

```

;CE31    ASM
;*****
;**                **
;
;    CAOS 3.1 reassembliert    **
;    von ML-Soft    **
;    **
;    14.05.95 bis 21.05.95    **
;    **
;*****

;    Speicher}bersicht    **
;    _____
;

;C000    BASIC-ROM

;-----

;E000    RESET und BASIC-Verteiler
;E024    DABR
;E055    Zeichenausgabe WPIX
;E0BF    PADR
;E13A    Punktroutinen PUDE, PUSE
;E189    KBD-Treiber
;E213    Tastaturtabelle KTAB
;E293    Interrupttabelle ISRTAB
;E29F    KBDZ, KBDS, INIEA, INIME, ERAM
;E2CF    SWITCH und MODUL
;E347    JUMP
;E36A    Tape-Treiber
;E527    TON-Ausgabe
;E581    BASIC-Token
;E62B    Sprungtabelle f. BASIC-Token
;E669    BASIC-Extension
;EE00    Zeichenbildtab. Gro~buchst. *)

;-----

;F000    PWRON und Programm-Verteiler
;F109    Kommandointerpreter (MENU)
;F1C6    Ein-/Ausgabe OSTR, INTB, OCHR
;F210    KBD, IRM-Default, INLIN
;F2C0    Hexzahlenverarbeitung
;F357    Systeminit, SIXD, Portinittab.
;F3D5    SAVE
;F43E    BRKT, LARG
;F45B    VERIFY, LOAD, COLOR
;F5D4    Unterprogrammtabelle SUTAB
;F662    LDMA, LDAM, MODIFY
;F6E3    Sprungtabelle f}r CRT-Routinen
;F707    CCTL-Belegungstabelle CCTAB
;F725    CRT-Treiber
;F8BE    Bildschirm-Steuercodes CRTTAB
;F8D0    FADR,
;F8E7    BASIC-I/O-Verteiler
;F956    MBOU, MBIN
;FAFB    DISPLAY
;FB69    CSTBT, KEYLIST, KEY
;FC1A    WININ, WINAK
;FC82    LINE, CIRCLE, SQR, MULT
;FE00    Zeichenbildtab. Kleinbuchst. *)
;-----

;*) nicht in diesem Listing aufgef}hrt

;**    EQU-Liste    **
;

LF    EQU    0AH
CR    EQU    0DH

VRAM    EQU    0B200H    ;VIDEO-RAM
CASSEQU    0B700H    ;Kass.-Puffer
ARGC    EQU    0B780H    ;UP-Nr. (PV2)
ARGN    EQU    0B781H    ;Anz. Argumente
ARG1    EQU    0B782H    ;1. Argument
ARG2    EQU    0B784H    ;2. Argument
ARG3    EQU    0B786H    ;3. Argument
ARG4    EQU    0B788H    ;4. Argument

NUMNX    EQU    0B796H    ;Zeichenanzahl
NUMVX    EQU    0B797H    ;Wert der Zahl
HCADR    EQU    0B799H    ;Hardcopy-Adr.
WINNR    EQU    0B79BH    ;Fensternummer
WINON    EQU    0B79CH    ;Fensteranfang
WINLG    EQU    0B79EH    ;Fensterl{nge
CURSO    EQU    0B7A0H    ;Cursorposition
STBT    EQU    0B7A2H    ;BS-Steuerbyte
COLOR    EQU    0B7A3H    ;Zeichenfarbe
WEND    EQU    0B7A4H    ;PAGE/SCROLL
CCTL0    EQU    0B7A6H    ;Zeichentab. 0
CCTL1    EQU    0B7A8H    ;Zeichentab. 1
CCTL2    EQU    0B7AAH    ;Zeichentab. 2
CCTL3    EQU    0B7ACH    ;Zeichentab. 3
SYSP    EQU    0B7AEH    ;System-Stack

```

SUTAB EQU 0B7B0H ;UP-Tabelle
CTAB EQU 0B7B2H ;Adresse CRTTAB
BJUMP EQU 0B7B4H ;JUMP-Adresse
OUTAB EQU 0B7B9H ;Ausgabe-UP
INTAB EQU 0B7BBH ;Eingabe-UP
UOUT1 EQU 0B7BDH ;USER-out #1
UIN1 EQU 0B7C0H ;USER-in #1
UOUT2 EQU 0B7C3H ;USER-out #2
UIN2 EQU 0B7C6H ;USER-in #2
IOERR EQU 0B7C9H ;-> ?IO ERROR

ZWEND EQU 0B7CFH ;WEND-Merker
FTAST EQU 0B7D1H ;F-Tastenzeiger
HOR EQU 0B7D3H ;hor. Koordin.
VERT EQU 0B7D5H ;vert. Koordin.
FARB EQU 0B7D6H ;Grafikfarbe
MIXIT EQU 0B7D7H ;IX-Bereich
ZEITB EQU 0B7D8H ;Zeichentabelle
VORTN EQU 0B7DAH ;Vortonl{nge
DTADR EQU 0B7DCH ;Datenzeiger
DTNR EQU 0B7DEH ;Datennummer
XSTEP EQU 0B7DFH ;X-Schrittma~
YSTEP EQU 0B7E1H ;Y-Schrittma~
XCNT EQU 0B7E3H ;X-Z{hler
YCNT EQU 0B7E5H ;Y-Z{hler
XDIR EQU 0B7E7H ;X-Richtung
YDIR EQU 0B7E9H ;Y-Richtung

WNDNFN EQU 0B99CH ;Fenstervektoren

;** CAOS 3.1 ROM E **

ORG 0E000H
;
BYE JP RESET ;Tasten-RESET
JP BEXP1 ;BASIC-
JP BEXP2 ;Expansionen
JP BEXP3
DW TOKTAB
BASIO JP BASPV

RESET ;Tasten-RESET
LD A,(IX+7)
AND 0C0H;RESET-Schutz?
JR Z,BYE1
CALL ERAM4 ;RAM4 l|schen!
JP POWER
;
BYE1 LD SP,1C4H
JP PWR2

DABR PUSH DE ;**32**
PUSH AF
CALL TCIF
JR C,IAD2
LD A,(WINON+1)
ADD D ;Cursor-Zeile
LD H,0
LD L,A
ADD HL,HL
ADD HL,HL
ADD HL,HL
PUSH BC
PUSH HL
POP BC
ADD HL,HL
ADD HL,HL
ADD HL,BC
LD A,(WINON)
ADD E ;Cursor-Spalte
ADD L
LD L,A
LD A,0
ADC H
LD H,A
LD BC,VRAM
ADD HL,BC
POP BC
POP AF
AND A
POP DE
RET
IAD2 POP AF
SCF
POP DE
LD HL,VRAM
RET

