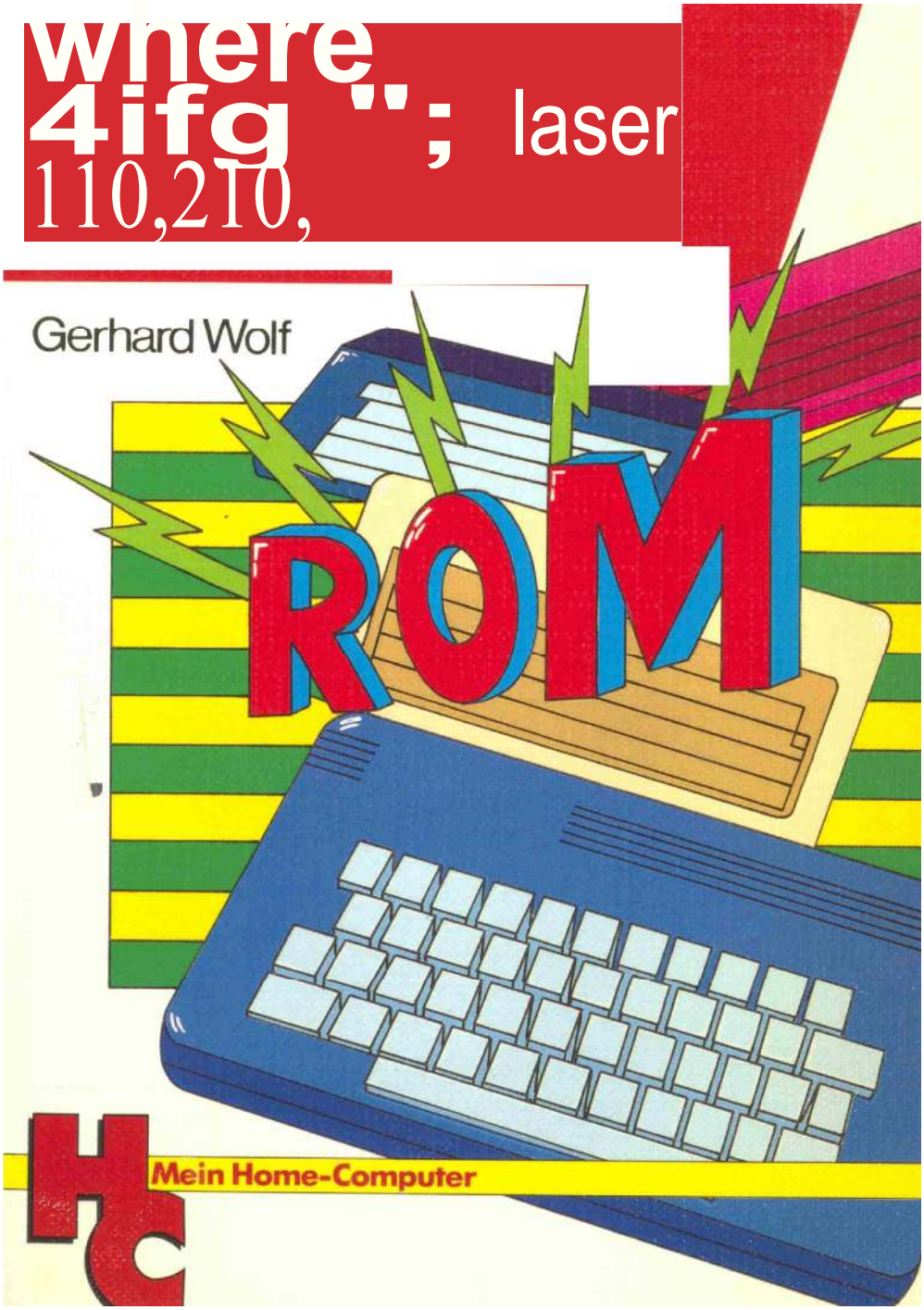
where 4ifg ":; laser 110,210,



andVZ200 ri

Gerhard Wolf
Laser 110, 210, 310 and VZ 200 ROM listings

Gerhard Wolf

Laser ROM Listings 110, 210, 310 and CZ 200

Full documented listing of BASIC Interpreter Version 2.0

VOGEL-BUCHVERLAG
WÜRZBURG

CIP short title recording of the Deutsche Bibliothek **Wolf,**

Gerhard:
Laser 110, 210, 310 and VZ 200 ROM listings: Fully. documented listing of BASIC interpreters version 2.0 / Gerhard Wolf.
Würzburg: Bird, 1985. [HC - My Home Computer) ISBN 3-8023-0852-2

ISBN 3-8023-0852-2 1.
Edition. 1985

Printed in Germany

Envelope design: Bernd Schröder, Böhl
Manufacturing: Alois Erdl KG, Trostberg

This book contains a complete documented listing of the BASfC-Interpreter, version 2.0 for the LASER - computers 11111, 210 uno 310 and above vZ:200.

The differences to version I,Z bez1enen sh only on the switching capability of the background colour between black una green 111, text mode. The required additional routines are almost completely iff, ROM area ao 3E00i1 housed.
Since no address shifts were made in the rest of the ROM, the tns collection would apply to this area for version 1.2.

First, some agreements:

The figure 1st decimal, 111if not a special marking is affixed. 'H' behind a lahi denotes it as hexadecimal and 'B' as binary representation.

In *the* register presentation, the general Z80 registers A, B, C, D, E, H, L, IX and IY have been used,
In *the* Arithmetic description 111, the register names X and Y are additionally applied. X is the working area used from address 791DH in the RAN for arithmetic operations of each type, I'lit Y sina for arithmetic with simple accuracy means the Z8@-Register B,CD,E {B=Exponent, C=t5B, DE=LSB), be1 of arithmetic with double accuracy finds door V of RAN workspace from address 7927H Usage.

MSB (most significant byte) = maximum byte of a number LSR
(least significant byte) :: low byte of a number

If individual bits of a register or address are addressed, the corresponding .bit is specified behind the register or address in Klallftler'n, e.g. A(7) = .Bit 7 in register A, MS 1(0) = Bit in the highest byte of the Arithmetic register X.

