```
                        C       INCLUDE CPORTS.INC
0028                    C       ERRORR  EQU     28H
05DC                    C       DELAY   EQU     1500
0006                    C       SNDPTO  EQU     6;+192
0003                    C       SNDPTI  EQU     3;+192
0006                    C       SENSE2  EQU     6;+192
0005                    C       DRIVE   EQU     5;+192
0008                    C       PORT    EQU      8;+192
0003                    C       CASPORT EQU      3;+192
FFF0                    C       IJTABLE EQU      0FFF0H
001F                    C       CASonOFF EQU     1FH     ; Change to true value..
                                ;
0028                            ERRRST  EQU     28H
0008                            DEHL    EQU     8
0010                            SCRRST  EQU     10H
                                ;
                                ;
                                ;********** START OF CASSETTE INTERFACE **********
                                ;
                                ;
                                ;Interrupt routine - toggles carry flag
                                ;
0A53    3F                      TOGGLE: CCF
0A54    FB                              EI
0A55    ED 4D                           RETI
                                ;
                                ;
0A57                            BLIP:
0A57    DB 06                           IN A,(SENSE2)
0A59    CB 47                           BIT 0,A
0A5B    20 05                           JR NZ,BLIP2             ;Jump if break key not pressed
0A5D    CD 0B06                         CALL RESint
0A60    EF                              RST ERRRST
0A61    3E                              DB 14+48
0A62    08                      BLIP2:  EX AF,AF'
0A63    EE 0F                           XOR 0FH
0A65    D3 06                           OUT (SNDPTO),A          ;Makes a noise
0A67    08                              EX AF,AF'
0A68    DB 03                           IN A,(SNDPTI)
0A6A    C9                              RET
                                ;
                                ;
0A6B                            INBIT::
0A6B    CD 0A57                         CALL BLIP
0A6E    FB                              EI
0A6F    AF                              XOR A                   ;Carry flag cleared
0A70    30 FE                   INBIT1: JR NC,INBIT1
0A72    3D                      INBIT2: DEC A
0A73    DA 0A72                         JP C,INBIT2             ;A bit faster
0A76    F3                              DI
0A77    DD BE 00                        CP (IX+0)               ;IX = MIDVAL
0A7A    C9                              RET
                                ;
                                ;
0A7B                            OUTBIT::
0A7B    CD 0A57                         CALL BLIP               ;Preserves carry flag
```

```
0A7E  3E 00                   LD A,0              ;Preserves carry flag
0A80  38 0A                   JR C,HIGH
                              ;
                              ;
                              ;
                              ;Both LOW and HIGH entered with A = 0
                              ;
0A82  30 FE                   LOW::   JR NC,LOW
0A84  D3 03                           OUT (CASPORT),A
0A86  38 FE                   LOW1:   JR C,LOW1
0A88  3C                              INC A
0A89  D3 03                           OUT (CASPORT),A
0A8B  C9                              RET
                              ;
                              ;
                              ;
0A8C  38 FE                   HIGH::  JR C,HIGH
0A8E  30 FE                   HIGH1:  JR NC,HIGH1
0A90  D3 03                           OUT (CASPORT),A
0A92  38 FE                   HIGH2:  JR C,HIGH2
0A94  30 FE                   HIGH3:  JR NC,HIGH3
0A96  3C                              INC A
0A97  D3 03                           OUT (CASPORT),A
0A99  C9                              RET
                              ;
                              ;
                              ;
0A9A                          INBYTE::
0A9A  06 08                           LD B,8
0A9C  CD 0A6B                 INBY1:  CALL INBIT
0A9F  CB 19                           RR C
0AA1  10 F9                           DJNZ INBY1
0AA3  C9                              RET
                              ;
                              ;
                              ;
0AA4                          OUTBYTE::
0AA4  06 08                           LD B,8
0AA6  CB 19                   OUTBY1: RR C
0AA8  CD 0A7B                         CALL OUTBIT
0AAB  10 F9                           DJNZ OUTBY1
0AAD  C9                              RET
                              ;
                              ;
                              ;
0AAE  7A                      INOUT:: LD A,D
0AAF  B3                              OR E
0AB0  C8                              RET Z               ;Return if size of block = 0
0AB1  CD 0B10                         CALL SETint
0AB4  08                              EX AF,AF'
0AB5  3E 90                           LD A,128+16
0AB7  08                              EX AF,AF'
0AB8  28 2C                           JR Z,OUTBLOCK       ;Jump if saving
                              ;
                              ;
                              ;
```

```
0ABA                    INBLOCK::
0ABA  06 00                     LD B,0
0ABC  CD 0A6B           INBLK1: CALL INBIT
0ABF  38 F9                     JR C,INBLOCK
0AC1  10 F9                     DJNZ INBLK1         ;Look for 50 low bits
0AC3  FB                STBIT:  EI
0AC4  AF                        XOR A
0AC5  3F                        CCF                 ;Set carry
0AC6  CD 0A72                   CALL INBIT2         ;Find long pulse
0AC9  30 F8                     JR  NC,STBIT
0ACB                    INBLK2::
0ACB  CD 0A9A                   CALL INBYTE         ;C = byte from tape
0ACE  3A FD67                   LD A,(VERIF)
0AD1  A7                        AND A
0AD2  28 09                     JR Z,INBLK3         ;Jump if loading
0AD4  7E                        LD A,(HL)           ;Here if verifying
0AD5  B9                        CP C
0AD6  28 06                     JR Z,INBLK4         ;Jump if good verify
0AD8  CD 0B06                   CALL RESint         ;Turn off interrupts
0ADB  EF                        RST ERRRST
0ADC  0B                        DB 11               ;"Mismatch"
0ADD  71                INBLK3: LD (HL),C           ;Store byte
0ADE  23                INBLK4: INC HL
0ADF  1B                        DEC DE
0AE0  7A                        LD A,D
0AE1  B3                        OR E
0AE2  20 E7                     JR NZ,INBLK2
0AE4  18 20                     JR RESint
                        ;
                        ;
                        ;
0AE6                    OUTBLOCK::
0AE6  01 05DC                   LD BC,DELAY
0AE9  AF                OTBLK1: XOR A
0AEA  CD 0A82                   CALL LOW
0AED  0B                        DEC BC
0AEE  78                        LD A,B
0AEF  B1                        OR C
0AF0  20 F7                     JR NZ,OTBLK1        ;Carry flag reset, A = 0
0AF2  30 FE             DEL1:   JR NC,DEL1          ;Short high
0AF4  D3 03                     OUT (CASPORT),A     ;A = 0
0AF6  38 FE             DEL2:   JR C,DEL2           ;One more interrupt
0AF8  3F                        CCF
0AF9  CD 0A92                   CALL HIGH2
0AFC                    OTBLK2::
0AFC  4E                        LD C,(HL)           ;C = byte to be output
0AFD  CD 0AA4                   CALL OUTBYTE
0B00  23                        INC HL
0B01  1B                        DEC DE
0B02  7A                        LD A,D
0B03  B3                        OR E
0B04  20 F6                     JR NZ,OTBLK2
                        ;
                        ;
0B06                    RESint::
0B06  CD 097F                   CALL IJINIT
```

```
0B09  3E 55                   LD A,55H
0B0B  D3 1F                   OUT (CASonOFF),A
0B0D  C3 0996                 JP INT125
                      ;
                      ;
                      ;
                      ;Routine to set CTC interrupts going
                      ;Returns Z if saving, NZ if loading/verifying
                      ;
0B10                  SETint::
0B10  CD 097F                 CALL IJINIT
0B13  E5                      PUSH HL
0B14  21 0A53                 LD HL,TOGGLE
0B17  22 FFF2                 LD (IJTABLE+2),HL
0B1A  22 FFF6                 LD (IJTABLE+6),HL
0B1D  E1                      POP HL
0B1E  DD 21 FD60              LD IX,MIDVAL
0B22  3E FE                   LD A,0FEH
0B24  D3 05                   OUT (DRIVE),A
0B26  3E AA                   LD A,0AAH
0B28  D3 1F                   OUT (CASonOFF),A
0B2A  0E 0B                   LD C,PORT+3
0B2C  06 C5                   LD B,0C5H           ;Assume loading
0B2E  3A FD68                 LD A,(TYPE)
0B31  A7                      AND A
0B32  20 05                   JR NZ,SETin1        ;Jump if A = 1, i.e. loading
0B34  0E 09                   LD C,PORT+1
0B36  3A FD5F                 LD A,(CASBAUD)
0B39  ED 41           SETin1: OUT (C),B
0B3B  ED 79                   OUT (C),A
0B3D  FB                      EI
0B3E  ED 4D                   RETI                ;Clear interrupts and return
                      ;
                      ;
                      ;********** END OF CASSETTE INTERFACE **********
                      ;
                      ;
                      ;Main routine for loading or verifying a BASIC program
                      ;
0B40  F5              LOAD::  PUSH AF             ;Save load / verify flag
0B41  AF                      XOR A
0B42  32 FD67                 LD (VERIF),A
0B45  3C                      INC A
0B46  32 FD68                 LD (TYPE),A         ;(TYPE) = 1 for loading or verifying
0B49  CD 0BEF                 CALL SETsound       ;Set up sound, call FINFF
0B4C  E5                      PUSH HL
0B4D  D9                      EXX                 ;DE' -> oldname, HL' -> newname
0B4E  E1                      POP HL
0B4F  11 0012                 LD DE,18
0B52  F5              NO1file:PUSH AF
0B53  D5              NOfile: PUSH DE
0B54  E5                      PUSH HL
0B55  CD 0AAE                 CALL INOUT
0B58  E1                      POP HL
0B59  D1                      POP DE
0B5A  7E                      LD A,(HL)
```

```
0B5B  FE FF                   CP 0FFH
0B5D  20 F4                   JR NZ,NOfile
0B5F  CD 0C23                 CALL COMPname
0B62  08                      EX AF,AF'
0B63  F1                      POP AF
0B64  28 03                   JR Z,NULLfile
0B66  08                      EX AF,AF'
0B67  20 E9                   JR NZ,NOlfile
0B69                  NULLfile:
0B69  19                      ADD HL,DE
0B6A  2B                      DEC HL
0B6B  2B                      DEC HL
0B6C  CF                      RST DEHL
0B6D  F1                      POP AF              ;Restore load / verify flag
0B6E  32 FD67                 LD (VERIF),A
0B71  A7                      AND A
0B72  20 0D                   JR NZ,VVER
0B74  D7                      RST SCRRST
0B75  89                      DB 80H+9
0B76  4C 4F 41 44             DB 'LOADING',13,10
0B7A  49 4E 47 0D
0B7E  0A
0B7F  18 0D                   JR JOIN1
0B81  D7              VVER:   RST SCRRST
0B82  8B                      DB 80H+11
0B83  56 45 52 49             DB 'VERIFYING',13,10
0B87  46 59 49 4E
0B8B  47 0D 0A
0B8E  18 20           JOIN1:  JR JOIN
                      ;
                      ;
                      ;
                      ;Main routine for saving a BASIC program
                      ;
0B90  AF              SAVE::  XOR A
0B91  32 FD68                 LD (TYPE),A         ;(TYPE) = 0 for saving
0B94  CD 0BEF                 CALL SETsound       ;Set up sound, call FINFF
0B97  ED 4B FA92              LD BC,(STKLIM)
0B9B  C5                      PUSH BC
0B9C  71                      LD (HL),C
0B9D  23                      INC  HL
0B9E  70                      LD (HL),B
0B9F  23                      INC HL
0BA0  EB                      EX DE,HL
0BA1  11 0014                 LD DE,20            ;DE=Length of filename.
0BA4  CD 0AAE                 CALL INOUT
0BA7  11 0000                 LD DE,0
0BAA  1B              SDELAY: DEC DE              ;Wait for a while after saving name
0BAB  7A                      LD A,D
0BAC  B3                      OR E
0BAD  20 FB                   JR NZ,SDELAY
0BAF  D1                      POP DE
                      ;
                      ;
0BB0  21 FA7A         JOIN:   LD HL,LSTPG
0BB3  7E                      LD A,(HL)
```

```
0BB4  F5                  PUSH AF           ;Save (LSTP6)
0BB5  E5                  PUSH HL
                          ;
0BB6  2A FA94             LD HL,(SYSTOP)
0BB9  A7                  AND A
0BBA  ED 52               SBC HL,DE
0BBC  EB                  EX DE,HL
0BBD  CD 0AAE             CALL INOUT        ;Save/load/verify system variables
                          ;
0BC0  E1                  POP HL            ;HL = LSTP6
0BC1  46                  LD B,(HL)         ;B = old LSTP6
0BC2  F1                  POP AF
0BC3  77                  LD (HL),A         ;Put true value back into LSTP6
0BC4  C5                  PUSH BC           ;Save old LSTP6
                          ;
0BC5                      GETBASIC::
0BC5  ED 5B FACC          LD DE,(ARRTOP)
0BC9  3A FACE             LD A,(ARRTOP+2)
0BCC  47                  LD B,A
0BCD  2A FD65             LD HL,(VAZERO)
0BD0  0E 00               LD C,0
0BD2  D9                  EXX               ;B'D'E' = size of program,
                          ;                 ;C'H'L' = start of program.
                          ;
0BD3  CD 35C9             MOREbas:CALL BLOCK
0BD6  20 05               JR NZ,ENDbas
0BD8  CD 0AAE             CALL INOUT        ;Save/load/verify program
0BDB  18 F6               JR MOREbasic
                          ;
0BDD  C1                  ENDbas: POP BC    ;B = old LSTP6
0BDE  CD 0C4F             CALL ADJVAL       ;Sets up pointers if dif config.
0BE1  2A FA81             LD HL,(CALCST)
0BE4  ED 5B FA7B          LD DE,(VARNAM)
0BE8  A7                  AND A
0BE9  ED 52               SBC HL,DE
0BEB  EB                  EX DE,HL
0BEC  C3 0AAE             JP INOUT          ;Save/load/verify BASIC variables, and return
                          ;
                          ;
                          ;
0BEF                      SETsound:
0BEF  AF                  XOR A
0BF0  32 FE14             LD (CHAN),A
0BF3  CD 093A             CALL KILSND
0BF6  3E 01               LD A,1
0BF8  32 FE16             LD (FREQ),A
0BFB  CD 08C6             CALL OFREQ
                          ;
                          ;Fall through to FINFF
                          ;
                          ;
0BFE  CD 3FE9             FINFF:: CALL  FIND1$    ;DE = start of string, HL = end of string
0C01  F5                  PUSH AF           ;Save flags
0C02  C5                  PUSH BC           ;Save length of name
0C03  79                  LD A,C
0C04  FE 0E               CP 14             ;Maximum length of filename + 1
```

```
0C06  30 03                   JR NC,NOTok
0C08  A0                      AND B
0C09  28 06                   JR Z,LENok          ;Filename is not greater than  17
0C0B  01 000E           NOTok: LD BC,14
0C0E  D5                      PUSH DE
0C0F  E1                      POP HL
0C10  09                      ADD HL,BC
0C11  79                LENok: LD A,C
0C12  0E 20                   LD C,' '
0C14  3C                BLOOP: INC A
0C15  FE 0F                   CP 15
0C17  71                      LD (HL),C
0C18  23                      INC HL
0C19  20 F9                   JR NZ,BLOOP
0C1B  1B                      DEC DE
0C1C  ED 53 FA81              LD (CALCST),DE
0C20  C1                      POP BC
0C21  F1                      POP AF              ;Restore flags
0C22  C9                      RET
                        ;                         ON EXIT....
                        ;                         HL = end of filename.
                        ;                         DE = start of filename.
                        ;                         BC = length of user filename.
                        ;
                        ;
                        ;COMPname preserves DE',HL'
                        ;
0C23                    COMPname:
0C23  D9                      EXX                 ;DE -> oldname, HL -> newname
0C24  D5                      PUSH DE
0C25  E5                      PUSH HL
0C26  06 0E                   LD B,14
0C28  1A                COM1: LD A,(DE)
0C29  BE                      CP (HL)
0C2A  20 04                   JR NZ,COOUT
0C2C  23                      INC HL
0C2D  13                      INC DE
0C2E  10 F8                   DJNZ COM1
0C30  E1                COOUT: POP HL
0C31  E5                      PUSH HL
0C32  F5                      PUSH AF             ;Save flags
0C33  D7                      RST SCRRST
0C34  87                      DB 80H+7
0C35  46 4F 55 4E             DB 'FOUND: '
0C39  44 3A 20
0C3C  06 0F                   LD B,15
0C3E  23                      INC HL
0C3F  7E                ACHAR: LD A,(HL)
0C40  23                      INC HL
0C41  CD 0CAB                 CALL PRINTX
0C44  10 F9                   DJNZ ACHAR
0C46  D7                      RST SCRRST
0C47  82 0D 0A                DB 80H+2,13,10
0C4A  F1                      POP AF
0C4B  E1                      POP HL
0C4C  D1                      POP DE
```

```
0C4D  D9                      EXX
0C4E  C9                      RET
                        ;
                        ;
                        ;
                        ;ADJVAL is called before loading/saving BASIC variables
                        ;HL -> one byte after last byte in BASIC program
                        ;B = old LSTP6
                        ;
0C4F  7C                ADJVAL::LD A,H
0C50  FE C0                    CP 0C0H
0C52  30 03                    JR  NC,DJVL0          ;Jump if HL >= C000H,
0C54  21 C000                  LD HL,0C000H          ;else HL = C000H
0C57  E5                DJVL0: PUSH HL               ;HL = new (VARNAM)
0C58  3A FA7A                  LD A,(LSTP6)          ;A = true (LSTP6)
0C5B  B8                       CP B                  ;B = old (LSTP6)
0C5C  28 37                    JR Z,DJVL3
                        ;
0C5E  CD 35F9                  CALL ADJDES           ;Adjust DESAVE (preserves flags)
                        ;
0C61  38 05                    JR C,DJVL1            ;Jump if true (LSTP6) < false (LSTP6)
0C63  78                       LD A,B
0C64  06 C0                    LD B,0C0H
0C66  18 02                    JR DJVL2
0C68  06 40             DJVL1: LD B,40H
0C6A  32 FAD3           DJVL2: LD (CRNTP6),A
0C6D  0E 00                    LD C,0                ;BC = 4000H or C000H
0C6F  3A FAA6                  LD A,(NBTP6)
0C72  CD 0671                  CALL P6CHK
0C75  20 1E                    JR NZ,DJVL3
0C77  21 FAA4                  LD HL,NBTOP
0C7A  CD 05D1                  CALL DDBCM            ;(HL) = (HL) + BC,  HL = HL + 2
0C7D  CD 0679                  CALL BSTPHL
0C80  CD 05D1                  CALL DDBCM            ;(BASTPN) = (BASTPN) + BC
0C83  3A FAA9                  LD A,(BASTP6)
0C86  CD 0671                  CALL P6CHK
0C89  20 0A                    JR NZ,DJVL3
0C8B  21 FAA7                  LD HL,BASTOP
0C8E  CD 05D1                  CALL DDBCM            ;(BASTOP) = (BASTOP) + BC
0C91  AF                       XOR A
0C92  CD 0647                  CALL P6SLC1           ;Adjusts P6TOP inter alia
0C95  E1                DJVL3: POP HL                ;HL = new (VARNAM)
0C96  ED 5B FA7B               LD DE,(VARNAM)        ;DE = old (VARNAM)
0C9A  A7                       AND A
0C9B  ED 52                    SBC HL,DE
0C9D  44                DJVL4:: LD B,H
0C9E  4D                       LD C,L
0C9F  21 FA7B                  LD HL,VARNAM
0CA2  3E 04                    LD A,4
0CA4  CD 05D1           DJLP:  CALL DDBCM            ;Add BC to all BASIC variable pointers
0CA7  3D                       DEC A
0CA8  20 FA                    JR NZ,DJLP
0CAA  C9                       RET
                        ;
                        ;
                        END
```

