

RCSL: 43-GL67
AUTHOR: JBP
EDITED: 74.02.20

INT00

; KEYWORD: MUS, MUSIL, INTERPRETER, LISTING.
; ABSTRACT: INTERPRETER FOR MUSIL CODE.
; ASCII PAPER TAPE RCSL: 43-GL68
; REL. BINARY PAPER TAPE RCSL: 43-GL69.

.TITL INT00

.RDX 10

.TXTM 1

; RC 73.04.18 INTERPRETER

PAGE 01

; PROCESSDESCRIPTOR OPTIONAL PART

```
; SAVE          ; OCTAL
; SAVE1=        0          ; 25
  SAVE2=        SAVE1+1   ; 26
  SAVE3=        SAVE2+1   ; 27
  SAVE4=        SAVE3+1   ; 30
  SAVE5=        SAVE4+1   ; 31
  R=            SAVE5+1   ; 32
  PC=           R+1       ; 33
  OP=           PC+1      ; 34
  .OPER=        OP+1      ; 35
  ZN=           .OPER+4   ; 41
```

; INITIALISATION

.NREL

```
.CL00= INTPRETE-GOS      ; DEFINE
.LOC .CL00              ; INTERPRETE ENTRY
CL00                    ;
.LOC .+1                ;
CL6                      ; INTGIVEUP
.LOC .+1                ;
CL7                      ; INTBREAK
.LOC MZSTART
```

.CL0: CL0
.CL1: CL1
.CL4: CL4
.CL5: CL5

```
CL11=. ; BASE OF TABLE
CL7    ; 0    ; STOP
CL100  ; 1    ; ANDD
CL101  ; 2    ; LOADD
CL102  ; 3    ; +D
CL103  ; 4    ; -D
CL104  ; 5    ; SHIFTD
CL105  ; 6    ; EXTRACTD
CL106  ; 7    ; *D
CL107  ; 8    ; /D

CL110  ; 9    ; ANDC
CL111  ; 10   ; LOADC
CL112  ; 11   ; +C
CL113  ; 12   ; -C
CL114  ; 13   ; SHIFTC
CL115  ; 14   ; EXTRACTC
CL116  ; 15   ; *C
CL117  ; 16   ; /C
```

```
;
CL120 ; 17 ; AND
CL121 ; 18 ; LOAD
CL122 ; 19 ; +
CL123 ; 20 ; -
CL124 ; 21 ; SHIFT
CL125 ; 22 ; EXTRACT
CL126 ; 23 ; *
CL127 ; 24 ; /

CL903 ; 25 ; LOAD NEGATIVE
CL901 ; 26 ; LOAD BYTE
CL902 ; 27 ; LOAD BYTEWORD
CL309 ; 28 ; JUMP
CL308 ; 29 ; LINK
CL301 ; 30 ; MOVEWORD
CL302 ; 31 ; MOVESTRING
0 ; 32 ; COMPAREWORD
CL304 ; 33 ; COMPARESTRING
CL305 ; 34 ; STORE REGISTER
CL322 ; 35 ; TRANSLATE
CL324 ; 36 ; CONVERT
CL310 ; 37 ; OPMESS
CL311 ; 38 ; OPIN
CL312 ; 39 ; OPWAIT
CL306 ; 40 ; CALL
CL314 ; 41 ; OPTEST
CL320 ; 42 ; MOVE
CL313 ; 43 ; OPSTATUS
CL307 ; 44 ; BINDEC
CL315 ; 45 ; DECBIN
CL316 ; 46 ; CHAR
CL360 ; 47 ; GOTO
CL303 ; 48 ; GO CODE
```

```
; END OF TABLE
```

```
.NREL
```