The collection contains a set of .Byte (DEFB) definitions, behind which a complete command with "Du111111Y" command is listed in the K0111111entar. These are used to override the command codes contained in the operand when the corresponding routine is run linear.

```
0000  F3          DI                  ; Turn lnterupts off
0001 AF           XOR   A             {text mode on
0002  32  00  68  LD    (6880H),A
0005 C3 74 06     JP    0674H         continue bel 674

                  ****************************************

                  Restart 8
0008 €3 80 78     JP    7800H         ;jump over RAM vector
                                      7811l0H ;to address 1C96H

000B  E1          POP   HL            ;unused
il0111C           JP    (HU)
E9
080D0 00 00 00

                  Restart 10
il01111 CJ 03     JP    7803H         Jump over RAl1 vector
78                                    7803H ;to address 1D78H

                  Read a character via Device Control Block (DCBl)
0013 CS 0014      PUSH BC             Save iBC
06 81 0816        LD    B,1           ;B Door Set DCB Check
18 2E             JR    '16H          ;to the DCB launch routine

                  Restart 18
0018 CJ 06 78     JP                  ;Jump over RAl1 vector 7806H
                  7806ti              to address 1C90H

                  Output a character via device control block IDCB>
01 C5             PUSH BC             ;BC rescue
001C 66 82        LD    B,2           Set 1B for DCB Check
001E 18 26        JR    46H           ;to the DCB call routine

                  Restart 28
0028 C3 09 78     JP                  ;Jump over RA!t vector
                  7809H               7809H ;to address 2SD9H

01123 CS                              ;not used
0024      06 04   PUSH BC
01126 18 lE       LD 3.4
                  JR    4H
```

```
                        Restart 28
002B C3 BC 78           JP      78'CH           ;Jump to RAl'I vector 78"CH

                        keyboard drop
                        Reg. A contains the ASCII code of a pressed key or 0
                        if none is pressed when jumping back.
0023 11 15 7B           LD      DE, 7815H       Load DCB Address for Keyboard
082E 18 E3              JR      13H             Continue at 13H

                        Restart 30
0830 0F 78              JP      780FH           { Jump to RA}t vector 7BNFH

                        DCB Video Output
                        • LASER 110-310 not used.
0833 11 1D 78           LD      DE, 781DH       ;Load DCB Address
00836 18 E3             JR      1BH             Continue at 1BH

                        Restart 3B              ;IH1 interrupt vector
0038 C3 118 2E          JP                      ;To interrupt service routine


                        Printer Output via Device Control Block (DB)
                        Reg, A 11ŭj contain the character to output
                        LD      DE, 7825H       Load iDCB Address
                        JR      1BH             ;continue at 1B
                        2EBBH

                                2EFDH           to the keyboard reader
003:S 11 25 78          JP                      routine
813E 18 DB

                        RET                     ;not used
0040 C3 FD 2E

                                74
0043 C9                 JP                      ;Jump to DCB call routine
044 08 80

084 C3 74 M

                        Keyboard Query
                        Wait until a key is pressed.
                        Exp.: A-Reg contains ASCII code of the Dead key
€0049 CD 2 80           CAU 2BH                 ;Evaluate keyboard
004 B7                  OR      A               <Print Screen?
004D C0                 RET     NZ              Yes, back
004E 18 F9              JR      49H             No, wait

                        %~1%1~HH~3H %i HF&f& M

                        Saving characters from cursor position
0050 2A4 20 78          LD lt., 7820H           ;Load cursor address
```

```
0e53  7E              LD    A, (H)              ;Load Characters
0854  32 JC 78        LD    (783CH>,A              ;secure to 783CH
0057  C9              RET

0058                                            ;not used

B0SF

%%%%#t lt i }#With ti t%

    Heading Time Part

       Eing: Reg. BC besti111111t Duration
0%68  0B              DEC   BC                  ;counter - 1
 001  78              LD    A+M                 i= II?
 002  Bl              OR    C
1116J  211 FB         JR    NZ.611H             No, back
0e65  C9              RET                       Yeah, done

fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

            Interrupt Vector for Non maskable interrupts

11 LASER 111-310 not used
O     31 011116       LD    SP, 600~
0069  3A EC 68        LD    A, (6BECHl
006C  3C              INC   A
IIJIIJ6J)  FE 02      CP    2
0116F  D2 80 08       JP    NC,111              ;cold start
0872  C3 CC 1116      JP    bCCH                Warning Startup

%%H111H% #t % tt # I # k #&

BASIC Initialisation Part 2

0075  11.8111.78      LD    DE,7888H            hterprope f, div, ut,        Inp
0078  21F718          LD    LT, 18F7H           ;u.a. to the RAi'! .
087B  01 27 0111      LD    BC, 27H
007E  ED IL           LDIR
111188  21 ES 79      LD    li.79E5H              Configure I0 Butter
0083  36.34           LD    (H),3H              ;before butter ':'    ,   ›'
11185  23             INC   ed                  ;Write
008   7111            LD    HL),B
0887  23              INC   ed
0188  €32             LD    (H,2                ,,'
0084  23              INC   HI
```

```
8B  22 A7-78          LD    OBA7Hl,HL     Save ilO Butfer Address
008E  11 20 1.1       LD    DE,12DH       ;Disk Command Vectors
0091  06 1C            LD    B.28          to 'DISK CONflAND' Error
0093  21.52.79         LD    HL.7952H      Antangsadr, the vectors in RAM
009   36 C3            LD    (H),0C3H      ;Jump command on adr. 12DH in
0098  23               INC   HL            to write to each vector
0099  73               LD    (H),E
009A  23               INC   HL
089B  72               LD    (Hll,D
009D  11H7             DJNZ  0096H         ;next vector
009F  6 15             LD    B.21          { RA/ Addresses for Extension
00A1  36 C9            LD    !HU,0C9H      ;existing BASIC command with
0043  23               INC   HL            i Return
00A4  23               INC   HL            then two bytes for possibly.
00A5  23               INC   HL            ;Keep jumping address
00A6  10 F9            DJNZ  00AlH         further
lrl0A8  21 ES 7A       LD    HL,7AEBH      Highlight < Start >
00AB  70               LD    (Hll, B       ; with rrl
IJ0AC  31 FB 79        LD    SP.79FBH      ;Load stack
0lr'lAF  CD BF 1B      CALL  1BBFH         Initialise Stack (via NEW>
00B2  CD C9 01         CALL  1C9H          ;Clear Screen
00B5  00 00 00 00 00                       not used
00BA  0 00 08 08
00BE  18 04            JR    00C4H         ;skip next 4 bytes
00C0  D7 B7 2112                           ;not used
00CA  21 4€ 7B         LD    H,7BCH        ;From 7B4DH Find storage end
lillC7  23             INC   ed            ;next byte
00c8  7C               LD    A,H           ;Address 1116100 reached?
0C9  BS                OR    L
80CA  28.1B            JR    1,00E 7H      Yes!
 0CC  7E               LD    A (H)         ;Load contents of byte
088CD  47              LD    B,            {works
00CE  2F               CPL                 to form a complement
00CF  77               LD    (HU,A         ;and Save
0D0  BE                CP    (Hl)          ;read value equal?
00D1  70               LD    HL)+B         {Old content restored,
00D2  28 F3            JR    1.0C7H        who equals, next byte
0004  18 11            JR    00E7H         Ern. Process Memory End
00D6  CD 5A 1E         CALL  1E5AH         ;not used
```

```
08D9  B7              OR    A
00DA  C2 97 19        JP    NZ,
00DD  EB              EX    EN, Hl
0EN   2               DEC   Hl…
00DF  PER 8F          LD    A, 8FH
00E1  4               LD    B (HL)
00E2  77              LD    (Hll, A
00E3  BE              CP    (HLI)
00E4  78              LD    (HU,B
08E5  20 CE           JR    NZ,
                            00B5H
                      Set the end of memory for BASIC
00E7  2B              DEC   Hl              {Last .Byte Address
08EB8 €11 14 7        LD    DE,7C14H        there must be at least 18 bytes
00EB  DF              RST   18H             ;be free
00EC  DA 7A 19        JP    C,197AH         ; otherwise, "OUT OF MEMORY"
00EF  11 CE FF        LD    EN-50
00r2  22 Bl 78        LD    (78B1Hl,Hl.     Remember {Memory End Address
00F5  19              ADD   HL, DE          {end of memory -50
00F6  22 A8 78        LD    (78A0Hl,Hl      ;=Beginning String-pace - 1
00F9  CD 4D 1         CALL  1BDH            ;Call the NEW routine
FC    CD 84 34        CALL  3484H           ;counter u. Initialise Pointer
0OFF  21 0F 01        LD    HL.010FH        Location Feed Text
0182  CD A7 28        CALL  28A7H           ;Print Text
0105  ED 5            II'!  1               Switch on interrupt node 1
0107  C3 BE 0b        JP    068EH           on Part 3 of the initialisation

                      ffffffffffffHHHHHHHHHHHHFFFFhffff

                      1st 11 text
010F  56.49.44.45     DEFl'I  'VIDEO TECHNOLOGY'
      4F 28 54 45
      43 48 4E 4F
      €4 4F 47 59
011F  0D              DEF.B   0DH
0120  42 41 53 49     DEFl'I  'BASIC V2.0'
      43 20 56 32
      2E 38
012A  0D 0D 0D        3x DEFB 0DH

                      tfffffffffffllllllHffffffffffffffffffffffffffffll

                      Issue "DISK CONMAND - ERROR"
012D  1E 2C           LD    E,2CH           failure code
```

```
11112F C3 A2.19          JP      19A2H          ;Jump to Nelli!

                         #t%#%%%%%%%%%%%%%%%~%%% %% #
                         t Wed Graphics Instructions POINT,
                         SET, and RESET

                         POINT Statement
111132
D7   0133                Determined if point in up. Graphic set
AF 111134                RST     10H            ;Next character d. Command
1111                     XOR A                  ;0 = Flag for Point
                         DEFB 01H               ;LD BC,803E simulated at POINT

                         SET - Statement
                         Sets point in high resolution graphics
111135 3E                LD      A,80H          ;80 = Flag for SET
80 111137 01             DEFB 1111H             ;si11.1st LD BC,1113E

                         RESET Statement
                         Deletes point in high resolution graphics
013B   3E 1111           LD      A,01H          ;1 = Flag for RESET

013A   F5                Geneinsan for POINT, SET and RESET
                         PUSH AF                Save flag
013B   CF                RST     B              ;next character='('?
013C   2B                DEFB    '('
0130   CD 1C             CALL    2B1CH          1. Evaluate expression in
       2B                                       ;X coordinate
0140   FE 8IIJ           CP      128            ;> 1277
0142   D2 4A LE          JP      NC, 1EH        ;Yes, SYNTAX-ERROR
Ill't5 F5                PUSH    AF             ;X coordinate on stack
014    CF                RST     8              Is there a comma?
0147   2C                DEFB    ','
0148   CD 1C             CALL    2B1CH          2. evaluate usdrurk in brackets
       2B                                       ;V coordinate
14R    RD 40             CP      64             ;> 63?
014D   D2 4A 1E          JP      NC,1E4AH       ;Yes, SYNTAX-ERROR
                         Determine From X,Y Image Address and
111150 5F                Bitmask
111151 AF                LD      E,A             Y41ert in DE
111152 57                XOR     A
111153 EB                LD      DA
0154   29                EX      DE,HL          Y x Line Length (x32)
111155 29                                       3x2
0156   29                                       $x
111157 29                                       $1x8
                                                $x16
                         ADD     HL, HL
                         ADD     HL, HL
                         ADD     HL, HL
                         ADD     H,H.
```

```
0158    29          ADD   HL, Hl.        5x32
0159    E           EX    DE, Hl.       = rel, Ze1lenant.address
IUSA    FI          POP   AF            Load X Value
015%    F5          PUSH  AF            ;and back to stack
015C    CB JF       SRL   A             ;X–J<ordiante / 4
015E    CB 3F       SRL   A
0160    83          ADD   A+E           ; + rel. Line lead, address
0161    SF          LD    E,A
0162    7A          LD    A, D          ;+ image start address
8163    F6 70       OR    70H           i (benm LASER 70080H)
11165   57          LD    D,A           i DE = Bi 1 daddress
11166   Fl          POP   AF            ;X coordinate
8167    E6 03       AND   3             Mask last 2 bits (0,152,3)
0169    87          ADD                 X 2 (0.244)
816A    47          LI>   BA            ;in B as displacement counter
0168    Fl          POP   AF            ;Load Function Flag
11116C  B7          OR    A             ;= @?
01      CA E7 38    JP    Z.38E7H       ;Yes - then run POINT
111170  FS          PUSH  AF            ;Secure Function Flag
                    For SET and RESET bits in Reg. Mask A and C
0171    O3F         LD    C,JFH         ;Load base value in C
0173    YES 46 78   LD    A,(7846H)     ;Colour code as base value in A
IH76    CB 27       SLA   A             ;Move to Top 2 Bits
1i'1178 CJ 27       SLA   A
017A    CJ OF       RRC   A             with B as shift counter
817C    CB 09       RRC   C             Values in erf, Position
017E    10 FA       DJNZ  017AH         A for SET (OR), C for RESET
0180    CJ 03 39    JP    3903H         Continue at 3983H
```