WPIX;Zeichen auf Grafikbildschirm

```
;sichtbar machen, PE: DE,A
PUSH HL
PUSH DE
PUSH BC
PUSH AF
PUSH AF
CALL FADR;Farbadresse
POP AF
PUSH HL ;wof}r?
CALL PADR
JR C,POP4 ;au~erhalb!
PUSH HL ;Pixeladresse
LD HL,CCTAB
LD C,0B7H ;High(CCTLx)
WPIX1 CP M
INC HL
JR NC,WPIX2
INC HL
INC HL
JR WPIX1
WPIX2 SUB M ;Offset
INC HL
LD L,M
LD H,C ;HL=CCTLx
LD C,M
INC HL
LD H,M
LD L,C ;Zeichentabelle
LD (ZEITB),HL
LD C,A
LD A,(STBT)
LD B,A
BIT 1,B
JR NZ,WPIX3;Farbe aus
LD A,(COLOR)
LD (DE),A ;oberes Byte
SET 5,E
LD (DE),A ;unteres Byte
WPIX3 POP DE ;Pixeladr.
POP HL ;Farbadr.
BIT 0,B
JR NZ,POP4 ;Pixel aus
LD B,0
LD A,3
WPIX4 SLA C
RL B ;*8
DEC A
JR NZ,WPIX4
LD HL,(ZEITB)
ADD HL,BC ;Zeichenadr.
PUSH DE
CALL WPIX5 ;4 Byte schreib.
POP DE
SET 5,E
CALL WPIX5 ;4 Byte schreib.
POP4 POP AF
JP POP3
;
WPIX5 LD BC,80H ;Offset f}r
LD A,4 ;n{chstes Byte
WPIX6 LDI
EX DE,HL
ADD HL,BC
EX DE,HL
INC BC ;BC korrigieren
DEC A
JR NZ,WPIX6
RET

PADR ;Pixel- und Farbadr berechnen
PUSH AF ;**34**
LD A,L ;Spalte
CP 40
JR NC,PADR5;zu gro~
PUSH BC
CP 32
JR NC,PADR6;rechte Seite
LD A,H
RLCA
RLCA
RLCA
AND 60H
OR L
BIT 0,H
JR Z,PADR1
SET 7,A
PADR1 LD C,A
LD A,H
RRA
RRA
RRA
AND 1EH
BIT 1,H
JR Z,PADR2
```

```

    SET 0,A
PADR2  OR 80H ;Pixel-RAM I
    LD B,A
    RRA
    RRA
    AND 7
    OR 0A8H;Color-RAM I
    LD D,A
    LD A,C
    AND 7FH
    BIT 4,H
    JR Z,PADR3
    SET 7,A
PADR3  LD E,A
PADR4  LD L,C
    LD H,B
    POP BC
    POP AF
    OR A ;CY=0
    RET
;
PADR5  POP AF
    SCF ;außerhalb
    RET
;
PADR6  AND 7
    LD L,A
    LD A,H
    RLCA
    RLCA
    RLCA
    AND 60H
    OR L
    LD L,A
    LD A,H
    RRA
    AND 18H
    BIT 0,H
    JR Z,PADR7
    SET 7,A
PADR7  OR L
    LD C,A
    LD A,H
    RLCA
    RLCA
    RLCA
    AND 6
    OR 0A0H;Pixel-RAM II
    BIT 1,H
    JR Z,PADR8
    SET 0,A
PADR8  LD B,A
    LD D,0B0H ;Color-RAM II
    BIT 2,A
    JR Z,PADR9
    INC D
PADR9  LD A,C
    AND 7FH
    BIT 1,B
    JR Z,PADR10
    SET 7,A
PADR10 LD E,A
    JR PADR4

PUDE;Punkt I|schen ;**2F**
    SCF
    JR PU1
PUSE;Punkt setzen ;**30**
    AND A
PU1  PUSH HL
    PUSH DE
    PUSH BC
    LD HL,(HOR)
    LD BC,(VERT)
    PUSH AF
    XOR A
    LD B,3
PU2  SRL H
    RR L
    RRA
    DJNZ PU2
    LD E,A
    LD A,C
    CPL
    LD H,A
    LD C,E
    CALL PADR
    JR C,PU5
    SCF
    LD A,C
    LD C,0
PU3  RR C
    SUB 20H
    JR NC,PU3
```

```
POP AF
LD A,C
JR C,PU6
OR M
LD M,A
LD A,(FARB)
AND 0F8H
LD B,A
LD A,(DE)
AND 7
OR B
LD (DE),A
PU4 JP POP3
PU5 POP AF
SCF
JR PU4
PU6 PUSH AF
LD B,M
CPL
AND M
LD M,A
POP AF
AND B
LD A,(DE)
JR PU4

;KBD-Treiber

ISR3C ;ISR CTC Kanal 3 (Tastatur)
EI
PUSH AF
LD A,23H ;DI,ZG256,Res
OUT 8FH ;CTC K3
SET 3,(IX+8);Timeout
LD (IX+13),0 ;Zeichen tot
JR TST4

ISRPB ;ISR PIO Kanal B (Tastatur)
EI
PUSH AF
IN A,8FH ;gemessene
PUSH AF ;Zeit retten
LD A,0A7H ;EI,ZG256,Res
OUT 8FH
LD A,8FH ;ZK
OUT 8FH
POP AF
BIT 3,(IX+8);Timeout?
RES 3,(IX+8);r}cksetzen
JR NZ,TST4
CP 101
JR NC,TST5 ;0-Bit
CP 68
JR NC,TST6 ;1-Bit
PUSH HL
PUSH DE
SRL (IX+12)
LD A,(IX+8)
AND 80H ;CAPS?
RLCA
XOR (IX+12) ;7bit-Scancode
LD H,0
LD E,(IX+14)
LD D,(IX+15)
LD L,A
ADD HL,DE ;Pos. in KTAB
LD A,M ;ASCII holen
POP DE
POP HL
CP (IX+13) ;=letzter Code?
JR Z,TST2
LD (IX+13),A ;eintragen
RES 4,(IX+8);Repeat r}cks.
TST1 SET 0,(IX+8);g}ltig machen
LD (IX+10),0 ;neuer Code
JR TST4
;
TST2 INC (IX+10) ;Zeit abwarten
BIT 4,(IX+8);Fast Repeat?
JR NZ,TST3
BIT 4,(IX+10) ;16 erreicht?
JR Z,TST4
SET 4,(IX+8);Repeat aktiv
JR TST1
;
TST3 BIT 1,(IX+10) ;nur jeder 2.
JR NZ,TST1 ;Interrupt!
TST4 POP AF
RETI
;
TST5 OR A ;0-Bit
JR TST7
TST6 SCF ;1-Bit
TST7 RR (IX+12) ;einschieben
```

```
JR      TST4

KTABDB  'Ww'
DB      'Aa'
DB      '2"'
DB      8,19H      ;CUL
DB      10H,0CH    ;HOME
DB      '-='
DB      0F2H,0F8H  ;F2
DB      'Yy'
DB      'Ee'
DB      'Ss'
DB      '3#'
DB      '^J'
DB      1,0FH      ;CLR
DB      ',*'
DB      0F3H,0F9H  ;F3
DB      'Xx'
DB      'Tt'
DB      'Ff'
DB      '5%'
DB      'Pp'
DB      1FH,2      ;DEL
DB      '0',40H
DB      0F5H,0FBH  ;F5
DB      'Vv'
DB      'Uu'
DB      'Hh'
DB      '7',27H
DB      'Oo'
DB      1AH,14H    ;INS
DB      '9)'
DB      3,3        ;BRK
DB      'Nn'
DB      'Ii'
DB      'Jj'
DB      '8('
DB      '['        ;SPC
DB      'Kk'
DB      ',<'
DB      13H,13H    ;STOP
DB      'Mm'
DB      'Zz'
DB      'Gg'
DB      '6&'
DB      ' '        ;freie Taste
DB      'Ll'
DB      ',>'
DB      0F6H,0FCH  ;F6
DB      'Bb'
DB      'Rr'
DB      'Dd'
DB      '4$'
DB      '\_
DB      '+;'
DB      '/?'
DB      0F4H,0FAH  ;F4
DB      'Cc'
DB      'Qq'
DB      16H,16H    ;CAPS
DB      '1!'
DB      LF,12H     ;CUD
DB      0BH,11H    ;CUU
DB      09H,18H    ;CUR
DB      0F1H,0F7H  ;F1
DB      CR,CR      ;ENTER

ISRTAB  DW  ISRPA   ;PIO A:Kassette
        DW  ISRPB   ;PIO B:Tastatur
        DW  0       ;CTC 0:frei
        DW  ISRC1   ;CTC 1:Kassette
        DW  ISRC2   ;CTC 2:Tondauer
        DW  ISRC3   ;CTC 3:Tastatur

KBDZ    ;Abfrage mit Quittung      ;**0E**
        CALL    KBDS
        RET     NC
        RES     0,(IX+8)
        RET

KBDS    ;Abfrage ohne Quittung      ;**0C**
        OR      A
        BIT     0,(IX+8);Code g}ltig?
        RET     Z
        LD      A,(IX+13)
        SCF     ;wenn ja
        RET

INIEA   PUSH     BC      ;**43**
        LD      C,M      ;1.Byte=Adr.
        INC     HL
        LD      B,M      ;2.Byte=Anz.
        INC     HL
```

```
OTIR
POP BC
RET

INIME CALL INIEA ;**44**
DEC D ;D Kan{le
JR NZ,INIME
RET

ERAM4 ;RAM I|schen
LD HL,4000H;RAM4
LD B,H
LD C,L
ERAM LD A,L
ERA4LD M,A ;I|schen
CPI ;INC HL, DEC BC
RET PO ;fertig
JR ERA4

DW 7F7FH
DB 'SWITCH',1
LD A,(ARGN)
LD D,E
CALL MODU
LD A,L ;Platz
CALL AHEX
CALL SPACE
LD A,H ;Kennbyte
CALL AHEX
CALL SPACE
LD A,E ;Steuerbyte
CALL AHEX
JP CRLF

MODU ;Lesen und Schalten ;**26**
CP 2
JR C,MODU5 ;nur lesen
LD A,L
CP 3
JR NC,MODU4
LD C,88H ;interne Module
IN H,(C)
AND A
JR NZ,MODU2
BIT 0,D ;f}r RAM0
RES 1,H
JR Z,MODU1
SET 1,H
MODU1 BIT 1,D
RES 3,H
JR Z,MODU3
SET 3,H
JR MODU3

MODU2 DEC A
JR NZ,MODU7
BIT 0,D ;f}r IRM
RES 2,H
JR Z,MODU3
SET 2,H
MODU3 OUT (C),H ;int. schalten
MODU4 LD A,D
LD C,80H
LD B,L
OUT (C),A ;Senden Steuerb.
MODU5 LD H,0
PUSH HL
PUSH AF
LD BC,0B800H
ADD HL,BC
POP AF
JR C,MODU6 ;nur lesen
LD M,A ;eintragen
MODU6 LD E,M ;R}ckgabe STB
POP HL ;(f}r Kommando)
LD C,80H
LD B,L
IN H,(C) ;Modultyp lesen
RET

MODU7 BIT 0,D ;f}r BASIC-ROM
RES 7,H
JR Z,MODU3
SET 7,H
JR MODU3

DW 7F7FH
DB 'JUMP',1
LD A,L
JUMPLD B,A ;**27**
LD C,80H
IN A,(C)
INC A ;Kennbyte FF?
```

```

    JP  Z,LOOP2    ;ja-Fehler
    LD  A,0FFH
    OUT (C),A      ;Ausgabe FF
    LD  H,0
    LD  DE,0B800H
    ADD HL,DE
    LD  M,A    ;Eintrag
    IN  A,88H
    AND 7EH
    JP  BJUMP      ;in IRM

;TAPE-Treiber

ISRO ;Init Bandausgabe    ;**08**
    LD  L,60H    ;Motor, LED ein
    CALL ISRI1
    LD  (IX+2),0 ;1. Block
    LD  BC,2000H ;Vortonl{nge
    ;
MBO  ;Ausgabe 1 Block      ;**01**
    INC (IX+2)
    DI
    XOR A
    LD  (IX+1),A;Pr}fsumme
    LD  A,87H    ;EI,ZG16,Res
    OUT 8DH
    LD  A,2FH    ;Vorton
    OUT 8DH
    EI
    LD  D,A    ;D=2Fh
MBO1    CALL HBITOT
    CPI
    JP  PE,MBO1    ;BC mal
    CALL ZTON      ;Trennzeichen
    LD  A,(IX+2)
    CALL BYTOT      ;Blocknummer
    LD  L,(IX+5)
    LD  H,(IX+6)
    LD  B,80H
BLKOT    LD  A,M
    CALL BYTOT      ;Datenbyte
    LD  A,(IX+1)
    ADD M    ;Pr}fsumme
    LD  (IX+1),A;berechnen
    INC HL
    DJNZBLKOT
    CALL BYTOT      ;und ausgeben
    LD  A,D
    CALL BITO1      ;noch ein Knacks
    LD  A,3
    OUT 8DH ;CTC stoppen
    RET

BYTOT    ;Ausgabe eines Bytes
    PUSH BC
    LD  C,A
    LD  B,8
BYTOUT    RRC C    ;mit Bit0
    PUSH AF    ;beginnend
    CALL C,HBITOT
    POP AF
    CALL NC,LBITOT
    DJNZBYTOUT ;8*
    POP BC
ZTON    LD  E,5DH    ;ZK Trennz.
    JR  BITOUT

LBITOT    LD  E,17H    ;ZK High-Bit
    JR  BITOUT

HBITOT    LD  E,2FH    ;ZK Low-Bit
BITOUT    LD  A,D
    CALL BITO1      ;aufrufen und
    LD  A,D ;reinlaufen
BITO1    LD  (IX),A
BITO2    LD  A,(IX)    ;Warten auf
    OR  A    ;Interrupt
    JR  NZ,BITO2
    LD  D,E
    RET

ISRI1    DI
    IN  A,88H
    OR  L    ;schalten
    OUT 88H
    EI
    LD  HL,(WEND)
    LD  (ZWEND),HL
    JP  PAGE

CSRO    ;Abschlu~ Bandausgabe    ;**09**
    LD  (IX+2),0FEH
    CALL MBO
```



```
CSROI  LD  HL,CASS  ;Kassettenpuffer
        LD  BC,7EH
        CALL  ERAM    ;||schen
        LD  HL,(ZWEND)
        LD  (WEND),HL
        IN  A,88H
        AND  9FH  ;Motor, LED aus
        OUT  88H
        JP  INIT  ;Ports init

ISRI  ;Init. Bandeingabe  ;**0A**
        LD  L,40H  ;Motor ein
        CALL  ISRI1
        CALL  MBI0 ;Block lesen
        RET  C  ;Fehler
        LD  DE,COM
        LD  H,(IX+6)
        LD  A,(IX+5)
        ADD  8
        LD  L,A
        LD  B,4
ISRI2  LD  A,(DE)
        SUB  M
        AND  A
        INC  HL
        INC  DE
        RET  NZ
        DJNZISRI2
        SET  7,(IX+7);COM-File
        RET

MBI  ;Einlesen 1 Block  ;**05**
        BIT  7,(IX+7)
        SCF  ;kein COM-File!
        RET  NZ
MBI0 LD  A,5  ;DI,ZG16
        OUT  8EH  ;CTC K2
        LD  A,83H  ;EI
        OUT  8AH  ;an PIO A
        PUSH  HL