;

```
CL00:  LDA      2      CUR      ; ENTRY TO INTERPRETER:
        STA      3      PC,2    ;
        SUB      0,0    ;
        STA      0      OP,2    ; OP,CUR:=0;
CL0:   LDA      2      CUR      ; CENTRAL LOOP;
        ;
CL1:   LDA*     3      PC,2    ; INST:=W(PC,CUR);
        ISZ      PC,2    ; PC,CUR:=PC,CUR+1;
        LDA      0      .255   ;
        AND      3,0    ; MODIF:=INST(8:15);
        LDA      1      .CL11  ;
        ADDZ    1,3     SZC     ; IF DISPACEMENT+OP>256 THEN
        JMP     CL20    ; GOTO ZONEPROC;
        SUBS    0,3     ; OP:=INST(0:7);
        JMP*   +0,3    ; GOTO OP+DISPACEMENT;
```

```
.CL11: CL11*256
```

```
; EXIT TO ACTUAL SUBINTERPRETER WITH
; AC0=MODIFICATION BITS
; AC2=CUR
```

PROCEDURE TAKEADDRESS(MODIF, ADDRESS);

```

; CALL RETURN
; AC0 MODIFBITS MODIFBITS SHIFT(-2)
; AC1 ADDRESS
; AC2 CUR CUR
; AC3 LINK DESTROYED
;
CL4: LDA# 1 PC,2 ; ADDRESS:=WORD(PC,CUR);
     ISZ PC,2 ; INCR(PC,CUR);
     MOVZR 0,0 ; MODIFBITS:= MODIFBITS/4;
     MOVZR 0,0 SMC ; IF BITS<>1X THEN
     JMP +0,3 ; RETURN;
     STA 3 SAVE1,2 ;
     LDA 3 .255 ;
     AND 1,3 ; FIELD:=ADDRESS(8:15);
     SUBS 3,1 ; ADDRESS:=ZN,CUR
     ADD 1,2 ; (ADDRESS(0:7));
     LDA 2 ZN,2 ; ADDRESS:=(ADDRESS,ZFIRST)
     LDA 1 ZFIRST,2; +FIELD;
     ADD 3,1 ;
     LDA 2 CUR ;
     JMP# SAVE1,2 ; RETURN;

```

PROCEDURE TAKEVALUE(MODIF, VALUE)

```

; CALL RETURN
; AC0 MODIF MODIF SHIFT (-2)
; AC1 VALUE
; AC2 CUR CUR
; AC3 LINK DESTROYED
;
CL5: MOVZR 0,0 SZC ;
     JMP CL52 ;
     MOVZR 0,0 SZC ; CASE MODIFBITS(14:15) OF
     JMP CL51 ;
     LDA# 1 PC,2 ; 0: VALUE:=INST(PC,CUR);
     ISZ PC,2 ; INCR(PC,CUR);
     JMP +0,3 ; RETURN;
CL51: LDA 2 PC,2 ; 2: VALUE:=WORD(INST(PC,CUR));
     LDA# 1 +0,2 ;
     LDA 2 CUR ;
     ISZ PC,2 ;
     JMP +0,3 ;
CL52: LDA 1 R,2 ; 3: VALUE:=R,CUR;
     MOVZR 0,0 ;
     JMP +0,3 ; RETURN;