Mi%tkt# #%# i%EttittHi

Error Handling at YERIFY

```
0183    21.39.78    LD    Hl.,7839H     ;Delete Verify Bit in Flag2
8186    CB 9E       RES   3, H.)
0188    21 84 83    LD    Hl...0384H    Addressing Error Message
018B    CD A7 28    CALL  28A7H         ;and spend
018     CJ CF 36    JP    36CFH         Continue at 36F
```

H¾lffff ★ fllflfffflfllllfllllllHH

IMKEY$ - Function

```
19D     D7          RST   11H           ;Next character
819E    IT          PUSH  K.            ;Save Pointer
```

```
019F    34.99.78    LD    A, 7899H      ;Load characters from
01A2    B7          OR    A             Characters present?
0143    20 0        JR    NZ,01ABH      ;Yes!
0145    CD 58 03    CALL  0358H         Keyboard Query
01A8    B7          OR    A             Pressed a new key?
01A9    28.11       JR    Z,011!CH      No, empty string in X-Reg.
014     F5          PUSH  AF            Character on stack
01AC    AF          XOR   A             Delete iINKEYf Memory
01AD    32.99.78    LD    (7899HJ,A
01B8    3C          INC   A             Stringlings = 1
11:1    CD 57 28    CALL  2857H         i1 ]!yte i• Reserve Strinqspace
11:4    Fl          POP   AF            Reload Character
1.BS    2A D4 78    LD    H, (78D4H)    Load URL in String Spare
1.B8    77          LD    (Hll, A       ;characters in string space.
01.B9   C3 84 28    JP    2884H         Continue at 284
01BC    212819      LD    H., 1928H     pointer aut Empty String
01BF    22 21 79    LD    ( 7921H) lt.  according to X
01C2    3E 83       LD    A+3           Type = Set String
014     32 AF 78    LD    (7BAM),A
01C7    El          POP   H             iPointer will load
lil1C8  C9          RET
```

%H~4HHHHHHHHHH EM M t Mt

CLS Statement

Delete the screen

```
019     3E 1C       LD    A, lCH        {cursor to top of image
01CB3   CD 3A 63    CALL  033AH
01CE    3E 1F       LD    A, 1FH        ;Delete Image to Finish
01Dlr'I C3 34 83    JP    033Atl
```

HffffHHHffHIIHIIIIIIIIIIIIHI

RA}ON - Statement

Random generator initialisation

```
01D3    ED SF       LD    M, R          Load {Retresh Register
01D5    32 AB 78    LD    (78ABHJ,A     ;in random number-6round value
lillDB  C9          RET
```

lllllllllllfHHHU.HHHHHHfHH

Keyboard - Tables

Key Codes without SHIFT

01D9	54 47 42 35 4E 36.59.48	DEFB 'T', 'G', 'B', '5', 'N', 'i,', 'Y', 'u'	Bitreir.e 0
01E1	57 53 58 32 2E 39 4F 4C	DEFB 'w's,',27.,'9, - O,'L'	bit series 1
01E9	00 080 80 00 00 2D 0D 3A	DEFB 00H, 00H, 00H4, 00H, 00H, '-' CR,':'	Series 2
01Fl	45 44 43 33 2C 38 49 4B	DEF.B 'E', 'D', 'C', '3',', ', '8', 'l',	bit series 3
01F9	51 41 5A 31 20 30 50 3B	DEFB '0,'4,'2,'1,' ','0,'P,'3'	bit series 4
0201	52 46 56 3 4D 37554A	DEFB 'R','F','V','4','N','7','U','J'	bit series 5

Keyboard codes with SHIFT (including semigraphics)

1209	SC 89 00 25 5 26 83 8	DEFB BCH,89H,10H,25H,SEH,2oH,83H,8bH	bit series
0211	8D 82 00 22 3E 29 SB 3F	DEFB 8DH, 82H, 00H, 22, 3EH, 291, 5H,	bit series 1
0219	00 III 00 lil1 O 3D 0D 2A	DEFB 0OH, O\, 00H, 00#, OO, 3DH, 8DH,	bit series 2
1221	BB841i1123 JC 28 85 2F	DEFB 8BH, 84#, OO\, 23H 3CH, 28H, 85,	bit series 3
0229	BE 81 80 21 20 40 5D 2B	DEFB 8EH, 81H, 80H, 21H, 20, 40\H4, 5DH,	bit series 4
0231	87 B88 80 24 SC 27 BA 8F	DEFB 87, 88H, OO\, 24, 5CH, 27H, BAH, FH	bit series 5

Keyboard Codes with CTRL (including CND tokens)