        PUSH  DE
        LD  A,0A3H  ;ZK
        OUT  8EH
        EI
MBI1 LD  B,16H
        XOR  A
        LD  (IX+1),A ;Pr}fsumme=0
MBI2 CALL  STOP1
        JR  C,MBI1
        CP  0BAH
        CALL  LEDOO
        JR  C,MBI1  ;10h korrekte
        DJNZMBI2 ;Schwingungen
MBI3 LD  B,2  ;erkennen
MBI4 XOR  A
        LD  C,A
        LD  (IX),A
        CALL  STOP21  ;2 halbe Trenn-
        CP  5DH  ;zeichen er-
        JR  NC,MBI3  ;kennen
        DJNZMBI4
        CALL  BYTIN  ;Blocknummer
        CALL  C,LEDOO
        LD  (IX+2),A
        LD  B,80H
        LD  L,(IX+5)
        LD  H,(IX+6)
BLKIN CALL  BYTIN  ;Datenbyte
        CALL  C,LEDOO
        LD  M,A
        LD  A,(IX+1)
        ADD  M  ;aufsummieren
        LD  (IX+1),A
        INC  HL
        DJNZBLKIN
        CALL  BYTIN  ;Pr}fsumme
        CALL  C,LEDOO
        LD  B,A
        LD  A,(IX+1)
        CP  B  ;gleich?
        POP  DE
        POP  HL
        LD  A,3  ;DI an PIO-A
        OUT  8AH
        JR  Z,LEDOO
        SCF  ;St|rung
LEDOO  PUSH  AF
        IN  A,88H
        SET  5,A  ;LED ein
        JR  NC,MBI5
        RES  5,A  ;LED aus
MBI5 OUT  88H
        POP  AF
        RET
```

```
COM DB 'COM',1

ISRC1 ;ISR CTC Kanal 1 (Kassette-Out)
    PUSH AF
    LD A,3 ;DI,ZG16,Res
    OUT 8DH
    LD A,87H ;EI,ZG16,Res
    OUT 8DH
    LD A,(IX) ;Uebergabezelle
    OUT 8DH ;ZK
    XOR A
    LD (IX),A ;Quittierung
    JR ISR2E

ISRPA ;ISR PIO Kanal A (Kassette-In)
    PUSH AF
    IN A,8EH
    LD (IX),A ;Uebergabezelle
    LD A,7
    OUT 8EH
    LD A,0A3H
ISR1E OUT 8EH
ISR2E POP AF
    EI
    RETI

ISRC2 ;ISR CTC Kanal 2 (Tondauer)
    PUSH AF
    LD A,3 ;Ton aus
    OUT 8CH ;CTC0
    OUT 8DH ;CTC1
    IN A,89H
    SET 7,A ;Blinken ein
    OUT 89H
    LD A,47H ;CTC2 auf
    OUT 8EH ;Blinken stellen
    LD A,20H ;ZK
    RES 1,(IX+8);Ton ist aus
    JR ISR1E

STOP1 XOR A ;1 Bit einlesen
    LD (IX),A
STOP11 LD A,(IX) ;Warten auf
    OR A ;Interrupt
    JR Z,STOP11
    LD C,A
    XOR A
    LD (IX),A
STOP21 LD A,(IX)
    OR A
    JR Z,STOP21
    ADD C ;beide 'Zeiten'
    RET ;addieren

BYTIN ;Einlesen eines Bytes
    LD D,8 ;8 Bit
    XOR A
    LD E,A
TRN1 CALL STOP1 ;Bit einlesen
    CCF
    JR NC,TRN2
    CP 0BAH ;zu intolerant
    RET C
    SCF
TRN2RR E ;Bit einschieben
    DEC D
    JR NZ,TRN1 ;8*
    CALL STOP1 ;Trennzeichen
    LD A,E ;mit CY-
    RET ;R}ckmeldung

TON CALL LARG ;**35**
TON1 BIT 1,(IX+8);alten Ton
    JR NZ,TON1 ;abwarten
    LD A,L
    AND A
    LD A,3
    JR Z,TON3 ;kein Ton1
    LD A,7
    BIT 0,H
    JR Z,TON2 ;VT16
    OR 20H ;VT256
TON2 OUT 8CH
    LD A,L ;ZK1
TON3 OUT 8CH
    LD A,E
    AND A
    LD A,3
    JR Z,TON5 ;kein Ton2
    LD A,7
    BIT 0,D
    JR Z,TON4 ;VT16
    OR 20H ;VT256
```

```
TON4      OUT  8DH
          LD   A,E   ;ZK2
TON5      OUT  8DH
          LD   A,C   ;Lautst rke
          XOR  1FH   ;da nullaktiv
          AND  1FH   ;maskieren
          SET  7,A   ;Blinken ein
          LD   C,A
          LD   A,B
          AND  A
          JR   Z,TON6 ;Dauerton
          RES  7,C   ;Blinken aus
          SET  1,(IX+8);neuer Ton
          LD   A,0C7H ;EI, Z hler 50Hz
          OUT  8EH   ;CTC2
          LD   A,B   ;Dauer
          OUT  8EH
TON6      LD   B,60H
TONB      IN   A,89H
          AND  B     ;maskieren
          OR   C
          OUT  89H   ;Ltst. ausgeben
          RET
```

```
ZKOUT     LD   A,M           ;**45**
          INC  HL
          AND  A
          RET  Z
          CALL OCHR
          JR   ZKOUT
```

```
TOKTAB    ;BASIC-Token
          ;(Bit 7 im 1. Byte gesetzt!)
          DB  ' NKEY$'
          DB  ' OYST'
          DB  ' TRING$'
          DB  ' NSTR'
          DB  ' ENUMBER'
          DB  ' ELETE'
          DB  ' AUSE'
          DB  ' EEP'
          DB  ' INDOW'
          DB  ' ORDER'
          DB  ' NK'
          DB  ' APER'
          DB  ' T'
          DB  ' OLOR'
          DB  ' OUND'
          DB  ' SET'
          DB  ' RESET'
          DB  ' LOAD'
          DB  ' PEEK'
          DB  ' POKE'
          DB  ' OCATE'
          DB  ' EYLIST'
          DB  ' EY'
          DB  ' WITCH'
          DB  ' TEST'
          DB  ' LOSE'
          DB  ' PEN'
          DB  ' ANDOMIZE'
          DB  ' GET$'
          DB  ' INE'
          DB  ' IRCLE'
          DB  ' SRLIN'
          DB  80H
```

```
TOKJP     ;Sprungtabelle f r BASIC-Token
          DW  INKEY
          DW  02FDH   ;JOYST
          DW  STRING
          DW  INSTR
          DW  RENUM
          DW  DELETE
          DW  PAUSE
          DW  BBEEP
          DW  WINDOW
          DW  BORDER
          DW  INK
          DW  PAPER
          DW  0C348H   ;AT
          DW  BCOLOR
          DW  SOUND
          DW  PSET
          DW  PRESET
          DW  BLOAD
          DW  0C348H   ;VPEEK
          DW  VPOKE
          DW  LOCATE
          DW  KEYLIST
          DW  BKEY
          DW  SWITCH
          DW  0C348H   ;PTEST
          DW  CLOSE
```

```
DW OPEN
DW RANDOM
DW 0C348H ;VGET
DW BLINE
DW CIRCLE

BEXP1 ;BASIC-Exp 1
LD A,B
SUB 9
JR C,SNERR1
CP 27
JR NC,SNERR1 ;>26
RLCA ;*2
LD C,A
LD B,0
EX DE,HL
LD HL,TOKJP+8
JP 0C8B7H ;R}cksprung

BEXP2 ;BASIC-Exp 2
LD A,M
CP 0DFH
RET C ;<DFH
CP 0E3H
RET NC ;>E2H
CP 0E1H;AT?
JP Z,PRAT
LD A,(3FDH)
AND A ;PRINT-Erw.?
JR NZ,SNERR1
INC A
LD (3FDH),A
PREX1 PUSH HL
LD HL,COLOR
CALL IRMRD
POP HL
LD (37EH),A;Farbe merken
LD A,M
CP 0DFH;INK?
JR Z,PRINK
CP 0E2H;COLOR?
JP Z,PCOL
CALL 0C8BDH
CALL PAPER
PREX2 LD A,M
CP ','
JR Z,PREX5
JR SNERR1

PCOL CALL 0C8BDH
CALL BCOLOR
JR PREX2

SNERR1 JP 0C348H ;SN-ERROR

PRINK CALL 0C8BDH
CALL INK
LD A,M
CP ','
JR Z,PREX5
CALL 0C8CCH ;Komma?
DB ','
CP 0E0H;PAPER?
JR NZ,SNERR1
CALL 0C8BDH
CALL PAPER
CALL 0C8CCH
DB ','
PREX3 CALL 0CB03H
LD A,(37EH);Farbe
PUSH HL
LD HL,COLOR
CALL IRMWR ;eintragen
POP HL
POP BC
RET
PREX4 PUSH BC
JR PREX1
PREX5 CALL 0C8BDH
JR PREX3

BEXP3 ;BASIC-Exp 3
LD A,C
CP 62H
JP Z,VPEEK
CP 6EH
JP Z,PTEST
CP 7CH
JP Z,CSRLN
CP 76H
JP Z,VGET
SUB 3EH
JR C,SNERR1
CP 7
```

```
JR    NC,SNERR1
EX    DE,HL
LD    BC,TOKJP ;Tokentab.
POP  HL
LD    L,A
ADD  HL,BC
LD    C,M
INC  HL
LD    H,M
LD    L,C
PUSH HL    ;Routine
EX    DE,HL    ;anspringen
RET
```

```
STRING    ;String vervielf{ltigen
CALL     0C8CCH
DB  '('    ;Klammer auf?
CALL     0D421H
PUSH  AF
CALL     0C8D6H    ;Komma?
CALL     0CD3AH
CALL     0C8DBH    ;Klammer zu?
POP  AF
PUSH  HL
PUSH  AF
CALL     0D330H
INC  HL
INC  HL
LD    E,M
INC  HL
LD    D,M
POP  BC
PUSH  BC
PUSH  AF
PUSH  DE
LD    C,A
XOR  A
CP    C
JR    Z,STR2
CP    B
JR    Z,STR2
LD    A,C
DEC  B
JR    Z,STR2
```

```
STR1ADD C
JR    C,STR5
DJNZ STR1
STR2LD  B,A
LD    C,0
PUSH  BC
CALL     0D1E1H    ;Str.-Arithm.
POP  BC
POP  BC
PUSH  BC
CALL     0D17EH
POP  HL
EX    (SP),HL
LD    A,H
POP  HL
EX    (SP),HL
LD    L,A
INC  H
```

```
STR3DEC H
PUSH  HL
PUSH  BC
JR    Z,STR4
CALL     0D2F2H
POP  BC
POP  HL
JR    STR3
```

```
STR4POP BC
POP  HL
POP  DE
CALL     0D302H
JP     0D1A9H
```

```
STR5LD  E,1CH    ;ST-
JP     0C356H    ;ERROR
```

```
RENUM    ;neu nummerieren
PUSH  HL
LD    HL,10
LD    (354H),HL;DISTAN
LD    HL,(35FH);Start
PUSH  AF
PUSH  HL
INC  HL
INC  HL
LD    A,M
INC  HL
LD    H,M
LD    L,A
LD    (34EH),HL;ZL-Nr
LD    (352H),HL;NANF
LD    DE,(3D7H)    ;Ende
```

```
DEC DE
DEC DE
REN1 POP HL
PUSH HL
LD A,M
INC HL
LD H,M
LD L,A
CALL 0C689H
EX (SP),HL
JR NZ,REN1
POP DE
INC HL
INC HL
LD A,M
INC HL
LD H,M
LD L,A
LD (350H),HL ; ZL-Abst
LD B,4
POP AF
LD HL,34EH
EX (SP),HL
REN2 JR Z,REN5
CALL 0C986H
PUSH AF
LD A,D
OR E
REN3 JP Z,0C967H ; FC-ERROR
POP AF
EX (SP),HL
LD M,E
INC HL
LD M,D
INC HL
JR Z,REN5
PUSH AF
DEC B
JR Z,REN4
POP AF
EX (SP),HL
CALL 0C8D6H ; Komma?
JR REN2
REN4 POP AF
JP NZ,SNERR2
REN5 LD HL,(350H)
LD DE,(34EH)
CALL 0C689H
REN6 JP C,0C967H
LD HL,(35FH)
REN7 CALL 0C4BEH
JR C,REN8
JR Z,REN3
JR REN7
REN8 POP HL
PUSH BC
LD DE,(350H)
LD HL,0
LD (350H),HL
REN9 LD H,B
LD L,C
LD C,M
INC HL
LD B,M
LD A,B
OR C
JR Z,REN3
INC HL
LD A,M
INC HL
LD H,M
LD L,A
CALL 0C689H
LD HL,(350H)
INC HL
LD (350H),HL
JR NZ,REN9
INC HL
INC HL
ADD HL,HL
INC HL
LD DE,(3D7H)
ADD HL,DE
JR C,REN6
CALL 0C327H ; genug RAM?
LD (3D7H),HL
XOR A
DEC HL
LD M,A
DEC HL
LD M,A
DEC DE
DEC DE
EX DE,HL
```

```
LD M,E
INC HL
LD M,D
INC HL
DEC A
LD M,A
INC HL
LD M,A
INC HL
EX DE,HL
LD HL,(352H)
LD (34EH),HL
REN10 POP HL
LD C,M
INC HL
LD B,M
INC HL
PUSH BC
LD BC,34EH
LD A,M
LD (DE),A
LD A,(BC)
LD M,A
INC HL
INC DE
INC BC
LD A,M
LD (DE),A
LD A,(BC)
LD M,A
INC DE
LD HL,(34EH)
LD BC,(354H)
ADD HL,BC
LD (34EH),HL
LD HL,(350H)
DEC HL
LD A,H
OR L
LD (350H),HL
JR NZ,REN10
LD (DE),A
POP HL
LD HL,(35FH)
PUSH HL
REN11 POP HL
LD C,M
INC HL
LD B,M
INC HL
PUSH BC
LD A,M
INC HL
AND M
INC A
JR Z,REN15
REN12 INC HL
REN13 LD A,M
OR A
JR Z,REN11
CP 88H ;GOTO
JR Z,REN16
CP 8CH ;GOSUB
JR Z,REN16
CP 8BH ;RESTORE
JR Z,REN14
CP 0D4H ;ELSE
JR Z,REN14
CP 0A9H;THEN
JR NZ,REN12
REN14 CALL 0C987H
LD A,E
OR D
CALL NZ,RENUP1
CALL NZ,RENUP2
JR REN13
REN15 DEC HL
LD (3D7H),HL
DEC HL
LD M,A
DEC HL
LD M,A
POP HL
POP HL
JP 0C48AH
REN16 CALL 0C987H
LD A,E
OR D
JR Z,REN13
CALL RENUP1
CALL NZ,RENUP2
LD A,M
CP ''
JR NZ,REN13
```

```
JR    REN16

RENUP1  PUSH    HL
PUSH    DE