A>

```
                        .Z80
0000'                   ASEG
                        ORG     0CABH
                        ;
                        ;
                        ;I/O SET UP ROUTINE TO SET UP BAUD RATES
                        ;
                        ;
0259                    BASIC   EQU     0259H
09F2                    BREAKM  EQU     09F2H
00BC                    GETSTR  EQU     00BCH
FA8F                    IOPL    EQU     0FA8FH
FD76                    IOPR    EQU     0FD76H
FD75                    PRORPL  EQU     0FD75H
2D62                    STOP    EQU     2D62H
                        ;
                        ;
                        PUBLIC BREAIO
                        PUBLIC SBAUD
                        PUBLIC BAUDST
                        PUBLIC PRINTX
                        ;
                        ;
                        ;RS232 PORTS AND CTC
                        ;
0030                    GETRST  EQU     30H
                        ;
0009                    CTCVA   EQU     09H
000A                    CTCVB   EQU     0AH
000E                    CTLRSA  EQU     0EH
000F                    CTLRSB  EQU     0FH
000C                    DATRSA  EQU     0CH
000D                    DATRSB  EQU     0DH
                        ;
                        ;
                        ;CENTRONICS PORTS
                        ;
0004                    CIFDAT  EQU     4;196
0000                    CIFLOW  EQU     0;192
0004                    CIFHI   EQU     4;196
0004                    CIFSTA  EQU     4;196
                        ;
                        ;
                        ;PARALLEL PORT
                        ;
0007                    PIN     EQU     7;199
0007                    POUT    EQU     7;199
                        ;
                        ;
0008                    DEHL    EQU     8
                        ;
                        ;
                        ;
                        ;I/O ROUTINES FOR 6 IN PORTS
                        ;    AND 6 OUT PORTS
                        ;
```

```
0CAB  E5          PRINTX: PUSH HL
0CAC  C5                  PUSH BC
0CAD  47                  LD B,A
0CAE  3A FD75             LD A,(PRORPL)
0CB1  A7                  AND A
0CB2  20 05               JR NZ,PRINX1
0CB4  CD 0CBF             CALL LISTOU
0CB7  18 03               JR PRINX2
0CB9  CD 0CC4     PRINX1: CALL LLISTO
0CBC  C1          PRINX2: POP BC
0CBD  E1                  POP HL
0CBE  C9                  RET
                  ;
                  ;
0CBF  3A FD76     LISTOU: LD A,(IOPR)
0CC2  18 03               JR FNDOUT
                  ;
                  ;
0CC4  3A FA8F     LLISTO: LD A,(IOPL)
                  ;
                  ;
0CC7  E6 03       FNDOUT: AND 3
0CC9  FE 01       FNDOU1: CP 1
0CCB  38 0F               JR C,OUTSTR         ;JUMP IF A = 0
0CCD  28 11               JR Z,CIFOUT         ;JUMP IF A = 1
                  ;
                  ;HERE IF A = 2
                  ;
0CCF  CD 0CF3     RSAOUT: CALL BREAIO
0CD2  DB 0E               IN A,(CTLRSA)
0CD4  CB 57               BIT 2,A
0CD6  28 F7               JR Z,RSAOUT
0CD8  78                  LD A,B
0CD9  D3 0C               OUT (DATRSA),A
0CDB  C9                  RET
                  ;
                  ;
0CDC  78          OUTSTR: LD A,B
0CDD  C3 00BC             JP GETSTR
                  ;
                  ;
0CE0  CD 0CF3     CIFOUT: CALL BREAIO
0CE3  DB 04               IN A,(CIFSTA)
0CE5  E6 0F               AND 0FH
0CE7  FE 0A               CP 0AH
0CE9  20 F5               JR NZ,CIFOUT
0CEB  78                  LD A,B
0CEC  D3 04               OUT (CIFDAT),A
0CEE  DB 00               IN A,(CIFLOW)
0CF0  DB 04               IN A,(CIFHI)
0CF2  C9                  RET
                  ;
                  ;
0CF3  CD 09F2     BREAIO: CALL BREAKMON
0CF6  C2 0259             JP NZ,BASIC
0CF9  C9                  RET
```

```
                        ;
                        ;
0CFA  00 4B 01          BAUDRT: DB      00H,4BH,1       ;19200
0CFD  80 25 02                  DB      80H,25H,2       ;9600
0D00  C0 12 04                  DB      0C0H,12H,4      ;4800
0D03  60 09 08                  DB      60H,09H,8       ;2400
0D06  B0 04 10                  DB      0B0H,04H,10H    ;1200
0D09  58 02 20                  DB      58H,02H,20H     ;600
0D0C  2C 01 40                  DB      2CH,01H,40H     ;300
0D0F  96 00 80                  DB      96H,00H,80H     ;150
0D12  6E 00 AF                  DB      6EH,00H,0AFH    ;110
0D15  4B 00 00                  DB      4BH,00H,0       ;75
0D18  18 01 00 04       RSWRR:  DB      18H,1,0,4,4CH,3,0E1H,5,68H
0D1C  4C 03 E1 05
0D20  68
                        ;
                        ;

                        ;
0D21  3E 03             BAUDS1: LD A,3
0D23  D3 0A                     OUT (CTCVB),A
0D25  3E 45                     LD A,45H
0D27  D3 0A                     OUT (CTCVB),A
0D29  7E                        LD A,(HL)
0D2A  D3 0A                     OUT (CTCVB),A
0D2C  3E 0F                     LD A,CTLRSB
                        ;
0D2E  4F                WRRS:   LD C,A
0D2F  21 0D18           WRRS32: LD HL,RSWRR
0D32  06 09                     LD B,9
0D34  ED B3                     OTIR
0D36  C9                        RET
0D37  05                          DB 5
0D38  F7                SBAUD:    RST GETRST
0D39  F5                          PUSH AF
0D3A  F7                          RST GETRST
0D3B  F1                          POP AF
                        ;

0D3C  D5                BAUDST: PUSH DE
0D3D  F5                        PUSH AF
0D3E  CD 0D56                   CALL FNDBUD
0D41  F1                        POP AF
0D42  D1                        POP DE
0D43  E6 01                     AND 1
0D45  20 DA                     JR NZ,BAUDS1
0D47  3E 03                     LD A,3
0D49  D3 09                     OUT (CTCVA),A
0D4B  3E 45                     LD A,45H
0D4D  D3 09                     OUT (CTCVA),A
0D4F  7E                        LD A,(HL)
0D50  D3 09                     OUT (CTCVA),A
0D52  3E 0E                     LD A,CTLRSA
0D54  18 D8                     JR WRRS
                        ;
                        ;
```

```
0D56  3E 0A           FNDBUD: LD A,0AH
0D58  F5                      PUSH AF
0D59  21 0CFA                 LD HL,BAUDRT
0D5C  CF              FNDBU1: RST DEHL
0D5D  EB                      EX DE,HL
0D5E  B7                      OR A
0D5F  ED 42                   SBC HL,BC
0D61  20 03                   JR NZ,FNDBU2
0D63  EB                      EX DE,HL
0D64  F1                      POP AF
0D65  C9                      RET
                      ;
0D66  F1              FNDBU2: POP AF
0D67  3D                      DEC A
0D68  20 03                   JR NZ,FNDBU3
0D6A  EF                      RST 28H
0D6B  22                      DB 34                ;BAD BAUD NO.
0D6C  C9                      RET
                      ;
0D6D  F5              FNDBU3: PUSH AF
0D6E  EB                      EX DE,HL
0D6F  23                      INC HL
0D70  18 EA                   JR FNDBU1
                      ;
                      ;
                      ;
                      END
```

```
                        .Z80
0000'                   ASEG
                        ORG     0D72H
                        ;
                        ;
                        PUBLIC COMPAD,GOTZER,INT64K
                        PUBLIC OVERFL,SETZER,ABS,CORECSIGN,EXP01
                        PUBLIC STAHLHLD,ROAHL,ROTR,DIV10,BCDEC,BIND1
                        PUBLIC DIVBY10
                        PUBLIC STBC,STHLHLD,TOKMIN,ZERACC,ROUNACC
                        PUBLIC OUTRANG,INTEGER,COPYACC,COMPARE
                        PUBLIC ACCSTACK,ADD,ALOP1,COMPACC,COMPBC
                        PUBLIC COSINE,DESTACK,DIVIDE,DONIB
                        PUBLIC EXCHANGE,FRAC,LARGE1,LARGEN
                        PUBLIC MOVDEOP1,MOVEDE,MULPLY,STRJP
                        PUBLIC NEGACC,OP1STACK,PIOP1,POLY2,POLYNOM
                        PUBLIC REMAINDER,RETACC,SAVEACC,SINE
                        PUBLIC SMALL1,SMALLEN,SQUARE,STA,STAC
                        PUBLIC STACKACC,STACKOP1,SUBTRACT
                        PUBLIC OPTABLE
                        ;
                        ;
20C0                    EQUAL   EQU     20C0H
20C8                    GREATER EQU     20C8H
20D2                    LESSTH  EQU     20D2H
20DB                    GREEQUL EQU     20DBH
20D2                    LESEQUL EQU     20D2H
20E9                    NOTEQU  EQU     20E9H
209A                    ANDACC  EQU     209AH
20A6                    ORACC   EQU     20A6H
20AF                    NOTACC  EQU     20AFH
208F                    SIGNACC EQU     208FH
                        ;
203D                    USR     EQU     203DH
3F4B                    ASC     EQU     3F4BH
3F95                    LEN     EQU     3F95H
3F9E                    VAL     EQU     3F9EH
2048                    RND     EQU     2048H
233F                    POWER   EQU     233FH
236E                    SQROOT  EQU     236EH
241D                    ATN     EQU     241DH
23EC                    EXP     EQU     23ECH
200A                    INT     EQU     200AH
2398                    LOG     EQU     2398H
2494                    LOGE2   EQU     2494H
2499                    ENUMBE  EQU     2499H
2452                    ATNT2   EQU     2452H
0A3A                    CHR$    EQU     0A3AH
153A                    CHRDS   EQU     153AH
3ECE                    INKEY$  EQU     3ECEH
3F20                    LEFT$   EQU     3F20H
3EDD                    MID$    EQU     3EDDH
3F1A                    RIGHT$  EQU     3F1AH
1541                    GRRDS   EQU     1541H
3FC6                    STR$    EQU     3FC6H
3F38                    TIME$   EQU     3F38H
```

```
                        ;
FDD2                    OP11    EQU     0FDD2H
FDCC                    ACC1    EQU     0FDCCH
FDD8                    SIGN    EQU     0FDD8H
FDD7                    YORN    EQU     0FDD7H
FDDE                    COPY    EQU     0FDDEH
FA81                    CALCST  EQU     0FA81H
FDD9                    MEM1    EQU     0FDD9H
                        ;
                        ;
0D72                    OPTABLE:
0D72    0E86                    DW      ADD
0D74    0E83                    DW      SUBTRACT
0D76    100B                    DW      MULPLY
0D78    109E                    DW      DIVIDE
0D7A    233F                    DW      POWER       ;Undefined
0D7C    20C0                    DW      EQUAL
0D7E    20C8                    DW      GREATER
0D80    20D2                    DW      LESSTH
0D82    20DB                    DW      GREEQUL
0D84    20D2                    DW      LESEQUL
0D86    20E9                    DW      NOTEQUAL
0D88    209A                    DW      ANDACC
0D8A    20A6                    DW      ORACC
                        ;
                        ;Single operators
                        ;
0D8C    20AF                    DW      NOTACC      ;220
0D8E    0E56                    DW      ABS
0D90    241D                    DW      ATN
0D92    137D                    DW      COSINE
0D94    23EC                    DW      EXP
0D96    23EC                    DW      EXP
0D98    200A                    DW      INT
0D9A    0030                    DW      UNDEFINE    ;**********
0D9C    2398                    DW      LOG
0D9E    1301                    DW      PEEK
0DA0    208F                    DW      SIGNACC
0DA2    13B2                    DW      SINE
0DA4    236E                    DW      SQROOT
0DA6    13B7                    DW      TANGENT
0DA8    1309                    DW      INP
0DAA    203D                    DW      USR
0DAC    0030                    DW      UNDEFINE
0DAE    3F4B                    DW      ASC
0DB0    3F95                    DW      LEN
0DB2    3F9E                    DW      VAL
0DB4    0030                    DW      UNDEFINE
0DB6    0FFA                    DW      REMAINDER ;2 numbers needed
0DB8    12A4                    DW      PIACC     ;242,0 numbers
0DBA    204B                    DW      RND       ;same
0DBC    0FEA                    DW      COMPACC
                        ;
                        ;String operators
                        ;
0DBE    0A3A            STRJP:  DW CHR$
```

```
0DC0  153A                    DW   CHRDS
0DC2  3ECE                    DW INKEY$
0DC4  3F20                    DW LEFT$
0DC6  3EDD                    DW   MID$
0DC8  3F1A                    DW RIGHT$
0DCA  1541                    DW GRRDS
0DCC  3FC6                    DW   STR$
0DCE  3F3B                    DW TIME$
                        ;
                        ;   New format for numbers
                        ;   ************************
                        ;
                        ;First numbers are stored without the
                        ;bit that specifies the one.i.e.
                        ;1 comes out as 0 with some exponent
                        ;
                        ;0 comes as 0 in the mantissa and 0 in the
                        ;exponent
                        ;
                        ;the exponent is calculated as a 2's complement
                        ;but with 81h as zero exponent
                        ;
                        ;All operations act on positive numbers
                        ;with a High bit set.
                        ;
0030                    FRONT     EQU  30H
0028                    ERROR     EQU  28H
0028                    ERRORS    EQU  28H
0038                    MATHSRST  EQU  38H
0028                    JT        EQU  28H
                        ;
                        ;
00FF                    ENDAR     EQU  0FFH
                        ;
00D2                    DIV       EQU  MULT+1
00D0                    SUB       EQU  PLUS+1
                        ;
                        ;
                        ;
                        ;This converts the number in BC
                        ;to a decimal number pointed to by
                        ;DE ended in a ENDMARK(0FFH)
                        ;
                        ;
0DD0  CD 10A6           BCDEC:    CALL GOTZER       ;BC not affected
0DD3  ED 43 FDCB                  LD   (ACC1-1),BC
0DD7  CD 0DFE                     CALL BINDEC
0DDA  13                          INC  DE
0DDB  1A                          LD   A,(DE)
0DDC  FE FF                       CP   0FFH
0DDE  C0                          RET  NZ
0DDF  1B                          DEC  DE
0DE0  3E 30                       LD   A,'0'
0DE2  12                          LD   (DE),A
0DE3  C9                          RET
```

Numbers Stored as 4 byte mantissa followed by 1 byte exponent. Mantissa Stored with least significant byte first. Sign bit in msb of m.s. byte 1 = -ve.

Uses ACC1. Result in COPY.