0239	CA 8D B5 B 97 BE 95 84	DEFB CA, 8DM, B5H, BM, 97M, 8EH, 95H, 84#	bit series
8241	BD CC Bl B9 1B BB SC 15	DEFB BDH,CCH,B1H,B9H,1.BH,S.BH,SCH,15H	bit series 1
0249	00 00 80 00 00 61 08 8	llEFB 00+, 00i, 110H, 110H, ll0Hi 11 H, 10H, li1BH	bit series 2
0251	87 BA BJ 9C 09 BB 89 BC	DEFB S7H,SAH,BJH,9CH,19H,BRA,89H,BCH	bit series 3
0259	81 9D E5 BA L!A88B27F	DEFB 81M, 9DH, E5H, BA, H, 88H, 82M,	bit series 4

```
0261  92 91 /If 98 08 80 8F 93   DEFB  92H, 91H, AFM, 98H, 08\, 80H, 8FH, 93H   bit series

                                  Keyboard codes for functions (with CTRL-ENTER)

829   FA 94 9E DF BF E0 F9 83     DEFB  FAH,94H,9EH,DFH,BFH,E0H,F9H,83H         bit series
0271  FS F4 A0 EI 00 D9 D3 00     DEFB  F5H, F4H, AMH, E 1H, 80H, 9H, 3H, 00H   bit series
0279  00 00 00 8 00 8! 00 00     DEFB  MH, 00H, 00H, 00\H, 00H4, 01H, 80H, 00H bit series
0281  F3 90 96 E3 00 DD D2 C     DEFB  F3H,91H,96H,E3H,80H,DDH,D2H,C6H         bit series
0289  F7 F6 DB E2 08 D8 CB 08    DEFB  F7H, F6H, DRH, E2H, 00H, D8H, CBH, OH   bit series
6291  F8 DE Cl E4 O D7 C9 82     DEFB  FB8M, DEH, CIH, E4H, 0OH, D7H, C9H, 82H bit series

                                  Tokens with the character ' ('

                                      are added to USD,

0299  E2 El E3 E4 DF EI! D7      DEFB  E2H,E1H,EJH,E4H,DFH,EBH,D7H
6240  D9 D8 F7 F5 FJ F8           DEFB  DDH, D9H, D8H, F7H, F5H, F3H, F8H
02A7  F7 F9 9D F6 F4 DE ES FA     llEFB F7H,F9H,9DH,F6H,F4H,DEH,E5H,FAH

                                  lllllllllHKtfllllllllllllffa.HH

                                  Print the graphic character output table on a printer.

                                  The table Zllll!i Byte contains each character.

82AF  89.88                       DEFB  88H, 80H          ;character 811H
e2B1  808 B8                      DEFB  881.8H            ;character 81H
12BJ  B8 81                       DEFB  B8H 81H           ;character 82H
I2B5  B8 BB                       DEFB  B8H B8H           ;character 83H
02B7  80 87                       DEFB  Blillit87H        ;character 84H
02B9  88 BF                       DEFB  B88H, BFH         ;character 85H
I2BB  B8 87                       DEFB  B8H, 87H          ;character 86H
02BD  B8 BF                       DEFB  BSH,BFH           ;character B7H
02BF  87 81                       DEFB  B7H,8"H           ;character B8H
'2Cl  B7 BB                       DEFB  87H,BFH           ;B9H
02c3  BF 81                       DEFB  BFH,8'H           ; character BliH
```

```
02  BF B8  DEFB        BFH:BSH              ;Bl!H
12  87.8   DEFB        87H.B7H              ;BOI character
02  87     DEFB        87H,BFH              Character BIH
08  BF     DEFR        BFH,87H              ;BEH character
02  BF     DEFB        BFH, BFH             ;sign SFH
CD  BF
```

***HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHFFL**

Fr1?11.uenz table for SOOND-Koaando

one 2-byte entry per note,

```
02  72     DEFII       626.591.558.526.97.449.43%.414                    A
CF  02                                                                    2
    2E
    Fl
    B7-
12  86.0   DEFW        390, 38, 397, 328, 309, 291, 275, 259             F
    5
    35
    13
02  F4     DEFW        244.231.217.215.193.182.171.161                   #
    D9
    Cl 10
    AB 8O
02  98     DEFW        152.143.135.127.128.112.116                       -
    87
    78
    6A
    8
```

%#Hit~Hi##%41%%1#±

Reset characters to cursor position!'II

(part of the screen output routine)

```
031 47     LD          B,A                  ;character to be output in B
83  3A     LD          A, (78JCH)           ;Load character to cursor
031 2A     LD          H, {7820)            ;Load cursor address
031 n      LD          (lt.>,A              !Print Ztichen
831 78     LD          A,                   Exp. Character again in A
031 C9     RET
6
```

#Mt#Mt# MM#MHHHMiMMMMMMMMMii

cursor address one line back

Ring: ed = .Current cursor address

```
'31 01     LD          BC.32                Line Length
7   28
```

```
031A B7          OR    A              Delete
031B ED 42       SBC   HL,BC          Carry ;Cursoradr. - 1
031D 22 20 78    LD    (7820H!,HL     line ;in cursor
0320 C9          RET                  pointer
```

****

S0\/ND command multipliers door 1 byte per same input code (1-9)

```
€8321 01 02 03 04 06 08 0€18 18    DEFB 12.35.6.812+16.24
```

4kt#ii% tlitt% i lt k } Mi t lt

Print characters on screen, printer, or cassette,
Eing: A = Character to output
789CH4 = Output Flag {8=Screen, 1=Printer **B** = Cassette)

```
032A  C5          PVSH BC          Save iBC
032B  4f          LD   C,A         ;Mark in C
032C  CD Cl 79    CALL 79C1H       ; RAI'I Extension Exp. 1 RET 1
032F  3A 9C 78    LD   A.789CH     {Load Output Flag
8332  B7          0R   A           ;and test
0333  79          LD   AC          ;Character back in
0334  CI          POP  BC          A ;Be Reload
0335  FA 54 3B    JP               ;Cassette? yes - continue with
8338  20 62       1'1.3B54H        3B54H ;Printer? yes - for print
                  JR   NZ.039CH    output
```

****

A character on dell Display Eing.: A = character to be issued

```
0833  D5          PUSH DE              ;Save Register
033B  F5          PUSH AF
0JC   C5          PUSH BC
033D  E5          PUSH lt.
033E  CD SB 31    CALL JIBBH           ;Output routine call
031   EI          POP  H               Reload the register
8342  CI          POP  BC
0343  80          NOP
034   08          NOP
```

```
0345  Fl              POP     AF
034   Dl              POP     EN
0347  C9              RET
```

%i% k It% #tt% t } #Mi tt 4} E } t }

cursor position in line er • medium

Not used on LASER 110-310

```
0348  3A 3D 78        LO      A, 1783
034B  E 08            AND     8
0340  34.20.78        LD      A,
0350  28.03           JR      Z,0355H
0352  0F              RCA
0353  E IF            AND     lFH
0355  E IF            AND     1FH
0357  C9              RET
```

 # #I # #'EH' Egg F # Egg Eli #Hi }Mi}M ME

keyboard drop

Exp.: A = ASCII code or 0

```
0358  CD C4 79        CALL    79C4H       ; RAl1-Er111eiter. output ( RET>
035B  05              PUSH    EN              Secure ;DE
035C  CD 2B 00        CALL    082RH           ;Evaluate keyboard
035F  D1              POP     EN              ;DE • Restore
0360  C9              RET
```

Mt# # i i # # i 'Hi # #litt#

Table of basic time values for each

Note of SOUND-COManr:los.

```
831   0A 0B 0C 0C     DEFB    108.11.12+12.13.14415.15        iA2 - EJ
      0D 0E 0F 0F
0369  10 11 12 13     DEFB    16.1718.19.21.22.23.25          iF3 - C4
      15 16 17 19
0371  1A 1C 10 1F     DEFB    26.28.25.31.33.35.37.39         Cll4 - 6#4
      21 23 25 27
0379  29 2C 2E 31     DEFB    41.49.52.5358                   iAA - 0#5
      34 35 3A
```