LD    DE,-1
CALL    0C4BBH ;n{chste Zeile
POP    DE
INC    BC
INC    BC
INC    BC
INC    BC
LD    H,B
LD    L,C
LD    BC,(352H)
RUP11   LD    A,M
INC    HL
PUSH    HL
OR     M
JR     Z,RUP13
LD    A,M
DEC    HL
LD    L,M
LD    H,A
CALL    0C689H
JR     Z,RUP12
LD    HL,(354H)
ADD    HL,BC
LD    B,H
LD    C,L
POP    HL
INC    HL
JR     RUP11
RUP12   LD    A,OFFH
OR     A
RUP13   POP    HL
POP    HL
RET

RENUP2  PUSH    BC
EX     DE,HL
LD     HL,(3D7H)
SBC    HL,DE
PUSH    HL
RUP21   POP    BC
LD     H,D
LD     L,E
DEC    DE
LD     A,(DE)
CP     ':'
JR     Z,RUP22
CP     ':'
JR     NC,RUP22
PUSH    BC
PUSH    DE
LDIR
POP    DE
JR     RUP21
RUP22   EX     DE,HL
POP    DE
PUSH    HL
PUSH    BC
XOR    A
LD     B,98H
CALL    0D6AEH
CALL    0D834H
POP    BC
POP    DE
INC    HL
INC    DE
RUP23   LD     A,M
OR     A
JR     Z,RUP24
PUSH    BC
PUSH    HL
EX     DE,HL
ADD    HL,BC
LD     D,H
LD     E,L
DEC    HL
LDDR
POP    HL
LDI
POP    BC
JR     RUP23
RUP24   PUSH    DE
LD     DE,(35FH)
CALL    0C493H
RUP25   INC    HL
LD     A,M
INC    HL
OR     M
JR     NZ,RUP25
EX     DE,HL
LD     M,E
```



```

INC HL
LD M,D
INC DE
INC DE
LD (3D7H),DE
POP HL
LD D,H
LD E,L
RUP26 LD A,M
OR A
INC HL
JR NZ,RUP26
POP BC
EX (SP),HL
PUSH BC
EX DE,HL
RET

DELETE ;Zeile(n) l|schen
RET Z
CALL 0C986H
JP Z,0C442H
CALL 0C8D6H
PUSH DE
CALL 0C986H
POP HL
RET NZ
EX DE,HL
PUSH HL
CALL 0C4BBH ;n{chste Zeile
JP NC,0C44DH ;UL-ERROR
POP DE
PUSH AF
PUSH BC
CALL 0C4BEH
JP NC,0C44DH ;UL-ERROR
POP BC
JP 0C450H ;einsortieren

PAUSE ;Programm unterbrechen
CALL 0C8BEH
JR NZ,PAUS2
PAUS1 CALL KBDS
JR NC,PAUS1
CP 3 ;BRK?
RET Z
CP 0AH ;CUU?
JR NZ,PAUS1
JP 0DDE4H ;Eingabe ASCII
PAUS2 CALL 0D421H
LD C,A
PAUS3 LD A,10H ;96 sek.
LD E,14H ;WAIT
CALL 0F015H ;PV5
PUSH BC
CALL KBDS
POP BC
JR NC,PAUS4
CP 3 ;BRK?
RET Z
CP 0AH ;CUU?
JR NZ,PAUS4
JP 0DDE4H ;Eingabe ASCII
PAUS4 DEC C
JR NZ,PAUS3
RET

BBEEP ;Signalton
LD B,1 ;ohne Angabe 1x
CALL 0C8BEH
JR Z,BBP1
CALL 0D421H
LD B,A ;Anzahl
BBP1 LD A,7 ;Beep
LD E,0 ;CRT
CALL 0F015H ;PV5
DJNZBBP1
RET

WINDOW ;Fenster einstellen
CALL 0C8BEH
JR Z,WINDOW1
PUSH HL
CALL 0D421H
EX (SP),HL ;in
LD HL,WNDFN+90 ;Fenster 9
CALL IRMWR ;erstellen
INC HL
EX (SP),HL
CALL 0C8D6H
CALL 0D421H
EX (SP),HL
CALL IRMWR
INC HL
```

```
EX (SP),HL
CALL 0C8D6H
CALL 0D421H
EX (SP),HL
CALL IRMWR
EX (SP),HL
CALL 0C8D6H
CALL 0D421H
EX (SP),HL
LD D,A
CALL 0F018H ;IRMON
CP 28H
JR NC,WINERR
LD A,(WNDNFN+92)
LD E,A
CP 28H
JR NC,WINERR
LD A,D
SUB E
JR C,WINERR
INC A
LD (WINLG),A
LD A,E
LD (WINON),A
LD A,(WNDNFN+91)
CP 20H
JR NC,WINERR
LD D,A
LD A,(WNDNFN+90)
CP 20H
JR NC,WINERR
LD E,A
LD A,D
SUB E
JR C,WINERR
INC A
LD (WINLG+1),A
LD A,E
LD (WINON+1),A
JR WINDOW2
WINDOW1 PUSH HL
CALL 0F018H ;IRMON
LD HL,100H
LD (WINON),HL
LD HL,1E28H
LD (WINLG),HL
WINDOW2 LD HL,0
LD (CURSO),HL
CALL 0F01BH ;IRMOF
POP HL
RET
WINERR CALL 0F01BH ;IRMOF
JR SNERR2

BLOAD ;MC-Programm laden
PUSH HL
LD HL,ARGN
XOR A
CALL IRMWR ;ARGN=0
POP HL
LD E,10H ;LOAD
JP 0F015H ;PV5

VPEEK ;Zeichen aus IRM lesen
CALL 0CDE1H
EX (SP),HL
LD DE,0CDF3H ;RET-Adr.
PUSH DE
CALL 0C96FH
PUSH HL
LD HL,8000H;IRM-Offset
ADD HL,DE
JR C,VPK ;Error
LD A,0BFH
CP H
VPK JR C,SNERR2
CALL IRMRD ;lesen
POP HL
JP 0D3E9H

SNERR2 JP 0C348H ;SN-ERROR

VPOKE ;Zeichen in IRM schreiben
CALL 0C96CH
PUSH DE
CALL 0C8D6H
CALL 0D421H
EX (SP),HL
LD DE,8000H;IRM-Offset
ADD HL,DE
JR C,VPK ;Error
LD D,A
LD A,0BFH
CP H
```

```
JR C,VPK ;nur bis BFFFH
LD A,D
CALL IRMWR ;schreiben
POP HL
RET

BORDER ;schaltet Bit 5,6 von PIO-B
;(beim KC85/4 RAM8!)
CALL 0D421H
RRCA
RRCA
RRCA
AND 60H ;Auswahl
LD C,A
LD B,9FH ;Maske
JP TONB

PRAT;PRINT AT
LD A,(3FDH)
BIT 1,A
SET 1,A
LD (3FDH),A
JR NZ,SNERR2
PUSH DE
PUSH HL
CALL 0F018H ;IRMON
LD HL,WINON
LD (WNDFN+90),HL
LD DE,WNDFN+80
LD BC,6 ;Fenster
LDIR ;retten
LD HL,0 ;Fenster gro~
LD (WINON),HL
LD HL,2028H
LD (WINLG),HL
CALL 0F01BH ;IRMOF
POP HL
CALL LOCAT
CALL 0C8DBH
CALL 0C8CCH
DB ','
LD A,M
CP 0DFH
JR C,PRAT2 ;<DFH
CP 0E3H
JR NC,PRAT2 ;>E2H
CALL PREX4 ;INK,PAPER,COLOR
PRAT1 PUSH HL
CALL 0F018H ;IRMON
LD HL,WNDFN+80
LD DE,WINON
LD BC,6 ;Fenster
LDIR ;regenerieren
CALL 0F01BH ;IRMOF
POP HL
POP DE
POP BC
RET

PRAT2 CALL 0CB03H
JR PRAT1

SNERR3 JP 0C348H ;SN-ERROR

INK ;Vordergrundfarbe
CALL 0D421H
CP 32
JR NC,SNERR3
SLA A
SLA A
SLA A
LD D,A
PUSH HL
LD HL,COLOR
CALL IRMRD
POP HL
AND 7
JR COL2

PAPER ;Hintergrundfarbe
CALL 0D421H
CP 8
JR NC,SNERR3
LD D,A
PUSH HL
LD HL,COLOR
CALL IRMRD
POP HL
AND 0F8H
COL2 OR D
PUSH HL
LD HL,COLOR
CALL IRMWR ;neuer Farbwert
POP HL
RET
```

```
BCOLOR      ;Farbe einstellen
CALL        0C8BEH
JR          Z,SNERR3
CALL        INK      ;Vordergrund
CALL        0C8BEH
RET Z
CALL        0C8D6H
JR          PAPER    ;Hintergrund
RET         ;???
```

```
LOCAT       ;Cursor positionieren
CALL        0C8BDH
CALL        0C8CCH
DB          '('
```

```
LOCATE      CALL        0D421H
LD          D,A
PUSH        HL
LD          HL,WINLG+1
CALL        IRMRD
POP HL
DEC A
CP          D
JR          C,SNERR3
CALL        0C8D6H
PUSH        DE
CALL        0D421H
POP DE
LD          C,A
PUSH        HL
LD          HL,WINLG
CALL        IRMRD
POP HL
DEC A
CP          C
JR          C,SNERR3
LD          A,C
PUSH        HL
LD          HL,CURSO
CALL        IRMWR
LD          A,D
INC HL
LOC1        CALL        IRMWR
POP HL
RET
```

```
INKEY       ;Eingabe ein Zeichen
PUSH        HL
CALL        KBDS
JR          NC,INKEY3
LD          A,1
CALL        0D17BH
CALL        0DDE4H      ;Eingabe ASCII
INKEY1      LD          HL,(3C2H)
LD          M,A
INKEY2      JP          0D1A9H
INKEY3      XOR A
CALL        0D17BH
JR          INKEY2
```

```
SOUND       ;Tonausgabe
CALL        0D421H
PUSH        HL
LD          HL,ARG1
LD          B,4      ;4 Argumente
SOUND1      CALL        IRMWR
INC HL
EX          (SP),HL
DEC B
JR          Z,SOUND2
PUSH        BC
CALL        0C8D6H
CALL        0D421H
POP BC
EX          (SP),HL
JR          SOUND1
SOUND2      CALL        0C8BEH
JR          Z,SOUND3
CALL        0C8D6H
CALL        0D421H
EX          (SP),HL
CALL        IRMWR
INC HL
EX          (SP),HL
CALL        0C8BEH
JR          Z,SOUND3
CALL        0C8D6H
CALL        0D421H
EX          (SP),HL
CALL        IRMWR
EX          (SP),HL
SOUND3      LD          E,35H      ;TON
POP BC
JP          0F015H      ;PV5
```

```
PSET ;Punkt setzen
LD B,1 ;setzen
POINT PUSH BC
CALL 0C96CH
PUSH HL
LD A,E
LD HL,HOR
CALL IRMWR
LD A,D
INC HL
CALL IRMWR
EX (SP),HL
CALL 0C8D6H
CALL 0D421H
EX (SP),HL
INC HL
CALL IRMWR
POP HL
CALL GFARB
LD E,30H ;PUSE
POP BC
DEC B
JR Z,POINT2
DEC E ;PUDE
POINT2 CALL 0F015H ;PV5
JP C,SNERR2
RET
```

```
PRESET ;Punkt I|schen
LD B,0 ;I|schen
JR POINT
```

```
GFARB ;Grafikfarbe setzen
CALL 0C8BEH
RET Z
CALL 0C8D6H
CALL 0D421H
PUSH HL
LD HL,FARB ;Grafik-Farbe
RLA
RLA
RLA
JP LOC1 ;in IRM schreib.
```

```
INSTR ;String1 in String2 suchen
CALL 0CD36H ;Klammer auf?
CALL 0C8D6H
PUSH HL
CALL 0D330H
JR Z,INSTR4
LD B,A
INC HL
INC HL
LD E,M
INC HL
LD D,M
POP HL
PUSH DE
PUSH BC
CALL 0CD3AH
CALL 0C8DBH ;Klammer zu?
POP BC
POP DE
PUSH HL
PUSH DE
PUSH BC
CALL 0D330H
JR Z,INSTR4
INC HL
INC HL
LD C,M
INC HL
LD H,M
LD L,C
POP BC
LD C,A
POP DE
PUSH HL
INSTR0 PUSH BC
PUSH DE
LD A,(DE)
INSTR1 CP M
JR Z,INSTR5
INC HL
DEC C
JR NZ,INSTR1
INSTR2 XOR A
POP HL
POP HL
POP HL
INSTR3 LD DE,0CDF3H ;RET-Adr.
PUSH DE
JP 0D0C0H
```

```
INSTR4  JP  0C967H  ;FC-ERROR
INSTR5  INC  HL
        PUSH  HL
        DEC  HL
INSTR6  INC  HL
        DEC  C
        JR   Z,INSTR7
        INC  DE
        DEC  B
        JR   Z,INSTR8
        LD   A,(DE)
        CP   M
        JR   Z,INSTR6
        POP  HL
        POP  DE
        LD   A,C
        POP  BC
        LD   C,A
        JR   INSTR0
INSTR7  INC  DE
        DEC  B
        POP  HL
        JR   NZ,INSTR2
        JR   INSTR9
INSTR8  POP  HL
INSTR9  POP  DE
        POP  DE
        POP  DE
        AND  A
        SBC  HL,DE
        LD   A,L
        JR   INSTR3

IRMWR   LD   E,28H  ;LDMA
        JR   JPV5

IRMRD   LD   E,29H  ;LDAM
        JR   JPV5

BKEY    ;F-Taste belegen
        CALL 0D421H
        AND  A
        JR   Z,SNERR4
        CP   0DH  ;F1...F12
        JR   NC,SNERR4
        LD   E,39H  ;KEY
JPV5     JP  0F015H  ;PV5

KEYLIST ;F-Tasten auflisten
        LD   E,3AH  ;KEYLI
        JR   JPV5

SWITCH  ;Module schalten
        CALL 0D421H
        PUSH AF
        CALL 0C8D6H
        CALL 0D421H
        LD   D,A  ;Steuerbyte
        POP  AF
        PUSH HL
        LD   L,A  ;Steckplatz
        LD   A,2  ;schalten
        LD   E,26H ;MODU
        CALL 0F015H ;PV5
        POP  HL
        RET

PTEST   ;Punkt testen
        CALL 0CDE1H
        EX   (SP),HL
        LD   DE,0CDF3H ;RET-Adr.
        PUSH DE
        CALL 0C96FH
        PUSH HL
        LD   A,E  ;nur
        LD   HL,HOR ;X-Koordinate
        CALL IRMWR ;eintragen
        LD   A,D
        INC  HL
        CALL IRMWR
        LD   E,2FH  ;PUDE
        CALL 0F015H ;PV5
        LD   B,0
        JR   Z,PTST1 ;war gel|scht
        INC  HL
        INC  HL
        CALL IRMWR
        LD   E,30H  ;PUSE
        CALL 0F015H ;PV5
        LD   B,1
PTST1   LD   A,B  ;R}ckgabewert
        POP  HL
        JP   0D3E9H
```