```

ZONEPROCESSES:

```

CL20:  SUBS      0,3      ; ZONEPROC:
      MOVZR     0,0      ;
      MOVZR     0,0      ; MODIF:=MODIF SHIFT -4;
      STA      3        SAVE4,2 ;
      LDA#     1        PC,2   ;
      ISZ      1        PC,2   ;
      STA      1        SAVE3,2 ; ZONE.CUR:=ZN;
      LDA      1        .CL20  ;
      ADD      1,3      ;
      JMP#     +0,3      ; GOTO SUBINTERPRETER;
      ;

CL12=256-CL11
.CL20:  .+1-CL11+128    ; TABSTART
      CL200      ; GETREC: AC0=ADDR
      CL201      ; PUTREC: AC0=VAL
      CL206      ; WAITTRANSFER
      CL203      ; REPEATSHARE
      CL205      ; TRANSFER: AC0=VAL, AC1=VAL
      CL206      ; INBLOCK
      CL206      ; OUTBLOCK
      CL202      ; INCHAR: AC1=ADDR
      CL206      ; RACKSPACE
      CL206      ; OUTSPACE
      CL204      ; OUTCHAR: AC1=VAL
      CL206      ; OUTNL
      CL204      ; OUTEND: AC1=VAL
      CL207      ; OUTTEXT: AC0=ADDR
      CL201      ; OUTOCTAL: AC0=VAL
      CL205      ; SETPOS: AC0=VAL, AC1=VAL
      CL204      ; CLOSE: AC1=VAL
      CL201      ; OPEN: AC0=VAL
      CL206      ; WAITZONE
      ;

CL205:  JSR      CL5      ; TAKEVALUE(MODIF,FILE);
      STA      1        SAVE5,2 ; SAVE(FILE);
      JSR#     .CL5     ; TAKEVALUE (MODIF,BLOCK);
      MOV      1,0      ;
      LDA      1        SAVE5,2 ;
      JMP      CL206    ; GOTO EXECUTE;
    
```

```

;
CL203: LDA      2      SAVE3,2 ;
        LDA      3      CUR      ;
        LDA      0      ZBUFFE,2; PC,CUR:=SAVEPC,ZONE;
        STA      0      PC,3      ;
        LDA      3      ZSIZE,2 ; LINK:=SAVELINK,ZONE;
        .REPEATSHARE ;
;
CL202: LDA#     3      PC,2      ;
        ISZ      PC,2      ;
        LDA      1      +0,3     ;
        LDA      3      SAVE4,2 ;
        LDA      2      SAVE3,2 ;
        JSR#     +CL12,3 ;
CL212: LDA      2      CUR      ;
        LDA      3      PC,2      ;
        STA#     1      -1,3     ;
        JMP#     .CL1      ; RETURN;
;
CL200: LDA#     3      PC,2      ;
        ISZ      PC,2      ;
        LDA      0      +0,3     ;
        LDA      3      SAVE4,2 ;
        LDA      2      SAVE3,2 ;
        JSR#     +CL12,3 ;
        MOV      0,1      ;
        JMP      CL212      ;
;
CL207: JSR#     .CL4      ; TAKEADDRESS(MODIF,AC1);
        JMP      .+2      ;
CL201: JSR#     .CL5      ; TAKEVALUE(MODIF,AC1);
        MOV      1,0      ; SKP ;
CL204: JSR#     .CL5      ; TAKEVALUE(MODIF,AC0);
CL206: LDA      3      SAVE4,2 ; EXECUTE;
        LDA      2      SAVE3,2 ;
        JSR#     +CL12,3 ;
        JMP#     .CL0      ;

```

```

;
CL901: JSR#           .CL4      ; BYTE:
      ; TAKEADDRESS(MODIF, ADDRESS);
      GETBYTE
      JMP           CL904      ; GETBYTE(ADDRESS, CHAR);

CL902: JSR#           .CL4      ; WORD:
      ; TAKEADDRESS(MODIF, ADDRESS);
      GETBYTE
      MOV5          0,0        ; GETBYTE(ADDRESS, BYTE);
      STA          0         R,2  ; R.CUR:=BYTE*256;
      INC          1,1        ;
      GETBYTE
      LDA          1         R,2  ; GETBYTE(ADDRESS+1, BYTE);
      ADD          1,0        ; R.CUR:=R.CUR+BYTE;
      JMP           CL904      ;

CL903: JSR#           .CL5      ; TAKEVALUE(MODIF, VAL);
      NEG          1,0        ;
CL904: STA          0         R,2  ; R.CUR:= -VALUE;
      JMP#         .CL1      ;

CL301:
      LDA#         1         PC,2  ; MOVEWORD:
      ISZ
      STA          1         PC,2  ; ADDRESS:= INST(PC.CUR);
      MOVZR        0,0        ;
      MOVZR        0,0        ;
      JSR#         .CL5      ; TAKEVALUE(MODIF, R.CUR);
      STA#         1         SAVE4,2 ; WORD(ADDRESS):=VALUE;
      JMP#         .CL1      ;