MM} ## # #### # With t # With # t IttMi %

OK and Error11Message of the VERIFY COMando

```
038 4F    DEFN      'OK'
038 D     DEFB      0DH,00H
038 45    DEFN      'ERROR'
038 O
a   O     DEFB      0DH,00H
          ********************************************************************************
          ***********************
          Output flag on screen.

          CR on printer if m1cht at start of line
038 AF    X0R       A                    {Output Flag on Screen
038 32    LD        (789CH1,A
038 34    LD        A, (789:BH)          {Printer Position in Row
039 B7    OR        A                    {= 8?
039 C8    RET       0r                   yes - ready
2
          Print Carriage Return on Printer

039 3E    LD        A,0DH                Load iCR
039 OS    PUSH      EN                   Secure ;DE
039 CD    CALL      039CH                ; Print CR
039 D1    POP       EN                   ;DE 111 Restore
039 C9    RET
B
          %it±~4$ 3 4 # # f t t #f kt i # #t kt#

          Print characters to printer.

          Ring: A = Character to output
                    789B = PHD
039 FS    PUSH      AF                   ;Save Register
039 D5    PUSH      EN
039 CS    PUSH      BC
039 4F    LD        CA                   ;Character in C
03A 1E    LD        E,0                  E = 0
03A FE    CP        0CH                  ;Is it a For Feed?
03A 28    JR        1,036                Yes!
03A FE    CP        BH                   ;Is it a line feed?
03A 20.   JR        N1.il3ADH            No!
03A 3E    LD        A, 0DH               Yes, replaced by carriage-return.
03A 4F    LD        C,A                  ;and in C
03A FE    CP        0DH                  ;Is it a carriage return?
03A 28    JR        Z,03BbH              Yes!
03B 3A    LD        A, (789BH)           ;Load PHD Position
03B 3C    INC       A                    ;+1
03B SF    LD        E,A                  ;in E
5
```

```
03B6 7
03B7 32 9B 78
03BA 79
03BB CO 3B i10
3RE CI
03BF Dl
03C0 F1
03C1 C9
```

```
LD      A,E
LD      (789BH1,        Save new Postings (CR=b")
A
LD A                    Characters in A
CALL a03BH              Print Character Register
POP     BC              contents restored,
POP     EN
POP     AF
RET                     ;finished
```