SNERR4 JP 0C348H ;SN-ERROR

CLOSE ;Kanal schlie~en

LD C,0
LD A,M
CP 'I' ;Eingabe?
JR Z,CLOS1
INC C
CP 'O' ;Ausgabe?
JR NZ,SNERR4

CLOS1 PUSH BC
INC HL
LD A,M
CP '#'

CLOS2 JR NZ,SNERR4
INC HL
CALL 0D421H
AND 3 ;4 Ger{te
POP BC
RET Z ;Console
PUSH HL
RLA ;*2
ADD C ;Ein/Ausgabe
PUSH AF
DEC A
LD B,0
SCF

CLOS3 RL B ;Bit platzieren
DEC A
JR NZ,CLOS3
LD HL,307H ;E/A-Flag
LD A,M
XOR B ;negieren
LD M,A
POP AF
POP HL
SET 6,A ;close
PUSH DE
LD E,A
LD D,3
CALL BASIO
POP DE
RET

RANDOM ;Zufallsgenerator

LD A,R
LD (31DH),A
RET

OPEN ;Kanal |ffnen

LD A,M
INC HL
CP 'I' ;Eingabe?
JR Z,OPEN1
CP 'O' ;Ausgabe?
JR NZ,CLOS2
CALL 0DE25H
CALL 0DDC8H
RET Z
LD A,0D5H
CALL 0DCB2H
LD HL,3EAH ;Druckpuffer
XOR A
CALL 0DDD5H
POP HL
RET