- Zero ACC1
- Store BC in ACC1
- Convert to decimal
- Pointer to beginning of string
- Exit if not a null string
- Print a single zero

```
                        ;
                        ;
                        ;
                        ;BIND1 CONVERTS A BINARY NO WITH AN
                        ;EXPONENT BETWEEN 0 AND 4.ACC IS
                        ;SET TO ZERO THE RESULT IS STORED
                        ;BACKWRDS IN COPY
                        ;
                        ;
0DE4  21 FDD0           BIND1:  LD   HL,ACC1+4
0DE7  3E A8                     LD   A,80H+40     40 bits ?
0DE9  96                        SUB  (HL)
0DEA  0E 00                     LD   C,0
0DEC  28 0C                     JR   Z,BIN3    ;CONVERT
0DEE  08                BIN2:   EX   AF,AF'    ;COUNTER
0DEF  06 01                     LD   B,1
0DF1  CD 0FBD                   CALL ROTR  - Rotate C6 byte mantissa
0DF4  CB 19                     RR   C         ;Catch carry
0DF6  08                        EX   AF,AF'
0DF7  3D                        DEC  A
0DF8  20 F4                     JR   NZ,BIN2
0DFA  79                BIN3:   LD   A,C
0DFB  32 FDCB                   LD   (ACC1-1),A  Store shifted bits as an extra mantissa byte
                        ;
                        ;
                        ;BINDEC CONVERTS A BINARY NO
                        ;IN ACC1 TO AN ASCII NO BCD
                        ;IN COPY
                        ;
                        ;
0DFE  06 00             BINDEC: LD   B,0
0E00  21 FDEA                   LD   HL,COPY+12 ;DESTINATION
0E03  36 FF                     LD   (HL),ENDMARK
0E05  2B                        DEC  HL
0E06  EB                        EX   DE,HL     ;store at DE
0E07  0E 50                     LD   C,50H     ;FOR SUBTRACTION
0E09  21 FDCB                   LD   HL,ACC1-1
0E0C  7E                SBINDEC: LD  A,(HL)    ;CHECK FOR ZERO
0E0D  23                        INC  HL
0E0E  B6                        OR   (HL)
0E0F  23                        INC  HL
0E10  B6                        OR   (HL)
0E11  23                        INC  HL
0E12  B6                        OR   (HL)
0E13  23                        INC  HL
0E14  B6                        OR   (HL)      ;HL IS AT ACC1+3
0E15  C8                        RET  Z
0E16  04                        INC  B
0E17  C5                        PUSH BC
0E18  CD 0E28                   CALL DIVBY10
0E1B  C1                        POP  BC
0E1C  0F                        RRCA
0E1D  0F                        RRCA           ;GET REMAINDER IN
0E1E  0F                        RRCA           ;LEAST NIBBLE OF A
0E1F  0F                        RRCA
0E20  E6 0F                     AND  0FH
```

```
0E22  C6 30                   ADD  A,'0'
0E24  12                      LD   (DE),A    ;SAVE ANSWER
0E25  1B                      DEC  DE
0E26  18 E4                   JR   SBINDEC   ;REPEAT
                      ;
                      ;
                      ;
                      ;ROUTINE TO DIVIDE A 5 BYTE NO IN
                      ;ACC BY 10 RESULT IN ACC.
                      ;
                      ;
0E28  3E 00           DIVBY10: LD   A,0  ;NO REMAINDER TO START
0E2A  06 05                    LD   B,5  ;COUNTER
0E2C  ED 6F           A6DIV:   RLD       ;GET MS NIBBLE IN A
0E2E  CD 0E3D                  CALL DIV10
0E31  ED 6F                    RLD       ;SAVE QUOTIENT
0E33  CD 0E3D                  CALL DIV10;GET NEXT NIBBLE
0E36  ED 6F                    RLD
0E38  2B                       DEC  HL
0E39  10 F1                    DJNZ A6DIV
0E3B  23                       INC  HL   ;HL POINTS AT ACC-1
0E3C  C9                       RET
                      ;
                      ;
                      ;DIV10 divides a no. in A by 10
                      ;result is in lower nibble of A
                      ;remainder in upper nibble of A.
                      ;C needs to have the value 50H.
                      ;
                      ;
0E3D  B9              DIV10:   CP   C
0E3E  38 01                    JR   C,ROTE1
0E40  91                       SUB  C
0E41  3F              ROTE1:   CCF
0E42  17                       RLA
0E43  B9                       CP   C
0E44  38 01                    JR   C,ROTE2
0E46  91                       SUB  C
0E47  3F              ROTE2:   CCF
0E48  17                       RLA
0E49  B9                       CP   C
0E4A  38 01                    JR   C,ROTE3
0E4C  91                       SUB  C
0E4D  3F              ROTE3:   CCF
0E4E  17                       RLA
0E4F  B9                       CP   C
0E50  38 01                    JR   C,ROTE4
0E52  91                       SUB  C
0E53  3F              ROTE4:   CCF
0E54  17                       RLA
0E55  C9                       RET
                      ;
                      ;
                      ;
                      ;This returns the absolute value of
```

At each step remainder is left in M.S. nibble of A (ie multiplied by 16) and next less significant nibble moved into lower nibble of A. Effectively performing modulo 16.

A must be < 160

Excess over 80 in lsb, shift remainder into msb

Excess over 40 in lsb, shift remainder towards msb

Excess over 20 in lsb, shift remainder towards msb

Excess over 10 in lsb, shift remainder into upper nibble

```
                        ;ACC, C set if was negative.Z set if
                        ;zero NC if was positive
                        ;
                        ;
0E56  21 FDD0           ABS:    LD   HL,ACC1+4
0E59  7E                        LD   A,(HL)
0E5A  A7                        AND  A
0E5B  C8                        RET  Z
0E5C  2B                        DEC  HL         ;Sign byte
0E5D  7E                        LD   A,(HL)
0E5E  17                        RLA             ;Sign flag
0E5F  CB BE                     RES  7,(HL)     ;Positive
0E61  23                        INC  HL
0E62  7E                        LD   A,(HL)
0E63  C9                        RET
                    ;
                    ;
                    ;This is a general subroutine it first
                    ;stores the sign of ACC and then xors it
                    ;with the sign of OP1,If either OP1 or
                    ;ACC is zero then Z is returned if OP1
                    ;is zero then C is returned.
                    ;
0E64  11 FDCF       SETOP1ACC:LD   DE,ACC1+3 ;Ms byte
0E67  21 FDD5             LD   HL,OP11+3
0E6A  1A                  LD   A,(DE)
0E6B  E6 80               AND  80H
0E6D  32 FDD8             LD   (SIGN),A
0E70  AE                  XOR  (HL)
0E71  E6 80               AND  80H
0E73  32 FDD7             LD   (YORN),A
0E76  13                  INC  DE
0E77  23                  INC  HL
0E78  EB                  EX   DE,HL
0E79  CD 0FB2             CALL ZEROCH
0E7C  EB                  EX   DE,HL
0E7D  C8                  RET  Z
0E7E  CD 0FB2             CALL ZEROCH
0E81  37                  SCF
0E82  C9                  RET
                    ;
                    ;
                    ;SUBTRACT SUBTRACTS TWO 4 BYTE NOS.
                    ;IN ACC AND OP1.ACC=OP1-ACC
                    ;GIVES A 4 BYTE RESULT.
                    ;
                    ;
0E83  CD 0FEA       SUBTRACT: CALL COMPACC
                    ;
                    ;
                    ;ADDITION works by first calculating
                    ;the sign of the two numbers then standardising
                    ;them and finally if they have the same sign
                    ;adding them or if different signs subtracting
                    ;them
                    ;
```

```
                        ;ADD adds the two numbers in ACC
                        ;and OP1 to give an answer in ACC
                        ;
                        ;
0E86  CD 0E64           ADD:    CALL SETOP1ACC ;Init test zero
0E89  3A FDD7                   LD   A,(YORN)
0E8C  20 07                     JR   NZ,CONADD
0E8E  38 53                     JR   C,CORECSIGN
0E90  CD 12E7                   CALL COPY6     ;ACC1=OP11
0E93  18 51                     JR   CORRECT
0E95  A7                CONADD: AND  A         ;NC to start
0E96  08                        EX   AF,AF'
0E97  1A                        LD   A,(DE)
0E98  47                        LD   B,A
0E99  96                        SUB  (HL)
0E9A  30 07                     JR   NC,GETROT
0E9C  46                        LD   B,(HL)
0E9D  ED 44                     NEG
0E9F  08                        EX   AF,AF'
0EA0  3F                        CCF
0EA1  08                        EX   AF,AF'
0EA2  EB                        EX   DE,HL
0EA3  28 0D             GETROT: JR   Z,STRET
0EA5  FE 21                     CP   33
0EA7  38 04                     JR   C,NOTSUF
0EA9  CD 10A9                   CALL SETZER
0EAC  AF                        XOR  A         ;Zero
0EAD  70                NOTSUF: LD   (HL),B    ;Same exponents
0EAE  47                        LD   B,A
0EAF  CD 0FBD                   CALL ROTR
0EB2  2B                STRET:  DEC  HL
0EB3  2B                        DEC  HL
0EB4  2B                        DEC  HL
0EB5  2B                        DEC  HL
0EB6  1B                        DEC  DE
0EB7  1B                        DEC  DE
0EBB  1B                        DEC  DE
0EB9  1B                        DEC  DE
0EBA  06 04                     LD   B,4
0EBC  08                        EX   AF,AF'    ;Get yorn
0EBD  CB 7F                     BIT  7,A
0EBF  20 2D                     JR   NZ,DOSUB
0EC1  08                        EX   AF,AF'
0EC2  1A                REPADD: LD   A,(DE)
0EC3  8E                        ADC  A,(HL)
0EC4  77                        LD   (HL),A    ;PUT RESULT IN ACC
0EC5  23                        INC  HL        ;NEXT ONES
0EC6  13                        INC  DE
0EC7  10 F9                     DJNZ REPADD    ;REPEAT
0EC9  2B                        DEC  HL
0ECA  11 FDCF                   LD   DE,ACC1+3
0ECD  01 0004                   LD   BC,4
0ED0  ED B8                     LDDR           ;Copy it
0ED2  30 0F                     JR   NC,CORECSIGN  ;No normal or round
0ED4  21 FDD0                   LD   HL,ACC1+4
0ED7  34                        INC  (HL)
```

; OP1 zero
ACC1 zero
so copy OP1
} Special cases for one element zero

; Toggle carry flag'

; Transpose operands
; Jump if exponents same

; It difference too large
set smaller value to zero

; Shift right smaller mantissa

; Pointers to least significant mantissa byte.

; Jump if signs of two terms differ

; Add and carry

; Copy result into ACC1
In case operands transposed

; No normal or round - Jump if no overflow in addition

; Increment exponent

```
0ED8  CA 121B            JP   Z,OVERFLOW     - Jump if floating overflow
0EDB  06 01              LD   B,1            ;1 rotate
0EDD  CD 0FBD            CALL ROTR           ;Carries vernt affected
0EE0  2B                 DEC  HL
0EE1  CB FE              SET  7,(HL)         - Set leading mantissa bit
0EE3  3A FDD8  CORECSIGN:LD   A,(SIGN)
0EE6  21 FDCF  CORRECT:  LD   HL,ACC1+3
0EE9  F6 7F              OR   7FH            ;Set to ones
0EEB  A6                 AND  (HL)
0EEC  77                 LD   (HL),A
0EED  C9                 RET
                         ;
                         ;
0EEE  08       DOSUB:    EX   AF,AF'
0EEF  1A       DOSUB1:   LD   A,(DE)
0EF0  9E                 SBC  A,(HL)
0EF1  77                 LD   (HL),A
0EF2  13                 INC  DE
0EF3  23                 INC  HL
0EF4  10 F9              DJNZ DOSUB1
0EF6  2B                 DEC  HL
0EF7  11 FDCF            LD   DE,ACC1+3
0EFA  01 0004            LD   BC,4
0EFD  ED B8              LDDR
0EFF  30 06              JR   NC,DONTNEG     - Jump if result positive
0F01  08                 EX   AF,AF'
0F02  3F                 CCF
0F03  08                 EX   AF,AF'
0F04  CD 0F13            CALL NEGACC         ;B=0 - Complement result
0F07  CD 0F47  DONTNEG:  CALL NORMAL         - Normalize result
0F0A  08                 EX   AF,AF'
0F0B  CB 19              RR   C
0F0D  3A FDD8            LD   A,(SIGN)
0F10  A9                 XOR  C
0F11  18 D3              JR   CORRECT
                         ;
                         ;This negates ACC1,4 bytes done
                         ;
0F13  37       NEGACC:   SCF  - Initial
0F14  21 FDCC            LD   HL,ACC1
0F17  7E                 LD   A,(HL)
0F18  2F                 CPL
0F19  88                 ADC  A,B
0F1A  77                 LD   (HL),A    ;1
0F1B  23                 INC  HL
0F1C  7E                 LD   A,(HL)
0F1D  2F                 CPL
0F1E  88                 ADC  A,B
0F1F  77                 LD   (HL),A    ;2
0F20  23                 INC  HL
0F21  7E                 LD   A,(HL)
0F22  2F                 CPL
0F23  88                 ADC  A,B
0F24  77                 LD   (HL),A    ;3
0F25  23                 INC  HL
0F26  7E                 LD   A,(HL)
```

```
0F27  2F                      CPL
0F28  88                      ADC  A,B
0F29  77                      LD   (HL),A    ;4
0F2A  C9                      RET

                        ;
                        ;This rounds the 5 byte number pointed
                        ;to by HL at LSB,Returns anywhere
                        ;Only positive numbers done,With high
                        ;bits set.
                        ;
0F2B  21 FDCB           ROUNACC: LD   HL,ACC1-1
0F2E  CB 16             ROHL:    RL   (HL)      ;Get carry
0F30  36 00                      LD   (HL),0    ;Nothing
0F32  D0                         RET  NC
0F33  23                         INC  HL
0F34  34                ADDROUN: INC  (HL)
0F35  C0                         RET  NZ
0F36  23                         INC  HL
0F37  34                         INC  (HL)
0F38  C0                         RET  NZ
0F39  23                         INC  HL
0F3A  34                         INC  (HL)
0F3B  C0                         RET  NZ
0F3C  23                         INC  HL        ;Special
0F3D  34                         INC  (HL)
0F3E  C0                         RET  NZ        ;Finshed
0F3F  36 80                      LD   (HL),80H
0F41  23                         INC  HL
0F42  34                         INC  (HL)
0F43  C0                         RET  NZ
0F44  C3 121B                    JP   OVERFLOW

                        ;
                        ;This normalises a 5 byte number in ACC
                        ;No rounding is done.C contains start
                        ;exponent.HL returns at ACC1+4
                        ;
0F47  0E 80             NORMAL:  LD   C,80H
0F49  3A FDCF           ANORM:   LD   A,(ACC1+3)
0F4C  CB 7F                      BIT  7,A
0F4E  20 21                      JR   NZ,QUNORM
0F50  2A FDCD                    LD   HL,(ACC1+1)
0F53  D9                         EXX
0F54  2A FDCB                    LD   HL,(ACC1-1)
0F57  D9                         EXX
0F58  47                STAHLHLD: LD  B,A  ;Save A
0F59  B4                         OR   H
0F5A  B5                         OR   L
0F5B  D9                         EXX
0F5C  B5                         OR   L
0F5D  B4                         OR   H
0F5E  D9                         EXX
0F5F  CA 10A6                    JP   Z,GOTZER
0F62  78                         LD   A,B
0F63  CD 0FB4                    CALL MDNORM
0F66  32 FDCF           SAVEAHL: LD   (ACC1+3),A
0F69  22 FDCD                    LD   (ACC1+1),HL
```

```
0F6C  D9                    EXX
0F6D  22 FDCB               LD   (ACC1-1),HL
0F70  D9                    EXX
0F71  21 FDD0        QUNORM: LD   HL,ACC1+4
0F74  79                    LD   A,C
                     ;
                     ;
                     ;This adds the twos complemt
                     ;number in A to the number in HL
                     ;
0F75  86             ADDA:  ADD  A,(HL)
0F76  38 05                 JR   C,CHECPOS
0F78  F2 10A9               JP   P,SETZER
0F7B  18 03                 JR   ADDBAC
0F7D  FA 121B        CHECPOS: JP  M,OVERFLOW
0F80  C6 80          ADDBAC: ADD  A,80H
0F82  77                    LD   (HL),A
0F83  C9                    RET
                     ;
                     ;This normalises A HL HL'
                     ;exponent in C,No rounding
                     ;takes place.
                     ;
                     ;
0F84  B7             MDNORM: OR   A
0F85  20 10                 JR   NZ,NOT8BIT
0F87  E5                    PUSH HL        ;Save H for A
0F88  65                    LD   H,L
0F89  D9                    EXX
0F8A  7C                    LD   A,H
0F8B  65                    LD   H,L
0F8C  2E 00                 LD   L,0
0F8E  D9                    EXX
0F8F  6F                    LD   L,A
0F90  79                    LD   A,C
0F91  D6 08                 SUB  8
0F93  4F                    LD   C,A
0F94  F1                    POP  AF        ;A=H
0F95  18 ED                 JR   MDNORM
0F97  F8             NOT8BIT: RET  M
0F98  0D                    DEC  C
0F99  D9                    EXX
0F9A  29                    ADD  HL,HL
0F9B  D9                    EXX
0F9C  ED 6A                 ADC  HL,HL
0F9E  8F                    ADC  A,A
0F9F  18 F6                 JR   NOT8BIT
                     ;
                     ;
                     ;This rounds off A HL HL'
                     ;Z set if A was zero
                     ;To round off A HL H' then
                     ;call ROHLD
                     ;
0FA1  D9             ROAHL:  EXX
0FA2  2C                    INC  L
```

```
0FA3  D9          EXX
0FA4  C0          RET  NZ
0FA5  D9          EXX
0FA6  24          INC  H
0FA7  D9          EXX
0FA8  C0          RET  NZ
0FA9  2C          INC  L
0FAA  C0          RET  NZ
0FAB  24          INC  H
0FAC  C0          RET  NZ
0FAD  3C          INC  A
0FAE  C9          RET
                  ;
                  ;
                  ;Zeroch checks to see if
                  ;the number pointed to by HL at
                  ;exponent is zero, If not then
                  ;negates HL.
                  ;
                  ;
0FAF  21 FDD0     ZERACC:  LD   HL,ACC1+4
0FB2  7E          ZEROCH:  LD   A,(HL)
0FB3  B6                   OR   (HL)
0FB4  C8                   RET  Z
0FB5  2B                   DEC  HL
0FB6  7E                   LD   A,(HL)
0FB7  17                   RLA            ;C set if it was negative
0FBB  CB FE                SET  7,(HL)    ;Z un affected
0FBA  23                   INC  HL
0FBB  7E                   LD   A,(HL)
0FBC  C9                   RET
                  ;
                  ;
                  ;ROTR ROTATES 5 BYTES RIGHT
                  ;INITIALLY HL POINTS AT EXP BYTE
                  ;HL RETURNS WITH EXP BYTE
                  ;B rotations are done rotates
                  ;
0FBD  2B          ROTR:    DEC  HL   ;MS byte
0FBE  78          DONIB:   LD   A,B
0FBF  A7                   AND  A
0FC0  28 26                JR   Z,ENROT
0FC2  FE 04                CP   4
0FC4  38 13                JR   C,DOBIT
0FC6  E5                   PUSH HL
0FC7  D6 04                SUB  4
0FC9  47                   LD   B,A
0FCA  AF                   XOR  A
0FCB  ED 67                RRD            ;1
0FCD  2B                   DEC  HL
0FCE  ED 67                RRD            ;2
0FD0  2B                   DEC  HL
0FD1  ED 67                RRD            ;3
0FD3  2B                   DEC  HL
0FD4  ED 67                RRD            ;4
0FD6  E1                   POP  HL
```

```
0FD7  18 E5                    JR   DONIB
0FD9  E5              DOBIT:   PUSH HL
0FDA  CB 3E                    SRL  (HL)
0FDC  2B                       DEC  HL
0FDD  CB 1E                    RR   (HL)
0FDF  2B                       DEC  HL
0FE0  CB 1E                    RR   (HL)
0FE2  2B                       DEC  HL
0FE3  CB 1E                    RR   (HL)
0FE5  E1                       POP  HL
0FE6  10 F1                    DJNZ DOBIT
0FE8  23              ENROT:   INC  HL
0FE9  C9                       RET
                      ;
                      ;
                      ;COMP TWOS COMPLEMENTS A NO.
                      ;HL ENTERS AT exponent BYTE EXIT
                      ;AT EXPONENT
                      ;
                      ;
                      ;
0FEA  21 FDD0         COMPACC: LD   HL,ACC1+4
0FED  F5              COMP1:   PUSH AF
0FEE  7E                       LD   A,(HL)
0FEF  A7                       AND  A
0FF0  28 06                    JR   Z,GETAF
0FF2  2B                       DEC  HL
0FF3  7E                       LD   A,(HL)
0FF4  EE 80                    XOR  80H
0FF6  77                       LD   (HL),A
0FF7  23                       INC  HL
0FF8  F1              GETAF:   POP  AF
0FF9  C9                       RET
                      ;
                      ;
                      ;This takes the remainder of the numbers
                      ;in ACC1 and OP11 to give a remainder in
                      ;ACC1
                      ;
0FFA  CD 1286         REMAINDER:CALL STACKACC  ;Save this one
0FFD  CD 109E                   CALL DIVIDE
1000  CD 112B                   CALL FRAC
1003  CD 129E                   CALL OP1STACK
1006  18 03                     JR   MULPLY
                      ;
1008  CD 12F3         SQUARE:   CALL COPYACC
                      ;
                      ;
                      ;MULPLY MULTIPLYS TWO 4 BYTE NOS
                      ;IN TOP BYTES OF ACC AND OP1
                      ;RESULT IS PUT IN ACC AS AN 8 BYTE
                      ;NO.
                      ;ACC=ACC*OP1
                      ;
                      ;
100B  CD 0E64         MULPLY:   CALL SETOP1ACC
```

