPUT.
05                ;     -    =1, MEANS TESTOUTPUT ON LPT.
06                ; THE LPT DRIVER SHOULD ALWAYS BE LOADED TOGETHER WITH THE INTER-
07                ; PRETER. IN CASE OF LPT ERRORS THE ACTION IS: REPETITTON UNTIL
08                ; LPT IS READY. IF TESTOUTPUT IS WANTED ON ANOTHER DRIVER THE
09                ; FILE NAME LPT SHOULD BE CHANGED. THIS NAME IS FOUND AS THE
10                ; FIRST WORD OF THE LOADED MODULE. A LIST COMMAND WILL GIVE
11                ; THE ADDRESS AS CURRENT NMAX.
12                ;
13                ; SWITCH 0 SET TESTOUTPUT WILL BE PRODUCED OF EVERY I/O
14                ; COMMAND CONTAINING SOME OF THE ZONE VARIABLES.
15                ; THE FORMAT IS:
16                ;     OP MODIF ZONE ZUSED ZSHAREL ZTOP ZREM ZFIRST ZLENGTH ZFORMA
17                ;
18                ; SWITCH 1 SET WILL PRODUCE TESTOUTPUT FOR
19                ;           GO CODEAND MOVE
20                ;           FORMAT: OPERATION MODIFICATION AND 6 MODIFWORDS (OCTAL
21                ; SWITCH 2 SET WILL PRODUCE TESTOUTPUT FOR
22                ;           MOVESTRING, COMPARE, TRANSLATE AND CONVERT
23                ;           FORMAT: OPERATION, MODIFICATION AND 4 MODIFWORDS (OCTAL
24                ; SWITCH 3 SET WILL PRODUCE TESTOUTPUT FOR
25                ;           OPSTATUS, BINDEC, DECBIN
26                ;           FORMAT: OPERATION, MODIFICATION AND 2 MODIFWORDS (OCTAL
27                ; SWITCH 4 SET WILL PRODUCE TESTOUTPUT FOR
28                ;           ARITC, ARIT, JUMP, LINK, CALL, OPMESS AND OTHERS
29                ;           FORMAT: OPERATION, MODIFICATION AND 1 MODIFWORD (OCTAL)
30                ; SWITCH 5 SET WILL PRODUCE TESTOUTPUT FOR
31                ;           STOP, OPTEST AND ARITD
32                ;           FORMAT: OPERATION AND MODIFICATION (OCTAL)
33
34
```

```
↑ 0003  INT10
01                .TITL  INT100
02        000012  .RDX     10
03        000001  .TXTM    1
04                .NREL
05                ;       *****     INTERPRETER      *****
06                ;
07
08
09        000225          .CL00=    INTPRETE-GOS              ; DEFINE IN MONITORS PAGE
10        000225          .LOC      .CL00                     ; INTPRETE ENTRY
11  00225 000403'         CL00                                ;
12        000227          .LOC      .+1                       ;
13  00227 074005'         CL6                                 ; INTGIVEUP
14        000231          .LOC      .+1                       ;
15  00231 074017'         CL7                                 ; INTBREAK
16        000234          .LOC      MZSTART
17
18  00234 000407'.CL0:    CL0                                 ; CENTRAL LOOP ENTRY 0.
19  00235 000410'.CL1:    CL1                                 ;    -     -     -  1.
20  00236 000423'.CL4:    CL4                                 ; TAKEADDRESS ENTRY.
21  00237 000443'.CL5:    CL5                                 ; TAKEVALUE ENTRY.
22
23                .NREL
24
25
```

```
↑ 0004  INT10
0KU                 .IFN TESTOUT
02                  ; ZONE USED TO WRITE THE TESTOUTPUT ON LPT
03                  ;
04                  IT53:                           ; OUTPUT ZONE DESCRIPTOR:
05                  .TXT   .LPT<0><0>.              ; NAME
    00000'046120
    00001'052000
    00002'000000
06  00003'000032            IT51-IT53               ; SIZE
07  00004'000003            3                       ; MODE
08  00005'000001            1                       ; KIND
09          000002  .RDX    2
10  00006'143773            1100011111111011        ; GIVEUPMASK
11          000012  .RDX    10
12  00007'000111'           IT99                    ; GIVEUPACTION
13  00010'000000            0                       ; FILE
14  00011'000000            0                       ; BLOCK
15  00012'000000            0                       ; CONVERSION
16  00013'000032'           IT51                    ; BUFFER
17  00014'000057            IT59-IT51               ; SIZE OF BUFFER
18  00015'000000            0                       ; FORMAT
19  00016'000000            0                       ; LENGTH
20  00017'000102"           IT52                    ; FIRST
21  00020'000102"           IT52                    ; TOP
22  00021'000032'           IT51                    ; USED SHARE
23  00022'000120            IT50                    ; SHARE LENGTH
24  00023'000120            IT50                    ; REMAINING
25          000006  .BLK    ZAUX                    ; AUXILIARY
26
27                  IT51:                           ; OUTPUT SHARE DESCRIPTOR:
28  00032'000000            0                       ; OPERATION
29  00033'000000            0                       ; COUNT
30  00034'000000            0                       ; ADDRESS
31  00035'000000            0                       ; SPECIAL
32  00036'000032'           IT51                    ; NEXT SHARE
33  00037'000000            0                       ; STATE
34  00040'000102"           IT52                    ; FIRST SHARE
35          000120  IT50=80                         ; SHARELENGTH=80
36          000102"IT52=.*2                         ; FIRST SHARE:
37          000050  .BLK    IT50+1/2                ; MAKES ROOM FOR SHARE;
38                  IT59:                           ; TOP OF BUFFER
39  00111'002203    IT99:   .REPEATSHARE            ; GIVEUPACTION;
40
41
```

```
↑ 0005  INT10
01                ; INTERPRETER TESTOUTPUT
02                ;
03 00112'064477 IT0:   READS  1              ; ENTRY TO TESTOUTPUT:
04 00113'020567        LDA    0    ICMAS    ;
05 00114'107405        AND    0,1  SNR      ;   IF NO TESTSWITCHES SET THEN
06 00115'000542        JMP         IT17     ;   RETURN;
07 00116'023033        LDA@   0    PC,2     ;
08 00117'034143        LDA    3    .255     ;
09 00120'117400        AND    0,3           ;   AC3:=MODIFBITS=WORD EXTRACT 8;
10 00121'054570        STA    3    ITMO     ;   SAVE MODIFBITS;
11 00122'162700        SUBS   3,0           ;   AC0:=OPERATIONS=WORD SHIFT (-8);
12 00123'040567        STA    0    ITOP     ;   SAVE OPERATION;
13
14 00124'034111        LDA    3    .128     ; SWITCH 0:
15 00125'125102        MOVL   1,1  SZC      ;   IF SWITCH 0 SET AND
16 00126'116032        ADCZ#  0,3  SZC      ;    OPERATION>=200(OCTAL) THEN
17 00127'000430        JMP         IT1      ;   BEGIN
18 00130'004542        JSR         IT18     ; OUTPUT(OP,MODIF);
19 00131'035033        LDA    3    PC,2     ; I/O-INSTRUCTION:
20 00132'021401        LDA    0    +1,3     ;     ZONE=WORD(PC+1);
21 00133'040551        STA    0    IT8      ;     SAVE ZONE;
22 00134'030547        LDA    2    IT05     ;     AC2:=TESTOUTPUTZONE;
23 00135'004525        JSR         IT16     ;     WRITE(TEST,DELIM,ZONE);
24 00136'021421        LDA    0    ZUSED,3  ;
25 00137'004523        JSR         IT16     ;     WRITE(TEST,DELIM,ZUSED.ZONE);
26 00140'021422        LDA    0    ZSHAREL,3;
27 00141'004521        JSR         IT16     ;     WRITE(TEST,DELIM,SHAREL.ZONE);
28 00142'021420        LDA    0    ZTOP,3   ;
29 00143'004517        JSR         IT16     ;     WRITE(TEST,DELIM,TOP.ZONE);
30 00144'021423        LDA    0    ZREM,3   ;
31 00145'004515        JSR         IT16     ;     WRITE(TEST,DELIM,REM.ZONE);
32 00146'021417        LDA    0    ZFIRST,3 ;
33 00147'004513        JSR         IT16     ;     WRITE(TEST,DELIM,FIRST.ZONE);
34 00150'021416        LDA    0    ZLENGTH,3;
35 00151'004511        JSR         IT16     ;     WRITE(TEST,DELIM,LENGTH.ZONE);
36 00152'021415        LDA    0    ZFORMAT,3;
37 00153'004507        JSR         IT16     ;     WRITE(TEST,DELIM,FORMAT.ZONE);
38 00154'021424        LDA    0    Z0,3     ;
39 00155'004505        JSR         IT16     ;     WRITE(TEST,DELIM,Z0.ZONE);
40 00156'000474        JMP         IT15     ;   END;
41
42
```

```
01                ; INTERPRETER TESTOUTPUT
02                ;
03 00157'034537 IT1:   LDA    3   IC57      ; NORMAL INSTRUCTIONS:
04 00160'125103        MOVL   1,1 SNC       ; SWITCH 1:
05 00161'000423        JMP        IT2       ;
06 00162'116423        SUBZ   0,3 SNC       ;   IF SWITCH 1 SET AND
07 00163'000404        JMP        .+4       ;   OP>=60 OR OP=52 THEN
08 00164'034531        LDA    3   IC52      ;   BEGIN
09 00165'116404        SUB    0,3 SZR       ;
10 00166'000416        JMP        IT2       ;
11 00167'004503        JSR        IT18      ;     OUTPUT(OP,MODIF);
12 00170'034123        LDA    3   .6        ;
13 00171'054523        STA    3   ITCOUNT   ;     ITCOUNT:=6;
14 00172'030040 IT12:  LDA    2   CUR       ; OUTMODIFWORDS:
15 00173'035033        LDA    3   PC,2      ;
16 00174'030507        LDA    2   IT05      ;
17 00175'054516        STA    3   ITPC      ;     ITPC:=PC;
18 00176'010515 IT13:  ISZ        ITPC      ;     WHILE ITCOUNT>0 DO
19 00177'022514        LDA@   0   ITPC      ;     BEGIN ITPC:=ITPC+1;
20 00200'004462        JSR        IT16      ;       WRITE(TEST,DELIM,WORD(ITPC);
21 00201'014513        DSZ        ITCOUNT   ;       ITCOUNT:=ITCOUNT-1;
22 00202'000774        JMP        IT13      ;     END;
23 00203'000447        JMP        IT15      ;   END;
24 00204'034513 IT2:   LDA    3   IC44      ; SWITCH 2:
25 00205'125102        MOVL   1,1 SZC       ;
26 00206'116423        SUBZ   0,3 SNC       ;   IF SWITCH 2 SET AND
27 00207'000410        JMP        IT3       ;    36<=OP<=44 THEN
28 00210'034510        LDA    3   IC35      ;   BEGIN
29 00211'116422        SUBZ   0,3 SZC       ;
30 00212'000405        JMP        IT3       ;
31 00213'004457        JSR        IT18      ;     OUTPUT(OP,MODIF);
32 00214'034116        LDA    3   .4        ;     ITCOUNT:=4;
33 00215'054477        STA    3   ITCOUNT   ;     GOTO MODIFWORDS;
34 00216'000754        JMP        IT12      ;   END;
35 00217'034477 IT3:   LDA    3   IC57      ; SWITCH 3:
36 00220'125102        MOVL   1,1 SZC       ;   IF SWITCH 3 SET AND
37 00221'116423        SUBZ   0,3 SNC       ;   45<=OP<=57 THEN
38 00222'000410        JMP        IT4       ;   BEGIN
39 00223'034474        LDA    3   IC44      ;
40 00224'116422        SUBZ   0,3 SZC       ;
41 00225'000405        JMP        IT4       ;
42 00226'004444        JSR        IT18      ;     OUTPUT(OP,MODIF);
43 00227'034121        LDA    3   .3        ;     ITCOUNT:=3;
44 00230'054464        STA    3   ITCOUNT   ;     GOTO MODIFWORDS;
45 00231'000741        JMP        IT12      ;   END;
46 00232'034466 IT4:   LDA    3   IC35      ; SWITCH 4:
47 00233'125102        MOVL   1,1 SZC       ;   IF SWITCH 4 SET AND
48 00234'116423        SUBZ   0,3 SNC       ;   11<=OP<=35 THEN
49 00235'000410        JMP        IT5       ;   BEGIN
50 00236'034463        LDA    3   IC10      ;
51 00237'116422        SUBZ   0,3 SZC       ;
52 00240'000405        JMP        IT5       ;
53 00241'004431        JSR        IT18      ;      OUTPUT(OP,MODIF);
54 00242'176520        SUBZL  3,3           ;     ITCOUNT:=1;
55 00243'054451        STA    3   ITCOUNT   ;     GOTO MODIFWORDS;
56 00244'000726        JMP        IT12      ;   END;
57
58
```