OPEN1 CALL 0DE5FH ;Eingabe
LD A,(309H);IN-Index
AND 3
RET Z
LD A,0D5H
CALL 0DCB2H
LD HL,3EAH ;Druckpuffer
CALL 0DDE4H ;Eingabe ASCII
POP HL
RET

BLINE ;Linie zeichnen

LD BC,43EH ;4 Par.,LINE
JR GRAPH

CIRCLE ;Kreis zeichnen

LD BC,33FH ;3 Par.,CIRCLE

GRAPH PUSH BC
CALL 0C96CH
POP BC
PUSH BC
PUSH HL
LD HL,ARG1
GRAPH1 LD A,E
CALL IRMWR
INC HL
LD A,D
CALL IRMWR

```

INC HL
DEC B
JR    Z,GRAPH2
EX    (SP),HL
PUSH BC
CALL    0C8D6H
CALL    0C96CH
POP BC
EX    (SP),HL
JR    GRAPH1
GRAPH2 POP HL
CALL    GFARB
POP DE
JP    0F015H    ;PV5

CSRLN    ;akt. Zeile holen
CALL    0CDE1H
EX    (SP),HL
LD    DE,0CDF3H ;RET-Adr.
PUSH DE
CALL    0D424H
PUSH HL
AND A
LD    A,0
JR    NZ,CSRLN1
LD    HL,WINON+1
CALL    IRMRD
CSRLN1 LD    B,A
LD    HL,CURSO+1
CALL    IRMRD
ADD B    ;akt. Zeile
POP HL
JP    0D3E9H

VGET;Zeichen von Bildschirm lesen
EX    (SP),HL
LD    A,1
CALL    0D17BH
CALL    0F018H    ;IRMON
LD    DE,(CURSO)
CALL    DABR
LD    A,M
CALL    0F01BH    ;IRMOF
JP    INKEY1    ;als Zeichen
                ;zur}ckgeben

DS    10,0FFH    ;frei

END
```



```
;CF31  ASM
; **  CAOS 3.1 ROM F      **

      ORG 0F000H

POWER  JP  PWRON
      JP  PV1  ;m. DEFB nr
      JP  PV2  ;(ARGC)=nr
      JP  PV3  ;E=nr
      JP  PV4  ;E=nr, IRM e/a
      JP  RCALL      ;m. DEFW offset
      JP  PWR3      ;JUMP-Einsprung
      JP  PV5  ;E=nr
      JP  IRMON
      JP  IRMOF
      PUSH BC      ;PV6, (ARGC)=nr
      CALL IRMON
      CALL PV2
      JR  PV51

PV5    PUSH BC
      CALL IRMON
      CALL PV3
PV51   CALL IRMOF
      POP BC
      RET

PV4    PUSH AF
      IN  A,88H
      SET 2,A
      OUT 88H ;IRM ein
      POP AF
      CALL PV3
      PUSH AF
      IN  A,88H
      RES 2,A
      OUT 88H ;IRM aus
      POP AF
      RET

PV1    DI
      PUSH HL
      POP HL
      POP HL
      INC HL      ;nr }bergehen
      PUSH HL
      DEC HL      ;(HL)=nr
      DEC SP
      DEC SP
      EI
      PUSH AF
      PUSH DE
      LD  E,M
PV22   LD  D,0
      LD  HL,(SUTAB)
      ADD HL,DE
      ADD HL,DE      ;HL=Pos. in Tab
      LD  E,M
      INC HL
      LD  D,M
      EX  DE,HL
      POP DE
      POP AF
      EX  (SP),HL
      RET      ;UP anspringen

PV2    AND A      ;VR: F
PV21   PUSH HL
      PUSH DE
      PUSH BC
      PUSH HL
      LD  HL,POP3      ;Returnadresse
      EX  (SP),HL      ;kellern
      PUSH HL
      PUSH AF
      PUSH DE
      JR  C,PV22      ;falls PV3
      LD  A,(ARGC);sonst E
      LD  E,A      ;beschaffen
      JR  PV22

PV3    SCF      ;VR: F
      JR  PV21

IRMON  POP BC      ;R}ckkehradr.
      PUSH IY
      LD  IY,0      ;SP in IY merken
      ADD IY,SP;(kein UP ver-
      DI      ;{ndert IY!)
      LD  (IX+11),A
      IN  A,88H
      SET 2,A      ;IRM ein
      OUT 88H
      LD  SP,(SYSP)
```

```
EI
LD  A,(IX+11)
PUSH BC
RET      ;wie JP (BC)

IRMOF  POP BC      ;analog IRMON
LD  (IX+11),A
IN  A,88H
RES 2,A      ;IRM aus
DI
OUT 88H
LD  SP,IY
EI
LD  A,(IX+11)
POP  IY
PUSH BC
RET

RCALL EX  (SP),HL      ;VR: DE
LD  E,M
INC  HL
LD  D,M
INC  HL      ;DE=offset
EX  DE,HL
ADD  HL,DE      ;zur Absolut-
EX  DE,HL      ;adr. addieren
EX  (SP),HL
PUSH DE
RET      ;wie JP (DE)

PWRON      ;Einschalt-Init
LD  SP,1C4H
XOR  A
LD  L,A
LD  H,A
LD  B,A
LD  C,A

PWR1  LD  M,A      ;Speicher
CPI      ;I|schen
JP  PE,PWR1
LD  BC,0FF80H
OTIR      ;Module aus
LD  B,8
IN  A,(C)
CP  0F4H      ;M022 in 08?
JR  NZ,PWR2
LD  A,2
LD  L,B
LD  D,43H      ;als RAM4
CALL  MODU      ;zuschalten
CALL  ERAM4      ;RAM4 I|schen

PWR2  LD  D,3
LD  HL,IOTAB
CALL  INIME      ;Ports init

PWR3  DI
CALL  SYSI      ;IRM init
LD  A,0CH
CALL  CRT      ;CLS
LD  HL,WINON+1
INC  M      ;ab Zeile 1
INC  HL
INC  HL
DEC  M      ;bis Zeile 30
DEC  M
LD  BC,880H      ;ROM-Modul mit
IN  A,(C)      ;Strukturbyte
DEC  A      ;01?
JR  NZ,MEN0      ;nein
LD  L,B
LD  D,43H      ;auf 4000H
LD  A,2      ;zuschalten
CALL  MODU
JP  4000H      ;anspringen

DW  7F7FH
DB  'MENU',1
MEN0  CALL  OSTR
DB  0CH
DB  '* HC-CAOS 3.1 *'
DB  LF,CR,0
MEN1  LD  HL,0
LD  C,L
LD  B,H
MENU  CALL  OSTR      ;**46**
DB  CR,2,'% ',0
CALL  BRKT
JR  C,LOOP1
LD  A,(IX+9)
MEN2  CPIR
JP  PO,LOOP1;ausgesucht
CPI      ;2. Mal?
JR  NZ,MEN2
MEN3  LD  A,M
CP  2      ;Epilog?
```

```
JR C,MEN4
CP 30H ;<30h?
JR C,MENU
CP 5FH ;>5Fh, Kleinb.?
JR NC,MENU
CALL OCHR
INC HL
DEC BC
JR MEN3
MEN4 CALL CRLF
JR MENU

LOOP CALL OSTR ;**12**
DB '%',0;Prompt
LOOP1 CALL INLIN
INC DE ;Zeichen danach
LD A,(DE)
CP ''
JR Z,LOOP
AND A
JR Z,LOOP
LD A,(IX+9)
CALL MSUCH
JR NC,LOOP2;nicht gef.
PUSH HL
CALL GARG ;Argumente
JR NC,LOOP3
POP HL
LOOP2 CALL ERRM
JR LOOP
LOOP3 LD HL,LOOP ;R}ckkehradr.
EX (SP),HL ;kellern
PUSH HL
CALL LARG
EX (SP),HL
LD A,M ;Epilog
AND A
JR NZ,LOOP4
IN A,88H
RES 2,A ;IRM off
OUT 88H
LOOP4 INC HL ;Epilog
EX (SP),HL ;}bergehen
RET

MSUCH LD BC,0
LD HL,0
ZSUCH PUSH AF ;**1D**
ZS1 POP AF
PUSH AF
ZS2 CPIR
JP PO,ZS6 ;ausgesucht
CPI
JR NZ,ZS2 ;falsch
PUSH DE
ZS3 LD A,(DE)
INC DE
CP '!' ;keine SPC und
JR C,ZS4 ;Steuerzeichen
CPI
JR Z,ZS3
POP DE
JR ZS1
ZS4 LD A,1
CP M ;Epilog?
JR NC,ZS5
POP DE
JR ZS1
ZS5 POP AF
POP AF
SCF ;CY=1
RET
ZS6 POP AF
AND A ;CY=0
RET

OSTR EX (SP),HL ;**23**
CALL ZKOUT
EX (SP),HL
NOT RET

NORM CALL NIN ;**13**
NOUT LD HL,ZEI0 ;**20**
SOUT PUSH HL ;**1E**
LD HL,(OUTAB)
EX (SP),HL
LD (OUTAB),HL
POP HL
RET

NIN LD HL,BUPTAB ;**21**
SIN PUSH HL ;**1F**
LD HL,(INTAB)
EX (SP),HL
```

```
LD (INTAB),HL
POP HL
RET

INTB PUSH HL ;**16**
LD HL,(INTAB)
JR INTA1

OCHR PUSH HL ;**24**
LD HL,(OUTAB)

INTA1 PUSH DE ;s.a. PV1
PUSH BC ;(sehr {hnlich)
PUSH AF
LD A,M
LD (ARGC),A
POP AF
CALL PV2

POP3 POP BC
POP DE
POP HL
RET

WTUP DEC A ;UP f}r WAIT
RET Z
PUSH AF
POP AF
JR WTUP

WAITLD B,A ;**14**
XOR A
WAIT1 CALL WTUP
DJNZ WAIT1
RET

KBD CALL CUCP ;**04**
PUSH HL
PUSH BC
BIT 6,(IX+8);F-Taste aktiv?
JR NZ,KBD6;ja
KBD1 CALL KBDZ
JR NC,KBD1
CP 0F1H;F-Taste?
JR C,KBD4
SET 6,(IX+8);ja - merken
SUB 0F0H
LD B,A
LD HL,0B900H
KBD2 LD A,M ;Pos. in F-
AND A ;Puffer best.
INC HL
JR NZ,KBD2
DJNZ KBD2
LD A,M ;erstes CHR
AND A
JR Z,KBD7 ;F-Taste frei
INC HL
KBD3 LD (FTAST),HL;Zeiger
KBD4 BIT 5,(IX+8);Click ein?
JR Z,KBD5
LD HL,19H
LD BC,0FH
LD E,H
PUSH AF
CALL TON1 ;Tastenclick
LD A,0FH
CALL WAIT
LD A,3
OUT 8CH ;Ton aus
POP AF
KBD5 POP BC
POP HL
JP CUCP
;
KBD6 CALL BRKT;Abbruch?
JR NC,KBD8
KBD7 RES 6,(IX+8);F-Taste inaktiv
JR KBD1
;
KBD8 LD HL,(FTAST);Zeiger auf
LD A,M ;n{chstes CHR
INC HL
AND A ;Ende?
JR NZ,KBD3
JR KBD7

KCTAB ;IRM-Defaulttabelle
DW NOT ;Hardcopy
DB 0 ;WINNR
DW 0 ;WINON
DW 2028H ;WINLG
DW 0 ;CURSO
DB 0 ;STBT
DB 39H ;COLOR
DW SCRLPG ;WEND
DW 0EE00H ;CCTL0
```

```
DW 0FE00H ;1
DW 0EE00H ;2
DW 0FE00H ;3
DW 01C4H ;SYSP
DW SUTB;SUTAB
DW CRTTAB ;CTAB
OUT 88H ;BJUMP
JP 0F012H
DW ZEIO ;OUTAB
DW ZEI4 ;INTAB
JP NOT ;UOUT1
JP NOT ;UIN1
JP NOT ;UOUT2
JP NOT ;UIN2
DW 0DC82H ;IOERR (BASIC)

INLIN      PUSH      HL      ;**17**
INLI1      CALL      INTB
          CALL      OCHR
CP CR      ;Enter?
JR NZ,INLI1
          CALL      CRLF
LD DE,(CURSO)
          CALL      DABR
LD DE,-40
          ADD HL,DE
EX DE,HL
          POP HL
          RET

HLHXLD A,H      ;**1A**
          CALL      AHEX
LD A,L
          CALL      AHEX
SPACE LD A,20H      ;**2B**
SPAC0      JP OCHR

ERRM      CALL      OSTR      ;**19**
          DB 'ERROR',7,0
          ;
CRLF      CALL      OSTR      ;**2C**
          DB CR,LF,0
          RET

HOME      LD A,10H      ;**2D**
          JR SPAC0

AHEX      PUSH      AF      ;**1C**
          RRA      ;Tetradentausch
          RRA
          RRA
          RRA
          CALL      AHEX0 ;erst rufen,
          POP AF      ;dann reinlaufen
AHEX0      AND 0FH ;Maske
          ADD 30H
          CP 3AH
          JR C,SPAC0 ;0..9
          ADD 7
          JR SPAC0 ;A..F

HLDE      CALL      HLDE1      ;**1B**
HLDE1      CALL      HLHX
          EX DE,HL
          RET

RHEX      LD A,(DE)      ;**18**
          CP ' ' ;Trennzeichen
          INC DE ;}bergehen
          JR Z,RHEX
          DEC DE
          LD HL,0
          LD (NUMVX),HL
          XOR A
          LD HL,NUMNX
          LD M,A
RH1 LD A,(DE)
          OR A ;Ende-Dummy?
          RET Z
          CP ' ' ;Ende-Space?
          RET Z
          SUB '0'
          RET C
          CP 10
          JR C,RH2
          SUB 7
          CP 10
          RET C
          CP 10H
          JR C,RH2
          SCF
          RET
RH2      INC DE
          INC M ;NUMNX erh|hen
```

```

    INC HL
    RLD      ;Tetrade
    INC HL   ;durchschieben
    RLD
    DEC HL
    DEC HL   ;RLD setzt auch
    JR  Z,RH1 ;Z-Flag!
    DEC DE
    SCF      ;Zahl zu gro~
    RET

GARG      ;bis 10 Arg. erfassen ;**22**
LD  BC,ARGN
    XOR A
LD  (BC),A ;(ARGN)=0
GARG1    CALL RHEX
    RET C   ;Fehler
LD  A,M
    OR  A
    RET Z   ;Ende-Dummy
    INC HL
    INC BC
LD  A,M ;L(NUMVX)
LD  (BC),A ;L(Argument)
    INC HL
    INC BC
LD  A,M ;H(NUMVX)
LD  (BC),A ;H(Argument)
LD  HL,ARGN
    INC M
LD  A,M
CP  11     ;10 Argumente?
JR  C,GARG1
SCF      ;zu viele
    RET

SYSI LD  DE,HCADR
LD  HL,KCTAB
LD  BC,32H
LDIR      ;IRM init
    XOR A
    CALL WINAK ;Fenster 0 init.
LD  HL,WNDFN
LD  DE,WNDFN+10
LD  BC,90
LDIR      ;Fenster 1..9
LD  A,1
;
SIXD LD  HL,ISRTAB;**31**
LD  (MIXIT),A
LD  E,0E4H
LD  D,A
LD  A,E
LD  BC,12 ;Interrupttab.
LDIR      ;kopieren
    ADD 12
LD  E,A
    PUSH DE ;IX-Arbeitsber.
    POP IX
LD  (IX+8),88H
LD  (IX+9),7FH
LD  (IX+14),KTAB ;Low
LD  (IX+15),KTAB/256;High
INIT    DI
    PUSH AF
    PUSH DE
    PUSH HL
IM  2
LD  A,(MIXIT)
LD  I,A
LD  D,5
LD  HL,IOTAB2
    CALL INIME ;Kan{le init.
    POP HL
    POP DE
    POP AF
EI
LD  (IX+5),CASS ;Low
LD  (IX+6),CASS/256 ;High
    RET

IOTAB    DB 8AH ;PIO-A-Steuerw.
DB 2
DB 0E4H;Int-Vektor
DB 4FH ;Byte-IN
DB 88H ;PIO-A-Daten
DB 1
DB 9FH ;IRM ein...
DB 8AH ;PIO-A-Steuerw.
DB 2
DB 0FH ;Byte-OUT
DB 3 ;DI

IOTAB2    DB 8BH ;PIO-B-Steuerw.
```

```

    DB 3
    DB 0FH ;Byte-OUT
    DB 83H ;EI
    DB 0E6H;Int-Vektor
DB 89H ;PIO-B-Daten
    DB 1
    DB 9FH ;Daten
DB 8CH ;CTC0
    DB 1
    DB 0E8H;Int-Vektor
DB 8EH ;CTC2 (blinken)
    DB 2
    DB 47H ;DI,Z{hler,RES
    DB 14H ;Zeitkonstante
DB 8AH ;PIO-A-Daten
    DB 1
    DB 0E4H;Int-Vektor
;
    DW 7F7FH
    DB 'SAVE',1
    CALL OSTR
    DB 'NAME :',0
    CALL INLIN
    LD HL,6 ;'NAME :'
    ADD HL,DE ;}bergehen
;
SAVELD DE,CASS ;**36**
    LD BC,11
    LDIR ;Name -> Puffer
    EX DE,HL
    LD M,0 ;Abschlu~-Null
    LD HL,ARGN
    LD A,M
    CP 2 ;mind 2 Arg.!!
    RET C
    LD DE,CASS+10H
    LD C,15H ;alle Argumente
    LDIR ;nach Puffer
    CALL ISRO
    LD HL,(ARG1)
SAVE1LD (IX+5),L
    LD (IX+6),H
    LD A,(IX+2)
    CALL AHEX;Blocknummer
    CALL OSTR
    DB ' ',0
    CALL BRKT
    LD BC,0A0H
    JR C,SAV2
    LD DE,80H
    ADD HL,DE
    LD DE,(ARG2)
    AND A
    SBC HL,DE ;Ende erreicht?
    JR NC,SAV2
    CALL MBO
    JR SAV1
;
SAV2CALL CRLF
    JP CSRO

BRKTBKTD ;**2A**
    RET NC
    CP 3 ;Break?
    SCF
    RET Z
    AND A
    RET

    LD (IX+9),7FH
LARGLD BC,(ARG3) ;**15**
    LD DE,(ARG2)
    LD HL,(ARG1)
    LD A,(ARGN)
    RET

    DW 7F7FH
    DB 'VERIFY',1
VERIF RES 0,(IX+7) ;**11**
    JR LOAD0

    DW 7F7FH
    DB 'LOAD',1
LOAD SET 0,(IX+7) ;**10**
LOAD0 CALL ISRI
LOAD1 LD A,(IX+2)
    JR NC,LOAD3
LOAD2 CALL AHEX
    CALL OSTR
    DB '*',CR,LF,0
    CALL MBI0
    JR LOAD1
;
```

```
LOAD3  CP  2      ;Block 0 oder 1?
      JR  NC,LOAD2
      INC A
      LD  E,A
      LD  HL,CASS
      LD  B,11
LOAD4  LD  A,M    ;Name anzeigen
      INC HL
      CALL      CRT
      DJNZ LOAD4
      LD  (IX+3),E;erwarteter Blk
      LD  A,(CASS+10H)
      BIT 0,(IX+7)
      JR  Z,LOAD6  ;wenn VERIFY
      LD  (IX+4),A;Anz. Arg.
      CP  2
      JR  C,NOMC   ;zu wenig Arg.
      CP  11
      JR  NC,NOMC  ;zu viel Arg.
      LD  DE,(CASS+13H)
      LD  HL,(CASS+11H)
      LD  A,(ARGN)
      AND A      ;Ladeoffset?
      JR  Z,LOAD5
      LD  BC,(ARG1)
      ADD HL,BC   ;AAdr, EAdr
      EX  DE,HL   ;umrechnen
      ADD HL,BC
      EX  DE,HL
      BIT 2,(IX+4);4..7 Args?
      JR  NZ,LOAD5
      PUSH HL
      LD  HL,(CASS+15H)
      ADD HL,BC   ;SAdr umrechnen
      LD  (CASS+15H),HL
      POP HL
LOAD5  PUSH      HL
      PUSH      DE
      LD  HL,CASS+11H
      LD  DE,ARG1
      LD  BC,20   ;alle Argumente
      LDIR      ;zur}ck kopieren
      POP DE
      POP HL
      CALL      SPACE
      CALL      HLDE;AAdr, Eadr
LOAD6  CALL      CRLF
      ;Load-Hauptschleife
LOAD7  CALL      BRKT
      BIT 7,(IX+7);COM-File?
      JR  Z,LOAD8
      JP  C,POWER ;BRK -> POWER
LOAD8  JR  C,CLJP ;BRK -> Abbruch
      CALL      MBIO
      LD  A,(IX+2);Blocknr.
      PUSH AF
      CP  (IX+3)
      JR  Z,LOAD9  ;erwarteter Blk.
      INC A
      JR  Z,LOAD9  ;letzter Blk.
      POP AF
      CALL      AHEX;Blocknummer
      CALL      OSTR
      DB  '* ',0
      JR  LOAD7
      ;
NOMC   CALL      OSTR
      DB  '???',19H,LF,0
CLJP   LD  (IX+4),2;kein Start!
      JR  CSRI
      ;
LOAD9  POP AF
      JR  C,LOAD14;Fehler
      PUSH AF
      CALL      AHEX
      CALL      OSTR
      DB  '> ',0
      POP AF
LOAD10 BIT 0,(IX+7)
      JR  Z,LOAD15;wenn VERIFY
      PUSH DE
      EX  DE,HL
      SBC HL,DE
      LD  BC,81H
      SBC HL,BC
      ADD HL,BC
      DEC BC
      JR  NC,LOAD11
      LD  C,L
LOAD11 LD  HL,CASS ;MC in Speicher
      LDIR      ;kopieren
      EX  DE,HL
      POP DE
      JR  C,CSRI  ;fertig!
```



```
LOAD12 INC (IX+3) ;n{chster Block
LOAD13 JR LOAD7
;
LOAD14 BIT 7,(IX+7)
JR NZ,LOAD7;COM-File!
CALL AHEX;Blocknummer
CALL OSTR ;fehlerhaft
DB ' ',19H,LF,0
BIT 0,(IX+7)
JR Z,LOAD7 ;wenn VERIFY
CALL OSTR
DB 'REPEAT (N)?',LF,CR,0
RES 6,(IX+8)
CALL KBD
CP 'N'
JR NZ,LOAD13
LD (IX+4),2;kein Start
JR LOAD10
;
LOAD15 INC A
JR NZ,LOAD12
CSRI CALL CSROI ;Abschlu~
CALL CRLF
BIT 0,(IX+7)
RET Z ;VERIFY
LD A,(IX+4)
LD (ARGN),A
CP 3 ;Autostart?
LD (IX+4),2
RET C ;nein
LD HL,(ARG3)
JP (HL) ;Start
```

```
DW 7F7FH
DB 'COLOR',1
COLRLD A,1FH
AND L
RLA
RLA
RLA
LD L,A
LD A,(ARGN)
CP 2
JR NC,COLR2
LD A,(COLOR)
COLR1 AND 7
OR L
LD (COLOR),A
RET
COLR2 LD A,E
JR COLR1
```