#M~%%%~i%% } %Mi%Hi %%%

```
Using Device Control Block to Call
Driver Routines
Ring: DE= DCB-Aclresse
        B = DCB type, A = excl. Character (only output} BC
        must be on the stack
```

```
fll3C2 ES 03C3
DD E5 03C5
D5 03C6 DD
E1 03C8 D5
03C9 21 D 83
03CC E5
03CD 4F
03CE 1A
03CF Ae
0300 B8
0301 C2 33 78
03D4 FE 02 03D6
DD 6E 01 03D9
DD 66 02 83DC
E9
```

```
PIJSH   Hl...           ;Save Register
PUSH    IX
PUSH    EN              DCB-Aclresse in IX
POP     IX              and on Stack ;Return Address
PUSH    EN              on Stack
LD      HL,0JODH
PIJSH   Hl...           Character by C
LD      C,A             ;Load DCB ID (1.Byte} with
LD      A,DE)           preset. Type undate real
AND     B               guy?
CP      B               no, via RAN 7833H back with
JP      NZ, 7833H       input carry
CP      2               ;Load from DCB driver address
ID      L, (IX+1)
LD      H,IX+2)         Mount the driver
JP      1.
```

```
Return VOii driver
```

```
83DD DI
0JDE    DD E1
03E0    E1
IIJEI   Cl
03E2 C
```

```
POP     EN              Restore Register
POP     IX
POP     Hl...
POP     BC
RET                     ;finish
                        ed
```

```
ffffffiHtfffllllllttffiHtffiHtfftHtfftHtfftHtffiHtffiHtfftH
tffiHtffiHtffiHtffl
```

```
Read a line from the keyboard.
Row is used until the RETURN or BREAK
```

```
                                Key scanned, displayed on screen and
                                then transfer to the I/@ buffer.

                                Preparing the Pointer and Flags

1113E3   21.39.78      LD      HL.7839H        {initialisation flag for
03E6     C8 EE         SET     5, HL.)         ;Set buffered output.
1113E8   2A 2111 78    LD      HL, (7820H)     ;Load cursor address
03E11    CD 53 1110    CALL    0053H           Save characters to cursor position
1113EE   7C            LD      A, H            ;cursor at start but last line?
03EF     FE 71         CP      71H
fll3Fl   20.10         JR      NZ,11403H       no
03F3     70            LD      +L
03F4     FE III        CP      0EH
03F6     20 8B         JR      NZ, 0403H       no
03F8     YES D7 7A     LD      A,(7AD7Hl       ;Check the status of the 1,line
03FB     B7            OR      A               ;= follow-up line?
1113FC   20.05         JR      NZ,11403H       No!
1113FE   3E 0D         LD      A, DH           Roll up image one line
0400     CD 8B 3111    CALL    31118BH
fll4fll3 41            LD      B,C             {Length Header text in B
11141114 CS            POStl   BC              ;on stack (B=Cl
041115   21.39.78      LD      tll, 7839H      ;Address flag 2
11141118 CB            RES     111, Hll        ;Reset CR Flag
04111A   CM 9          RES     2, (tll)        ;Reset BREAK Flag
11140C   CB 46         BIT     0, HL)          wait until CR flag set
040E     28 FC         JR      Z, 04111CH

                                Determine the initial address of the input line

041fll   3A A 7        LD      A,(78A6Hl       ;Load column to input line
0413     4F            LD      C,A             ;in BC
111414   F             XOR     A
0415     32M 78        LD      478AH),A        ;Column counter= 1  !Top of Line)
0418     47            LD      B,A
0419     24 20 78      LD      HL, (7820Hl     ;Load cursor address
€041     ED 42         SBC     HL,BC           ;- Column= Line
041E     22 20 78      LD      (782fllH L, HL  ;back to cursor pointer pointer

                                Load Butter and Row Address

0421     11 ES 79      LD      DE,79E8H        Initial address of the 1/0 Butfer
0424     Cl            POP     BC              ;Header text character counter
0425     21.39.78      LD      HL.7839H        Address flag 2
0428     CB 66         BIT     4, (H)          ;Is this an INPUT conand?
042A     2A 20.78      LD      HL, (7B2111Hl)  ;Load initial address of row
042D     28.42         JR      Z,0471H         {no INPUT-Cd, continue at 471H
```

```
                                  Set the INPUT text pointer to the default text
042F   CS              PUSH   .BC            {Register Secure
4/30   IT              PUSH   HL
0431   CD A8 33        CALL   33A8H          Determine the status of the row
0434   El              POP    HL            Reload iHL + BC
0435   Cl              POP    BC
0436   B7              OR     A             Follow? (status=00)
0437   20.08           JR     NZ,0441H      No!
0439   7D              LD     A,L           ;Line address in H- 1 line
043A   20              SUB    32
043C   6F              LD     n
043D   7C              LD     A,H
043E   EN 00           SBC    A,0
440    67              LD     H,A
0441   48              LD     ,M            Number of Shear
0442   1A              LD     A, {DE)       ;Pointer behind header text
0443   BE              CP     (Hi...)       ;compare if not changed
@44    20.87           JR     NZ.044DH      Not Right, Stop
446    23              INC    HL            Image Pointer +1
0447   13              INC    EN            Butfer Pointer +1
0448   10F             DJNZ   0442H         Ready?
0444   CS              PUSH   BC            if equal, Work length
044B   18 04           JR     0451H         continue at 0451
044D   61 00 00        LD     BC,           unequal, length = @
0450   CS              PUSH   BC            to the stack
0451   IT              PUSH   HI...         H. rescue
0452   CD AB 33        CALL   33A8H         ;Read the status of the line
0455   El              POP    HI...         Hi... + Reload BC
0456   Cl              PP     BC
0457   or              PUSH   BC            Remember Length
0458   FE 80           CP     80H           ;Single Line?
045A   28 84           JR     Z,0466H       Yes!
045C   3E 40           LD     + 64          Max. Character count= 64 - Header
045E   91              SUB    C
045F   47              LD     B,A
0460   01              POP    EN            Number of bias in the stack
0461   1E e            LD     E,0           -
0463   D5              PUSH   EN
O      18.05           JR     046.BH        i2 Retain lines
06     6 20            LD     B.32          il Retain row
0468   24 20 78        LD     HEI... (7820H)  ;Text Attachment load
04B    11 EB 79        LD     DE,79E8H      I/0 Buffer Address
OE     C3 A8 JE        JP     3EA8H         ;Check background colour
```

```
                                        {be green background.next b,@4R8