```
↑ 0007  INT10
01                ; INTERPRETER TESTOUTPUT
02                ;
03 00245'034454 IT5:   LDA     3   IC10       ; SWITCH 5:
04 00246'125102        MOVL    1,1 SZC        ;   IF SWITCH 5 SET AND OP<=10 THEN
05 00247'116423        SUBZ    0,3 SNC        ;   BEGIN
06 00250'000407        JMP         IT17       ;     OUTPUT(OP,MODIF);
07 00251'014421        JSR         IT18       ;   END;
08
09 00252'031431 IT15:  LDA     2   IT05       ; PRINT:
10 00253'024126        LDA     1   .10        ;
11 00254'006212        OUTCHAR                ;
12 00255'024113        LDA     1   .32        ;
13 00256'006214        OUTEND                 ;
14
15 00257'030040 IT17:  LDA     2   CUR        ; RETURN:
16 00260'027032        LDA@    1   PC,2       ;
17 00261'000530        JMP         CL1+1      ;
18
19 00262'054423 IT16:  STA     3   IT9        ; WRITE(TEST,DELIM,CHAR):
20 00263'044424        STA     1   IT11       ;   SAVE SWITCHES;
21 00264'024422        LDA     1   IT10       ;
22 00265'006212        OUTCHAR                ;   OUTCHAR(AC1);
23 00266'006216        OUTOCTAL               ;   OUTOCTAL(AC0);
24 00267'034415        LDA     3   IT8        ;   GET SAVED ZONE;
25 00270'024417        LDA     1   IT11       ;   GET SAVED SWITCHES;
26 00271'002414        JMP@        IT9        ; JUMP BACK;
27
28 00272'054416 IT18:  STA     3   IT19       ; OUTPUT(OP,MODIF):
29 00273'020417        LDA     0   ITOP       ;   GET OPERATION;
30 00274'030407        LDA     2   IT05       ;   GET OUTPUTZONE;
31 00275'004765        JSR         IT16       ;   WRITE;
32 00276'020413        LDA     0   ITMO       ;   GET MODIFICATION;
33 00277'004763        JSR         IT16       ;   WRITE;
34 00300'030040        LDA     2   CUR        ;
35 00301'002407        JMP@        IT19       ; RETURN;
36
37        000002 .RDX   2                     ;
38 00302'176000 ICMAS: 1111110000000000      ; SWITCH MASK;
39        000012 .RDX   10                    ;
40 00303'000000'IT05:  IT53                   ; TESTZONE;
41 00304'000000 IT8:   0                      ; SAVED ZONE;
42 00305'000000 IT9:   0                      ; WORK.LINK;
43 00306'000055 IT10:  45                     ; CHARACTER -
44 00307'000000 IT11:  0                      ; SAVED SWITCHES;
45 00310'000000 IT19:  0                      ; WORK.LINK;
46 00311'000000 ITMO:  0                      ; SAVED MODIFICATION;
47 00312'000000 ITOP:  0                      ; SAVED OPERATION;
48 00313'000000 ITPC:  0                      ; WORK PC;
49 00314'000006 ITCOUNT: 6                    ; NO OF MODIFWORDS;
50 00315'000052 IC52:  42                     ;
51 00316'000057 IC57:  47                     ;
52 00317'000044 IC44:  36                     ;
53 00320'000035 IC35:  29                     ;
54 00321'000010 IC10:   8                     ;
55K               .ENDC
56
57
```

```
↑ 0008  INT10
01        000322'CL11=.          ; BASE OF TABLE CONTAINING SUBINTERPRETERS.
02
03 00322'074017'          CL7     ; 0     : STOP
04
05 00323'073642'          CL100   ; 1     : ANDD        DIRECT OPERATIONS
06 00324'073652'          CL101   ; 2     : LOADD
07 00325'073661'          CL102   ; 3     : +D
08 00326'073672'          CL103   ; 4     : -D
09 00327'073703'          CL104   ; 5     : SHIFTD
10 00330'073740'          CL105   ; 6     : EXTRACTD
11 00331'073753'          CL106   ; 7     : *D
12 00332'074003'          CL107   ; 10    : /D
13
14 00333'073640'          CL110   ; 11    ; ANDC        OPERATIONS ON CONSTANTS
15 00334'073650'          CL111   ; 12    : LOADC
16 00335'073657'          CL112   ; 13    : +C
17 00336'073670'          CL113   ; 14    : -C
18 00337'073701'          CL114   ; 15    : SHIFTC
19 00340'073736'          CL115   ; 16    : EXTRACTC
20 00341'073751'          CL116   ; 17    : *C
21 00342'073761'          CL117   ; 20    : /C
22
23 00343'073635'          CL120   ; 21    ; AND         OPERATIONS ON VARIABLES
24 00344'073645'          CL121   ; 22    : LOAD
25 00345'073654'          CL122   ; 23    : +
26 00346'073665'          CL123   ; 24    : -
27 00347'073676'          CL124   ; 25    : SHIFT
28 00350'073733'          CL125   ; 26    : EXTRACT
29 00351'073746'          CL126   ; 27    : *
30 00352'073756'          CL127   ; 30    : /
31
32 00353'073730'          CL903   ; 31    : LOAD NEGATIVE
33 00354'073716'          CL901   ; 32    : LOAD BYTE
34 00355'073721'          CL902   ; 33    : LOAD BYTEWORD
35 00356'073511'          CL309   ; 34    : JUMP
36 00357'073465'          CL308   ; 35    : LINK
37 00360'073262'          CL301   ; 36    : MOVEWORD
38 00361'073276'          CL302   ; 37    : MOVESTRING
39 00362'000000           0       ; 40    : COMPAREWORD
40 00363'073425'          CL304   ; 41    ; COMPARESTRING
41 00364'073420'          CL305   ; 42    : STORE REGISTER
42 00365'073360'          CL322   ; 43    : TRANSLATE
43 00366'073323'          CL324   ; 44    : CONVERT
44 00367'073537'          CL310   ; 45    : OPMESS
45 00370'073544'          CL311   ; 46    : OPIN
46 00371'073614'          CL312   ; 47    : OPWAIT
47 00372'073444'          CL306   ; 50    : CALL
48 00373'073626'          CL314   ; 51    : OPTEST
49 00374'073267'          CL320   ; 52    : MOVE
50 00375'073554'          CL313   ; 53    : OPSTATUS
51 00376'073452'          CL307   ; 54    : BINDEC
52 00377'073606'          CL315   ; 55    : DECBIN
53 00400'073410'          CL316   ; 56    ; INSERT
54 00401'073476'          CL360   ; 57    : GOTO
55 00402'073460'          CL303   ; 60    : GO CODE
56
57              ; END OF TABLE
58
```

```
↑ 0009  INT10
01                ; A JUMP TO THE REENTRANT INTERPRETER IS EXECUTED BY A JSR@ INTPR.
02                ; INSTR:=WORD(PC.CUR).
03                ; OPERATION:=INSTR(0:7), MODIFICATIONBITS:=INSTR(8:15).
04                ; IF OPERATION>=200(OCTAL) THEN I/O-INSTRUCTION: PROC(ZONE,...).
05                ;                              ELSE NORMAL INSTRUCTION.
06                ; EXIT TO THE ACTUEL SUBINTERPRETER WITH
07                ;   AC0=MODIFICATION BITS
08                ;   AC1=OPERATIONS BITS.
09                ;   AC2=CURENT PROCESS.
10
11
12 00403'030040 CL00:   LDA     2       CUR     ; ENTRY TO INTERPRETER:
13 00404'055033         STA     3       PC,2    ;
14 00405'102400         SUB     0,0             ;
15 00406'041034         STA     0       OP,2    ; OP.CUR:=0:
16 00407'030040 CL0:    LDA     2       CUR     ; CENTRAL LOOP:
17                                              ;
18                CL1:                          ; CENTRAL LOOP 1:
1KU               .IFN    TESTOUT
20 00410'002411         JMP@            .IT0    ; GOTO TESTOUTPUT;
21K               .ENDC
2KU               .IFE    TESTOUT
23 00411'027033         LDA@    1       PC,2    ; INST:=W(PC.CUR);
24                .ENDC
25 00412'011033         ISZ             PC,2    ; PC.CUR:=PC.CUR+1:
26 00413'020143         LDA     0       .255    ;
27 00414'123400         AND     1,0             ; MODIF:=INST(8:15);
28 00415'106700         SUBS    0,1             ; OPERATION:=INSTR(0:7);
29 00416'004401         JSR             .+1     ; AC3:=CL14;
30 00417'137000 CL14:   ADD     1,3             ; AC3:=CL14+OPERATION;
31 00420'003703         JMP@            CL15,3  ; GOTO SUBINTPR.=CL11-CL14+CL14+OP
32       177703 CL15=CL11-CL14                  ;
3KU               .IFN    TESTOUT
34 00421'000112'.IT0:   TT0                     ;
35K               .ENDC
3KU               .IFE    TESTOUT
37 00422'000000         0                       ; DUMMY
38K               .ENDC
39
```