SUTB;Unterprogrammtabelle

```
DW CRT ;00
DW MBO ;01
DW UOUT1 ;02
DW UOUT2 ;03
DW KBD ;04
DW MBI ;05
DW UIN1 ;06
DW UIN2 ;07
DW ISRO ;08
DW CSRO ;09
DW ISRI ;0A
DW CSRI ;0B
DW KBDS ;0C
DW BYE ;0D
DW KBDZ ;0E
DW COLR;0F
DW LOAD ;10
DW VERIF ;11
DW LOOP ;12
DW NORM ;13
DW WAIT;14
DW LARG;15
DW INTB ;16
DW INLIN ;17
DW RHEX ;18
DW ERRM ;19
DW HLHX;1A
DW HLDE;1B
DW AHEX;1C
DW ZSUCH ;1D
DW SOUT ;1E
DW SIN ;1F
DW NOUT ;20
DW NIN ;21
DW GARG ;22
DW OSTR ;23
DW OCHR ;24
DW CUCP;25
DW MODU ;26
DW JUMP;27
```

```
DW LDMA ;28
DW LDAM ;29
DW BRKT;2A
DW SPACE ;2B
DW CRLF;2C
DW HOME ;2D
DW MODI ;2E
DW PUDE;2F
DW PUSE;30
DW SIXD;31
DW DABR ;32
DW TCIF ;33
DW PADR ;34
DW TON ;35
DW SAVE;36
DW MBIN;37
DW MBOUT ;38
DW KEY ;39
DW KEYLI ;3A
DW DISP;3B
DW WININ ;3C
DW WINAK ;3D
DW LINE ;3E
DW CIRCL ;3F
DW SQR ;40
DW MULT ;41
DW CSTBT ;42
DW INIEA ;43
DW INIME ;44
DW ZKOUT ;45
DW MENU ;46

LDMA LD M,A ;**28**
RET

LDAM LD A,M ;**29**
RET

DW 7F7FH
DB 'MODIFY',1
MODI0 CALL LARG
MODI LD A,0BFH ;**2E**
CP H
RET C ;ab C000h gesp.
LD (ARG1),HL
CALL HLHX;Adresse
LD A,M
CALL AHEX;Datenbyte
CALL OSTR
DB ',8,8,8,0'
MODI1 CALL INLIN
CALL RHEX ;Adresse
JR C,MODI9 ;Fehler
LD A,M
OR A
JR Z,MODI9 ;kein Zeichen
INC HL
PUSH DE
LD E,M
INC HL
LD D,M
EX DE,HL ;HL=Adresse
POP DE
LD B,0
MODI2 PUSH HL
CALL RHEX ;Datenbyte
LD A,M
OR A
JR Z,MODI4 ;kein Zeichen
INC HL
LD A,M
POP HL
MODI3 LD M,A ;eintragen
INC B
INC HL
PUSH HL
MODI4 POP HL
LD A,(DE)
CP ':' ;Abschlu~?
RET Z
CP ':' ;1 Zeichen
JR NZ,MODI5
INC DE
LD A,(DE) ;vom Video-RAM
INC DE
JR MODI3 ;eintragen
;
MODI5 LD A,B
OR A
JR NZ,MODI6
INC HL
MODI6 LD A,(DE)
CP ':' ;R}ckschritt?
JR NZ,MODI7
```

```

    DEC HL
    DEC HL
    JR    MODI
;
MODI7  CP    '/' ;neue Adresse?
    JR    NZ,MODI8
    INC DE
    CALL    RHEX ;Adresse
    JR    C,MODI9 ;Fehler
    LD HL,(NUMVX)
    JR    MODI
;
MODI8  OR    A ;Ende-Dummy?
    JR    Z,MODI
    CP    ''
    JR    Z,MODI2 ;noch ein Byte
MODI9  CALL    ERRM
    JR    MODI0

JPTAB  ;Sprungtabelle f}r CRT-Routinen
    DW CLR ;01
    DW CAPS;16
    DW CLICK ;14
    DW BEEP;07
    DW HCOPY ;0F
    DW CBL ;0D
    DW CLS ;0C
    DW CUL ;08
    DW DEL ;1F
    DW CUD ;0A
    DW INS ;1A
    DW CBL ;19
    DW CLLN;02
    DW CUU ;0B
    DW CUR ;09
    DW HOMEPG ;10
    DW SCROL ;12
    DW PAGE;11

CCTAB: ;CCTL-Tabelle
;Aufbau: Zeichen, Offset, CCTL
    DB 0E0H,0C0H,CCTL3
    DB 0A0H,0A0H,CCTL2
    DB 080H,080H,CCTL3
    DB 060H,040H,CCTL1
    DB 020H,020H,CCTL0
    DB 000H,000H,CCTL1

TCIF ;Test Cursor im Fenster    **33**
    LD A,(WINLG)
    DEC A
    SUB E ;Cursor-Spalte
    RET C
    LD A,(WINLG+1)
    DEC A
    SUB D ;Cursor-Zeile
    RET

CRT  PUSH    HL ;**00**
    PUSH    DE
    PUSH    BC
    PUSH    AF
    PUSH    AF
    LD A,(STBT)
    LD DE,(CURSO)
    LD L,A
    POP AF
    CP 20H ;Steuerfunktion?
    JR    NC,CRT1
    BIT 3,L ;darstellen?
    JR    NZ,CRT1
    LD HL,(CTAB)
    LD BC,18
    CPIR ;Funktion def.?
    JR    NZ,CRT3;nein
    LD HL,JPTAB
    ADD HL,BC
    ADD HL,BC
    LD C,M
    INC HL
    LD H,M
    LD L,C ;HL=Sprungadr.
    CALL    JPHL
    JR    CRT2
;
CRT1  CALL    PCHR
CRT2  LD    (CURSO),DE
CRT3  POP AF
    JP    POP3
;
PCHR  PUSH    AF
    CALL    DABR
    POP AF
    LD M,A ;ASCII eintragen
```

```
CALL WPIX; darstellen
CUR INC E
LD A,(WINLG)
CP E
RET NZ
LD E,0
CUD INC D
LD A,(WINLG+1)
CP D
RET NZ
LD HL,(WEND)
JPHL JP (HL) ;PAGE o. SCROLL

HCOPIY LD HL,(HCADR) ;ShCLR
JP (HL)

SCRLPG LD DE,0 ;Cursor li/oben
CALL DABR
INC D
PUSH HL
CALL DABR ;n{chste Zeile
POP DE
JR C,SCRL9
LD A,(WINLG) ;Spalten
LD C,A
LD A,(WINLG+1); Zeilen
DEC A
LD B,0
SCRL1 PUSH HL
PUSH BC
LDIR ;im VRAM eine
POP BC ;Zeile rollen
POP DE
LD HL,40 ;1 Zeile tiefer
ADD HL,DE
DEC A
JR NZ,SCRL1
LD A,(WINON+1)
RLCA
RLCA
RLCA
LD H,A ;Zeile*8
LD A,(WINLG+1)
DEC A
RLCA
RLCA
RLCA ;L{nge*8
PUSH AF
LD A,(WINON);Spalte
CP 32
LD C,A
LD A,(WINLG);Breite
JR NC,SCRL2 ;rechter Teil
ADD C
CP 32
JR C,SCRL3
LD B,32
SUB B
LD D,C
LD C,A
LD A,B
JR SCRL4
SCRL2 LD B,C
LD C,A
LD DE,0
JR SCRL5
SCRL3 LD D,C
LD BC,0
SCRL4 SUB D
LD E,A
SCRL5 POP AF
SCRL6 PUSH AF
LD A,E
AND A
JR Z,SCRL7
LD L,D
CALL SCRLUP
SCRL7 LD A,C
AND A
JR Z,SCRL8
LD L,B
CALL SCRLUP
SCRL8 INC H
POP AF
DEC A
JR NZ,SCRL6
SCRL9 LD A,(WINLG+1)
DEC A
LD D,A
JR CLLN

CLR PUSH DE
CALL DABR
LD A,' ' ;Leerzeichen
```

```
LD M,A
CALL WPIX
POP DE
CUL LD A,E ;Cursor links
SUB 1
JR C,CUL1
LD E,A
RET
CUL1 LD A,D ;Zeile hoch
SUB 1
RET C
LD D,A
LD A,(WINLG)
DEC A
LD E,A
RET

CLLN LD E,0 ;ShDEL
PUSH DE
CALL DABR
LD A,(WINLG)
LD B,A ;Spalten
CLLN1 LD M,0 ;VRAM l|schen
LD A,' '
CALL WPIX;Space anzeigen
INC HL
DJNZ CLLN1
POP DE
RET

CLS LD D,0 ;ShHOME
LD A,(WINLG+1)
CLS1 PUSH AF
PUSH DE
CALL CLLN ;Zeile f|r
POP DE ;Zeile l|schen
POP AF
INC D
DEC A
JR NZ,CLS1
HOMEPG LD D,0
CBL LD E,0
RET

CUU LD A,D
SUB 1
RET C
LD D,A
RET

SCROL LD HL,SCRLPG ;ShCUD
WADR LD (WEND),HL
RET

PAGE LD HL,HOMEPG ;ShCUU
JR WADR

DEL CALL DABR
LD A,M
AND A ;Ende?
RET Z
PUSH DE ;Cursor
PUSH HL ;VRAM
DEL1 INC E ;n{chste Spalte
CALL DABR ;raus?
JR NC,DEL2 ;nein
LD E,0 ;Spalte 0 in
INC D ;neuer Zeile
CALL DABR ;raus?
JR C,DEL3 ;ja, Ende
DEL2 LD A,M
AND A ;Ende?
JR Z,DEL3
EX (SP),HL
LD M,A ;eintragen
CALL WPIX;zeichnen
JR DEL1 ;von vorn
DEL3 POP HL
LD M,0 ;letztes CHR
LD A,' ' ;l|schen
CALL WPIX
POP DE
RET

INS PUSH DE ;Cursor
LD A,' '
CALL DABR
INS1 LD B,M ;altes
LD M,A ;neues Zeichen
CALL WPIX;zeichnen
LD A,B
AND A ;Dummy?
JR Z,INS2
INC E ;CUR
```

```
CALL DABR ;raus?
JR NC,INS1 ;nein
LD E,0 ;Spalte 0 in
INC D ;neuer Zeile
CALL DABR ;raus?
JR NC,INS1 ;nein
POP DE
RET
;
INS2 PUSH HL
INC E ;CUR
CALL DABR ;raus?
JR NC,INS3 ;nein
LD E,0
INC D
CALL DABR
JR C,DEL3
INS3 LD A,M
AND A
JR NZ,DEL3
POP HL
POP DE
RET

CLICK LD A,(IX+8) ;ShINS
XOR 20H
XOR8 LD (IX+8),A
RET

CAPS LD A,(IX+8)
XOR 80H
JR XOR8
```

```
CRTTAB ;Bildschirmsteuercodes
DB 11H ;PAGE
DB 12H ;SCROL
DB 10H ;HOME
DB 09H ;CUR
DB 0BH ;CUU
DB 02H ;CLLN
DB 19H ;CBL
DB 1AH ;INS
DB 0AH ;CUD
DB 1FH ;DEL
DB 08H ;CUL
DB 0CH ;CLS
DB 0DH ;ENTER
DB 0FH ;HCOPY
DB 07H ;BEEP
DB 14H ;CLICK
DB 16H ;SHLOK
DB 01H ;CLR
```

```
FADR;Berechnet Farbadresse
;PE: HL=VRAM-Adresse
;PA: HL=Farbadresse
LD BC,-VRAM
ADD HL,BC
LD B,0
LD DE,40 ;Zeilenbreite
FADR1 INC B
XOR A
SBC HL,DE
JR NC,FADR1
ADD HL,DE
ADD B
DEC A
RLA
RLA
RLA
LD H,A
RET
```

```
BASPV ;BASIC-I/O-Verteiler
PUSH HL
PUSH BC
CALL IRMON
BIT 5,E
PUSH DE
JR NZ,BASBYE
INC HL
INC HL
BIT 7,E
JR NZ,BSA2
PUSH HL
LD A,E
AND 7
LD HL,BUPTAB
ADD L
LD L,A
LD A,D
LD D,E
LD E,M ;UP-Nummer
POP HL
```

```
CALL PV3 ;Aufruf
POP DE
LD D,A
LD A,E
AND 4FH
XOR 43H
JR NZ,BSA1
CALL OSTR
DB 'VERIFY ?(Y):',0
CALL KBD
CP 'Y'
CALL Z,VERIF
BSA1LD A,D
RES 3,E
CALL IRMOF
POP BC
POP HL
RET

BUPTAB ;H-Teil konstant!
DB 16H ;INTB #0
DB 24H ;OCHR
DB 37H ;MBIN #1
DB 38H ;MBOUT
DB 06H ;USIN1 #2
DB 02H ;UOT1
DB 07H ;USIN2 #3
DB 03H ;UOUT2

BSA2CALL KBDS
POP DE
LD D,A
JR NC,BSA1
RES 7,E
JR BSA1

BASBYE LD A,(35EH)
AND A
JR Z,BASB1
CALL IRMOF
CALL 0C641H ;NEW
CALL IRMON
BASB1 JP LOOP

MBOUT ;Byteweise MB-Ausgabe ;**38**
LD E,D ;Steuerbyte
LD D,A ;Datenbyte
PUSH DE
PUSH BC
BIT 3,E
JR Z,NOINIT
SET 1,(IX+7);Init merken
LD A,'U'+80H
CP M
PUSH DE
LD DE,0A0H ;Vortonl{nge
JR NZ,NOHIU;bei 'U'
LD DE,400H ;gedehnt
NOHIU LD (VORTN),DE
LD DE,CASS
LD BC,11
LDIR ;Name
EX DE,HL
POP DE
LD M,D ;1. Byte
LD A,74H ;noch Bytes
MBOUT1 CALL SETDT
MBOUTE POP BC
POP DE
RES 3,E
RET

NOINIT LD HL,(DTADR)
LD M,D ;Byte eintragen
BIT 6,E
JR NZ,MCLOS
LD A,(DTNR)
DEC A
JR NZ,MBOUT1
BIT 1,(IX+7);schon Init?
JR NZ,MBOUT2;nein
CALL MBLO ;ja-Blockout
CALL SETDTN
JR MBOUTE

MBOUT2 RES 1,(IX+7);r}cksetzen
CALL ISRO ;Init
CALL MBLNR ;Blocknr.
CALL SETDTN
JMBOUTE JR MBOUTE

MCLOS BIT 1,(IX+7);Init?
JR Z,MBOUT3
CALL ISRO ;aber jetzt!
```

```
JR      MBOU4

MBOU3  CALL      MBLO      ;Blockout
MBOU4  LD      BC,(VORTN)
        CALL      CSRO
MBOU5  PUSH      DE
        CALL      CSRI
        RES 5,(IX+7);'U'-Merker
        POP DE      ;r}cksetzen
JR      JMBOU4E

MBIN;Byteweise MB-Eingabe      ;**37**
LD      E,D      ;Steuerbyte
LD      D,A      ;Datenbyte (PA!)
        PUSH      DE
        PUSH      BC
        BIT 6,E      ;Close?
JR      NZ,MBOU5
        BIT 3,E      ;Init?
JR      Z,MBIU
LD      A,1      ;erwarteter Blk
LD      (IX+3),A
        PUSH      HL
        CALL      ISRI
MBIN1  POP      HL
        JP      C,JBLERR;Fehler
LD      A,(IX+2);gelesener Blk.
CP      1
        JP      NZ,JBLERR;Fehler
RES 2,(IX+7);Kopfblk geladen
        INC (IX+3)      ;erw. Block
LD      DE,CASS
LD      B,11
LD      A,(DE)
CP      'U'+80H      ;hohes 'U'?
JR      NZ,MBIN2
SET 5,(IX+7);Merker zur
;Blocknummernausgabeunterdr}ckung
MBIN2  ADD 29H      ;gesch}tzte
JR      NC,MBIN3;BASIC-
LD      A,(DE)      ;Programme
        SUB 4      ;als unge-