                                        srcnu, Behind9r. b.3E6

                    Text start address and maximum length of 1111ttles,
                    if n1cht INPUT command
0471   II1 00 00    LD      BC,0            {Pretext Length = Set
0474   c5           PUSH    BC              ;on stack
0475   IT           PUSH    HL              ;Save HL
0476   CD AS 33     CALL    33A8H           Determine the status of the row
0479   E1           P0P     HL              Load HL 1111eder
047A   FE 80        CP      8M              ;Single?
047(   28 IU        JR      Z.048CH         3a!
047E   FE 81        CP      81H             i2 lines?
0480   28 06        JR      Z,0488H         Yes!
0482   01 20 O      LD      BC.32           {be1 follow-up line a lenght back
0485   117          0R      A
0486   ED 42        SBC     HL,BC
0488   6 40         LD      B,6             ;Leave 2 lines
048A   18 02        JR      048EH
li!48C 6 20         LD      3.32            ;1 Record Row
048E   3A 18.78     LD      A, !7818H1      ;Check background colour
0491   B7           0R      A               ili! = green, 1 = black
0492   CA 40 3E     JP      Z,3E40H         at green u,eiter at 3E40H

                    Transfer data from image to I/0 buffer

0495   7E           LD      #, (H)          ;Character VOii Load Image
049    FE 4         CP      64              ;Graphic or Inverse?
0498   DA AE 04     JP      C,04AW          {no, take over
049B   C1           POP     BC              if it is INPUT, then
                                            ;Graphics and Inverse only in
                                            ;Strings allowed
049C   11A          LD      DE, 0444H       ;Backjump address in stack
049F   D5           PUSH    EN
04AI   CS           PUSH    BC
lll4A1 C3 02 05     JP      li!51il2H       ;Check text ID (BREAK?J
li!4A4 Profit       RET     C               ;BREAK, back to BASIC
li!4A5 21 1A 3E     LD      li,3E1AH        ;Text "SYNTAX ERROR"
04A8   CD A7 28     CALL    28A7H           ;issue
AB     C3 E3 83     JP      03E3H           ;back to line input
04AE   RD 22        CP      22H             ;String flag?
04111  20 31        JR      NZ.04E3         No, continue
04B2   12           LD       (EN), A        ;Character in 1/0 Bufter
```

```
04B3   23          INC   Hl          Image Address +1
04B    13          INC   EN          butter address +1
045    05          DEC   B           ;Character counter -1
04B    28 3        JR    Z,04EEH     Bill 0, Stop Pick
04B8   7E          LD    A, (HL)     ;Load character from image
04B9   FE 40       CP    64          ;normal. Text characters?
04BR   DA C9 04    JP    C,04C9H     Yes!
04BE   RD 89       CP    128         ;Inverse text character?
04C0   DA C5 o     JP    C, 04C5H    yes'
04C3   E 8F        AND   BFH         ;Delete graphic character, bits
04CS   F6 80       OR    B0H         ;Set Bit 7
04C7   18 13       JR    04DCH
04C9   RD 22       CP    22H         ;String delimiter '''?
04CB   20 09       JR    NZ ,        {nemn!
04CD   IT          PUSH  HL          Hi. rescue
04CE   21.39.78    LD    Hl,7839H    ;Address flag 2
04D1   C3 6        BIT   4, HL.)     INPUT-KOlllando?
04D3   El          POP   Hl.         H. Load again
M4D4   28 0D       JR    Z.04E3H     ;no - from now on graphic u.
                                     Inverse not allowed.
04D    CB 6F       BIT   5.A         ;characters in real ASCII code
04D8   20 02       JR    NZ.04DCH    convert, e.g., 'A' from @1 to 41
04DA   F 4         OR    40H         ;concerns codes 01 - 1FH
04DC   12          LD    (EN),A      ;Character in l/0 Butfer
04DD   23          INC   Hl.         Image Address + 1
04EN   13          INC   EN          ;butter address+ 1
04DF   11 D7       DJNZ  04B8H       ;counter - 1
04E1   18B         JR    04EEH       {=, then finished
04E3   CB 6F       BIT   5.A         ;characters in real ASCII code
04E5   20.82       JR    NZ,         jU111Trade, e.g. 'C' of 13 in 43
04E7   F 40        OR    40H         ;concerns codes 011 lFH
04E9   12          LD    (DEl,A      ;Character in I/0 Butfer
04EA   23          INC   llkh        ;Image Address + 1
1.14EB 13          INC   EN          ;butter address + 1
04EC   10 A7       DJNZ  0495H       ;counter - 1

                   Transfer finished, complete butter content

84EE   1           DEC   EN          ;Blanks a1 Butterend el11in.
04EF   7A          LD    A,D         to the beginning of the buffer?
OF     FE 79       CP    79H
04F2   20%         JR    NZ.04FAH    ;no
```

```
04F4   7B          LD    A,E
04F5   FE E8       CP    0EBH
04F7   DA FF 04    JP    C,04FFH        Yeah, done
04FA   lA          LD    A, (DE)        ;Load Characters
04FB   RD 20       CP    20H            = Blank?
04FD   28 EF       JR    1.04EEH        ija, go back
04FF   13          INC   EN             Buttering end 11it X'00
05080  Af          XOR   A              iIdentify
0501   12          LD    KDE),A

                   Dependent on w0111 Row status 1 or 2 Blank lines

111502 CD A8 33    CALL  33A8H          ;Get Line Status
11151115 2A 20.78  LD    1-L,(7820H)    ;Load cursor pointer
111508 FE 81       CP    81H            i2 lines?
0504   CD 53 00    CALL  0053H          ;Character from cursor position.
11150D 28,084      JR    NZ,0513H       ;one-line
O50F   AF          XOR   A              ;1 Print blank
111510 CD 8 30     CALL  388BH
111513 WF          XOR   A              ! Print blank line
111514 CD 8B 30    CALL  308BH
111517  3A 38.78   LD    A, (783810     ;Load flag
11151A E FD        AND   OFDH           ;INVERSE-f"reset
11151C 32 38 78    LD     (7838H)/A     Flag 1 • back
11151F 21.39.78    LD    HL.7839H       Address flag 2
 0522  C3 5        BIT   2 HL)          ;BREAK flag set?
 11524 28 05       JR    Z,052BH        No!
111526 3E 01       LD    A, 1           BREAK, A=1
 111528 37         SCF                  ;+ See Carry
 111529 18,111     JR    152CH
 052B  A           XOR   4              ;no BREAK, A=II
11152C 21.39.78    LD    1-L,7839H      ;Address flag 2
11152F CB A        RES   4,(H)          ;Reset INPUT-Cd Flas9
111531 21 ES 79    LD    H,             ;Addressing I/0 buffer
 0534  Cl          POP   BC             ; at the beginning of the input
 111535 F5         PUSH  AF             ;Secure BREAK connoisseur
 053   89          ADD   HL,BC
 0537  C3 29 3E     JP    3E29H          continue at JE29H
```