```
↑ 0010  INT10
01                ; PROCEDURE TAKEADDRESS(MODIF,ADDRESS);
02                ; GET THE ADDR OF AN INTEGER OR STRING ADDRESSED BY PC AND INCR(PC
03                ; 2 BITS MODIF: 00 ADDR=WORD(PC). !INTEGER!
04                ;     -      -  : 01 ADDR=   -       !STRING!
05                ;     -      -  : 10 ADDR=   -       !FILE!
06                ;     -      -  : 11 ADDR=   -      .ADDR=ZN(CUR+ADDR(0:7)).ZFIRST
07                ;                                        +ADDR(8:15).
08                ;        CALL                RETURN
09                ; AC0    MODIFBITS           MODIFBITS SHIFT(-2)
10                ; AC1                        ADDRESS
11                ; AC2    CUR                 CUR
12                ; AC3    LINK                DESTROYED
13                ;
14  00423'027033  CL4:    LDA@     1          PC,2     ; ADDRESS:=WORD(PC.CUR);
15  00424'011033          ISZ                 PC,2     ; INCR(PC.CUR);
16  00425'101223          MOVZR    0,0        SNC      ; BITS:=MODIFBITS EXTRACT 2;
17  00426'101221          MOVZR    0,0        SKP      ; MODIFBITS:=MODIFBITS SHIFT (-2);
18  00427'101223          MOVZR    0,0        SNC      ; IF BITS<>11 THEN
19  00430'001400          JMP                 +0,3     ; RETURN;
20  00431'055025          STA      3          SAVE1,2  ;
21  00432'034143          LDA      3          .255     ;
22  00433'137400          AND      1,3                 ; FIELD:=ADDRESS(8:15);
23  00434'166700          SUBS     3,1                 ; ADDRESS:=ZN.CUR
24  00435'133000          ADD      1,2                 ;          (ADDRESS(0:7));
25  00436'031041          LDA      2          ZN,2     ; ADDRESS:=(ADDRESS.ZFIRST)
26  00437'025017          LDA      1          ZFIRST,2 ;          +FIELD;
27  00440'167000          ADD      3,1                 ;
28  00441'030040          LDA      2          CUR      ;
29  00442'003025          JMP@                SAVE1,2  ; RETURN;
30
31                ; PROCEDURE TAKEVALUE(MODIF,VALUE)
32                ; GETS THE VALUE OF AN INTEGER.
33                ; 2 BITS MODIF: 00 VALUE=INST(PC); INC(PC) ;
34                ;     -      -  : 01 VALUE=R;
35                ;     -      -  : 10 VALUE=WORD(INST(PC)); INC(PC);
36                ;     -      -  : 11 VALUE=R;
37                ;        CALL                RETURN
38                ; AC0    MODIF               MODIF SHIFT (-2)
39                ; AC1                        VALUE
40                ; AC2    CUR                 CUR
41                ; AC3    LINK                DESTROYED
42                ;
43  00443'101222  CL5:    MOVZR    0,0        SZC      ;
44  00444'000413          JMP                 CL52     ;
45  00445'101222          MOVZR    0,0        SZC      ; CASE MODIFBITS(14:15) OF
46  00446'000404          JMP                 CL51     ;
47  00447'027033          LDA@     1          PC,2     ; 0: VALUE:=INST(PC.CUR):
48  00450'011033          ISZ                 PC,2     ;    INCR(PC.CUR);
49  00451'001400          JMP                 +0,3     ;    RETURN;
50  00452'031033  CL51:   LDA      2          PC,2     ; 2: VALUE:=WORD(INST(PC.CUR));
51  00453'027000          LDA@     1          +0,2     ;
52  00454'030040          LDA      2          CUR      ;
53  00455'011033          ISZ                 PC,2     ;    INCR(PC.CUR);
54  00456'001400          JMP                 +0,3     ;    RETURN;
55  00457'025032  CL52:   LDA      1          R,2      ; 1 AND 3:
56  00460'101220          MOVZR    0,0                 ;    VALUE:=R.CUR
57  00461'001400          JMP                 +0,3     ;    RETURN;
```

```
↑ 0011  INT10
01                ; ZONEPROCESSES;
02                ; CALL: PROCEDURE(ZONE,PARAM1,PARAM2);
03
04
05                ; PROCEDURE GETZONEADDR;
06  00462'101220 CL20:   MOVZR   0,0              ;
07  00463'101220         MOVZR   0,0              ;   MODIF:=MODIF SHIFT -2;
08  00464'045030         STA     1       SAVE4,2  ;   SAVE4:=OP;
09  00465'027033         LDA@    1       PC,2     ;   AC1:=WORD(PC.CUR);
10  00466'011033         ISZ             PC,2     ;   INCR(PC.CUR);
11  00467'045027         STA     1       SAVE3,2  ;   ZONE.CUR:=ZN;
12  00470'001400         JMP             +0,3     ; RETURN;
13
14                ; TRANSFER(Z,LENGTH,OPERATION), SETPOSITION(Z,FILE,BLOCK);
15  00471'004771 CL205:  JSR             CL20     ; GETZONEADDR;
16  00472'004751         JSR             CL5      ; TAKEVALUE(MODIF,PARAM1);
17  00473'045031         STA     1       SAVE5,2  ; SAVE(PARAM1);
18  00474'004747         JSR             CL5      ; TAKEVALUE (MODIF,PARAM2);
19  00475'121000         MOV     1,0              ; AC0:=PARAM2;
20  00476'025031         LDA     1       SAVE5,2  ; AC1:=PARAM1;
21A00477'000000         JMP             CL209    ; GOTO EXECUTE;
22
23                ; REPEATSHARE(Z);
24  00500'004762 CL203:  JSR             CL20     ; GETZONEADDR;
25  00501'031027         LDA     2       SAVE3,2  ;
26  00502'034040         LDA     3       CUR      ;
27  00503'021013         LDA     0       ZBUFFE,2 ; PC.CUR:=SAVEPC.ZONE;
28  00504'041433         STA     0       PC,3     ;
29  00505'035014         LDA     3       ZSIZE,2  ; LINK:=SAVELINK.ZONE;
30  00506'002203         .REPEATSHARE             ; CALL: AC2=ZONE, AC3=LINK;
31
32                ; INCHAR(Z,INTEGER VARIABLE NAME);
33  00507'004753 CL202:  JSR             CL20     ; GETZONEADDR;
34  00510'037033         LDA@    3       PC,2     ;
35  00511'011033         ISZ             PC,2     ;
36  00512'025400         LDA     1       +0,3     ; AC1:=PARAM1;
37  00513'035030         LDA     3       SAVE4,2  ;
38  00514'031027         LDA     2       SAVE3,2  ; AC2:=ZONE;
39  00515'007400         JSR@            +0,3     ; EXIT TO MONITOR ENTRY;
40  00516'030040 CL212:  LDA     2       CUR      ;
41  00517'035033         LDA     3       PC,2     ;
42  00520'047777         STA@    1       -1,3     ;
43  00521'002235         JMP@            .CL1     ; RETURN;
```

```
01                ; SUBINTERPRETERTABLE CONTINUED;
02 00522'000000           0                              ; EMPTY ELEMENTS;
03 00523'000000           0                              ;
04         000524'CL21=.                                  ; I/O-TABLESTART:
05 00524'073227'          CL200     ; 200                : GETREC:       AC0=ADDR
06 00525'073243'          CL201     ; 201                : PUTREC:       AC0=VAL
07 00526'073255'          CL206     ; 202                : WAITTRANSFER
08 00527'000500'          CL203     ; 203                : REPEATSHARE
09 00530'000471'          CL205     ; 204                : TRANSFER:     AC0=VAL, AC1=VAL
10 00531'073255'          CL206     ; 205                : INBLOCK
11 00532'073255'          CL206     ; 206                : OUTBLOCK
12 00533'000507'          CL202     ; 207                : INCHAR:       AC1=ADDR
13 00534'073255'          CL206     ; 210                : BACKSPACE
14 00535'073255'          CL206     ; 211                : OUTSPACE
15 00536'073247'          CL204     ; 212                : OUTCHAR:      AC1=VAL
16 00537'073255'          CL206     ; 213                : OUTNL
17 00540'073247'          CL204     ; 214                : OUTEND:       AC1=VAL
18 00541'073240'          CL207     ; 215                : OUTTEXT:      AC0=ADDR
19 00542'073243'          CL201     ; 216                : OUTOCTAL:     AC0=VAL
20 00543'000471'          CL205     ; 217                : SETPOS        AC0=VAL, AC1=VAL
21 00544'073247'          CL204     ; 220                : CLOSE:        AC1=VAL
22 00545'073243'          CL201     ; 221                : OPEN:         AC0=VAL
23 00546'073255'          CL206     ; 222                : WAITZONE
24                                                       ; THE NEXT ENTRY IN THE I/O-TABLE
25                                                       ; IS 346-200(OCTAL) FOR CREATEENTR
26                                                       ; SEE REL.ADDR 146.
27                                                       ; THE EMTHY TABLEELEMENTS ARE USED
28                                                       ; FOR CODE.
29
30         000002 .IFN      CL21-CL11+GOS-GETREC         ; IF ADDR USED FOR MONITORENTRIES
31         072460           .BLK 30000                   ; ARE NOT CORRECT BECAUSE OF
32                  .ENDC                                ; CHANGES IN MONTTOR OR INTERPRE-
33                                                       ; TER, A BLOCK OF 30 K IS CREATED
34                                                       ; TO INDICATE AN ERROR.
```

```
↑ 0013  INT10
01                ; GETREC(Z,VARIABLE NAME);
02A73227'004000 CL200:  JSR              CL20     ; GETREC:
03 73230'037033         LDA@     3       PC,2     ;
04 73231'011033         ISZ              PC,2     ;
05 73232'021400         LDA      0       +0,3     ;
06 73233'035030         LDA      3       SAVE4,2  ; GET OP;
07 73234'031027         LDA      2       SAVE3,2  ; GET ZONE;
08 73235'007400         JSR@             +0,3     ; EXIT TO MONITOR ENTRY;
09 73236'105000         MOV      0,1             ;
10A73237'000000         JMP              CL212    ;
11
12                ; OUTTEXT(Z,STRING VARIABLE NAME);
13A73240'004000 CL207:  JSR              CL20     ;
14 73241'006236         JSR@             .CL4     ; TAKEADDRESS(MODIF,AC1);
15 73242'000403         JMP              .+3      ;
16A73243'004000 CL201:  JSR              CL20     ;
17 73244'006237         JSR@             .CL5     ; TAKEVALUE(MODIF,AC1);
18 73245'121000         MOV      1,0             ;
19 73246'000410         JMP              CL209    ; GOTO EXECUTE;
20A73247'004000 CL204:  JSR              CL20     ;
21 73250'006237         JSR@             .CL5     ; TAKEVALUE(MODIF,AC0);
22 73251'000405         JMP              CL209    ; GOTO EXECUTE;
23A73252'004000 CL208:  JSR              CL20     ; GETZONEADDR;
24 73253'006236         JSR@             .CL4     ; TAKEADDR(MODIF,FILE);
25 73254'125221         MOVZR    1,1     SKP      ; ADDR:=FILE//2;
26A73255'004000 CL206:  JSR              CL20     ;
27
28                ; EXECUTE;
29                ;              CALL:     EXIT TO MONITOR I/O-TABLE
30                ; AC0          PARAM2    PARAM2
31                ; AC1          PARAM1    PARAM1
32                ; AC2          CUR       ZONE
33                ; AC3                    OP
34 73256'035030 CL209:  LDA      3        SAVE4,2  ; EXECUTE:
35 73257'031027         LDA      2        SAVE3,2  ;
36 73260'007400         JSR@             +0,3     ; EXIT TO MONITOR ENTRY;
37 73261'002234         JMP@             .CL0     ; RETURN AND GET CUR;
38                ;
39                ; END ZONEPROCESSES;
```