```


CL320:

```

LDA# 0 PC,2 ;
ISZ PC,2 ;
JSR CL333 ; TAKEADDRESS AND MODIFY;
STA 1 SAVE4,2 ; SAVE4.CUR:= VAL;
JSR CL333 ; TAKEADDRESS AND MODIFY;
STA 1 SAVE3,2 ; SAVE3.CUR:=VAL;
JMP .+2 ;

```

CL302:

```

JSR CL321 ; MOVESTRING;
JSR# .CL5 ; TAKEVALUE(SAVE,VALUE);
STA 1 SAVE2,2 ; SAVE2.CUR+= VALUE;
LDA 1 .S2 ;
ADD 1,2 ; PARAMADDR:= .SAVE2+(CUR);
MOVE ; MOVE(PARAMADDR);
JMP# .CL0 ; RETURN;

```

.S2:

SAVE2 ;

CL321:

```

STA 3 SAVE5,2 ; TAKEADDRESSES;
JSR# .CL4 ; TAKEADDRESS(MODIF,SAVE4.CUR);
STA 1 SAVE3,2 ;
JSR# .CL4 ; TAKEADDRESS(MODIF,SAVE3.CUR);
STA 1 SAVE4,2 ;
JMP# .CL4 ; TAKEADDRESS(MODIF,SAVE4.CUR);

```

CL333:

```

STA 3 SAVE5,2 ; TAKEADDRESS AND MODIFY;
JSR# .CL4 ; TAKEADDRESS(MODIF,ADDRESS);
STA 1 SAVE3,2 ;
JSR# .CL5 ; TAKEVALUE(MODIF,VALUE);
LDA 3 SAVE3,2 ;
ADD 3,1 ; VAL:=ADDRESS+VALUE;
JMP# .CL4 ;