Shift one bit

Exit if zero exponent

Toggle sign bit

Test operands

```
100E  CA 10A6                 JP   Z,GOTZER
1011  CD 1094                 CALL SWAP      ;Largest in ACC
1014  3A FDCF          V5MULPLY: LD   A,(ACC1+3)
1017  5F                       LD   E,A
1018  16 00                    LD   D,0
101A  62                       LD   H,D
101B  6A                       LD   L,D
101C  D9                       EXX
101D  ED 4B FDCD               LD   BC,(ACC1+1)
1021  3A FDCC                 LD   A,(ACC1)
1024  DD 21 0000              LD   IX,0

                   ;To start put result in HL' ix
                   ;E' BC A HAVE ACC1 goes into
                   ;DE' BC
                   ;
1028  2A FDD4                 LD   HL,(OP11+2)
102B  ED 5B FDD2             LD   DE,(OP11)
                   ;
                   ;First multypy by the low two bytes
                   ;i.e OP11 and OP11+2
                   ;
102F  CB 3B          MTOP1:  SRL  E
1031  30 07                 JR   NC,NOLS
1033  D9                     EXX
1034  DD 19                  ADD  IX,DE
1036  30 01                 JR   NC,NOCARY
1038  23                     INC  HL
1039  D9           NOCARY:   EXX
103A  CB 3D          NOLS:   SRL  L
103C  30 06                 JR   NC,NOES
103E  DD 09                  ADD  IX,BC
1040  D9                     EXX
1041  ED 5A                  ADC  HL,DE
1043  D9                     EXX
1044  17           NOES:     RLA            ;There was NC
1045  CB 11                  RL   C
1047  CB 10                  RL   B
1049  D9                     EXX
104A  CB 13                  RL   E
104C  CB 12                  RL   D
104E  D9                     EXX
104F  F2 102F                JP   P,MTOP1
                   ;HL' IX result so far
                   ;A=0 L=0 E=0
                   ;DE' BC have ACC1
                   ;Now put into A HI' IX
                   ;C' De' BC have ACC1
1052  D9                       EXX
1053  4F                       LD   C,A       ;BC'=0 for rotating and add
1054  47                       LD   B,A
1055  D9                       EXX
1056  CB 3A          MTOP2:    SRL  D         ;High bytes
1058  30 08                    JR   NC,NODS
105A  D9                       EXX
105B  DD 19                    ADD  IX,DE
```

```
105D  ED 4A              ADC  HL,BC
105F  D9                 EXX
1060  CE 00              ADC  A,0
1062  CB 3C        NODS: SRL  H
1064  30 07              JR   NC,NOHS
1066  DD 09              ADD  IX,BC
1068  D9                 EXX
1069  ED 5A              ADC  HL,DE
106B  89                 ADC  A,C
106C  D9                 EXX
106D  CB 21        NOHS: SLA  C
106F  CB 10              RL   B
1071  D9                 EXX
1072  CB 13              RL   E
1074  CB 12              RL   D
1076  CB 11              RL   C
1078  D9                 EXX
1079  F2 1056            JP   P,MTOP2
107C  DD E5              PUSH IX
107E  E1                 POP  HL
107F  D9                 EXX
                   ;
                   ;
1080  0E 80              LD   C,80H
1082  CD 0F58            CALL STAHLHL
1085  3A FDD6      DCON: LD   A,(OP11+4)
1088  CD 0F75            CALL ADDA
108B  CD 0F2B            CALL ROUNACC
108E  3A FDD7            LD   A,(YORN)
1091  C3 0EE6            JP   CORRECT
                   ;
                   ;This compares the byte pointed to by
                   ;HL, with the one pointed to by DE
                   ;if DE is>hl then left otherwise swappded
                   ;(HL at OP11 DE at ACC1)
                   ;
1094  1A          SWAP: LD   A,(DE)
1095  BE                 CP   (HL)
1096  D0                 RET  NC        ;DE)=HL
1097  C5                 PUSH BC
1098  4E                 LD   C,(HL)
1099  77                 LD   (HL),A
109A  79                 LD   A,C
109B  12                 LD   (DE),A
109C  C1                 POP  BC
109D  C9                 RET
                   ;
                   ;
                   ;DIVIDE DIVIDES ACC1 BY OP11 TO
                   ;GIVE A 4 BYTE ANSWER IN ACC1
                   ;
                   ;
109E  CD 0E64      DIVIDE: CALL SETOP1ACC
10A1  20 12              JR   NZ,DIV3
10A3  D2 1217            JP   NC,DIVZERO
                   ;
```

A HL' IX = A HL' IX + BC' DE'

Shift and test LSB of H

A HL' IX = A HL' IX + C' DE' BC

Shift left C' DE' BC

Result in A HL HL'

Repeat 8 times

- Normalize
- Get the exponent
- Add exponents
- Round to 4 bytes
- Set sign bit

- Test operand
- Jump if non zero
- Zero divide error

```
                        ;
                        ;SETZER SETS A 5 BYTE NO.
                        ;TO 0 POINTED TO BY HL AT EXPONENT
                        ;RETURNS WITH A=0,B=0
                        ;AND HL AT EXPONENT
                        ;
                        ;
10A6  21 FDD0           GOTZER:  LD   HL,ACC1+4
                        ;
10A9  E5                SETZER:  PUSH HL
10AA  C5                         PUSH BC
10AB  06 06                      LD   B,6
10AD  AF                         XOR  A
10AE  77                REPZER:  LD   (HL),A
10AF  2B                         DEC  HL
10B0  10 FC                      DJNZ REPZER
10B2  C1                         POP  BC
10B3  E1                         POP  HL
10B4  C9                         RET
                      ;
                      ;
                      ;DIV3 DIVIDES OP1 BY ACC TO
                      ;GIVE A RESULT IN ACC
                      ;
                      ;
10B5  1A              DIV3:    LD   A,(DE)        ;Negate exponent
10B6  ED 44                    NEG
10B8  12                       LD   (DE),A
10B9  CD 1094                  CALL SWAP      ;Largest in ACC
10BC  34                       INC  (HL)
10BD  CA 121B                  JP   Z,OVERFLOW ;Overflow
10C0  06 01                    LD   B,1       ;USED TO HOLD THE CURRENT RESULT BYTE
10C2  0E 05                    LD   C,5     ;COUNTER FOR BYTES DONE.
10C4  2A FDD4                  LD   HL,(OP11+2);2 MOST SIGNIFICANT BYTES
10C7  ED 5B FDCE               LD   DE,(ACC1+2);2 MS BYTES OF DIVISOR
10CB  D9                       EXX            ;DASH REGISTERS
10CC  2A FDD2                  LD   HL,(OP11) ;LS 2 BYTES
10CF  ED 5B FDCC               LD   DE,(ACC1)
10D3  01 FDCF                  LD   BC,ACC1+3 ;ADDRESS FOR RESULT
10D6  D9                       EXX
10D7  18 0D                    JR   ENTERDIV  ;ENTRE DIVIDE PROGRAM
10D9  D9              DIVROTATE:EXX           ;ENTRY ON DASH REGS
10DA  3F              DIVR3:   CCF           ;WAS 0 IF A ONE 1 IF A ZERO
10DB  CB 10                    RL   B         ;PUT INTO RESULT
10DD  38 2C                    JR   C,SAVEBYTE;C MEANS D HAS BEEN FULLY ROTATED
10DF  D9              DIVR2:   EXX
10E0  29                       ADD  HL,HL
10E1  D9                       EXX
10E2  ED 6A                    ADC  HL,HL
10E4  38 1B                    JR   C,DIV2SUB
                      ;
                      ;COMPARE ROUTINE
                      ;
10E6  7C              ENTERDIV: LD   A,H
10E7  BA                        CP   D
10E8  38 F0                     JR   C,DIVR3   ;NO Irge enough
```

Marker bit

OP1 → HL HL'

ACC1 → DE DE'