```
↑ 0014  INT10
01                ; MOVEWORD(TO ADDR=PC,VALUE=PC+1);
02                ; TO ADDR  ,MODIFBITS(14:15)
03                ; VALUE    ,    -    (12:13)
04                CL301:                              ; MOVEWORD:
05  73262'006236          JSR@              .CL4      ;   ADDRESS:=INST(PC.CUR);
06  73263'045030          STA       1       SAVE4,2   ;
07  73264'006237          JSR@              .CL5      ;   TAKEVALUE(MODIF,R.CUR);
08  73265'047030          STA@      1       SAVE4,2   ;   WORD(ADDRESS):=VALUE;
09  73266'002235          JMP@              .CL1      ; RETURN;
10
11                ; MOVE(FROM STRING,FROMINDEX,TO STRING,TOINDEX,COUNT);
12                ; GETS A NEW MODIFBITS IN PC.CUR;
13                ; SAVE1: WORK
14                ; SAVE2: COUNT                              , MODIFBITS(6:7)
15                ; SAVE3: ADDR OF TOSTRING(TOINDEX)      ,     -     (14:15),(12:13)
16                ; SAVE4:   -  -  FROMSTRING(FROMINDEX) ,     -     (10:11),(8:9)
17                ; SAVE5: WORK
18                CL320:                              ; MOVE:
19  73267'023033          LDA@      0       PC,2      ;   GET NEW MODIFBITS IN PC.CUR;
20  73270'011033          ISZ               PC,2      ;
21  73271'004423          JSR               CL333     ;   TAKEADDRESS AND MODIFY;
22  73272'045030          STA       1       SAVE4,2   ;   SAVE4.CUR:= VAL;
23  73273'004421          JSR               CL333     ;   TAKEADDRESS AND MODIFY;
24  73274'045027          STA       1       SAVE3,2   ;   SAVE3.CUR:=VAL;
25  73275'000402          JMP               .+2       ;
26
27                ; MOVESTRING(FROM ADDR, TO ADDR,COUNT);
28                ; SAVE2:  COUNT     , MODIFBITS(10:11)
29                ; SAVE3:  FROM ADDR,      -     (14:15)
30                ; SAVE4:  TO ADDR  ,      -     (12:13)
31                ; SAVE5:  WORK
32  73276'004410  CL302:  JSR               CL321     ; MOVESTRING:
33  73277'006237          JSR@              .CL5      ;   TAKEVALUE(SAVE,COUNT);
34  73300'045026          STA       1       SAVE2,2   ;   SAVE2.CUR:= COUNT;
35  73301'024404          LDA       1       .S2       ;
36  73302'133000          ADD       1,2               ;   PARAMADDR:= .SAVE2+(CUR);
37  73303'006224          MOVE                        ;   MOVE(PARAMADDR);
38  73304'002234          JMP@              .CL0      ; RETURN;
39  73305'000026  .S2:                      SAVE2     ;
```

```
↑ 0015  INT10
01                ; TAKEADDRESSES(ADDR1,ADDR2);
02                ; SAVE3:  ADDR1     , MODIFBITS(14:15)
03                ; SAVE4:  ADDR2     ,     -     (12:13)
04 73306'055031 CL321:  STA     3       SAVE5,2 ; TAKEADDRESSES:
05 73307'006236         JSR@            .CL4    ;   TAKEADDRESS(MODIF,SAVE4.CUR);
06 73310'045027         STA     1       SAVE3,2 ;
07 73311'006236         JSR@            .CL4    ;   TAKEADDRESS(MODIF,SAVE3.CUR);
08 73312'045030         STA     1       SAVE4,2 ;
09 73313'003031         JMP@            SAVE5,2 ; RETURN TO LINK;
10
11                ; TAKEADDRES AND MODIFY;
12                ; ADDRESS           , MODIFBITS(14:15)
13                ; VALUE             ,     -     (12:13)
14                ; SAVE3,   WORK
15                ; AC1:=RESULT=ADDRESS+VALUE
16 73314'055031 CL333:  STA     3       SAVE5,2 ; TAKEADDRESS AND MODIFY:
17 73315'006236         JSR@            .CL4    ;   TAKEADDRESS(MODIF,ADDRESS);
18 73316'045027         STA     1       SAVE3,2 ;
19 73317'006237         JSR@            .CL5    ;   TAKEVALUE(MODIF,VALUE);
20 73320'035027         LDA     3       SAVE3,2 ;
21 73321'167000         ADD     3,1             ;   VAL:=ADDRESS+VALUE;
22 73322'003031         JMP@            SAVE5,2 ; RETURN TO LINK;
23
24                ; CONVERT(FROMSTRING,TO STRING,CONVERTTABLE,COUNT).
25                ; SAVE1: ADDR OF CONVERTTABLE   , MODIFBITS(10:11)
26                ; SAVE2:  -   -  COUNT          ,     -     (8:9)
27                ; SAVE3:  -   -  FROM STRING    ,     -     (14:15)
28                ; SAVE4:  -   -  TO STRING      ,     -     (12:13)
29                ; SAVE5: WORK
30 73323'004763 CL324:  JSR             CL321   ; CONVERT:
31 73324'006236         JSR@            .CL4    ;   TAKEADDRESSES(INSTRING,OUTSTR
32 73325'045025         STA     1       SAVE1,2 ;   TAKEADDRESS(TABLE);
33 73326'006237         JSR@            .CL5    ;   TAKEVALUE(COUNT);
34 73327'125400         INC     1,1             ;
35 73330'045026         STA     1       SAVE2,2 ;
36 73331'004414 CL325:  JSR             CL327   ; WHILE COUNT>0 DO
37 73332'004410         JSR             CL326   ; BEGIN
38 73333'025025         LDA     1       SAVE1,2 ;   GETBYTE(INCR(INSTRING),BYTE);
39 73334'107000         ADD     0,1             ;   GETBYTE(TABLE+BYTE,BYTE);
40 73335'006174         GETBYTE                 ;
41 73336'025030         LDA     1       SAVE4,2 ;   PUTBYTE(OUTSTRING,BYTE);
42 73337'006175         PUTBYTE                 ;
43 73340'011030         ISZ             SAVE4,2 ;   INCR(OUTSTRING)
44 73341'000770         JMP             CL325   ; END;
45 73342'025027 CL326:  LDA     1       SAVE3,2 ; INCR(ADDR);
46 73343'011027         ISZ             SAVE3,2 ;
47 73344'002174         .GETBYTE                ;
48 73345'015026 CL327:  DSZ             SAVE2,2 ; COUNT:=COUNT-1;
49 73346'001400         JMP             +0,3    ;
50 73347'002235         JMP@            .CL1    ; RETURN;
```

```
↑ 0016  INT10
01                ; SUBINTERPRETER-TABLE CONTINUED:
02 73350'000000           0                          ; EMPTY ELEMENT;
03 73351'000000           0                          ;
04        073352'CL22=     .                          ; CAT-ENTRIES:
05 73352'000471'           CL205     ; 346           : CREATE ENTRY;
06 73353'073252'           CL208     ; 347           : LOOKUP ENTRY;
07 73354'073252'           CL208     ; 350           : CHANGE ENTRY;
08 73355'073255'           CL206     ; 351           : REMOVE ENTRY;
09 73356'000471'           CL205     ; 352           : INIT CAT
10 73357'073247'      CL208 CL204    ; 353           : UPDATE CAT
11                ; END OF I/O-TABLE;                   SSTENTRY;
12
13
14        072462 .IFN      CL22-CL11+GOS-CREATE       ; IF ADDR USED FOR MONITORENTRIES
15L                        .BLK 30000                 ; ARE NOT CORRECT BECAUSE OF
16                .ENDC                               ; CHANGES IN MONITOR OR INTERPRE-
17                                                    ; TER, A BLOCK OF 30 K IS CREATED
18                                                    ; TO INDICATE AN ERROR.
```

```
↑ 0017  INT10
01                ; TRANSLATE(FROMBYTE,TOBYTE,CONVERTTABLE);
02                ; SAVE: TABLEADDR//2              , MODIFBITS(10:11)
03                ; SAVE1:
04                ; SAVE2:
05                ; SAVE3: ADDR OF FROMBYTE         ,      -     (14:15)
06                ; SAVE4:  -    -  TOBYTE          ,      -     (12:13)
07                ; SAVE5: WORK
08                CL322:                                ; TRANSLATE:
09 73360'004726           JSR               CL321     ;  TAKEADDRESSES(INSTRING,OUTSTRIN
10 73361'006236           JSR@              .CL4      ;  TAKEADDRESS(TABLE);
11 73362'125220           MOVZR   1,1                 ;
12 73363'045024           STA     1         SAVE,2    ;
13 73364'025027           LDA     1         SAVE3,2   ;
14 73365'006174           GETBYTE                       ;
15 73366'101300           MOVS    0,0                 ;  AC0(BIT0-7)=BYTE
16 73367'041027           STA     0         SAVE3,2   ;
17 73370'035024           LDA     3         SAVE,2    ;  TABLEADDR:=TABLEADDR//2;
18 73371'021400 CL323:    LDA     0         +0,3      ;  GET TABLEBYTE:
19 73372'025401           LDA     1         +1,3      ;  VALUE(ARG):=DEFAULTVALUE;
20 73373'101005           MOV     0,0       SNR       ;  IF WORD(TABLEADDR)!ARG,VALUE!<>
21 73374'000410           JMP               CL329     ;  BEGIN
22 73375'175400           INC     3,3                 ;    TABLEADDR:=TABLEADDR+1;
23 73376'025027           LDA     1         SAVE3,2   ;    GET FROMBYTE; !FROMBYTE=ARG I
24 73377'122400           SUB     1,0                 ;    ARG:=ARG-FROMBYTE;
25 73400'024143           LDA     1         .255      ;
26 73401'107400           AND     0,1                 ;    AC1:=VALUE;
27 73402'122404           SUB     1,0       SZR       ;    IF ARG<>0 THEN GOTO GET TABLE
28 73403'000766           JMP               CL323     ;  END;
29 73404'121000 CL329:    MOV     1,0                 ;
30 73405'025030           LDA     1         SAVE4,2   ;
31 73406'006175           PUTBYTE                       ;  TOBYTE:=VALUE(ARG);
32 73407'002235           JMP@              .CL1      ; RETURN;
33
34                ; INSERT(BYTE,TO RECORD,RECORD INDEX);
35                ; SAVE1:
36                ; SAVE2: BYTE VALUE               , MODIFBITS(14:15)
37                ; SAVE3: ADDR OF RECORD(INDEX)    ,      -     (12:13)+(10:11)
38                ; SAVE4:
39                ; SAVE5: WORK
40 73410'006237 CL316:    JSR@              .CL5      ; INSERT:
41 73411'034143           LDA     3         .255      ;   TAKEVALUE(MODIF,VALUE);
42 73412'137400           AND     1,3                 ;
43 73413'055026           STA     3         SAVE2,2   ;
44 73414'004700           JSR               CL333     ;   TAKEADDRESS AND MODIFY;
45 73415'021026           LDA     0         SAVE2,2   ;
46 73416'006175           PUTBYTE                       ;
47 73417'002235           JMP@              .CL1      ; RETURN;
48
49                ; STORE REGISTER;
50                ; STORE R.CUR IN ADDRESS=WORD(PC.CUR) AND INCR(PC.CUR);
51                CL305:                                ; STORE:
52 73420'037033           LDA@    3         PC,2      ;   ADDRESS:=WORD(PC.CUR);
53 73421'011033           ISZ               PC,2      ;
54 73422'021032           LDA     0         R,2       ;
55 73423'041400           STA     0         +0,3      ;   WORD(ADDRESS):=R.CUR;
56 73424'002235           JMP@              .CL1      ; RETURN;
```