```

```

;
CL324: JSR          CL321 ; CONVERT:
        JSR@       ,CL4  ; TAKEADDRESSES(INSTRING,OUTSTRING);
        STA        1     SAVE1,2 ; TAKEADDRESS(TABLE);
        JSR@       .CL5  ; TAKEVALUE(COUNT);
        INC        1,1   ;
        STA        1     SAVE2,2 ;

CL325: JSR          CL327 ; WHILE COUNT>0 DO
        JSR          CL326 ; BEGIN
        LDA        1     SAVE1,2 ; GETBYTE(INCR(INSTRING),BYTE);
        ADD        0,1   ; GETBYTE(TABLE+BYTE,BYTE);
        GETBYTE    ;
        LDA        1     SAVE4,2 ; PUTBYTE(OUTSTRING,BYTE);
        PUTBYTE    ;
        ISZ        SAVE4,2 ; INCR(OUTSTRING)
        JMP        CL325 ; END;

CL326: LDA        1     SAVE3,2 ;
        ISZ        SAVE3,2 ;
        .GETBYTE    ;

CL327: DSZ        SAVE2,2 ;
        JMP        +0,3  ;
        JMP@       .CL1  ;

CL322: JSR          CL321 ; TRANSLATE:
        JSR@       ,CL4  ; TAKEADDRESSES(INSTRING,OUTSTRING);
        MOVZR     1,1   ; TAKEADDRESS(TABLE);
        STA        1     SAVE,2  ;
        LDA        1     SAVE3,2 ;
        GETBYTE    ;
        MOVS      0,0   ;
        STA        0     SAVE3,2 ;
        LDA        3     SAVE,2  ;
CL323: LDA        0     +0,3  ;
        LDA        1     +1,3  ;
        MOV        0,0   SNR    ;
        JMP        .+8    ;
        INC        3,3   ;
        LDA        1     SAVE3,2 ;
        SUB        1,0   ;
        LDA        1     .255  ;
        AND        0,1   ;
        SUB        1,0   SZR    ;
        JMP        CL323 ;
        MOV        1,0   ;
        LDA        1     SAVE4,2 ;
        PUTBYTE    ;
        JMP@       .CL1  ;

CL316: JSR@       .CL5  ; INSERT:
        LDA        3     .255  ;
        AND        1,3   ;
        STA        3     SAVE2,2 ;
        JSR        CL333 ; TAKEADDRESS AND MODIFY;
        LDA        0     SAVE2,2 ;
        PUTBYTE    ;
        JMP@       .CL1  ;
    
```

```

;
CL303:  MOV      2,3      ; GO CODE;
        SUB      0,3      ; ADDR:=CUR-BITS;
        LDA#     0        PC,2 ; BITS:=W(PC.CUR);
        ISZ     PC,2      ; INCR(PC.CUR);
        JMP#    +0,3      ; GOTO W(ADDR);

CL304:                                     ; COMPARESTRING;
        JSR     CL321     ; TAKEADDRESSES(STRING1,STRING2);
        JSR#    .CLS     ; TAKEVALUE(MODIF,VALUE);
        STA     1        SAVE2,2 ; SAVE2.CUR:= VALUE;
CL34:   JSR     CL326     ; REPEAT
        STA     0        SAVE,2  ; GETBYTE(SAVE3,SAVE);
        LDA     1        SAVE4,2 ;
        GETBYTE                                     ; GETBYTE(SAVE4,R);
        ISZ     SAVE4,2  ; INCR(SAVE4.CUR);
        LDA     1        SAVE,2  ;
        SUB     0,1      ; IF R.CUR <> 0 THEN
        STA     1        R,2     ;
        MOV     1,1      SZR     ;
        JMP#    .CL1     ; RETURN;
        JSR     CL327     ; UNTIL COUNT=0;
        JMP     CL34     ;
                                     ; RETURN;
CL305:                                     ; STORE;
        LDA#    3        PC,2   ;
        ISZ     PC,2     ;
        LDA     0        R,2    ; TAKEADDRESS(MODIF,ADDRESS);
        STA     0        +0,3   ; WORD(ADDRESS):= R.CUR;
        JMP#    .CL1     ; RETURN;
    
```

```

;
CL306: LDA# 3 PC,2 ; LINK:=W(PC,CUR);
        ISZ PC,2 ; INCR(PC,CUR);
        LDA 0 PC,2 ;
        STA 0 +0,3 ; LINKWORD.LINK:=PC.CUR;
        INC 3,3 ; PC.CUR:=LINK+1;
        JMP CL360+1 ;
;
CL307: JSR# .CL5 ;
        STA 1 SAVE4,2 ;
        JSR# .CL4 ;
        LDA 0 SAVE4,2 ;
        BINDEC ;
        JMP# .CL1 ;
;
CL308: LDA# 3 PC,2 ; LINK;
        LDA 3 +0,3 ; POINT:=W(PC,CUR);
        MOVL# 3,3 SMC ; IF POINT >=0 THEN
        JMP CL360+1 ; EXECUTE JUMP;
        NEG 3,2 ; ZONE:=-POINT;
        LDA 3 CUR ;
        LDA 0 ZBUFFF,2;
        STA 0 PC,3 ; PC.CUR:=SAVEPC.ZONE;
        JMP# ZSIZE,2 ; GOTO SAVEDLINK.ZONE;
; EXECUTE JUMP;
CL360: LDA# 3 PC,2 ;
        STA 3 PC,2 ;
        JMP# .CL1 ;
CL309: JSR CL399 ; INDEX:=0;
        1B0+1B9+1B10+1B11 ; COND(0) "=",>=",<=";
        1B0+1B11+1B12+1B13 ; COND(1) "<=",<>,"<"
        1B0+1B10+1B12+1B14 ; COND(2) ">=",<>,">";
CL399: LDA 1 R,2 ;
        MOV 1,1 SNR ; IF R.CUR<>0 THEN
        JMP .+4 ; BEGIN
        INC 3,3 ; INDEX:=1;
        MOVL 1,1 SMC ; IF R.CUR>0 THEN
        INC 3,3 ; INDEX:=2;
        LDA 1 +0,3 ; END; C:=COND(INDEX);
        MOV 0,3 ;
        LDA 0 BIT,3 ; M:=BIT(MODIF);
        AND 0,1 SZR ; IF M AND C THEN
        JMP CL360 ; GOTO EXECUTE JUMP;
        ISZ PC,2 ; SKIP;
        JMP# .CL1 ;