LD      (DE),A      ;sch}tzt
        INC DE      ;anzeigen
LD      (DE),A      ;und Setzen
        INC DE      ;LIST/EDIT/BYE-
LD      (DE),A      ;Schutz
LD      (35EH),A;<=hier
SET 6,(IX+7);RESET-Schutz
LD      DE,CASS
MBIN3  LD      A,(DE)
        CALL      MBCRT      ;anzeigen
LD      A,(DE)
CP      M      ;stimmt Name?
JR      Z,MBIN4
SET 2,(IX+7);nein-merken
MBIN4  INC      HL
        INC DE
        DJNZ MBIN3
LD      A,LF
        CALL      MBCRT
LD      A,CR
        CALL      MBCRT
        BIT 2,(IX+7)
        JP      NZ,JIOERR ;?IO ERROR
EX      DE,HL
LD      A,75H      ;noch Bytes
MBIN5  LD      D,M      ;auszulesen
        CALL      SETDT
LD      A,D
        POP BC
        POP DE
RES 3,E
        RET

MBIU1  LD      HL,(DTADR)
LD      A,(DTNR)
        DEC A
JR      NZ,MBIN5
MBIU1  CALL      MBI
JR      C,MERR
        CALL      SETDTN
LD      A,0FFH
CP      (IX+2)      ;Block FF?
JR      Z,MBIU3
LD      A,(IX+3)
CP      (IX+2)      ;erwarteter?
JR      Z,MBIU3
LD      A,'*'
MBIU2  CALL      MBCRT      ;nicht
        CALL      BLNR;erwartet
        CALL      JBRKT
JR      MBIU1
```



```
MBIU3      INC (IX+3)      ;n{chsten
          DEC A            ;erwarteten
          BIT 5,(IX+7)
          JR  NZ,MBIU4
          CALL BLNR;Blocknr.
          LD  A,'>'
          CALL MBCRT
          LD  A,' '
          CALL MBCRT
MBIU4      CALL SETDTN
          LD  HL,(DTADR)
          JR  MBIN5

MERR      LD  A,'?' ;Fehler in Block
          JR  MBIU2

JBLERR    LD  A,'*' ;nicht erw. Blk.
          CALL MBCRT
          CALL JBRKT
          CALL MBI
          PUSH HL
          JP  MBIN1

BLNR      LD  A,(IX+2)
          PUSH AF          ;siehe AHEx!
          RRA
          RRA
          RRA
          RRA
          CALL BLNR1
          POP AF
BLNR1     AND 0FH
          ADD '0'
          CP  '9'+1
          JR  C,BLNR2
          ADD 7
BLNR2     CALL CRT
          RET

          PUSH AF          ;Space-Ausgabe
          LD  A,' ' ;nicht benutzt!
CRTA      CALL CRT
          POP AF
          RET

MBCRT     PUSH AF
          JR  CRTA

MBLO      LD  BC,(VORTN)
          CALL MBO
MBLNR     CALL BLNR;AHEx
          LD  A,'>'
          CALL MBCRT
          LD  A,' '
          CALL MBCRT
JBRKT     CALL BRKT;BRK?
          RET NC
JIOERR    LD  HL,(IOERR)
          JP  (HL)

SETDTN    ;Setzen Datenzeiger neu
          LD  A,80H
          LD  HL,CASS
          JR  SETDT1

SETDT     ;n{chsten Datenzeiger setzen
          INC HL
SETDT1    LD  (DTADR),HL
          LD  (DTNR),A
          RET

          DW  7F7FH
          DB  'DISPLAY',1
DISP      LD  A,(ARGN) ;**3B**
          CP  3
          JR  NC,DISP1
          LD  C,8 ;default
DISP1     PUSH BC
DISP2     CALL HLHX;Adresse
          LD  A,0BFH
          CP  H
          JR  C,DISP6 ;C000H...
          PUSH HL
          LD  B,8 ;8 Byte
DISP3     LD  A,M
          CALL AHEx;hex anzeigen
          DEC B
          JR  Z,DISP4
          CALL SPACE
          INC HL
          JR  DISP3
DISP4     LD  A,9 ;CUR
          CALL OCHR
```

```
CALL CSTBT
POP HL
LD B,8 ;8 Byte
DISP5 LD A,M
CALL OCHR ;ASCII anzeigen
INC HL
DJNZ DISP5
CALL CSTBT
CALL CRLF
CALL BRKT
JR C,DISP6 ;BRK
PUSH DE
EX DE,HL
SBC HL,DE ;EAdr erreicht?
EX DE,HL
POP DE
JR C,DISP6
DEC C
JR NZ,DISP2
CALL KBD ;Warten Taste
CP 3 ;BRK?
POP BC
RET Z
CP 13H ;STOP?
JR NZ,DISP1
CALL OSTR
DB 0BH,9,9,9,9,9,0
JP MODI1 ;-> MODIFY
```

```
DISP6 POP BC
RET
```

```
CSTBT PUSH AF
LD A,(STBT)
XOR 8
LD (STBT),A
POP AF
RET
```

```
KEYUP LD HL,0B900H
LD B,A
KUP1 LD A,M
AND A ;Dummys
INC HL ;ausz{hlen
JR NZ,KUP1
DJNZ KUP1
KUP2 LD A,M
AND A ;Ende-Dummy?
RET Z
INC HL
CALL CSTBT
CALL OCHR ;anzeigen
CALL CSTBT
JR KUP2
```

```
DW 7F7FH
DB 'KEYLIST',1
KEYLI LD C,1 ;**39**
KEYL1 LD A,'F'
CALL OCHR ;F
LD A,C
CALL AHEX0 ;Nr.
CALL OSTR
DB ': ',0
LD A,C
CALL KEYUP ;pr{sentieren
CALL CRLF
INC C
LD A,13
CP C ;F1-FC
RET Z
JR KEYL1
```

```
DW 7F7FH
DB 'KEY',1
LD A,L
KEY AND A ;**3A*
RET Z ;0
CP 13
RET NC ;>12
CALL KEYUP ;pr{sentieren
LD DE,0B99BH
EX DE,HL
AND A
SBC HL,DE
RET C
EX DE,HL
LD C,E ;BC=L{nge bis
LD B,D ;Pufferende
KEY1 CALL KBD
AND A
JR Z,KEY1 ;keine Taste
CP 13H
JP Z,CRLF ;STOP-Ende
```

```
CP      1
JR      Z,KEY2      ;CLR
PUSH    HL
LD      HL,0B99AH;vorletztes
INC     M      ;Zeichen 0?
DEC     M
JR      NZ,KEY3;nein-voll!
EX      (SP),HL
CALL    CSTBT
POP     HL
LD      D,H
LD      E,L
INC     DE
PUSH    BC      ;Pufferrest
LDDR    ;hinterziehen
POP     BC
INC     HL
LD      M,A      ;eintragen
INC     HL      ;und
CALL    CRT      ;anzeigen
CALL    CSTBT
DEC     BC      ;L{nge-1
JR      KEY1
;
KEY2    DEC     HL
LD      A,M
AND     A      ;Leere Belegung
INC     HL      ;nicht l|schen!
JR      Z,KEY1
DEC     HL
PUSH    HL
INC     HL
INC     BC      ;L{nge+1
PUSH    BC
LD      D,H
LD      E,L
DEC     DE      ;Pufferrest
LDIR    ;vorziehen
POP     BC
CALL    OSTR
DB      8,' ',8,0
KEY3    POP     HL
JR      KEY1

WININ   CP      10      ;**3C**
RET     NC      ;Nr. zu gro~
LD      B,A
LD      A,L
CP      40
RET     NC      ;Spalte zu gro~
ADD     E
CP      41
RET     NC      ;Ende zu gro~
LD      A,H
CP      32
RET     NC      ;Zeile zu gro~
ADD     D
CP      33
RET     NC      ;Ende zu gro~
PUSH    BC
PUSH    HL
PUSH    DE
LD      DE,WNNR
CALL    WIN1
EX      DE,HL
LDIR    ;Daten kopieren
POP     HL
LD      (WINLG),HL
POP     HL
LD      (WINON),HL
POP     AF
LD      (WINNR),A
XOR     A
LD      HL,CURSO
LD      M,A      ;Spalte=0
INC     HL
LD      M,A      ;Zeile=0
INC     HL
LD      M,A      ;STBT=0
INC     HL
LD      M,39H      ;COLOR
INC     HL
LD      M,SCRLPG
INC     HL      ;WEND
LD      M,SCRLPG/256
SCF
RET

WIN1    ;Adresse von Fenstervektor ber.
LD      A,(DE)      ;Nummer
WIN2    LD      HL,WNDFN-10
LD      BC,10      ;L{nge Vektor
INC     A
WIN3    ADD     HL,BC
```

```
DEC A
JR NZ,WIN3
INC DE ;DE=WINON
RET

WINAK CP 10 ;**3D**
RET NC ;zu gro~
PUSH AF
LD DE,WINNR
PUSH DE
CALL WIN1
EX DE,HL ;akt. Vektor
LDIR ;retten
POP DE
POP AF
LD (DE),A
CALL WIN2 ;neuen Vektor
LDIR ;holen
SCF
RET

LINE ;Linie zeichnen ;**3E**
; von (ARG1)/(ARG2) nach (ARG3)/(ARG4)
LD HL,(ARG3) ;XEND
LD BC,(ARG1) ;XANF
LD (HOR),BC ;X-Start
AND A
SBC HL,BC
LD BC,0
JR NC,LINE1;wenn XANF>=XEND
DEC BC ;r}ckl{ufig
LINE1 LD (XDIR),BC
LD B,7
LINE2 SLA L ;XSTEPS
RL H ;*128
DJNZ LINE2
LD (XCNT),HL
LD (XSTEP),HL
LD A,(ARG4) ;YEND
LD L,A
LD A,(ARG2) ;YANF
LD (VERT),A ;Y-Start
LD C,A
LD H,0
LD B,H
XOR A
SBC HL,BC
JR NC,LINE3
DEC A ;r}ckl{ufig
LINE3 LD (YDIR),A
LD B,7
LINE4 SLA L ;YSTEPS
RL H ;*128
DJNZ LINE4
LD (YSTEP),HL
LD (YCNT),HL
LD BC,200H
;LINE-Hauptschleife
LINE5 CALL PUSE ;Punkt setzen
LD HL,(XCNT)
LD DE,(XSTEP)
ADD HL,DE
LD (XCNT),HL
LD HL,(HOR)
LD DE,(XDIR)
ADC HL,DE
LD (HOR),HL;X neu
LD HL,(YCNT)
LD DE,(YSTEP)
ADD HL,DE
LD (YCNT),HL
LD A,(VERT);Y neu
LD E,A
LD A,(YDIR)
ADC E
LD (VERT),A
DEC BC
LD A,B
OR C
JR NZ,LINE5
RET

CIRCL ;Kreis zeichnen ;**3F**
;um (ARG1)/(ARG2)
LD A,(ARG3);Radius
LD C,A
LD D,182 ;1/SQR(2) * 256
CALL MULT
LD A,B
LD (YCNT),A
LD D,C
CALL MULT
LD C,A
LD (XCNT),BC
```

```

;CIRCLE-Hauptschleife
CIR1 LD HL,(XCNT)
LD A,(YCNT)
LD C,A
LD D,A
PUSH DE ;Achtelkreis
CALL MULT ;berechnen
LD C,A
AND A
SBC HL,BC
CALL SQR
POP DE
LD E,A
AND A ;und jeweils 8
CIR2 CCF ;Punkte setzen
PUSH AF
LD HL,(ARG1)
LD B,0
LD C,D
ADD HL,BC
LD (HOR),HL
LD A,(ARG2)
PUSH AF
ADD E
LD (VERT),A
CALL PUSE;1. Punkt
POP AF
PUSH AF
SUB E
LD (VERT),A
CALL PUSE;2. Punkt
SBC HL,BC
SBC HL,BC
LD (HOR),HL
CALL PUSE;3. Punkt
POP AF
ADD E
LD (VERT),A
CALL PUSE;4. Punkt
LD A,E
LD E,D ;D/E vertauscht
LD D,A
POP AF
JR C,CIR2 ;5. bis 8. Punkt
LD A,(YCNT)
SUB 1
LD (YCNT),A
JR NC,CIR1 ;weiter
RET

SQR ;A=SQR(HL) ;**40**
XOR A
LD A,0FFH
LD E,A
LD D,A
SQR1 INC A
INC DE
INC DE
SBC HL,DE
JP P,SQR1
RET

MULT ;BA=C*D ;**41**
LD B,0
LD E,9
MUL1 LD A,D
RRA
DEC E
RET Z
LD D,A
LD A,B
JR NC,MUL2
ADD C
MUL2 RRA
LD B,A
JR MUL1 ;8*

SCRLUP PUSH HL ;Scroll-UP
PUSH DE
PUSH BC
PUSH HL
CALL PADR
POP BC
PUSH HL ;Pixel-neu
PUSH DE ;Farbe-neu
LD L,A
LD A,8
ADD B
LD H,A
LD A,L
LD L,C
CALL PADR
EX (SP),HL ;Pixel-alt
EX DE,HL ;Farbe-alt
```

```
LD C,A
LD B,0
PUSH BC ;Breite
LDIR ;Farbscrolling
POP BC
POP HL
POP DE
LDIR ;Pixelscrolling
JP POP3

CUCP PUSH HL ;**25**
PUSH DE
PUSH BC
PUSH AF
LD DE,(CURSO)
CALL DABR
JR C,CUCP2 ;au~erhalb!
CALL FADR
CALL PADR
LD A,(DE)
CPL ;1. Farbbyte
LD (DE),A
SET 5,E
LD A,(DE)
CPL ;2. Farbbyte
BIT 7,(IX+8);CAPS?
JR NZ,CUCP1
XOR 4

CUCP1 LD (DE),A
CUCP2 JP POP4

BEEP LD HL,30H ;CHR(7)
LD BC,0FH
PUSH DE
LD E,H ;E=0
CALL TON1
LD A,1EH ;Tondauer
CALL WAIT
LD A,3
OUT 8CH ;Ton aus
LD A,10H
POP DE ;Cursor
JP WAIT;Tonpause

ZEI0 DB 0 ;CRT
DB 38H ;MBOUT
DB 2 ;UOUT1
DB 3 ;UOUT2
ZEI4 DB 4 ;KBD
DB 37H ;MBIN
DB 6 ;UIN1
DB 7 ;UIN2

END
```