```
                ; COMPARESTRING(STRING1,STRING2);
                ; SAVE : BYTE1
                ; SAVE1:
                ; SAVE2: COUNT                    , MODIFBITS(10:11)
                ; SAVE3: ADDR OF STRING1          ,      -    (14:15)
                ; SAVE4:  -   -  STRING2          ,      -    (12:13)
                ; SAVE5: WORK
                CL304:                                ; COMPARESTRING:
 73425'004661           JSR             CL321     ;   TAKEADDRESSES(STRING1,STRING2)
 73426'006237           JSR@            .CL5      ;   TAKEVALUE(MODIF,VALUE);
 73427'045026           STA     1       SAVE2,2   ;   SAVE2.CUR:= VALUE;
 73430'004712 CL34:     JSR             CL326     ;   REPEAT
 73431'041024           STA     0       SAVE,2    ;     GETBYTE(SAVE3,SAVE);
 73432'025030           LDA     1       SAVE4,2   ;
 73433'006174           GETBYTE                   ;     GETBYTE(SAVE4,R);
 73434'011030           ISZ             SAVE4,2   ;     INCR(SAVE4.CUR);
 73435'025024           LDA     1       SAVE,2    ;
 73436'106400           SUB     0,1               ;     R.CUR:=RESULT OF COMPARE;
 73437'045032           STA     1       R,2       ;
 73440'125004           MOV     1,1     SZR       ;     IF R.CUR<>0 THEN
 73441'002235           JMP@            .CL1      ;     RETURN;
 73442'004703           JSR             CL327     ;   UNTIL COUNT=0;
 73443'000765           JMP             CL34      ; RETURN;

                ; CALL
                ;   1: PC POINTS AT THE FIRST CELL IN THE PROCEDURE, WHICH CONTAI
                ;      THE ADDR OF THE CELL USED TO SAVE THE LINK=CALLED FROM.
                ;   2: LINK=PC+1 (JUMP BACK TO) IS WRITEN IN THIS SAVE LINKCELL.
                ;   3: PC:=SAVELINKCELL+1 (=NOW CONTINUE HERE)
 73444'037033 CL306:    LDA@    3       PC,2      ; LINK:=W(PC.CUR);
 73445'011033           ISZ             PC,2      ; INCR(PC.CUR);
 73446'021033           LDA     0       PC,2      ;
 73447'041400           STA     0       +0,3      ; LINKWORD.LINK:=PC.CUR;
 73450'175400           INC     3,3               ; PC.CUR:=LINK+1:
 73451'000426           JMP             CL360+1   ;

                ; BINDEC(BIN.VALUE NAME, DEC.VALUE NAME);
 73452'006237 CL307:    JSR@            .CL5      ; TAKEVALUE(MODIF,VALUE);
 73453'045030           STA     1       SAVE4,2   ;
 73454'006236           JSR@            .CL4      ; TAKEADDRESS(MODIF,ADDRESS);
 73455'021030           LDA     0       SAVE4,2   ;
 73456'006232           BINDEC                    ;
 73457'002235           JMP@            .CL1      ; RETURN;

                ; GO CODE;
                ;           CALL:                   RETURN:
                ; AC0       MODIFBITS=-INDEX        WORD(PC.CUR)
                ; AC1
                ; AC2       CUR                     CUR
                ; AC3                               ADDR=CUR+INDEX
 73460'155000 CL303:    MOV     2,3               ; GO CODE:
 73461'116400           SUB     0,3               ;   ADDR:=CUR+INDEX;
 73462'023033           LDA@    0       PC,2      ;   MODIFBITS:=WORD(PC.CUR);
 73463'011033           ISZ             PC,2      ;   INCR(PC.CUR);
 73464'003400           JMP@            +0,3      ; GOTO W(ADDR);
```

```
↑ 0019  INT10
                ; LINK;
                ; PC POINTS AT THE CELL, WHICH CONTAINS THE ADDRESS OF THE RETURN
                ; ADDRESSCELL.
                ; IF RETURNADDRESS>=0 THEN PC:=POINT AND CONTINUE HERE
                ;                      ELSE EXIT FROM BOTTOM OF A
                ;                      GIVEUPPROCEDURE. NOTE THAT STDGIVEUP IS AL-
                ;                      WAYS CALLED BEFORE ENTRY TO A USER GIVEUP-
                ;                      PROCEDURE.
                ;                      ZONE:=-POINT;
                ;                      PC.CUR:=SAVEPC.ZONE( PC IN PROGRAM).
                ;                      GOTO SAVELINK.ZONE=RETURN FROM WAITTRANSFER
73465'037033 CL308:  LDA@    3        PC,2     ; LINK:
73466'035400         LDA     3        +0,3     ; POINT:=WORD(WORD(PC.CUR));
73467'175113         MOVL#   3,3      SNC      ; IF POINT >=0 THEN
73470'000407         JMP              CL360+1  ; GOTO "GOTO";
73471'170400         NEG     3,2              ; ZONE:=-POINT;
73472'034040         LDA     3        CUR      ;
73473'021013         LDA     0        ZBUFFE,2;
73474'041433         STA     0        PC,3     ; PC.CUR:=SAVEPC.ZONE;
73475'003014         JMP@             ZSIZE,2  ; GOTO SAVELINK.ZONE;

73476'037033 CL360:  LDA@    3        PC,2     ; GOTO:
73477'055033         STA     3        PC,2     ;
73500'002235         JMP@             .CL1     ; RETURN;

                ; JUMP;
                ; IF R.CUR SATISFIES THEN CONDITION AT THE CONTENT OF PC
                ;                     ELSE INCR(PC) AND CONTINUE.
73501'115005 CL309:  MOV     0,3      SNR      ; JUMP:
73502'000774         JMP              CL360    ;   IF MODIF=0 THEN GOTO "GOTO".
73503'021501         LDA     0        BIT,3    ;   AC0:=1BIT(MODIF);
73504'004404         JSR              CL399    ;   INDEX:=0;
73505'100160         1B0+1B9+1B10+1B11         ;   COND(0) "=,>=,<=" R=0;
73506'100034         1B0+1B11+1B12+1B13        ;   COND(1) "<=,<>,<" R<0;
73507'100052         1B0+1B10+1B12+1B14        ;   COND(2) ">=,<>,>" R>0;
73510'025032 CL399:  LDA     1        R,2      ;
73511'125005         MOV     1,1      SNR      ;   IF R.CUR<>0 THEN
73512'000404         JMP              CL398    ;   BEGIN
73513'175400         INC     3,3               ;     INDEX:=1;
73514'125103         MOVL    1,1      SNC      ;     IF R.CUR>0 THEN INDEX:=2;
73515'175400         INC     3,3               ;   END;
73516'025400 CL398:  LDA     1        +0,3     ;   C:=COND(INDEX);
73517'107404         AND     0,1      SZR      ;   IF MODIF AND C THEN
73520'000756         JMP              CL360    ;   GOTO "GOTO";
73521'011033         ISZ              PC,2     ; SKIP:
73522'002235         JMP@             .CL1     ;
```

```
01                ;PROCEDURE SENDTO OPERATOR(COMMAND,ADDRESS,COUNT,BUF);
02                ;
03                ;         CALL:              RETURN:
04                ; AC0     COMMAND            IRRELEVANT
05                ; AC1     ADDRESS              -
06                ; AC2     CUR                BUF
07                ; AC3     LINK               CUR
08
09 73523'055024 CL400:  STA     3       SAVE,2  ; SEND TO OPERATOR:
10 73524'060277          INTDS                   ; DIRTY WORK!
11 73525'040407          STA     0       CL40A   ; ONLY ONE PROCESS AT TIME,
12 73526'044410          STA     1       CL40C   ; THE ENTERPRETER IS REENTRANT.
13 73527'024404          LDA     1       CL401   ; AC1:=ADDR, AC2:=NAME OF RECEIVE
14 73530'031035          LDA     2       .OPER,2 ; SENDMESSAGE(MESS0.CUR,OPERATOR)
15 73531'006004          SENDMESSAGE             ;
16 73532'003424          JMP@            SAVE,3  ;
17 73533'073534'CL401:   .+1     ; MESSAGE BUILD UP IN NEXT 3 CELLS:
18 73534'000000 CL40A:   0       ; OPERATION
19 73535'000120 CL40B:   80      ; COUNT
20 73536'000000 CL40C:   0       ; ADDRESS
21
22                ; PROCEDURE OPMESS(STRING VARIABLE NAME);
23                CL310:                          ; OPMESS:
24 73537'006236          JSR@            .CL4    ;   TAKEADDRESS(MODIF,ADDRESS);
25 73540'020121          LDA     0       .3      ;   OUTPUTMODE;
26 73541'004762          JSR             CL400   ;
27 73542'006005          WAITANSWER              ;
28 73543'002234          JMP@            .CL0    ; RETURN;
29                                               ;
30                ; PROCEDURE OPIN(STRING VARIABLE NAME);
31                CL311:                          ; OPIN:
32 73544'006236          JSR@            .CL4    ;   TAKEADDRESS(MODIF,ADDRESS);
33 73545'021034          LDA     0       OP,2    ;   OP.CUR CONTAINS BUFFERADDR, SE
34 73546'101004          MOV     0,0     SZR     ;   IF OP.CUR <> 0 THEN
35 73547'002235          JMP@            .CL1    ;   RETURN1;
36 73550'020120          LDA     0       .1      ;   INPUTMODE;
37 73551'004752          JSR             CL400   ;
38 73552'051434          STA     2       OP,3    ;   OP.CUR:=BUFERADDRESS;
39 73553'002234          JMP@            .CL0    ; RETURN;
40                                               ;
41                ; OPSTATUS(ZONE.Z0,ERRORS);
42                ; SAVE1:
43                ; SAVE2: VALUE             , MODIFBITS(14:15)
44                ; SAVE3: ADDRESS           ,      -     (12:13)
45                ; SAVE4: LINK
46                ; SAVE5:
47                CL313:                          ; OPSTATUS:
48 73554'006237          JSR@            .CL5    ;   TAKEVALUE(MODIF,VALUE);
49 73555'045026          STA     1       SAVE2,2 ;   SAVE2.CUR:= VALUE;
50 73556'006236          JSR@            .CL4    ;   TAKEADDRESS(SAVE3,ADDRESS);
51 73557'045027          STA     1       SAVE3,2 ;
52 73560'004402          JSR             CL402   ;   OUTMESSAGE;
53 73561'002234          JMP@            .CL0    ; RETURN;
```