```

; PROCEDURE SENDTO OPERATOR(COMMAND,ADDRESS,COUNT,BUF);

```

; CALL:          RETURN:
; AC0  COMMAND   IRRELEVANT
; AC1  ADDRESS   -
; AC2  CUR       BUF
; AC3  LINK      CUR
; SAVE1 COUNT
    
```

```

CL400:  STA      3      SAVE,2 ; SEND TO OPERATOR:
        INTDS                    ; DIRTY WORK!
        STA      0      CL40A ;
        STA      1      CL40C ;
        LDA      1      CL401 ;
        LDA      2      .OPER,2; SENDMESSAGE(MESS0,CUR,OPERATOR);
        SENDMESSAGE ;
        JMP#      SAVE,3 ;
    
```

```

CL401:  .+1      ; MESSAGE:
CL40A:  0        ; OPERATION
CL40B:  80       ; COUNT
CL40C:  0        ; ADDRESS
    
```

```

CL310: ; OPMESS:
        JSR#      .CL4 ; TAKEADDRESS(MODIF,ADDRESS);
        LDA      0      .3 ;
        JSR      CL400 ;
        WAITANSWER ;
        JMP#      .CL0 ; RETURN;
    
```

```

CL311: ; OPIN:
        JSR#      .CL4 ; TAKEADDRESS(MODIF,ADDRESS);
        LDA      0      OP,2 ;
        MOV      0,0    SZR ; IF OP,CUR <> 0 THEN
        JMP#      .CL1 ; RETURN;
        LDA      0      .1 ;
        JSR      CL400 ;
        STA      2      OP,3 ;
        JMP#      .CL0 ; RETURN;
    
```

```

CL313: ; OPSTATUS:
        JSR#      .CL5 ; TAKEVALUE(MODIF,VALUE);
        STA      1      SAVE2,2 ; SAVE2.CUR:= VALUE;
        JSR#      .CL4 ; TAKEADDRESS(SAVE3,ADDRESS);
        STA      1      SAVE3,2 ;
        JSR      CL402 ; OUTMESSAGE;
        JMP#      .CL0 ; RETURN;
    
```

; PROCEDURE OUTMESSAGE;

```

;
CL402: STA      3          SAVE4,2 ;
CL403:          LDA      0          SAVE2,2 ; REP:
          MOV      0,0        SNR      ; WORD:=SAVE2.CUR;
          JMP@     SAVE4,2      ; IF WORD=0 THEN
          MOVZL   0,0        SNC      ; RETURN;
          JMP     CL404        ; IF WORD(0) THEN
          STA      0          SAVE2,2 ; BEGIN
          LDA      1          SAVE3,2 ; ADDRESS:=SAVE3.CUR;
          LDA      0          .3      ; COMMAND:=3;
          JSR     CL400        ; SEND TO OPERATOR;
          WAITANSWER          ; WAITANSWER(IRR,IRR,BUF);
          MOV      3,2        SKP      ; END
CL404: STA      0          SAVE2,2 ; ELSE SAVE2.CUR:=WORD SHIFT 1;
          LDA      1          SAVE3,2 ; SKIP:
          GETBYTE          ;
          INC      1,1        ;
          MOV      0,0        SZR      ;
          JMP     .-3         ;
          STA      1          SAVE3,2 ;
          JMP     CL403        ;

CL315: JSR@     .CL4         ; DECBIN:
          DECBIN          ;
          LDA@     3          PC,2     ;
          ISZ     PC,2       ;
          STA      1          +0,3    ;
          JMP@     .CL1       ;

CL312:          ; OPWAIT:
          ISZ     PC,2       ;
          LDA      2          OP,2     ;
          MOV      2,2        SNR      ;
          JMP@     .CL0       ;
          WAITANSWER          ;
          MOV      3,2        ;
          LDA      3          PC,2     ;
          STA@    1          =1,3     ;
          STA      0          OP,2     ; OP.CUR:=0;
          JMP@     .CL1       ;

CL314:          ; OPTEST:
          LDA      3          OP,2     ;
          LDA      3          RECEIV,3;
          SUB     0,0        ;
          MOVZL   3,3        SZC      ;
          CGM     0,0        ;
          STA      0          R,2      ;
          JMP@     .CL1       ;