Rotate carry into B
jump if marker bit
overflows

Double HL HL'

```
10EA  20 16                    JR   NZ,DIVSUB ;SUBTRACT
10EC  7D              TESTLC:  LD   A,L      ;NEXT BYTE
10ED  BB                       CP   E        ;COMPARE
10EE  38 EA                    JR   C,DIVR3
10F0  20 10                    JR   NZ,DIVSUB
10F2  D9                       EXX           ;DASH
10F3  7C                       LD   A,H
10F4  BA                       CP   D
10F5  38 E2                    JR   C,DIVROTATE
10F7  20 0A                    JR   NZ,DIV1SUB
10F9  7D                       LD   A,L
10FA  BB                       CP   E
10FB  38 DC                    JR   C,DIVROTATE
10FD  20 04                    JR   NZ,DIV1SUB
10FF  18 1A                    JR   ZERORESLT
1101  A7              DIV2SUB: AND  A        ;Clear carry
1102  D9              DIVSUB:  EXX           ;DASH REGS
1103  ED 52           DIV1SUB: SBC  HL,DE        ;NO CARRY SET
1105  D9                       EXX               ;NORMAL
1106  ED 52                    SBC  HL,DE
1108  A7                       AND  A        ;clear carry
1109  18 CF                    JR   DIVR3
110B  78              SAVEBYTE:LD   A,B      ;GET RESULT TO SAVE
110C  06 01                    LD   B,1      ;NEW COUNTER AND RESULT
110E  D9                       EXX           ;DE' POINTS TO ADDRESS
110F  02                       LD   (BC),A   ;SAVE
1110  0B                       DEC  BC       ;NEXT ONE
1111  D9                       EXX
1112  0D                       DEC  C        ;COUNTER
1113  20 CA                    JR   NZ,DIVR2
1115  CD 0F47          FINSDIV: CALL NORMAL
1118  C3 1085                  JP   DCON
                       ;
111B  D9               ZERORESLT:EXX      ;GET NORMAL REGS
111C  37                       SCF           ;ROTATE IN CARRY
111D  78                       LD   A,B      ;GET CURRENT BYTE
111E  17               DIVREPROT:RLA
111F  30 FD                    JR   NC,DIVREPROT;PUT BITS IN CORRECT PLACE
1121  D9               PUTANOT: EXX
1122  02                       LD   (BC),A
1123  0B                       DEC  BC
1124  D9                       EXX
1125  97                       SUB  A        ;A=0
1126  0D                       DEC  C
1127  20 F8                    JR   NZ,PUTANOT
1129  18 EA                    JR   FINSDIV
                       ;
                       ;
                       ;This takes the fractional part
                       ;of ACC
                       ;
112B  CD 0FAF          FRAC:    CALL ZERACC
112E  C8                       RET  Z
112F  1F                       RRA
1130  32 FDD8                   LD   (SIGN),A
1133  17                       RLA
```

```
1134  D6 A0                  SUB  80H+32
1136  D2 10A9                JP   NC,SETZER
1139  FE E1                  CP   -31
113B  DA 0EE3                JP   C,CORECSIGN
113E  C6 20          NONEG:  ADD  A,32      ;Number of rots
1140  47                     LD   B,A       ;Save it
1141  2B                     DEC  HL
1142  7E                     LD   A,(HL)
1143  2B                     DEC  HL
1144  2A FDCD                LD   HL,(ACC1+1)
1147  D9                     EXX
1148  2A FDCB                LD   HL,(ACC1-1)
114B  D9                     EXX
114C  D9             ROTREM: EXX
114D  29                     ADD  HL,HL
114E  D9                     EXX
114F  ED 6A                  ADC  HL,HL
1151  8F                     ADC  A,A
1152  10 F8                  DJNZ ROTREM
1154  01 8080                LD   BC,80H*256+80H
1157  CD 136B                CALL STHLHLD
115A  C3 0EE3                JP   CORECSIGN
                       ;
115D  11 FDCF        COMPAO: LD   DE,ACC1+3
1160  21 FDD5                LD   HL,OP11+3
                       ;
                       ;
                       ;COMPARE COMPARES TWO 4 BYTE NUMBERS
                       ;POINTED TO BY DE AND HL AT MS BYTE
                       ;RETURNS Z=1,A=0 IF THE SAME,C=0,A=1
                       ;IF DE>HL or C=1,A=ff if DE<HL
                       ;
                       ;
1163  E5             COMPARE:  PUSH HL
1164  D5                       PUSH DE
1165  1A                       LD   A,(DE)
1166  AE                       XOR  (HL)
1167  17                       RLA
1168  38 10                    JR   C,WHCHGRT ;different signs
116A  13                       INC  DE   ;EXPONENT
116B  23                       INC  HL
116C  06 06                    LD   B,6       ;Check 5+1 bytes
116E  1A             COMREP:   LD   A,(DE)
116F  BE                       CP   (HL)
1170  20 08                    JR   NZ,WHCHGRT
1172  1B                       DEC  DE
1173  2B                       DEC  HL
1174  10 F8                    DJNZ COMREP
1176  D1                       POP  DE
1177  E1                       POP  HL
1178  9F                       SBC  A,A       ;NC,Z
1179  C9                       RET
117A  D1             WHCHGRT:  POP  DE   ;NC if mod DE>mod HL
117B  E1                       POP  HL
117C  1F                       RRA        ;put in carry
117D  AE                       XOR  (HL) ;Different to HL
```

```
117E  17                    RLA
117F  9F                    SBC  A,A
1180  D8                    RET  C    ;NZ,C
1181  3C                    INC  A    ;NZ,NC;DE>HL
1182  C9                    RET
                    ;
                    ;
                    ;This exchanges OP1 with ACC
                    ;
                    ;
1183  D5            EXCHANGE: PUSH DE
1184  E5                      PUSH HL
1185  C5                      PUSH BC
1186  F5                      PUSH AF
1187  11 FDD1                 LD   DE,OP11-1
118A  21 FDCB                 LD   HL,ACC1-1
118D  06 06                   LD   B,6
118F  1A              EXOPACC:  LD   A,(DE)
1190  4E                      LD   C,(HL)
1191  77                      LD   (HL),A
1192  79                      LD   A,C
1193  12                      LD   (DE),A
1194  13                      INC  DE
1195  23                      INC  HL
1196  10 F7                   DJNZ EXOPACC
1198  F1                      POP  AF
1199  C1                      POP  BC
119A  E1                      POP  HL
119B  D1                      POP  DE
119C  C9                      RET
                    ;
                    ;This tests to see if ACC1 is
                    ;an integer between 0 and 255
                    ;
119D  21 FDD0         INTEGER:  LD   HL,ACC1+4
11A0  CD 11F6                 CALL TESTEXP
11A3  C0                      RET  NZ
11A4  D0                      RET  NC       ;Zero
11A5  D6 08                   SUB  8
11A7  D8                      RET  C        ;Too large
11A8  5F                      LD   E,A      ;Save rotates
11A9  79                      LD   A,C
11AA  2B                      DEC  HL
11AB  B6                      OR   (HL)
11AC  2B                      DEC  HL
11AD  B6                      OR   (HL)
11AE  C0                      RET  NZ       ;Mantissa( )0
11AF  78                      LD   A,B      ;Top byte
11B0  CB FF                   SET  7,A
11B2  1C                      INC  E
11B3  1D              INLOP:  DEC  E        ;NC's
11B4  C8                      RET  Z
11B5  0F                      RRCA
11B6  D8                      RET  C        ;mantissa<>0
11B7  18 FA                   JR   INLOP
                    ;
```

```
                        ;This converts the floating point number
                        ;in ACC1 to a number in A
                        ;
11B9  EF                SMALL1:  RST  JT
11BA  81                         DB   81H
11BB  EF                SMALLEN: RST  JT
11BC  9D                         DB   9DH
                        ;CALL LARGEN
11BD  C0                         RET  NZ
11BE  79                         LD   A,C
11BF  30 01                      JR   NC,NOTSNEG
11C1  04                         INC  B
11C2  04                NOTSNEG: INC  B
11C3  05                         DEC  B
11C4  C9                         RET
                        ;
                        ;This is a routine to convert the floating
                        ;point number in ACC1 to an integer in BC
                        ;If the number is negative then C is returned
                        ;If out of range then NZ is returned.
                        ;
11C5  CD 128F           LARGE1:  CALL ACCSTACK
11C8  21 FDD0           LARGEN:  LD   HL,ACC1+4
11CB  01 0000                    LD   BC,0
11CE  CD 11F6                    CALL TESTEXP
11D1  28 02                      JR   Z,SAFE
11D3  D0                         RET  NC        ;Too large
11D4  AF                         XOR  A         ;Small Z,NC
11D5  D0                SAFE:    RET  NC        ;Got zero
11D6  58                         LD   E,B
11D7  3C                         INC  A
11D8  CB F8                      SET  7,B       ;This is set
11DA  3D                MOROT:   DEC  A
11DB  28 06                      JR   Z,CORBC
11DD  CB 38                      SRL  B
11DF  CB 19                      RR   C
11E1  18 F7                      JR   MOROT
11E3  CB 13             CORBC:   RL   E
11E5  9F                         SBC  A,A
11E6  C8                         RET  Z
11E7  CD 11EC                    CALL COMPBC
11EA  3C                         INC  A         ;A=0FFh
11EB  C9                         RET
                        ;
                        ;This does a 2's complement on BC
                        ;
11EC  F5                COMPBC:  PUSH AF
11ED  78                         LD   A,B
11EE  2F                         CPL
11EF  47                         LD   B,A
11F0  79                         LD   A,C
11F1  2F                         CPL
11F2  4F                         LD   C,A
11F3  03                         INC  BC
11F4  F1                         POP  AF
11F5  C9                         RET
```

```
                        ;
                        ;This tests the exponet of the number
                        ;pointed to by HL to see if it
                        ;is in the 0 to 65535 range NZ if
                        ;out of range C if zero
                        ;
11F6  7E                TESTEXP: LD   A,(HL)
11F7  A7                         AND  A
11F8  C8                         RET  Z    ;Z,NC
11F9  D6 81                      SUB  80H+1
11FB  D8                         RET  C    ;NZ,C
11FC  4F                         LD   C,A
11FD  3E 0F                      LD   A,15
11FF  91                         SUB  C    ;Number of rotations
1200  3F                         CCF
1201  D0                         RET  NC   ;NZ,NC
1202  4F                         LD   C,A
1203  AF                         XOR  A    ;Z
1204  79                         LD   A,C
1205  37                         SCF       ;C
1206  2B                         DEC  HL
1207  46                         LD   B,(HL)
1208  2B                         DEC  HL
1209  4E                         LD   C,(HL)
120A  C9                         RET
                        ;
                        ;This loads PI into OP11 HL
                        ;returns pointing at the exponent
                        ;byte BC DE undestroyed
                        ;
1208  D5                PIOP1:   PUSH DE
120C  11 141F                    LD   DE,PI
120F  CD 12B9                    CALL MOVDEOP1
1212  D1                        POP  DE
1213  21 FDD6                   LD   HL,OP11+4 ;Exponent
1216  C9                        RET
                        ;
                        ;
                        ;
1217  EF                DIVZERO: RST  ERROR
1218  21                       DB   33
1219  EF                OUTRANG: RST  ERROR
121A  22                       DB   34
121B  EF                OVERFLOW: RST  ERROR
121C  20                       DB   32
                        ;
                        ;
                        ;
                        ;IMATHS.MAC
                        ;
                        ;
                        ;
                        ;
0005                    BIOS     EQU  5
00FF                    ENDMARK  EQU  0FFH
                        ;
```

```
0030                    UNDEFINE  EQU  FRONT
0030                    FRONT     EQU  30H
000D                    CR        EQU 13
000A                    LF        EQU 10
00FF                    INPUT     EQU 0FFH
0002                    CONOUT    EQU 2
0009                    STRNG     EQU 9
0006                    DIRIO     EQU 6
0030                    ASCINO    EQU  '0'
                        ;
                        ;
                        ;This reads the number pointed
                        ;to by DE into BC if there is a
                        ;point then Z will not be set
                        ;A is deystroyed.
                        ;
                        ;
121D  E5                INT64K:   PUSH HL
121E  CD 1225                     CALL INT641
1221  44                          LD   B,H
1222  4D                          LD   C,L
1223  E1                          POP  HL
1224  C9                          RET
                        ;
1225  21 0000           INT641:   LD   HL,0
1228  1A                          LD   A,(DE)
1229  D6 30                       SUB  '0'
122B  D8                          RET  C
122C  FE 0A                       CP   10
122E  38 0D                       JR   C,INT6
1230  A7                          AND  A
1231  C9                          RET
1232  AF                SETZ:     XOR  A
1233  C9                          RET
1234  1A                INTOP:    LD   A,(DE)
1235  D6 30                       SUB  '0'
1237  38 F9                       JR   C,SETZ
1239  FE 0A                       CP   10
123B  30 F5                       JR   NC,SETZ   ;Nz is set
123D  13                INT6:     INC  DE
123E  29                          ADD  HL,HL
123F  D8                          RET  C
1240  44                          LD   B,H
1241  4D                          LD   C,L
1242  29                          ADD  HL,HL
1243  D8                          RET  C
1244  29                          ADD  HL,HL
1245  D8                          RET  C
1246  09                          ADD  HL,BC
1247  D8                          RET  C
1248  4F                          LD   C,A
1249  06 00                       LD   B,0
124B  09                          ADD  HL,BC
124C  30 E6                       JR   NC,INTOP
124E  C9                          RET
                        ;
```

```
                        ;
                        ;Tokmin looks at the byte pointed to
                        ;by DE to see if it is +- tokenised
                        ;increments if it is,Z if a +/-
                        ;
124F  1A                TOKMIN:  LD   A,(DE)
1250  FE D0                       CP   MINUS
1252  28 04                      JR   Z,ISMIN
1254  FE CF                      CP   PLUS
1256  37                         SCF
1257  C0                         RET  NZ
1258  13                ISMIN:   INC  DE
1259  C9                         RET
                        ;
                        ;This puts the number pointed to
                        ;by DE in increasing memory onto
                        ;the stack
                        ;
                        ;
125A  E5                STACKDE: PUSH HL
125B  2A FA81                    LD   HL,(CALCSTK)
125E  C5                         PUSH BC
125F  01 0005                    LD   BC,5
1262  EB                        EX   DE,HL
1263  ED B0                      LDIR
1265  EB                        EX   DE,HL
1266  22 FA81                    LD   (CALCSTK),HL
1269  C1                         POP  BC
126A  E1                         POP  HL
126B  C9                         RET
                        ;
                        ;This takes the number on the stack
                        ;and moves it to DE low bye set to
                        ;zero
                        ;
126C  CD 1274           ZDESTACK: CALL DESTACK
126F  EB                         EX   DE,HL
1270  36 00                      LD   (HL),0
1272  EB                        EX   DE,HL
1273  C9                         RET
                        ;
                        ;
                        ;This takes the current number on
                        ;the stack and moves it to DE
                        ;
                        ;
1274  E5                DESTACK: PUSH HL
1275  2A FA81                    LD   HL,(CALCSTK)
1278  C5                         PUSH BC
1279  01 0005                    LD   BC,5
127C  2B                         DEC  HL
127D  ED B8                      LDDR
127F  23                         INC  HL
1280  22 FA81                    LD   (CALCSTK),HL
1283  C1                         POP  BC
1284  E1                         POP  HL
```

```
1285  C9                        RET
                        ;
                        ;
                        ;This puts the number in ACC1
                        ;onto the stack
                        ;
                        ;
1286  D5                STACKACC: PUSH DE
1287  11 FDCC                     LD   DE,ACC1
128A  CD 125A           STA23:    CALL STACKDE
128D  D1                          POP  DE
128E  C9                          RET
                        ;
                        ;This moves the number on the
                        ;stack to ACC1
                        ;
                        ;
128F  D5                ACCSTAK:  PUSH DE
1290  11 FDD0                    LD   DE,ACC1+4
1293  CD 126C            STA24:    CALL ZDESTACK
1296  D1                          POP  DE
1297  C9                          RET
                        ;
                        ;
                        ;This puts the number in OP11
                        ;onto the stack
                        ;
                        ;
1298  D5                STACKOP1: PUSH DE
1299  11 FDD2                     LD   DE,OP11
129C  18 EC                       JR   STA23
                        ;
                        ;This moves the number on the stack
                        ;to OP11
                        ;
                        ;
129E  D5                OP1STACK: PUSH DE
129F  11 FDD6                     LD   DE,OP11+4
12A2  18 EF                       JR   STA24
                        ;
                        ;
                        ;This moves the number pointed to
                        ;by DE to ACC.
                        ;
12A4  11 141F           PIACC:    LD   DE,PI
12A7  E5                MOVEDE:   PUSH HL
12A8  21 FDCB                     LD   HL,ACC1-1
12AB  C5                DOMOVE:   PUSH BC
12AC  01 0005                     LD   BC,5
12AF  36 00                       LD   (HL),0
12B1  23                          INC  HL
12B2  EB                          EX   DE,HL
12B3  ED B0                       LDIR
12B5  EB                          EX   DE,HL
12B6  C1                          POP  BC
12B7  E1                          POP  HL
```

```
12B8  C9                      RET
                        ;
                        ;This moves the number pointed to
                        ;by DE to OP11
                        ;
12B9  E5                MOVDEOP1: PUSH HL
12BA  21 FDD1                     LD   HL,OP11-1
12BD  18 EC                       JR   DOMOVE
                        ;
                        ;
                        ;This saves the number pointed to
                        ;by HL at exponent
                        ;
                        ;
12BF  21 FDD0           SAVEACC:  LD   HL,ACC1+4
12C2  D5                SAVEF1:   PUSH DE
12C3  11 FDDD                     LD   DE,MEM1+4
12C6  C5                SAVE:     PUSH BC
12C7  E5                          PUSH HL
12C8  01 0005                     LD   BC,5
12CB  ED B8                       LDDR
12CD  E1                          POP  HL
12CE  C1                          POP  BC
12CF  D1                          POP  DE
12D0  C9                          RET
                        ;
                        ;
                        ;This returns the number to
                        ;the place pointed to by HL
                        ;at exponent
                        ;
12D1  21 FDD0           RETACC:   LD   HL,ACC1+4
12D4  D5                RETF1:    PUSH DE
12D5  11 FDDD                     LD   DE,MEM1+4
12D8  E5                REF:      PUSH HL
12D9  C5                          PUSH BC
12DA  01 0005                     LD   BC,5
12DD  EB                          EX   DE,HL
12DE  ED B8                       LDDR
12E0  EB                          EX   DE,HL
12E1  36 00                       LD   (HL),0
12E3  C1                          POP  BC
12E4  E1                          POP  HL
12E5  D1                          POP  DE
12E6  C9                          RET
                        ;
                        ;This copies the six bytes pointed
                        ;to by HL to DE
                        ;
12E7  C5                COPY6:    PUSH BC
12E8  E5                          PUSH HL
12E9  D5                          PUSH DE
12EA  01 0006                     LD   BC,6
12ED  ED B8                       LDDR
12EF  D1                          POP  DE