```
↑ 0021  INT10
01                ; PROCEDURE OUTMESSAGE;
02                ; THE PROCEDURE OUTPUTS A STRING CORRESPONDING TO EACH BIT IN
03                ; WORD, IF THE BIT IS SET;
04                ;              CALL:
05                ; SAVE2:       WORD
06                ; SAVE3:       ADDRESS OF STRING
07                ; SAVE4:       LINK
08 73562'055030 CL402:  STA      3        SAVE4,2 ;
09                CL403:                            ; REP:
10 73563'021026         LDA      0        SAVE2,2 ;   WORD:=SAVE2.CUR;
11 73564'101005         MOV      0,0      SNR     ;   IF WORD=0 THEN
12 73565'003030         JMP@             SAVE4,2 ;   RETURN;
13 73566'101123         MOVZL    0,0      SNC     ;   IF WORD(0)=1 THEN
14 73567'000407         JMP              CL404   ;   BEGIN
15 73570'041026         STA      0        SAVE2,2 ;     SAVE2:=WORD SHIFT 1;
16 73571'025027         LDA      1        SAVE3,2 ;     ADDRESS:=SAVE3.CUR;
17 73572'020121         LDA      0        .3      ;     COMMAND:=OUT;
18 73573'004730         JSR              CL400   ;     SEND TO OPERATOR;
19 73574'006005         WAITANSWER                ;     WAITANSWER(IRR,IRR,BUF);
20 73575'171001         MOV      3,2      SKP     ;   END
21 73576'041026 CL404:  STA      0        SAVE2,2 ;   ELSE SAVE2.CUR:=WORD SHIFT 1;
22
23                                                ; SKIP:
24 73577'025027         LDA      1        SAVE3,2 ;   ADDRESS:=SAVE3.CUR;
25 73600'006174         GETBYTE                   ;   REPEAT
26 73601'125400         INC      1,1              ;     GETBYTE; ADDR:=ADDR+1;
27 73602'101004         MOV      0,0      SZR     ;   UNTIL BYTE=0;
28 73603'000775         JMP              .-3     ;   SAVE3.CUR:=ADDR;
29 73604'045027         STA      1        SAVE3,2 ;
30 73605'000756         JMP              CL403   ;
31
32                ; PROCEDURE DECBIN(DEC.VAL NAME, BIN.VAL NAME);
33 73606'006236 CL315:  JSR@             .CL4    ; DECBIN:
34 73607'006233         DECBIN                    ;   TAKEADDRESS(MODIF,ADDR);
35 73610'037033         LDA@     3        PC,2    ;   ADDR:=WORD(PC.CUR);
36 73611'011033         ISZ              PC,2    ;   W(ADDR):=BIN.VALUE;
37 73612'045400         STA      1        +0,3    ;   INCR(PC.CUR);
38 73613'002235         JMP@             .CL1    ; RETURN;
```

```
↑ 0022  INT10
01                ; PROCEDURE OPWAIT(LENGTH);
02                CL312:                                  ; OPWAIT:
03 73614'011033           ISZ               PC,2        ;   INCR(PC);
04 73615'031034           LDA       2       OP,2        ;
05 73616'151005           MOV       2,2     SNR         ;   IF OP.CUR<>0 THEN
06 73617'002234           JMP@              .CL0        ;   BEGIN
07 73620'006005           WAITANSWER                    ;     WAITANSWER
08 73621'171000           MOV       3,2                 ;     LENGTH=W(PC.CUR-1):=BYTECOUN
09 73622'035033           LDA       3       PC,2        ;     OP.CUR:=0(=STATUS);
10 73623'047777           STA@      1       -1,3        ;                                   /
11 73624'041034           STA       0       OP,2        ;   END;                  STATUS C74
12 73625'002235           JMP@              .CL1        ; RETURN;
13                                                      ;
14                ; PROCEDURE OPTEST;
15                CL314:                                  ; OPTEST:
16 73626'035034           LDA       3       OP,2        ;   OP.CUR:=BUFFERADDR;
17 73627'035405           LDA       3       RECEIV,3;
18 73630'102400           SUB       0,0                 ;
19 73631'175122           MOVZL     3,3     SZC         ;   IF RECEIVER.BUF<0 !ANSWERED!
20 73632'100000           COM       0,0                 ;   THEN !ANSWERED! R:=1
21 73633'041032           STA       0       R,2         ;   ELSE R:=0;
22 73634'002235           JMP@              .CL1        ; RETURN;
```

```
↑ 0023  INT10
01                ; AT DIRECT OPERATIONS: AC0=MODIF=VALUE;
02 73635'037033 CL120:  LDA@    3       PC,2    ; AND :     AC0:=R AND WORD(WORD(PC
03 73636'021400         LDA     0       +0,3    ;
04 73637'000402         JMP             .+2     ;
05 73640'023033 CL110:  LDA@    0       PC,2    ; ANDC:     AC0:=R AND WORD(PC);
06 73641'011033         ISZ             PC,2    ;
07 73642'025032 CL100:  LDA     1       R,2     ; ANDD:     AC0:=R AND AC0;
08 73643'123400         AND     1,0             ;
09 73644'000406         JMP             CL101
10
11 73645'037033 CL121:  LDA@    3       PC,2    ; LOAD :    AC0:=WORD(WORD(PC));
12 73646'021400         LDA     0       +0,3    ;
13 73647'000402         JMP             .+2     ;
14 73650'023033 CL111:  LDA@    0       PC,2    ; LOADC:    AC0:=WORD(PC);
15 73651'011033         ISZ             PC,2    ;
16 73652'041032 CL101:  STA     0       R,2     ; LOADD:    R:=AC0;
17 73653'002235         JMP@            .CL1    ; RETURN WITH R=AC0;
18
19 73654'037033 CL122:  LDA@    3       PC,2    ; + :       AC1:=R+WORD(WORD(PC));
20 73655'021400         LDA     0       +0,3    ;
21 73656'000402         JMP             .+2     ;
22 73657'023033 CL112:  LDA@    0       PC,2    ; +C:       AC1:=R+WORD(PC);
23 73660'011033         ISZ             PC,2    ;
24 73661'025032 CL102:  LDA     1       R,2     ; +D:       AC1:=R+AC0;
25 73662'107000         ADD     0,1             ;
26 73663'045032 CL132:  STA     1       R,2     ;
27 73664'002235         JMP@            .CL1    ; RETURN WITH R=AC1;
28
29 73665'037033 CL123:  LDA@    3       PC,2    ; - :       AC1:=R-WORD(WORD(PC));
30 73666'021400         LDA     0       +0,3    ;
31 73667'000402         JMP             .+2     ;
32 73670'023033 CL113:  LDA@    0       PC,2    ; -C:       AC1:=R-WORD(PC);
33 73671'011033         ISZ             PC,2    ;
34 73672'025032 CL103:  LDA     1       R,2     ; -D:       AC1:=R-AC0;
35 73673'106400         SUB     0,1             ;
36 73674'045032         STA     1       R,2     ;
37 73675'002235         JMP@            .CL1    ; RETURN WITH R=AC1;
38
39 73676'037033 CL124:  LDA@    3       PC,2    ; SHIFT :   AC0:=R SHIFT WORD(WORD(
40 73677'021400         LDA     0       +0,3    ;
41 73700'000402         JMP             .+2     ;
42 73701'023033 CL114:  LDA@    0       PC,2    ; SHIFTC:   AC0:=R SHIFT WORD(PC);
43 73702'011033         ISZ             PC,2    ;
44 73703'115005 CL104:  MOV     0,3     SNR     ; SHIFTD:   AC0:=R SHIFT AC0;
45 73704'002235         JMP@            .CL1    ;
46 73705'025032         LDA     1       R,2     ;
47 73706'101113         MOVL#   0,0     SNC     ; IF SHIFTS>0 THEN
48 73707'174400         NEG             3,3     ; COUNT:=-SHIFTS;
49 73710'101113         MOVL#   0,0     SNC     ; FOR COUNT:=COUNT STEP -1 UNTIL 0
50 73711'125121         MOVZL   1,1     SKP     ;  R:= R SHIFT SIGN(SHIFTS);
51 73712'125220         MOVZR   1,1             ;
52 73713'175404         INC     3,3     SZR     ;
53 73714'000774         JMP             .-4     ;
54 73715'000746         JMP             CL132   ;
```

```
↑ 0024  INT10
01 73716'006236 CL901:  JSR@            .CL4     ; LOAD BYTE:
02                                               ; TAKEADDRESS(MODIF,ADDRESS);
03 73717'006174         GETBYTE                  ; GETBYTE(ADDRESS,VALUE);
04 73720'000732         JMP             CL101    ;
05
06 73721'006236 CL902:  JSR@            .CL4     ; LOAD WORD;
07                                               ; TAKEADDRESS(MODIF,ADDRESS);
08 73722'006174         GETBYTE                  ; GETBYTE(ADDRESS,BYTE);
09 73723'101300         MOVS    0,0              ;
10 73724'041032         STA     0       R,2      ; VALUE:=BYTE*256;
11 73725'125400         INC     1,1              ;
12 73726'006174         GETBYTE                  ; GETBYTE(ADDRESS+1,BYTE);
13 73727'000732         JMP             CL102    ; VALUE:=VALUE+BYTE;
14
15                                               ; LOAD NEGATIVE;
16 73730'006237 CL903:  JSR@            .CL5     ; TAKEVALUE(MODIF,VAL);
17 73731'120400         NEG     1,0              ; VALUE:=-VALUE;
18 73732'000720         JMP             CL101    ;
19
20 73733'037033 CL125:  LDA@    3       PC,2     ; EXTRACT : AC0:=R EXTRACT W(W(PC)
21 73734'021400         LDA     0       +0,3     ;
22 73735'000402         JMP             .+2      ;
23 73736'023033 CL115:  LDA@    0       PC,2     ; EXTRACTC: AC0:=R EXTRACT WORD(PC
24 73737'011033         ISZ             PC,2     ;
25 73740'034114 CL105:  LDA     3       .16      ; EXTRACTD: AC0:=R EXTRACT AC0;
26 73741'116400         SUB     0,3              ;
27 73742'021501         LDA     0       BIT,3    ;
28 73743'100520         NEGZL   0,0              ;
29 73744'100000         COM     0,0              ;
30 73745'000675         JMP             CL100    ;
31
32 73746'037033 CL126:  LDA@    3       PC,2     ; * :          AC0:=R*WORD(WORD(PC))
33 73747'021400         LDA     0       +0,3     ;
34 73750'000402         JMP             .+2      ;
35 73751'023033 CL116:  LDA@    0       PC,2     ; *C:          AC0:=R*WORD(PC);
36 73752'011033         ISZ             PC,2     ;
37 73753'025032 CL106:  LDA     1       R,2      ; *D:          AC0:=R*AC0;
38 73754'006176         MULTIPLY                 ;
39 73755'000706         JMP             CL132    ;
```