```

```

;
CL120: LDA# 3 PC,2 ;
      LDA 0 +0,3 ;
      JMP .+2 ;
CL110: LDA# 0 PC,2 ;
      ISZ PC,2 ;
CL100: LDA 1 R,2 ;
      AND 1,0 ;
      JMP CL101 ;
CL121: LDA# 3 PC,2 ;
      LDA 0 +0,3 ;
      JMP .+2 ;
CL111: LDA# 0 PC,2 ;
      ISZ PC,2 ;
CL101: STA 0 R,2 ; LOAD
      JMP# .CL1 ;

CL122: LDA# 3 PC,2 ;
      LDA 0 +0,3 ;
      JMP .+2 ;
CL112: LDA# 0 PC,2 ;
      ISZ PC,2 ;
CL102: LDA 1 R,2 ;
      ADD 0,1 ;
CL132: STA 1 R,2 ;
      JMP# .CL1 ;

CL123: LDA# 3 PC,2 ;
      LDA 0 +0,3 ;
      JMP .+2 ;
CL113: LDA# 0 PC,2 ;
      ISZ PC,2 ;
CL103: LDA 1 R,2 ;
      SUB 0,1 ;
      STA 1 R,2 ;
      JMP# .CL1 ;

CL124: LDA# 3 PC,2 ;
      LDA 0 +0,3 ;
      JMP .+2 ;
CL114: LDA# 0 PC,2 ;
      ISZ PC,2 ;
CL104: MOV 0,3 SHR ; IF SHIFTS=0 THEN
      JMP# .CL1 ;
      LDA 1 R,2 ;
      MOVL# 0,0 SNC ; IF SHIFTS>0 THEN
      NEG 3,3 ; COUNT:=-SHIFTS;
      MOVL# 0,0 SNC ; FOR COUNT:=COUNT STEP -1 UNTIL 0 DO
      MOVZL 1,1 SKP ; R:= R SHIFT SIGN(SHIFTS);
      MOVZR 1,1 ;
      INC 3,3 SZR ;
      JMP .-4 ;
      JMP CL132 ;

```

```
CL125: LDA# 3 PC,2 ;  
LDA 0 +0,3 ;  
JMP .+2 ;  
CL115: LDA# 0 PC,2 ;  
ISZ PC,2 ;  
CL105: LDA 3 .16 ;  
SUB 0,3 ;  
LDA 0 BIT,3 ;  
NEGZL 0,0 ;  
LDA 1 R,2 ;  
AND 1,0 ;  
JMP CL103+1 ;  
  
CL126: LDA# 3 PC,2 ;  
LDA 0 +0,3 ;  
JMP .+2 ;  
CL116: LDA# 0 PC,2 ;  
ISZ PC,2 ;  
CL106: LDA 1 R,2 ;  
MULTIPLY ;  
JMP CL132 ;  
  
CL127: LDA# 3 PC,2 ;  
LDA 1 +0,3 ;  
JMP .+2 ;  
CL117: LDA# 1 PC,2 ;  
ISZ PC,2 ;  
CL137: LDA 0 R,2 ;  
DIVIDE ;  
JMP CL101 ;  
CL107: MOV 0,1 ;  
JMP CL137 ;
```


;

; STDGIVEUP

```
CL6:   STA     3      ZSIZE,2 ; SAVEDLINK.ZONE:=LINK;
        LDA     3      SIZE,2  ; PROC:=SIZE.ZONE;
        NEG     2,0    ;
        STA     0      +0,3    ; PROC(0):=-ZONE;
        INC     3,1    ; START:=PROC+1;
        LDA     3      CUR     ;
        LDA     0      PC,3    ;
        STA     0      ZBUFFE,2; SAVEPC.ZONE:=PC.CUR;
        STA     1      PC,3    ; PC.CUR:=START;
        JMP     .CL0    ; GOTO NEXT0;
```