12F0  E1                          POP  HL
```

```
12F1  C1                    POP  BC
12F2  C9                    RET
                    ;
                    ;
                    ;This copies ACC1 to OP1
                    ;
12F3  D5            COPYACC: PUSH DE
12F4  E5                     PUSH HL
12F5  21 FDD0                LD   HL,ACC1+4
12F8  11 FDD6                LD   DE,OP11+4
12FB  CD 12E7                CALL COPY6
12FE  E1                     POP  HL
12FF  D1                     POP  DE
1300  C9                     RET
                    ;
1301  EF            PEEK:    RST  JT
1302  9D                     DB   9DH
1303  C2 1219       OUT112:  JP   NZ,OUTRANG
1306  0A                     LD   A,(BC)
1307  18 08                  JR   STA

1309  CD 11BB       INP:     CALL SMALLEN
130C  20 F5                  JR   NZ,OUT112
130E  4F                     LD   C,A
130F  ED 78                  IN   A,(C)
                    ;
                    ;
                    ;This stores A in floating point
                    ;not 2's complement.
                    ;
1311  4F            STA:     LD   C,A
1312  AF                     XOR  A
1313  79                     LD   A,C
1314  F5                     PUSH AF
1315  18 07                  JR   GOTPOS
                    ;
                    ;This converts the number in A
                    ;to a floting point number in ACC1
                    ;as 2's complement
                    ;
1317  CB 7F         STAC:    BIT  7,A
1319  F5                     PUSH AF
131A  28 02                  JR   Z,GOTPOS
131C  ED 44                  NEG
131E  21 0000       GOTPOS:  LD   HL,0
1321  22 FDCC                LD   (ACC1),HL
1324  CD 132D                CALL ONZ
1327  F1                     POP  AF
1328  C8                     RET  Z
1329  CB FE                  SET  7,(HL)
132B  C9                     RET
                    ;
132C  79            ONEBYZ:  LD   A,C
132D  A7            ONZ:     AND  A
132E  28 14                  JR   Z,DOZ
1330  0E 88                  LD   C,80H+8
```

```
1332  17           NOEL:  RLA          } Rotate m.s. 1 into
1333  0D                  DEC  C       } carry decrementing exponent
1334  30 FC               JR   NC,NOEL } at each step -
1336  0C                  INC  C       - Increment exponent
1337  3F                  CCF          } Clear carry and rotate
1338  1F                  RRA          } it back into A,
1339  21 FDD0             LD   HL,ACC1+4 } Store exponent
133C  71                  LD   (HL),C  }
133D  2B                  DEC  HL      } Store leading bits
133E  77                  LD   (HL),A  }
133F  2B                  DEC  HL      } Zero next byte
1340  36 00              LD   (HL),0  }
1342  23                  INC  HL      - Restore pointer
1343  C9                  RET
1344  67            DOZ:  LD   H,A     } Zero HL
1345  6F                  LD   L,A     }
1346  18 1A               JR   DOZ1

                    ;
                    ;This stores the number in BC to ACC1
                    ;
1348  21 0000       STBC:  LD   HL,0     } Zero 2 L.S bytes
134B  22 FDCC              LD   (ACC1),HL }
134E  78                   LD   A,B      } If 0 zero last
134F  A7                   AND  A        } as a single byte
1350  28 DA                JR   Z,ONEBYZ }
1352  60                   LD   H,B      } Copy to HL
1353  69                   LD   L,C      }
1354  3E 90                LD   A,80H+16 - Initial exponent
1356  CB 7C                BIT  7,H      } Test for already
1358  20 06                JR   NZ,MISOUT } normalised
135A  29            OUT1:  ADD  HL,HL    }
135B  3D                   DEC  A        } Normalise
135C  CB 7C                BIT  7,H      } Number
135E  28 FA                JR   Z,OUT1   }
1360  CB BC          MISOUT: RES  7,H
1362  22 FDCE        DOZ1:  LD   (ACC1+2),HL } Store result
1365  21 FDD0              LD   HL,ACC1+4 }
1368  77                   LD   (HL),A    }
1369  2B                   DEC  HL        }
136A  C9                   RET

                    ;
                    ;This converts the number in A HL,HL'
                    ;to a floating point number in
                    ;ACC1, A HL HL' is assumed to be
                    ;positive initially.C is set for start
                    ;exponent.B has the exponent to go in ACC.
                    ;No rounding is done.
                    ;
136B  ED 43 FDCF    STHLHLD:  LD   (ACC1+3),BC
136F  C3 0F58                 JP   STAHLHL
                    ;

                    ;
                    ;
                    ;
                    ;IMATHS.MAC
```

```
                        ;
                        ;
                        ;
00CF                    PLUS EQU  207
00D1                    MULT EQU  209
00D0                    MINUS    EQU  208
                        ;
                        ;
                        ;
                        ;This calculates the exponent
                        ;of 2 to the power of
                        ;number between 0 and 1
                        ;
1372  11 2494           EXP01:    LD   DE,LOGE2
1375  EF                          RST  JT
1376  CE 88                       DB   0CEH,88H
                        ;         CALL MOVDEOP1
                        ;         CALL MULPLY    ;ACC1*LOG e 2
1378  21 2499                     LD   HL,ENUMBERS
137B  18 55                       JR   POLYNOM
                        ;
                        ;
                        ;
                        ;This calculates the cosine of an
                        ;angle
                        ;
137D  EF                COSINE:   RST  JT
137E  94                          DB   94H
                        ;CALL PIOP1
137F  35                          DEC  (HL)      ;PI/2
1380  EF                          RST  JT
1381  8B                          DB   8BH
                        ;         CALL ADD
                        ;
                        ;
                        ;This calculates the SINE of an
                        ;angle in ACC between 0 and PI/2
                        ;
1382  CD 0E56           SINE:     CALL ABS
1385  F5                          PUSH AF
1386  CD 120B                     CALL PIOP1
1389  34                          INC  (HL)
138A  CD 1183                     CALL EXCHANGE
138D  CD 0FFA                     CALL REMAINDER
1390  CD 120B                     CALL PIOP1
1393  CD 115D                     CALL COMPAO
1396  38 06                       JR   C,NOTNEG
1398  F1                SUBACC:   POP  AF
1399  3F                          CCF
139A  F5                          PUSH AF
139B  EF                          RST  JT
139C  C6 90                       DB   0C6H,90H
                        ;         CALL EXCHANGE
                        ;         CALL SUBTRACT
139E  CD 120B           NOTNEG:   CALL PIOP1
13A1  35                          DEC  (HL)
```

```
13A2  CD 115D                    CALL COMPAD
13A5  38 05                      JR   C,NOTPI2
13A7  23              SUBPI2:    INC  HL
13A8  34                         INC  (HL)      ;pi
13A9  CD 0E83                    CALL SUBTRACT
13AC  21 13FB         NOTPI2:    LD   HL,SNUMBERS
13AF  CD 13C1                    CALL POLY2
13B2  F1                         POP  AF
13B3  D0                         RET  NC
13B4  C3 0FEA                    JP   COMPACC
                      ;
                      ;
                      ;This calculates the real TAN value
                      ;by first calculating the SIN
                      ;then COS and then dividing them
                      ;
13B7  EF              TANGENT:   RST  JT
13B8  C5 D7 C6 C1                DB   0C5H,0D7H,0C6H,0C1H
13BC  D6 D8 C4 8A                DB   0D6H,0D8H,0C4H,8AH
                      ;          CALL STACKACC
                      ;          CALL SINE
                      ;          CALL EXCHANGE
                      ;          CALL ACCSTACK
                      ;          CALL STACKOP1
                      ;          CALL COSINE
                      ;          CALL OP1STACK
                      ;          CALL DIVIDE
13C0  C9                         RET
                      ;
                      ;This calculates a polynomial with
                      ;terms of x+x^3/2+x^5/3
                      ;
13C1  E5              POLY2:     PUSH HL
13C2  CD 1286                    CALL STACKACC
13C5  CD 1008                    CALL SQUARE    ;Got square terms
13C8  E1                         POP  HL
13C9  CD 13D2                    CALL POLYNOM
13CC  CD 129E                    CALL OP1STACK
13CF  C3 100B                    JP   MULPLY
                      ;
                      ;This executes the polynomial pointed
                      ;to by HL the first byte gives how many
                      ;there are.Generally a 1+x/2+x/3+x/4 type
                      ;
13D2  46              POLYNOM:   LD   B,(HL)
13D3  23                         INC  HL
13D4  EB                         EX   DE,HL
13D5  CD 12BF                    CALL SAVEACC   ;Save X
13D8  CD 12A7                    CALL MOVEDE    ;Get first one
13DB  C5              PLOOP:     PUSH BC
13DC  D5                         PUSH DE
13DD  21 FDD6                    LD   HL,OP11+4
13E0  CD 12D4                    CALL RETF1
13E3  CD 100B                    CALL MULPLY
13E6  D1                         POP  DE
13E7  CD 12B9         ALOOP:     CALL MOVDEOP1
```

```
13EA  D5                    PUSH DE
13EB  CD 0E86               CALL ADD
13EE  D1                    POP  DE
13EF  C1                    POP  BC
13F0  10 E9                 DJNZ PLOOP
13F2  C9                    RET
                      ;
13F3  11 2452         ALOP1:  LD   DE,ATNT2
13F6  06 03                 LD   B,3
13F8  C5                    PUSH BC
13F9  18 EC                 JR   ALOOP
                      ;
                      ;
                      ;
                      ;This calculates the half tan value
                      ;
                      ;
                      ;HALFTAN:  PUSH DE
                      ;          LD   HL,PI2
                      ;          LD   DE,OP11
                      ;          LD   BC,5
                      ;          LDIR
                      ;          CALL EXCHANGE
                      ;          CALL REMAINDER
                      ;          LD   HL,OP1+8
                      ;          CALL SETZER
                      ;          LD   (HL),0E3H
                      ;          DEC  HL
                      ;          LD   (HL),40H
                      ;          CALL EXCHANGE
                      ;          CALL DIVIDE
                      ;          POP  DE
                      ;          LD   HL,ACC1+4
                      ;          LD   B,(HL)
                      ;          LD   A,0DEH+6
                      ;          SUB  B
                      ;          CP   7
                      ;          PUSH AF
                      ;          JR   NC,NOSUB
                      ;          LD   B,A
                      ;          DEC  HL
                      ;          LD   A,(HL)
                      ;DOMORE:   SRL  A
                      ;          DJNZ DOMORE
                      ;          DEC  A
                      ;          POP  BC
                      ;          PUSH AF
                      ;          PUSH BC
                      ;          LD   C,A
                      ;          LD   B,0
                      ;          PUSH BC
                      ;          LD   HL,OP1+8
                      ;          CALL SETZER
                      ;          POP  BC
                      ;          LD   HL,TANTABL
                      ;          ADD  HL,BC
```

```
                ;       ADD  HL,BC
                ;       LD   BC,2
                ;       PUSH DE
                ;       LD   DE,OP11+3
                ;       LDIR
                ;       PUSH HL
                ;       CALL ADD
                ;       POP  HL
                ;       POP  DE
                ;NOSUB: CALL SMALLTAN
                ;       CALL STACKACC  ;Save it
                ;       LD   HL,OP1+8
                ;       CALL SETZER
                ;       POP  AF
                ;       JR   NC,LESSTO
                ;       POP  AF
                ;       LD   HL,VALTAN
                ;       LD   B,0
                ;       LD   C,A
                ;       ADD  HL,BC
                ;       ADD  HL,BC
                ;       ADD  HL,BC
                ;       ADD  HL,BC
                ;       ADD  HL,BC
                ;       ADD  HL,BC
                ;       LD   DE,OP11
                ;       LD   BC,5
                ;       LDIR
                ;LESSTO: CALL STACKOP1
                ;       CALL ADD
                ;       CALL OP1STACK
                ;       LD   HL,ACC1+4
                ;       CALL SAVEF1
                ;       CALL ACCSTACK
                ;       CALL MULPLY
                ;       LD   HL,OP1+8
                ;       CALL SETZER
                ;       LD   (HL),0E2H
                ;       DEC  HL
                ;       LD   (HL),40H
                ;       LD   HL,ACC
                ;       CALL COMP1
                ;       CALL ADD
                ;       LD   HL,OP11+4
                ;       CALL RETF1
                ;       CALL DIVIDE
                ;       RET
                ;
                ;This is a routine to calculate TAN X
                ;X is assumed to be in ACC1
                ;using a cubic to approximate
                ;B*X+C*X^2+D*X^3
                ;calculated as
                ;TAN X=X*(B+X*(C+D*X))
                ;The numbers A,B,C,D are stored these
                ;are then put onto the calculator stack
                ;
```

```
                        ;TOP OF STACK      D   *
                        ;                  X   +
                        ;                  C   *
                        ;                  X   +
                        ;                  B   *
                        ;                  X
                        ;Bottom stack
                        ;
                        ;
00FF                    FINAR    EQU  0FFH
                        ;
                        ;DE points to the top of the
                        ;operator stack
                        ;
                        ;SMALLTAN: LD   HL,TNUMBERS
                        ;          INC  DE
                        ;          LD   B,(HL)
                        ;          INC  HL
                        ;          EX   DE,HL
                        ;          LD   (HL),FINAR
                        ;          INC  HL
                        ;STNUM:    CALL STACKACC
                        ;          LD   (HL),MULT
                        ;          INC  HL
                        ;          CALL STACKDE
                        ;          LD   (HL),PLUS
                        ;          INC  HL
                        ;          DJNZ STNUM
                        ;          DEC  HL
                        ;          DEC  HL
                        ;          EX   DE,HL
                        ;          CALL ARITH
                        ;          RET
                        ;
                        ;
                        ;This calculates 1-(ACC)^2
                        ;
                        ;ONEMACC2: CALL SQUARE
                        ;          LD   HL,ACC
                        ;          CALL COMP1
                        ;          LD   HL,OP1+8
                        ;          CALL SETZER
                        ;          LD   (HL),0E2H
                        ;          DEC  HL
                        ;          LD   (HL),40H
                        ;          CALL ADD
                        ;          RET
                        ;
                        ;
                        ;This calculates 1+(ACC)^2
                        ;
                        ;ONEPACC2: CALL SQUARE
                        ;          LD   HL,OP1+8
                        ;          CALL SETZER
                        ;          LD   (HL),0E2H
                        ;          DEC  HL
```

```
                        ;       LD   (HL),40H
                        ;       CALL ADD
                        ;       RET
                        ;
                        ;
                        ;
                        ;
                        ;
0010                    .RADIX  16
                        ;
                        ;
13FB  06                SNUMBERS: DB   6
13FC  9D 30 92 30               DB   9D,30,92,30,60     ; 1/13!
1400  60
1401  40 2B 32 D7               DB   40,2B,32,0D7,67    ;-1/11!
1405  67
1406  2B 1D EF 3B               DB   2B,1D,0EF,3B,6E    ; 1/9!
140A  6E
140B  D0 00 0D D0               DB   0D0,0,0D,0D0,74    ;-1/7!
140F  74
1410  89 88 88 08               DB   89,88,88,8,7A      ; 1/5!
1414  7A
1415  AB AA AA AA               DB   0AB,0AA,0AA,0AA,7E ;-1/3!
1419  7E
141A  00 00 00 00               DB   0,0,0,0,81
141E  81
                        ;
141F  A2 DA 0F 49       PI:     DB   0A2,0DA,0F,49,82   ;PI
1423  82
                        ;
000A                    .RADIX  10
                        ;
                        ;
                        END
```

```
                        .Z80
0000'                   ASEG
                        ORG     1424H
                        ;
                        ;
FF5B                    CURSCR  EQU     0FF5BH
00BC                    GETSTR  EQU     00BCH
FD76                    IOPR    EQU     0FD76H
FA83                    KBDBUF  EQU     0FA83H
0079                    KBD     EQU     0079H
FD75                    PRORPL  EQU     0FD75H
0CAB                    PRINTX  EQU     0CABH
FE1A                    WKAREA  EQU     0FE1AH
                        ;
                        ;
                        PUBLIC EDEND
                        PUBLIC EDITOR
                        PUBLIC EDIT1
                        ;
                        ;
00FF                    ENDMARK EQU 255
0010                    SCRRST EQU 10H
007F                    DEL EQU 127
000D                    CR EQU 13
                        ;
                        ;
1424  CD 1440     EDIT1:    CALL SINS      ;SWITCH OFF INSERT
1427  18 08                 JR JEDIT1      ;ENTER IF KEY HAS BEEN PRESSED
1429  CD 1440     EDITOR:   CALL SINS      ;SWITCH OFF INSERT
142C  CD 0079     JEDIT:    CALL KBD       ;ENTRY IF NO KEY HAS BEEN PRESSED
142F  28 FB                 JR Z,JEDIT
1431  FE 0D       JEDIT1:   CP CR          ;FINISH EDIT
1433  28 0B                 JR Z,SINS
1435  FE 7F                      CP DEL
1437  20 02                      JR NZ,EDIT4
1439  3E 16                      LD A,22
143B  CD 00BC     EDIT4:    CALL GETSTR
143E  18 EC                 JR JEDIT
1440  E5          SINS:     PUSH HL
1441  2A FF5B                LD HL,(CURSCR)
1444  CB 9E                  RES 3,(HL)
1446  E1                     POP HL
1447  C9                     RET
1448  D7          EDEND:    RST 10H
1449  3A 3D 1F               DB 3AH,3DH,1FH
144C  2A FF5B                LD HL,(CURSCR)
144F  11 0005                LD DE,5
1452  19                     ADD HL,DE
1453  5E                     LD E,(HL)
1454  23                     INC HL
1455  46                     LD B,(HL)
1456  21 0000                LD HL,0
1459  19          ED1:      ADD HL,DE
145A  10 FD                DJNZ ED1
145C  E5                   PUSH HL
145D  E5                   PUSH HL
```

```
145E  C1
145F  2A FA83
1462  D7
1463  82 1B 4C
1466  3A FE1A
1469  77
146A  23
146B  0B
146C  78
146D  B1
146E  20 F2
1470  C1
1471  C5
1472  2B
1473  7E
1474  FE 20
1476  20 06
1478  0B
1479  78
147A  B1
147B  20 F5
147D  2B
147E  23
147F  36 FF
1481  E5
1482  ED 4B FA83
1486  A7
1487  ED 42
1489  EB
148A  7B
148B  E1
148C  EB
148D  C1
148E  C9
```

```
                        POP BC
                        LD HL,(KBDBUF)
PRO62:                  RST SCRRST
                        DB 82H,27,'L'
                        LD A,(WKAREA)
                        LD (HL),A
                        INC HL
                        DEC BC
                        LD A,B
                        OR C
                        JR NZ,PRO62
                        POP BC
                        PUSH BC
PRO63:                  DEC HL          ;REMOVE SPACES FROM END OF LINE
                        LD A,(HL)
                        CP 32
                        JR NZ,PRO64
                        DEC BC
                        LD A,B
                        OR C
                        JR NZ,PRO63
                        DEC HL
PRO64:                  INC HL
                        LD (HL),ENDMARK
                        PUSH HL
                        LD   BC,(KBDBUF)
                        AND  A
                        SBC  HL,BC
                        EX DE,HL
                        LD   A,E
                        POP HL
                        EX DE,HL
                        POP BC
                        RET
                ;
                ;
                END
```