```
↑ 0025  INT10
01 73756'037033 CL127:  LDA@     3       PC,2     ; / :          AC0:=R/WORD(WORD(PC)
02 73757'025400         LDA      1       +0,3     ;
03 73760'000402         JMP              .+2      ;
04 73761'027033 CL117:  LDA@     1       PC,2     ; /C:          AC0:=R/WORD(PC);
05 73762'011033         ISZ              PC,2     ;
06 73763'102520 CL137:  SUBZL    0,0              ;      SAVE2=FORTEGN:=1(POS);
07 73764'041026         STA      0       SAVE2,2  ;      DIVIDEND:=R.CUR;
08 73765'021032         LDA      0       R,2      ;      IF DIVIDEND<0 THEN
09 73766'101133         MOVZL#   0,0     SNC      ;      BEGIN
10 73767'000403         JMP              CL147    ;        DIVISOR:=-DIVISOR;
11 73770'124400         NEG      1,1              ;        DIVIDEND:=-DIVIDEND;
12 73771'100400         NEG      0,0              ;      END;
13 73772'125133 CL147:  MOVZL#   1,1     SNC      ;      IF DIVISOR<0 THEN
14 73773'000403         JMP              CL157    ;      BEGIN
15 73774'045026         STA      1       SAVE2,2  ;        FORTEGN:=NEGATIV;
16 73775'124400         NEG      1,1              ;        DIVISOR:=-DIVISOR;
17 73776'006177 CL157:  DIVIDE                   ;      END;
18 73777'025026         LDA      1       SAVE2,2  ;      DIVISION AF TO POS. TAL;
19 74000'125122         MOVZL    1,1     SZC      ;      IF FORTEGN=NEGATIV THEN
20 74001'100400         NEG      0,0              ;      QVOTIENT:=-QVOTIENT;
21 74002'000650         JMP              CL101    ;
22 74003'105000 CL107:  MOV      0,1              ; /D:          AC0:=R/AC0;
23 74004'000757         JMP              CL137    ;
```

```
↑ 0026  INT10
01                ; STDGIVEUP
02                ; STDGIVEUP IS ALWAYS CALLED BEFORE JUMPING FURTHER TO THE USER-
03                ; GIVEUPPROCEDURE.
04                ; ENTRY: A JUMP TO STDGIVEUP IS DONE FROM WAITTRANSFER BY A JMP@
05                ;        ZGIVEUP. ZGIVEUP=CL6 AND SIZE=ADDR OF USER GIVEUPPROCEDU
06                ;        SAVELINK.ZONE=ZSIZE:=WAITTRANSFERS RETURNLINK,
07                ;        SAVEPC.ZONE  =ZBUF :=PC+1.
08                ;        THE TWO SAVELOCATIONS ARE USED AT RETURN FROM USER GIVEU
09                ; EXIT : PC POINTING AT USER GIVEUPPROCEDURE AND RETURNLINK FROM
10                ;        GIVEUPPROCEDURE IS SET TO -ZONE TO INDICATE THAT STDGIVE
11                ;        HAVE BEEN CALLED. (IT IS POSSIBLE TO CALL A USER GIVEUP-
12                ;        PROCEDURE DIRECTLY).
13                ;         CALL        RETURN
14                ; AC0                 PC.CUR
15                ; AC1
16                ; AC2      ZONE
17                ; AC3      WAITTRANSFER LINK
18                ; SIZE.Z   ZONE GIVEUP ADDRESS
19
20
21 74005'055014 CL6:     STA     3       ZSIZE,2 ; SAVELINK.ZONE:=LINK;
22 74006'035003          LDA     3       SIZE,2  ; GIVEUPPROCEDURE:=SIZE.ZONE;
23 74007'140400          NEG     2,0             ;
24 74010'041400          STA     0       +0,3    ; PROC(0):=-ZONE;
25 74011'165400          INC     3,1             ; START IN USERGIVEUPPROC:=PROC+1
26 74012'034040          LDA     3       CUR     ;
27 74013'021433          LDA     0       PC,3    ;
28 74014'041013          STA     0       ZBUFFE,2; SAVEPC.ZONE:=PC.CUR;
29 74015'045433          STA     1       PC,3    ; PC.CUR:=START;
30 74016'002234          JMP@            .CL0    ; GOTO NEXT0;
```

```
↑ 0027  INT10
01                ; STDBREAK, STOP
02                ;          CALL          RETURN
03                ; AC0      ERRORNO
04                ; AC1      STATUS
05                ; AC2      CUR           CUR
06                ; AC3                    PROG
07                ; SAVE     ZONE
08 74017'101005 CL7:    MOV       0,0       SNR      ; IF ERRORNO.=0 THEN GOTO STOP;
09 74020'000470         JMP       CL72      ; "OWN CLEAN!"
10 74021'041025 cl7;    STA       0         SAVE1,2  ; SAVE ERRORNO;
11 74022'045026         STA       1         SAVE2,2  ; SAVE STATUS;
12 74023'021024         LDA       0         SAVE,2   ; SAVE=ADDR OF ZONE USED ON DEVICE
13 74024'041031         STA       0         SAVE5,2  ; WHICH IS BREAKED;
14 74025'034475         LDA       3         .ZN      ; ZN.CUR=START OF ZONEADDR.TABLE;
15 74026'157000         ADD       2,3                ;
16 74027'055030         STA       3         SAVE4,2  ; SAVE4:=START OF ZONES IN CUR;
17 74030'025003 CL70:   LDA       1         SIZE,2   ; SIZE.CUR=SIZE OF PROCESSDESCRIPT
18 74031'147000         ADD       2,1                ;
19 74032'166405         SUB       3,1       SNR      ;  WHILE ADDR OF ZONES<>CUR+SIZE.C
20 74033'000432         JMP                 CL710    ;  BEGIN
21 74034'033030         LDA@      2         SAVE4,2  ;     SAVE4=ADDR OF ZONE;
22 74035'024054         LDA       1         PROCESS  ;
23 74036'006010         SEARCHITEM                   ;     SEARCHITEM(PROC,ZN,RESULT);
24 74037'151005         MOV       2,2       SNR      ;     IF RESULT=0 THEN
25 74040'000421         JMP                 CL715    ;     GOTO SKIP;
26 74041'033430         LDA@      2         SAVE4,3  ;
27 74042'126520         SUBZL     1,1                ;
28 74043'045004         STA       1         ZMODE,2  ;     ZMODE:=INPUT=1;
29 74044'020457         LDA       0         .CL701   ;     RETURN FROM GIVEUP=CL701 IF
30 74045'041007         STA       0         ZGIVE,2  ;     CALLED BY CLOSE;
31 74046'021005         LDA       0         ZKIND,2  ;
32 74047'041427         STA       0         SAVE3,3  ;     SAVE KIND.ZONE;
33 74050'101120         MOVZL     0,0                ;     IF 1B0=COROUTINE SET IN KIND
34 74051'101220         MOVZR     0,0                ;     CLOSE MUST NOT CALL CWANSWER
35 74052'041005         STA       0         ZKIND,2  ;     WAITTRANSFER);
36 74053'006220         CLOSE                        ;     CLOSE(ZN);
37 74054'020450 CL701:  LDA       0         .INTGIVE;
38 74055'041007         STA       0         ZGIVE,2  ;
39 74056'034040         LDA       3         CUR      ;
40 74057'021427         LDA       0         SAVE3,3  ;     GET SAVED KIND;
41 74060'041005         STA       0         ZKIND,2  ;
42 74061'030040 CL715:  LDA       2         CUR      ; SKIP:
43 74062'011030         ISZ                 SAVE4,2  ;     INCR(ADDR OF ZONES);
44 74063'035030         LDA       3         SAVE4,2  ;
45 74064'000744         JMP                 CL70     ;  END;
46 74065'006011 CL710:  CLEANPROCESS                 ;
47 74066'021025         LDA       0         SAVE1,2  ;  GET ERRORNO;
48 74067'034122         LDA       3         .5       ;
49 74070'116405         SUB       0,3       SNR      ; IF ERRNO=5 THEN
50 74071'000434         JMP                 CL816    ; GOTO WTBREAK;
51 74072'034423         LDA       3         CL813    ; TEXT:='BREAK ';
52 74073'024420         LDA       1         CL811    ; CHAR:="+";
53 74074'101133         MOVZL#    0,0       SNC      ; IF ERRNO<0 THEN
54 74075'000404         JMP                 .+4      ; BEGIN
55 74076'041032         STA       0         R,2      ;     POSBREAK:=FALSE;
56 74077'100400         NEG       0,0                ;     ERRNO:=-ERRNO;
57 74100'024414         LDA       1         CL812    ;     CHAR:="-";
58 74101'123000         ADD       1,0                ; END;
59 74102'041403         STA       0         +3,3     ; TEXT(BYTE7:8):=CHAR,ERRORNO;
```

```
        LDA     0       ZONE,3  ;       START OF ZONE := CUR + STAR OF ZONE ADDR;
        LDA     1       .INTZONE;
        SUB     1,0     SNR     ;       DISP:=0
        JMP             CL71    ;       IF  ZONE.GIVEUP <> STANDARD THEN
        LDA     0       ,7      ;       BEGIN   ! COROUTINES!
        LDA     1       TRECORD,2;        DISP:=7
        MOVL    1,1     SZC     ;         IF  TRECORD(0:0)=1 THEN
        LDA     0       .12     ;          DISP:=12 ;  ! NEW COMOS!
        ADD     0,3             ;
CL711:  STA     3       SAV4,2  ;       STARTOF ZONE:= START OF ZONE + DISP;
```