; STDBREAK, STOP

```

CL7:   MOV     0,0      SNR     ; IF ERRORNO.=0 THEN GOTO STOP;
      JMP     CL72     ; "OWN CLEAN!"
      LDA     3        .5     ;
      SUB     0,3      SNR     ; IF ERRNO=5 THEN
      JMP     CL816    ; GOTO WTBREAK;
      LDA     1        CL811   ; CHAR:="+";
      MOVZL#  0,0      SNC     ; IF ERRNO<0 THEN
      JMP     .+3      ; BEGIN
      NEG     0,0      ; ERRNO:=ERRNO*-1;
      LDA     1        CL812   ; CHAR:="-";
      ADD     1,0      ; END;
      STA     0        SAVE5,2 ; TEXT,CHAR:=CHAR,ERRNO;
      LDA     0        CL813+2 ; TEXT:="BREAK ";
      STA     0        SAVE4,2 ;
      LDA     0        CL813+1 ;
      STA     0        SAVE3,2 ;
      LDA     0        CL813+0 ;
      STA     0        SAVE2,2 ;
      LDA     1        CL814   ;
      ADDZL   2,1      ;
      SUB     0,0      ;
      STA     0        R,2     ;
      LDA     0        .3     ;
      JSR#    .CL400    ;
      WAITANSWER      ;
      MOV     3,2      ;
      LDA     0        SAVE5,2 ;
      LDA     1        CL812   ;
      AND     1,0      ;
      SUB     1,0      SNR     ; IF ERRORNO="-" THEN
      JMP     CL710    ; GOTO STOPPROCESS;
CL71:  LDA     3        .ZN    ;
      ADD     2,3      ;
      STA     3        SAVE4,2 ; ZN:=BASE OF ZONES;
      LDA     1        SIZE,2  ;
      ADD     2,1      ;
      SUB     3,1      SNR     ; WHILE COUNT OF ZONES<>0 DO
      JMP     CL710    ; BEGIN
      STA     1        SAVE3,2 ;
CL70:  LDA#    2        SAVE4,2 ;
      SUBZL   1,1      ;
      STA     1        ZMODE,2 ;
      CLOSE      ; CLOSE(ZN);
      LDA     2        CUR     ;
      ISZ     SAVE4,2  ; INCR(ZN);
      DSZ     SAVE3,2  ; DECR(COUNT OF ZONES)
      JMP     CL70     ; END;
CL710: CLEANPROCESS      ;
CL72:  STOPPROCESS      ; STOPPROCESS(CUR);
      LDA     3        PROG,2  ; GOTO
      JMP#    PSTART,3 ; PSTART,PROG,CUR;

```

```

;
CLB11:  "+*256+48          ) +,0
CLB12:  "-*256+48          ) -,0
CLB13:  .TXT      .BREAK . )
        .LOC      .-1      )
CLB14:  SAVE2          )
; PROCEDURE WAIT TRANSFER BREAK(5,STATUS,CUR);

```

```

CLB16:          ) WTBREAK:
        LDA      3      SAVE,2 )
        STA      1      SAVE2,2 ) STATUS:
        LDA      1      CLB18 )
        STA      1      SAVE3,2 )
        MOVZL    3,1    )
        LDA      0      .3      )
        JSR@     .CL400  )
        WAITANSWER )
        MOV      3,2    )
        JSR@     .CL42   )
        JMP      CLB1   ) GOTO STOP;

```

```

.ZN:     ZN
.CL40:   CL400
.CL42:   CL402
CLB18:   .+1*2          )
        .TXT      . B0<0> OFFLINE<0> BUSY<0>
<0><0><0> B6<0>
EOF<0> BLOCK<0> B9<0>
PARITY<0> EDT<0> B12<0>
B13<0> TIMER<0> ?.

```

.END