A>

```
                        .Z80
0000'                   ASEG
                        ORG     148FH
                        ;
                        ;
                C       INCLUDE SCBPORTS.INC
0000            C               PGPORT EQU 0;255
                        ;
                        ;
FDCC                    ACC1    EQU     0FDCCH
0A41                    CHR1    EQU     0A41H
11EC                    COMPBC  EQU     11ECH
FE52                    DELSPR  EQU     0FE52H
0FBE                    DONIB   EQU     0FBEH
00BC                    GETSTR  EQU     00BCH
0F13                    NEGACC  EQU     0F13H
FAD2                    PAGE    EQU     0FAD2H
00CB                    SCENTR  EQU     00CBH
FFD5                    TYPTBL  EQU     0FFD5H
FE1A                    WKAREA  EQU     0FE1AH
                        ;
                        ;
                        PUBLIC  SYNPROC
                        PUBLIC  SINCOS
                        PUBLIC  SCENT
                        PUBLIC  SCLS
                        PUBLIC  SVS
                        PUBLIC  SCRVS
                        PUBLIC  SATTR
                        PUBLIC  SCOLOUR
                        PUBLIC  SINK
                        PUBLIC  SPAPER
                        PUBLIC  SADJSPR
                        PUBLIC  SMVSPR
                        PUBLIC  SSPRITE
                        PUBLIC  SCTLSPR
                        PUBLIC  SGENPAT
                        PUBLIC  SCURSOR
                        PUBLIC  SWINDOW
                        PUBLIC  SDRAW
                        PUBLIC  SARC
                        PUBLIC  SCIRCLE
                        PUBLIC  SLINE
                        PUBLIC  SDSI
                        PUBLIC  GRRDS
                        PUBLIC  CHRDS
                        PUBLIC  SPLOT
                        PUBLIC  SAANGLE
                        PUBLIC  SRANGLE
                        PUBLIC  SCENT4
                        ;
                        ;
0010                    RSTSCR  EQU     10H
0028                    RSTJT   EQU     28H
0030                    RSTGN   EQU     30H
                        ;
```

```
                        ;
                        ;
148F  21 FDD0           X256:  LD HL,ACC1+4;   MULTIPLIES ACC1 BY 256
1492  7E                       LD A,(HL)
1493  A7                       AND A
1494  C8                       RET Z
1495  C6 08                    ADD A,8
1497  77                       LD (HL),A
1498  C9                       RET
1499  D5                GEPC:  PUSH DE;       FINDS AND LOADS PLOT COORD
149A  CD 152A                  CALL EVALPO
149D  C3 1619                  JP RANGL4
14A0  21 14A9           NOTEDE:  LD HL,DEEXIT
14A3  ED 53 FE2A                 LD (WKAREA+16),DE
14A7  E3                         EX (SP),HL
14A8  E9                         JP (HL)
14A9  ED 5B FE2A        DEEXIT: LD DE,(WKAREA+16)
14AD  C9                        RET
14AE  FE 20             ESCOUT: CP 32
14B0  38 04                     JR C,ESCOU2
14B2  F5                        PUSH AF
14B3  D7                        RST RSTSCR
14B4  1B                        DB 27
14B5  F1                        POP AF
14B6  C3 00BC           ESCOU2: JP GETSTR
14B9  E1                SYNPROC:       POP HL
14BA  ED 53 FE2A                LD (WKAREA+16),DE
14BE  7E                        LD A,(HL)
14BF  CD 14AE                   CALL ESCOUT
14C2  E5                        PUSH HL
14C3  E1                SYNPR4: POP HL
14C4  23                        INC HL
14C5  7E                        LD A,(HL)
14C6  FE C9                     CP 201
14C8  28 DF                     JR Z,DEEXIT
14CA  E5                SYNPR5: PUSH HL
14CB  21 14C3                   LD HL,SYNPR4
14CE  E5                        PUSH HL
14CF  D6 10                     SUB 16
14D1  20 0F                     JR NZ,SYNF
14D3  3E FF                     LD A,255
14D5  CD 1530           SYNX:   CALL SYNGEN
14D8  3C                        INC A
14D9  28 03                     JR Z,SYNX2
14DB  B9                        CP C
14DC  38 4A                     JR C,EVPMER
14DE  79                SYNX2:  LD A,C
14DF  C3 00BC                   JP GETSTR
14E2  3D                SYNF:   DEC A;         NO MUST BE 0-15
14E3  20 04                     JR NZ,SYNS
14E5  3E 0F                     LD A,15
14E7  18 EC                     JR SYNX
14E9  3D                SYNS:   DEC A;         NO MUST BE 0-7
14EA  20 04                     JR NZ,SYNP
14EC  3E 07                     LD A,7
14EE  18 E5                     JR SYNX
```

```
14F0  3D              SYNP:   DEC A;          NO MUST BE 0-127
14F1  20 04                   JR NZ,SYNPN
14F3  3E 7F                   LD A,127
14F5  18 DE                   JR SYNX
14F7  3D              SYNPN:  DEC A
14F8  20 0C                   JR NZ,SYNL
14FA  CD 151E                 CALL EVALPN
14FD  30 01                   JR NC,SYNPN2
14FF  04                      INC B
1500  04              SYNPN2: INC B
1501  05                      DEC B
1502  20 24                   JR NZ,EVPMER
1504  18 D8                   JR SYNX2
1506  3D              SYNL:   DEC A
1507  20 11                   JR NZ,SYNALP
1509  CD 151E                 CALL EVALPN
150C  30 04                   JR NC,SYNL2
150E  78                      LD A,B
150F  C6 10                   ADD A,16
1511  47                      LD B,A
1512  78              SYNL2:  LD A,B
1513  FE 10                   CP 16
1515  30 11                   JR NC,EVPMER
1517  D7                      RST RSTSCR
1518  C0                      DB 0C0H
1519  C9                      RET
151A  3E 01           SYNALP: LD A,1;         NO MUST BE 0-1
151C  18 B7                   JR SYNX
151E  ED 5B FE2A      EVALPN: LD DE,(WKAREA+16)
1522  F7                      RST RSTGN
1523  ED 53 FE2A              LD (WKAREA+16),DE
1527  C8                      RET Z
1528  EF              EVPMER: RST 28H
1529  30                      DB 30H
152A  CD 151E         EVALPO: CALL EVALPN;    ENSURES A POSITIVE NO RETURNED
152D  D0                      RET NC
152E  18 F8                   JR EVPMER
1530  F5              SYNGEN: PUSH AF
1531  CD 152A                 CALL EVALPO
1534  05                      DEC B
1535  04                      INC B
1536  20 F0                   JR NZ,EVPMER
1538  F1                      POP AF
1539  C9                      RET
153A  3E 4C           CHRDS:  LD A,'L'
153C  CD 14AE                 CALL ESCOUT
153F  18 1E                   JR GRRDS5
1541  3E 43           GRRDS:  LD A,'C'
1543  CD 14AE                 CALL ESCOUT
1546  CD 1566                 CALL GSTOA
1549  3D                      DEC A
154A  FE 08                   CP 8
154C  30 DA                   JR NC,EVPMER
154E  3C                      INC A
154F  F5                      PUSH AF
1550  CD 1566                 CALL GSTOA
```

```
1553  F5                    PUSH AF
1554  CD 1566               CALL GSTOA
1557  F1                    POP AF
1558  47                    LD B,A
1559  D7                    RST RSTSCR
155A  C0                    DB 0C0H
155B  F1                    POP AF
155C  CD 00BC               CALL GETSTR
155F  21 FE1A         GRRDS5: LD HL,WKAREA
1562  7E                    LD A,(HL)
1563  C3 0A41               JP CHR1
1566  EF              GSTOA:  RST RSTJT
1567  9E                    DB 09EH
1568  38 BE                 JR C,EVPMER
156A  20 BC                 JR NZ,EVPMER
156C  04                    INC B
156D  05                    DEC B
156E  20 B8                 JR NZ,EVPMER
1570  79                    LD A,C
1571  C9                    RET
1572  04 0E                 DB 4,14
1574  EF              SLINE:  RST RSTJT
1575  A1                    DB 0A1H
1576  02 10 10 10           DB 2,16,16,16,16
157A  10
157B  C9                    RET
157C  D5              SDSI:   PUSH DE
157D  EF                    RST RSTJT
157E  A9                    DB 0A9H
                      ;CALL EDITOR
157F  D1                    POP DE
1580  C9                    RET
1581  02 0E                 DB 2,14
1583  EF              SCURSOR:        RST RSTJT
1584  A1                    DB 0A1H
1585  03 10 10              DB 3,16,16
1588  C9                    RET
1589  02                    DB 2
158A  EF              SPAPER: RST RSTJT
158B  A1                    DB 0A1H
158C  04 11                 DB 4,17
158E  C9                    RET
158F  02                    DB 2
1590  EF              SINK:   RST RSTJT
1591  A1                    DB 0A1H
1592  06 11                 DB 6,17
1594  C9                    RET
1595  EF              SCLS:   RST RSTJT
1596  A1                    DB 0A1H
1597  0C                    DB 12
1598  C9                    RET
1599  02 0E                 DB 2,14
159B  EF              SCTLSPR:        RST RSTJT
159C  A1                    DB 0A1H
159D  0E 12 10              DB 14,18,16
15A0  C9                    RET
```

```
15A1  0A 0E                   DB 10,14
15A3  EF                SGENPAT:    RST RSTJT
15A4  A1                      DB 0A1H
15A5  0F 12 10 10             DB 15,18,16,16,16,16,16,16,16,16,16
15A9  10 10 10 10
15AD  10 10 10
15B0  C9                      RET
15B1  02 0E                   DB 2,14
15B3  EF                SCOLOUR:    RST RSTJT
15B4  A1                      DB 0A1H
15B5  10 12 11                DB 16,18,17
15B8  C9                      RET
15B9  03 0E                   DB 3,14
15BB  EF                SADJSPR:    RST RSTJT
15BC  A1                      DB 0A1H
15BD  11 12 13 10             DB 17,18,19,16
15C1  C9                      RET
15C2  07 0E                   DB 7,14
15C4  EF                SSPRITE:    RST RSTJT
15C5  A1                      DB 0A1H
15C6  12 13 13 15             DB 18,19,19,21,21,20,20,17
15CA  15 14 14 11
15CE  C9                      RET
15CF  03 0E                   DB 3,14
15D1  EF                SMVSPR: RST RSTJT
15D2  A1                      DB 0A1H
15D3  13 11 13 12             DB 19,17,19,18
15D7  C9                      RET
15D8  02 0E                   DB 2,14
15DA  EF                SWINDOW:    RST RSTJT
15DB  A1                      DB 0A1H
15DC  14 12 10                DB 20,18,16
15DF  C9                      RET
15E0  02 0E                   DB 2,14
15E2  EF                SATTR:  RST RSTJT
15E3  A1                      DB 0A1H
15E4  41 12 16                DB 65,18,22
15E7  C9                      RET
15E8  07 0E                   DB 7,14
15EA  EF                SCRVS:  RST RSTJT
15EB  A1                      DB 0A1H
15EC  59 12 12 10             DB 89,18,18,16,16,16,16,16
15F0  10 10 10 10
15F4  C9                      RET
15F5  02                      DB 2
15F6  EF                SVS:    RST RSTJT
15F7  A1                      DB 0A1H
15F8  5A 12                   DB 90,18
15FA  C9                      RET
15FB  02                      DB 2
15FC  CD 14A0           SAANGLE: CALL NOTEDE
15FF  CD 151E                 CALL EVALPN
1602  11 FE30                 LD DE,WKAREA+22
1605  D5                AAEX:   PUSH DE
1606  18 11                   JR RANGL4;      ABSOLUTE ANGLE NOW IN WKAREA+18
1608  C9                      RET
```

```
1609  02                    DB 2
160A  CD 14A0           SRANGLE: CALL NOTEDE
160D  CD 151E                 CALL EVALPN;    RELATIVE ANGLE
1610  11 FE2C           RANGL2: LD DE,WKAREA+18
1613  EF                RANGL3: RST RSTJT
1614  8E                        DB 0BEH
                        ;CALL MOVDEOP1
1615  1B                        DEC DE
1616  D5                        PUSH DE
1617  EF                        RST RSTJT
1618  8B                        DB 08BH
                        ;CALL ADD
1619  EF                RANGL4: RST RSTJT
161A  85                        DB 085H
                        ;CALL STACKACC
161B  D1                RANGL5: POP DE
161C  EF                        RST RSTJT
161D  9F                        DB 09FH
161E  C9                        RET
                        ;JP DESTACK
161F  11 FE2C           SINCOS: LD DE,WKAREA+18;        L ON STACK
1622  D5                        PUSH DE
1623  EF                        RST RSTJT
1624  DC D7 C4 D6               DB 0DCH,0D7H,0C4H,0D6H,0C8H,0C4H,0C5H,096H
1628  C8 C4 C5 96

                        ;CALL MOVEDE
                        ;CALL SINE
                        ;CALL OP1STACK
                        ;CALL STACKOP1
                        ;CALL MULPLY
                        ;CALL OP1STACK
                        ;CALL STACKACC
                        ;CALL STACKOP1
162C  D1                        POP DE
162D  EF                        RST RSTJT
162E  DC D8 C4 C8               DB 0DCH,0D8H,0C4H,0C8H,085H
1632  85
1633  C9                        RET
                        ;CALL MOVEDE
                        ;CALL COSINE
                        ;CALL OP1STACK
                        ;CALL MULPLY
                        ;JP STACKACC
1634  EF                CUPT:   RST RSTJT
1635  9C                        DB 09CH
                        ;CALL MOVEDE;   WORKS OUT CURRENT PLOT POSITION
1636  18 07                     JR NEWPT2
1638  D5                NEWPT:  PUSH  DE;       CALCS NEW PT AS OLD+L*SINA OR L*COSA
1639  EF                        RST RSTJT
163A  81                        DB 081H
                        ;CALL ACCSTACK
163B  D1                        POP DE
163C  CD 1613                   CALL RANGL3
163F  CD 148F           NEWPT2: CALL X256
1642  EF                        RST RSTJT
1643  9D                        DB 09DH
```

```
                        ;CALL LARGEN
1644  38 08                     JR C,NEWPTE
1646  20 06                     JR NZ,NEWPTE
1648  79                        LD A,C
1649  07                        RLCA
164A  78                        LD A,B
164B  CE 00                     ADC A,0
164D  D0                        RET NC
164E  EF                  NEWPTE: RST 28H;    POINT OFF SCREEN
164F  32                        DB 50
1650  CD 152A             PULL2:  CALL EVALPO;    PULLS A NO IN RANGE 0-256 & MULTS BY 256
1653  18 EA                     JR NEWPT2
1655  CD 152A             GLSTPT: CALL EVALPO
1658  EF                        RST RSTJT
1659  85                        DB 085H
165A  11 FE31              STSTPT: LD DE,WKAREA+23;       STORES CURRENT POINT IN (LSTPT)
165D  CD 1634                   CALL CUPT
1660  6F                        LD L,A
1661  E5                        PUSH HL
1662  11 FE36                   LD DE,WKAREA+28
1665  CD 1634                   CALL CUPT
1668  E1                        POP HL
1669  67                        LD H,A
166A  22 FE3D                   LD (WKAREA+35),HL
166D  C9                        RET
166E  02 0E                     DB 2,14
1670  CD 14A0              SPLOT:  CALL NOTEDE
1673  11 FE35                   LD DE,WKAREA+27;       PLOT A POINT & RESET XCORD & YCORD
1676  CD 1499                   CALL GEPC
1679  11 FE3A                   LD DE,WKAREA+32
167C  CD 1499                   CALL GEPC
167F  CD 165A                   CALL STSTPT
1682  EB                        EX DE,HL
1683  18 2B                     JR PTOUT
1685  C9                        RET
1686  02                        DB 2
1687  CD 14A0              SDRAW:  CALL NOTEDE
168A  CD 1655                   CALL GLSTPT;    DRAWS A LINE LENGTH L FROM CURRENT COORD
168D  EF                  DRAW2:  RST RSTJT
168E  BF                        DB 0BFH
                        ;CALL SINCOS
168F  11 FE31                   LD DE,WKAREA+23
1692  CD 1638                   CALL NEWPT
1695  5F                        LD E,A
1696  D5                        PUSH DE
1697  11 FE36                   LD DE,WKAREA+28
169A  CD 1638                   CALL NEWPT
169D  D1                        POP DE
169E  57                        LD D,A
169F  2A FE3D             DRAW3:  LD HL,(WKAREA+35)
16A2  A7                        AND A
16A3  ED 52                     SBC HL,DE
16A5  C8                        RET Z;          SEES IF END THE SAME AS START
16A6  23                        INC HL
16A7  24                        INC H;          CHECK IF CAN GET AWAY WITH POINT
16A8  3E 02                    LD A,2
```

```
16AA  BC                        CP H
16AB  38 07                     JR C,LNOUT
16AD  BD                        CP L
16AE  38 04                     JR C,LNOUT
16B0  D7              PTOUT:    RST RSTSCR;     PLOTS A POINT
16B1  01                        DB 1
16B2  18 0D                     JR DEOUT
16B4  2A FE3D         LNOUT:    LD HL,(WKAREA+35);      DRAWS A LINE ON SCREEN
16B7  EB                        EX DE,HL
16B8  CD 16B0                   CALL PTOUT;     PLOT POINT IN CASE OVER FLAG SET
16BB  D7                        RST RSTSCR
16BC  02                        DB 2
16BD  CD 16C5                   CALL DEOUT2
16C0  EB                        EX DE,HL
16C1  ED 53 FE3D      DEOUT:    LD (WKAREA+35),DE
16C5  C5              DEOUT2:   PUSH BC
16C6  42                        LD B,D
16C7  4B                        LD C,E
16C8  D7                        RST RSTSCR
16C9  C0                        DB 0C0H
16CA  C1                        POP BC
16CB  C9                        RET
16CC  02 0E                     DB 2,14
16CE  CD 14A0         SARC:     CALL NOTEDE
16D1  CD 1802                   CALL CLWK
16D4  CD 1655                   CALL GLSTPT;    DRAWS AN ARC
16D7  CD 151E                   CALL EVALPN
16DA  21 FDD0                   LD HL,ACC1+4
16DD  7E                        LD A,(HL)
16DE  A7                        AND A
16DF  28 AC                     JR Z,DRAW2
16E1  EF                        RST RSTJT
16E2  85                        DB 085H
                      ;CALL STACKACC
16E3  CD 148F                   CALL X256
16E6  EF                        RST RSTJT
16E7  9D                        DB 09DH
                      ;CALL LARGEN
16E8  C2 1528                   JP NZ,EVPMER
16EB  21 FE52                   LD HL,DELSPR
16EE  36 00                     LD (HL),0
16F0  30 05                     JR NC,ARC2
16F2  CB F6                     SET 6,(HL);     REVERSE IF NEGATIVE
16F4  CD 11EC                   CALL COMPBC
16F7  C5              ARC2:     PUSH BC
16F8  EF                        RST RSTJT
16F9  C1 C4 C5 CA               DB 0C1H,0C4H,0C5H,0CAH,0C5H,0C5H,0FFH,0C1H,0A2H
16FD  C5 C5 FF C1
1701  A2

                      ;CALL ACCSTACK; CALCULATE EFFECTIVE RADIUS
                      ;CALL OP1STACK
                      ;CALL STACKACC;   DIVIDE ACC=OP1/ACC
                      ;CALL DIVIDE
                      ;CALL STACKACC
                      ;CALL STACKACC
                      ;CALL SINCOS
```

```
                        ;CALL ACCSTACK
                        ;CALL COMPACC
1702  21 FE1A                   LD HL,WKAREA
1705  01 0005                   LD BC,5
1708  CD 1776                   CALL SUARCW;    PUTS Y START POS IN WKAREA
170B  EF                        RST RSTJT
170C  A2                        DB 0A2H
                        ;CALL COMPACC
170D  11 FE36                   LD DE,WKAREA+28
1710  CD 1613                   CALL RANGL3;    YCORD CONTAINS YC
1713  21 FE24                   LD HL,WKAREA+10
1716  01 0003                   LD BC,3
1719  CD 1776                   CALL SUARCW;    YC IN WKAREA
171C  EF                        RST RSTJT
171D  81                        DB 081H
                      ;CALL ACCSTACK
171E  21 FE1F                   LD HL,WKAREA+5
1721  01 FFFB                   LD BC,-5
1724  CD 1776                   CALL SUARCW;    X START IN WKAREA
1727  EF                        RST RSTJT
1728  A2                        DB 0A2H
                      ;CALL COMPACC
1729  11 FE31                   LD DE,WKAREA+23
172C  CD 1613                   CALL RANGL3
172F  21 FE27                   LD HL,WKAREA+13
1732  01 FFFD                   LD BC,-3
1735  CD 1776                   CALL SUARCW;    XC IN WKAREA
1738  EF                        RST RSTJT
1739  C4 C1 96                  DB 0C4H,0C1H,096H
                     ;CALL OP1STACK
                     ;CALL ACCSTACK; SETTING UP TO CALCULATE END ANGLE
                     ;CALL STACKOP1
173C  CD 1610                   CALL RANGL2
173F  EF                        RST RSTJT
1740  FF C1 A2                  DB 0FFH,0C1H,0A2H
                     ;CALL SINCOS
                     ;CALL ACCSTACK
                     ;CALL COMPACC
1743  11 FE36                   LD DE,WKAREA+28
1746  CD 1613                   CALL RANGL3
1749  EF                        RST RSTJT
174A  81                        DB 081H
                     ;CALL ACCSTACK
174B  11 FE31                   LD DE,WKAREA+23
174E  CD 1613                   CALL RANGL3
1751  3A FE52                   LD A,(DELSPR)
1754  C1                        POP BC
1755  CB 6F                     BIT 5,A
1757  20 0E                     JR NZ,ARC4
1759  78                        LD A,B
175A  B1                        OR C
175B  28 0A                     JR Z,ARC4
175D  21 FFFF                   LD HL,-1
1760  ED 53 FE3B                LD (WKAREA+33),DE
1764  CD 1836                   CALL CIRC2;     ARC IS PLOTTED AS CIRCUM OF CIRC
1767  2A FE3D             ARC4: LD HL,(WKAREA+35)
```

```
176A  E5                    PUSH HL
176B  CD 165A               CALL STSTPT
176E  EB                    EX DE,HL
176F  E1                    POP HL
1770  22 FE3D               LD (WKAREA+35),HL
1773  C3 169F               JP DRAW3
1776  3A FE52         SUARCW: LD A,(DELSPR);  SETS UP WORK AREA DURING ARC
1779  CB 6F                 BIT 5,A
177B  C0                    RET NZ
177C  CB 77                 BIT 6,A;        ARE X & Y REVERSED
177E  20 01                 JR NZ,SUARC2
1780  09                    ADD HL,BC
1781  11 FDD0         SUARC2: LD DE,ACC1+4
1784  1A                    LD A,(DE)
1785  FE 90                 CP 144;         IS CENTRE MILES OFF SCREEN
1787  38 06                 JR C,SUARC3
1789  21 FE52               LD HL,DELSPR
178C  CB EE                 SET 5,(HL)
178E  C9                    RET
178F  E5                SUARC3: PUSH HL
1790  D5                    PUSH DE
1791  EF                    RST RSTJT
1792  85                    DB 085H
                        ;CALL STACKACC
1793  E1                    POP HL
1794  3E 90                 LD A,144
1796  96                    SUB (HL)
1797  47                    LD B,A
1798  2B                    DEC HL
1799  E5                    PUSH HL
179A  4E                    LD C,(HL)
179B  CB FE                 SET 7,(HL)
179D  CD 0FBE               CALL DONIB
17A0  CB 79                 BIT 7,C
17A2  C4 0F13               CALL NZ,NEGACC
17A5  E1                    POP HL
17A6  D1                    POP DE
17A7  06 03                 LD B,3
17A9  7E                SUARCL: LD A,(HL)
17AA  12                    LD (DE),A
17AB  2B                    DEC HL
17AC  13                    INC DE
17AD  10 FA                 DJNZ SUARCL
17AF  EF                    RST RSTJT
17B0  81                    DB 081H
17B1  C9                    RET
                        ;JP ACCSTACK
17B2  21 FE29           ABCOX2: LD HL,WKAREA+15;FINDS ABS X/Y PTS DURING CIRC &ARC
17B5  01 FE21                 LD BC,WKAREA+7
17B8  CD 17C0                 CALL ABCORD
17BB  53                      LD D,E
17BC  0B                      DEC BC
17BD  0B                      DEC BC
17BE  0B                      DEC BC
17BF  2B                      DEC HL
17C0  0A                ABCORD: LD A,(BC);      CONVERT XY TO ABS
```

```
17C1  86                    ADD A,(HL)
17C2  2B                    DEC HL
17C3  0B                    DEC BC
17C4  D5                    PUSH DE
17C5  57                    LD D,A
17C6  0A                    LD A,(BC)
17C7  8E              ABC02:ADC A,(HL)
17C8  5F                    LD E,A
17C9  2B                    DEC HL
17CA  0B                    DEC BC
17CB  0A                    LD A,(BC)
17CC  8E                    ADC A,(HL)
17CD  CB 7A                 BIT 7,D
17CF  28 04                 JR Z,ABC03
17D1  1C                    INC E
17D2  20 01                 JR NZ,ABC03
17D4  3C                    INC A
17D5  A7              ABC03:AND A
17D6  C2 164E               JP NZ,NEWPTE
17D9  7B                    LD A,E
17DA  D1                    POP DE
17DB  5F                    LD E,A
17DC  C9                    RET
17DD  CD 17B2         NXTPT:CALL ABCOX2;NEXT PT FOR CIRCLE & ARC
17E0  3A FE52               LD A,(DELSPR)
17E3  07                    RLCA
17E4  07                    RLCA
17E5  30 04                 JR NC,NXTPT2
17E7  7B                    LD A,E
17E8  5A                    LD E,D
17E9  57                    LD D,A
17EA  A7                    AND A
17EB  2A FE3B         NXTPT2:LD HL,(WKAREA+33)
17EE  ED 52                 SBC HL,DE
17F0  C8                    RET Z
17F1  C3 169F               JP DRAW3;       TO DO THE PLOTTING
17F4  AF              PU2LD:XOR A;          PICKS UP NXT COORD & PUTS IN WKAREA
17F5  12                    LD (DE),A
17F6  13                    INC DE
17F7  D5                    PUSH DE
17F8  CD 1650               CALL PULL2
17FB  E1                    POP HL
17FC  70                    LD (HL),B
17FD  23                    INC HL
17FE  71                    LD (HL),C
17FF  23                    INC HL
1800  EB                    EX DE,HL
1801  C9                    RET
1802  21 FE1A         CLWK: LD HL,WKAREA
1805  06 10                 LD B,16
1807  36 00           CLWK2:LD (HL),0
1809  23                    INC HL
180A  10 FB                 DJNZ CLWK2
180C  C9                    RET
180D  03 0E                 DB 3,14
180F  CD 14A0       SCIRCLE:CALL NOTEDE
```

```
1812  CD 1802                   CALL CLWK
1815  11 FE24                   LD DE,WKAREA+10
1818  CD 17F4                   CALL PU2LD
181B  CD 17F4                   CALL PU2LD
181E  11 FE1A                   LD DE,WKAREA
1821  CD 17F4                   CALL PU2LD
1824  21 FE52                   LD HL,DELSPR
1827  CB B6                     RES 6,(HL)
1829  CD 17B2                   CALL ABCOX2
182C  CD 16B0                   CALL PTOUT
182F  ED 53 FE3B                LD (WKAREA+33),DE
1833  01 0647                   LD BC,1607
1836  C5              CIRC2:    PUSH BC
1837  11 FE1D         CIRC2A:   LD DE,WKAREA+3
183A  21 FE23                   LD HL,WKAREA+9
183D  06 04                     LD B,4
183F  A7                        AND A;          CALC Y=Y+KX
1840  1A              CIRC3:    LD A,(DE)
1841  8E                        ADC A,(HL)
1842  77                        LD (HL),A
1843  2B                        DEC HL
1844  1B                        DEC DE
1845  10 F9                     DJNZ CIRC3
1847  7E                        LD A,(HL);      TIDY UP HI-BYTE
1848  CE 00                     ADC A,0
184A  77                        LD (HL),A
184B  13                        INC DE
184C  1A                        LD A,(DE)
184D  07                        RLCA
184E  9F                        SBC A,A
184F  86                        ADD A,(HL)
1850  77                        LD (HL),A
1851  11 FE1E                   LD DE,WKAREA+4
1854  21 FE22                   LD HL,WKAREA+8
1857  06 04                     LD B,4
1859  A7                        AND A
185A  1A              CIRC4:    LD A,(DE)
185B  9E                        SBC A,(HL)
185C  12                        LD (DE),A
185D  2B                        DEC HL
185E  1B                        DEC DE
185F  10 F9                     DJNZ CIRC4;     X=X-KY
1861  1A                        LD A,(DE)
1862  DE 00                     SBC A,0
1864  12                        LD (DE),A
1865  23                        INC HL
1866  7E                        LD A,(HL)
1867  07                        RLCA
1868  1A                        LD A,(DE)
1869  CE 00                     ADC A,0
186B  12                        LD (DE),A
186C  CD 17DD                   CALL NXTPT
186F  C1                        POP BC
1870  0B                        DEC BC
1871  78                        LD A,B
1872  B1                        OR C
```

```
1873  20 C1                    JR NZ,CIRC2
1875  C9                       RET
1876  E5              SCENT:   PUSH HL
1877  21 00CB                  LD HL,SCENTR
187A  E3                       EX (SP),HL
187B  E5                       PUSH HL;        COMMON ENTRY POINT FOR
187C  D5                       PUSH DE;        DIFFERENT SCREEN TYPES
187D  F5                       PUSH AF
187E  DD 7E 00                 LD A,(IX+0);    DEDUCE SCREEN TYPE FROM UPPER
1881  07                       RLCA;           3 BITS
1882  07                       RLCA
1883  07                       RLCA
1884  E6 07                    AND 7
1886  5F                       LD E,A;         SCREEN TYPE
1887  16 00                    LD D,0
1889  21 FFD5                  LD HL,TYPTBL
188C  19                       ADD HL,DE
188D  19                       ADD HL,DE
188E  19                       ADD HL,DE
188F  7E              SCENT4:  LD A,(HL)
1890  23                       INC HL
1891  5E                       LD E,(HL);      FIND ENTRY POINT FOR SPECIFIC
1892  23                       INC HL;         ROUTINE (SUPPLIED IN A),FOR
1893  56                       LD D,(HL);      THIS SCREEN TYPE
1894  2A FAD2                  LD HL,(PAGE)
1897  67                       LD H,A
1898  7D                       LD A,L
1899  E6 8F                    AND 08FH
189B  84                       ADD A,H
189C  32 FAD2                  LD (PAGE),A
189F  D3 00                    OUT (PGPORT),A
18A1  EB                       EX DE,HL
18A2  F1                       POP AF
18A3  16 00                    LD D,0
18A5  87                       ADD A,A
18A6  5F                       LD E,A
18A7  19                       ADD HL,DE
18A8  5E                       LD E,(HL)
18A9  23                       INC HL
18AA  56                       LD D,(HL)
18AB  EB                       EX DE,HL
18AC  D1                       POP DE
18AD  E3                       EX (SP),HL
18AE  C9                       RET
                        ;
                        ;
                        END
```