```
↑ 0029  INT10
01 74103'165120 CL71:   MOVZL   3,1             ; OPERATORMESS:
02 74104'020121         LDA     0       .3      ; OPERATION:=OUTPUT;
03 74105'006451         JSR@            .CL400  ; SEND TO OPER(3,TEXT,ADDR);
04 74106'006005         WAITANSWER              ; WAITANSWER(BUF,IRR,IRR);
05 74107'171000         MOV     3,2             ;
06 74110'006013 CL72:   STOPPROCESS             ; STOPPROCESS(CUR);
07 74111'035012         LDA     3       PROG,2  ; GOTO
08 74112'003401         JMP@            PSTART,3; PSTART.PROG.CUR;
09 74113'020060 CL811:  32*256+48               ; SP,0
10 74114'026460 CL812:  "-*256+48               ; -,0
11 74115'074116'CL813:  .+1                     ; ADDR:
12                      .TXT    .<10>BREAK .    ;
   74116'005102
   74117'051105
   74120'040513
   74121'020000
13 74122'000041 .ZN:    ZN                      ; NOTE: USED IN TEXT
14 74123'074054'.CL701: CL701                   ;
15 74124'000226 .INTGIVE: INTGIVEUP             ; INTERPRETER GIVEUPADDRESS;
16
17              ;WAIT TRANSFER BREAK:
18              CL816:                          ; WTBREAK:
19 74125'035031         LDA     3       SAVE5,2 ; ZONE:=SAVE5.CUR;
20 74126'020126         LDA     0       .10     ;
21 74127'041777         STA     0       -1,3    ; ZONE(-1):=' <10>';
22 74130'126000         ADC     1,1             ; AC1:=-1;
23 74131'167120         ADDZL   3,1             ; TEXT:=ZONE.ZNAME*2-1;
24 74132'020121         LDA     0       .3      ; SEND TO OPER(OUT,TEXT,ADDR);
25 74133'006423         JSR@            .CL400  ;
26 74134'006005         WAITANSWER              ; WAITANSWER(BUF,IRR,IRR);
27 74135'171000         MOV     3,2             ;
28 74136'025026         LDA     1       SAVE2,2 ; GET STATUS;
29 74137'102000         ADC     0,0             ; COUNT:=-1;
30 74140'101440         INCO    0,0             ; REPEAT
31 74141'125103         MOVL    1,1     SNC     ;  COUNT:=COUNT+1; STATUS:=STATUS*
32 74142'000776         JMP             .-2     ; UNTIL CARRY;
33 74143'024422         LDA     1       .50     ; CHAR1:='2';
34 74144'034126         LDA     3       .10     ;
35 74145'162423         SUBZ    3,0     SNC     ; IF COUNT>=10 THEN
36 74146'163001         ADD     3,0     SKP     ; BEGIN  COUNT:=COUNT-10;
37 74147'125400         INC     1,1             ;  CHAR:='3'
38 74150'034407         LDA     3       CL814   ; END;
39 74151'045403         STA     1       +3,3    ; TEXT(7:8):= ' ' CON CHAR;
40 74152'024135         LDA     1       .48     ;
41 74153'107300         ADDS    0,1             ; TEXT(9:10):= COUNT+48;
42 74154'045404         STA     1       +4,3    ;
43 74155'000726         JMP             CL71    ; GOTO OPERATORMESS;
44 74156'073523'.CL400: CL400                   ; SEND TO OPERATOR;
45 74157'074160'CL814:  .+1
46                      .TXT    . ERROR <50><48>.
   74160'020105
   74161'051122
   74162'047522
   74163'020062
   74164'030000
47 74165'020062 .50:    32*256+50               ; SP,2
48
49              CL820:
50                      .END
```

0031 INT10

CL0	000407'	3/18	9/16						
CL00	000403'	3/11	9/12						
CL1	000410'	3/19	7/17	9/18					
CL100	073642'	8/05	23/07	24/30					
CL101	073652'	8/06	23/09	23/16	24/04	24/18	25/21		
CL102	073661'	8/07	23/24	24/13					
CL103	073672'	8/08	23/34						
CL104	073703'	8/09	23/44						
CL105	073740'	8/10	24/25						
CL106	073753'	8/11	24/37						
CL107	074003'	8/12	25/22						
CL11	000322'	8/01	9/32	12/30	16/14				
CL110	073640'	8/14	23/05						
CL111	073650'	8/15	23/14						
CL112	073657'	8/16	23/22						
CL113	073670'	8/17	23/32						
CL114	073701'	8/18	23/42						
CL115	073736'	8/19	24/23						
CL116	073751'	8/20	24/35						
CL117	073761'	8/21	25/04						
CL120	073635'	8/23	23/02						
CL121	073645'	8/24	23/11						
CL122	073654'	8/25	23/19						
CL123	073665'	8/26	23/29						
CL124	073676'	8/27	23/39						
CL125	073733'	8/28	24/20						
CL126	073746'	8/29	24/32						
CL127	073756'	8/30	25/01						
CL132	073663'	23/26	23/54	24/39					
CL137	073763'	25/06	25/23						
CL14	000417'	9/30	9/32						
CL147	073772'	25/10	25/13						
CL15	177703	9/31	9/32						
CL157	073776'	25/14	25/17						
CL20	000462'	11/06	11/15	11/24	11/33	13/02	13/13	13/16	13/20
		13/23	13/26						
CL200	073227'	12/05	13/02						
CL201	073243'	12/06	12/19	12/22	13/16				
CL202	000507'	11/33	12/12						
CL203	000500'	11/24	12/08						
CL204	073247'	12/15	12/17	12/21	13/20	16/10			
CL205	000471'	11/15	12/09	12/20	16/05	16/09			
CL206	073255'	12/07	12/10	12/11	12/13	12/14	12/16	12/23	13/26
		16/08							
CL207	073240'	12/18	13/13						
CL208	073252'	13/23	16/06	16/07					
CL209	073256'	11/21	13/19	13/22	13/34				
CL21	000524'	12/04	12/30						
CL212	000516'	11/40	13/10						
CL22	073352'	16/04	16/14						
CL301	073262'	8/37	14/04						
CL302	073276'	8/38	14/32						
CL303	073460'	8/55	18/51						
CL304	073425'	8/40	18/08						
CL305	073420'	8/41	17/51						
CL306	073444'	8/47	18/30						
CL307	073452'	8/51	18/38						
CL308	073465'	8/36	19/12						
CL309	073501'	8/35	19/29						

CL310	073537'	8/44	20/23				
CL311	073544'	8/45	20/31				
CL312	073614'	8/46	22/02				
CL313	073554'	8/50	20/47				
CL314	073626'	8/48	22/15				
CL315	073606'	8/52	21/33				
CL316	073410'	8/53	17/40				
CL320	073267'	8/49	14/18				
CL321	073306'	14/32	15/04	15/30	17/09	18/09	
CL322	073360'	8/42	17/08				
CL323	073371'	17/18	17/28				
CL324	073323'	8/43	15/30				
CL325	073331'	15/36	15/44				
CL326	073342'	15/37	15/45	18/12			
CL327	073345'	15/36	15/48	18/22			
CL329	073404'	17/21	17/29				
CL333	073314'	14/21	14/23	15/16	17/44		
CL34	073430'	18/12	18/23				
CL360	073476'	8/54	18/35	19/15	19/22	19/30	19/44
CL398	073516'	19/38	19/42				
CL399	073510'	19/32	19/36				
CL4	000423'	3/20	10/14				
CL400	073523'	20/09	20/26	20/37	21/18	29/44	
CL401	073533'	20/13	20/17				
CL402	073562'	20/52	21/08				
CL403	073563'	21/09	21/30				
CL404	073576'	21/14	21/21				
CL40A	073534'	20/11	20/18				
CL40B	073535'	20/19					
CL40C	073536'	20/12	20/20				
CL5	000443'	3/21	10/43	11/16	11/18		
CL51	000452'	10/46	10/50				
CL52	000457'	10/44	10/55				
CL6	074005'	3/13	26/21				
CL7	074017'	3/15	8/03	27/08			
CL70	074030'	27/17	27/45				
CL701	074054'	27/37	29/14				
CL71	074103'	29/01	29/43				
CL710	074065'	27/20	27/46				
CL715	074061'	27/25	27/42				
CL72	074110'	27/09	29/06				
CL811	074113'	27/52	29/09				
CL812	074114'	27/57	29/10				
CL813	074115'	27/51	29/11				
CL814	074157'	29/38	29/45				
CL816	074125'	27/50	29/18				
CL820	074166'	29/49					
CL901	073716'	8/33	24/01				
CL902	073721'	8/34	24/06				
CL903	073730'	8/32	24/16				
IC10	000321'	6/50	7/03	7/54			
IC35	000320'	6/28	6/46	7/53			
IC44	000317'	6/24	6/39	7/52			
IC52	000315'	6/08	7/50				
IC57	000316'	6/03	6/35	7/51			
ICMAS	000302'	5/04	7/38				
IT0	000112'	5/03	9/34				
IT05	000303'	5/22	6/16	7/09	7/30	7/40	
IT1	000157'	5/17	6/03				

IT10	000306'	7/21	7/43						
IT11	000307'	7/20	7/25	7/44					
IT12	000172'	6/14	6/34	6/45	6/56				
IT[illegible]	000176'	6/18	6/22						
IT15	000252'	5/40	6/23	7/09					
IT16	000262'	5/23	5/25	5/27	5/29	5/31	5/33	5/35	5/37
		5/39	6/20	7/19	7/31	7/33			
IT17	000257'	5/06	7/06	7/15					
IT18	000272'	5/18	6/11	6/31	6/42	6/53	7/07	7/28	
IT19	000310'	7/28	7/35	7/45					
IT2	000204'	6/05	6/10	6/24					
IT3	000217'	6/27	6/30	6/35					
IT4	000232'	6/38	6/41	6/46					
IT5	000245'	6/49	6/52	7/03					
IT50	000120	4/23	4/24	4/35	4/37				
IT51	000032'	4/06	4/16	4/17	4/22	4/27	4/32		
IT52	000102"	4/20	4/21	4/34	4/36				
IT53	000000'	4/04	4/06	7/40					
IT59	000111'	4/17	4/38						
IT8	000304'	5/21	7/24	7/41					
IT9	000305'	7/19	7/26	7/42					
IT[illegible]	000111'	4/12	4/39						
IT[illegible]U	000314'	6/13	6/21	6/33	6/44	6/55	7/49		
ITMO	000311'	5/10	7/32	7/46					
ITOP	000312'	5/12	7/29	7/47					
ITPC	000313'	6/17	6/18	6/19	7/48				
TESTO	000000U	4/01	9/19	9/22	9/33	9/36			
.50	074165'	29/33	29/47						
.CL0	000234	3/18	13/37	14/38	20/28	20/39	20/53	22/06	26/30
.CL00	000225	3/09	3/10						
.CL1	000235	3/19	11/43	14/09	15/50	17/32	17/47	17/56	18/21
		18/43	19/24	19/46	20/35	21/38	22/12	22/22	23/17
		23/27	23/37	23/45					
.CL4	000236	3/20	13/14	13/24	14/05	15/05	15/07	15/17	15/31
		17/10	18/40	20/24	20/32	20/50	21/33	24/01	24/06
.CL40	074156'	29/03	29/25	29/44					
.CL5	000237	3/21	13/17	13/21	14/07	14/33	15/19	15/33	17/40
		18/10	18/38	20/48	24/16				
.CL70	074123'	27/29	29/14						
.INTG	074124'	27/37	29/15						
.IT0	000421'	9/20	9/34						
.S[illegible]	073305'	14/35	14/39						
.ZN	074122'	27/14	29/13						