A>

```
                        .Z80
0000'                   ASEG
                        ORG     18AFH
                        ;
                        ;
FACF                    BASELI  EQU     0FACFH
1B6A                    CHOUT   EQU     1B6AH
FD6A                    CONTAD  EQU     0FD6AH
FD69                    CONTFL  EQU     0FD69H
02CB                    EFFENT  EQU     02CBH
19DD                    FIND    EQU     19DDH
FA83                    KBDBUF  EQU     0FA83H
2BC9                    NEWLINE EQU     2BC9H
02C0                    NOGOOD  EQU     02C0H
29AF                    ONELINE EQU     29AFH
FD75                    PRORPL  EQU     0FD75H
0744                    STACKS  EQU     0744H
00F4                    SWITCH  EQU     00F4H
FF5B                    VINTFG  EQU     0FF58H
                        ;
                        ;
                        PUBLIC ERROR
                        PUBLIC OUTSTR
                        ;
                        ;
0010                    SCRRST    EQU  10H
                        ;
18AF    CD 00F4         ERROR:: CALL SWITCH0   - Page 0 ROM
18B2    CB 6F                   BIT 5,A        - Test bit 5
18B4    F5                      PUSH AF  } Stack bytes
18B5    F5                      PUSH AF  } and flags
18B6    D7                      RST SCRRST
18B7    A3 1B 56 FF             DB 0A3H,1BH,'V',0FFH,6FH,29    <ESC> V <-FF>, VS 7, CLS, Page On
18BB    6F 1D
18BD    AF                      XOR A
18BE    32 FF58                 LD (VINTFG),A   } Real screen mode
18C1    32 FD75                 LD (PRORPL),A
18C4    F1                      POP AF          } Jump if bit 5 reset
18C5    28 2D                   JR Z,SYNT10       (syntax error ?)
18C7                    RUNTIME:
18C7    3A FD69                 LD A,(CONTFLG)  } Jump if CONTFLG
18CA    A7                      AND A             not zero
18CB    20 16                   JR NZ,SYNT9
18CD    2A FD6A                 LD HL,(CONTAD)
18D0    A7                        AND A
18D1    ED 4B FA83                LD BC,(KBDBUF)  } Jump if continue
18D5    E5                        PUSH HL           address in
18D6    ED 42                     SBC HL,BC         KBD BUFF.
18D8    E1                        POP HL
18D9    30 08                   JR NC,SYNT9
18DB    D7                      RST SCRRST      } VS 0, <Home>
18DC    60 1A                   DB 60H,26
18DE    2B                      DEC HL          } Decrement HL twice
18DF    2B                      DEC HL
18E0    CD 29AF                 CALL ONELINE  - Print line containing error
18E3    21 FFFE         SYNT9:  LD HL,0FFFEH  - Clear continuation line
```

```
18E6  22 FD6A                    LD (CONTAD),HL
18E9  F1                         POP AF
18EA  EE 30                      XOR 30H
18EC  CD 18FB                    CALL SYNT11
18EF  0E 00                      LD C,0
18F1  C3 02CB                    JP EFFENT
18F4  F1                 SYNT10: POP AF
18F5  CD 18FB                    CALL SYNT11
18F8  C3 02C0                    JP NOGOOD

18FB  F5                 SYNT11:   PUSH AF
18FC  D7                 SYNTAX: RST SCRRST
18FD  6F 0D 0A                   DB 6FH,13,10
1900  21 1913           SYNT1:  LD HL,ERR0
1903  F1                         POP AF
1904  CD 19DD                   CALL FIND
                         ;
                         ;
1907  7E                 OUTSTR: LD  A,(HL)
1908  23                         INC  HL
1909  CB 7F                      BIT  7,A
190B  CB BF                      RES  7,A
190D  CD 1B6A                    CALL CHOUT
1910  28 F5                      JR   Z,OUTSTR
1912  C9                         RET

                         ;***** CHECK THET BASELINEW IS IN CORRECT PAGE FOR  DATA STATS
                         ;
1913  50 61 72 61        ERR0:    DC 'Params'
1917  6D F3
1919  4D 69 73 74        ERR1:    DC 'Mistake'
191D  61 6B E5
1920  C1                 ERR2:    DC 'A'
1921  53 45 2E C1        ERR3:    DC 'SE.A'
1925  53 45 2E C2        ERR4:    DC 'SE.B'
1929  53 79 6D 62        ERR5:    DC 'Symbol?'
192D  6F 6C BF
1930  4E 6F 74 20        ERR6:    DC 'Not numeric'
1934  6E 75 6D 65
1938  72 69 E3
193B  4E 6F 74 20        ERR7:    DC 'Not a string'
193F  61 20 73 74
1943  72 69 6E E7
1947  42 6F 6F 6C        ERR8:    DC 'Boolean?'
194B  65 61 6E BF
194F  53 45 2E C3        ERR9:    DC 'SE.C'
1953  53 45 2E C4        ERR10:   DC 'SE.D'
1957  4D 69 73 6D        ERR11:   DC 'Mismatch'
195B  61 74 63 E8
195F  53 45 2E C5        ERR12:   DC 'SE.E'
1963  C7                 ERR13:   DC 'G'
1964  42 CB              ERR14:   DC 'BK'
1966  4E 6F 20 64        ERR15:   DC 'No data'
196A  61 74 E1
196D  4F 76 65 72        ERR32:   DC 'Overflow'
```

```
1971  66 6C 6F F7
1975  44 69 76 20          ERR33:   DC 'Div /0'
1979  2F B0
197B  4F 75 74 20          ERR34::  DC 'Out of range'
197F  6F 66 20 72
1983  61 6E 67 E5
1987  4E 6F 20 73          ERR35:   DC 'No space'
198B  70 61 63 E5
198F  53 75 62 73          ERR36:   DC 'Subscript'
1993  63 72 69 70
1997  F4
1998  47 6F 73 75          ERR37:   DC 'Gosub'
199C  E2
199D  55 6E 64 65          ERR38:   DC 'Undefined'
19A1  66 69 6E 65
19A5  E4
19A6  41 72 72 61          ERR39:   DC 'Array exists'
19AA  79 20 65 78
19AE  69 73 74 F3
19B2  4E 6F 20 46          ERR40:   DC 'No FOR'
19B6  4F D2
19B8  4E 6F 20 63          ERR41:   DC 'No call'
19BC  61 6C EC
19BF  4E 6F 20 6C          ERR42:   DC 'No line'
19C3  69 6E E5
                           ;
                           END

A>
```