WH#MN k MM k} M # i REM%H}Hi}MM #It

Part of the INPVT Caching routine

```
                                Imoort a line into the I/0 buffer
 053A   3A AF 7A        LD      A,17AAFHl       wait until text output is
 053    B7              OR      A               i7AAFH contains number of
 053E   28 FA           JR      NZ,053AH        in the print buffer; at 8 = empty

054lll  @6 40           LD                      ;Delete I/0 buffer (length 64)

 0542   21 E8 79        LD      HL.79E8H        ;Butter start address
111545  3E 2111         LD      A,' '           spaces in A
 0547   77              LD      (H),A           Transfer to Buffer
 0548   23              INC     HL              iBufferadl esse + 1
111549  10 FC           DJNZ    0547H           Counter - 1, if - ready!
 054B   AF              XOR     A               @ in A
 054C   77              LD      (HL),A          End of buffer ait X'00' mark.

 054D   CD A8 33         CALL    33A8H           ;Get Line Status

 0550   B7              OR      A               ;Sequential?
 0551   3A From 78      LD      (A, 17BA6H)     ;Load column counter
 0554   20.82           JR      NZ,0558H        jl:a followup!
111556  C6-2111         ADD     A, 32           Add a row in the next row.
 0558   4F              LD      CA              Pass column counters to BC
 0559   AF              XOR     A               Add to this= III
 055A   47              LD      B,A
 055B8  2A4 20 78        LD      HL, (7820H)     Load iCUrsor Pointer
 055E   ED 42            SBC     HL,BC           ;- Column = Top of Line
 056111 11 ES 79         LD      DE,79E8H        ; I/0 Bufter-Adl Load Fair
 0563   c5              PUSH    BC              ;Remember column counter
 054    ED B0            LDIR                    ;previous. Text from row to butter
 @5     Cl              POP     BC              ;Load column counter
 0567   21.39.78         LD      HL.7839H        ;Address flag 2
 0564   CB E            SET     4,(H)           ;Set INPUT Cad Flag
11156C  CD E3 03         CALL    lll3E3H         ;Read line
 05F    C9              RET

                         ##ti#%}} t# # il MM H # i i Hi

                         RUN-#pomando for autom. CRUH Start

 0570   52 55 4E        DEFM    'RIJN'
 0573   00              DEFII   0

                         HFFF

                         Printer - Driver

058D    79              LD      A, C            ;Load character to output
```

```
058E   B7              OR     A              := blank?
11:58F 28 33           JR     Z,05C4H        $J Determine Printer Status Only
0591   FE B            CP     0BH            {page feed?
0593   28.04           JR     059FH          Yes - Run
0595   FE 11.C         CP     @CH            {conditional page feed?
0597   20.14           JR     NZ.05ADH       {nenn'
0599   AF              XOR    A              nrd executed only if number
059A   DD B 03         OR     iIX+3)         {Lines/Page is not equal
059D   28 0E           JR     Z.05ADH        ;Otherwise output 0C aut printer
059F   DD 7E 03        LD     A,             ;lei len/Sei te
05A2   DD 9 04         SUB    (1X+4)         ;- Number of printed lines
05A5   47              LD     B,             1n B as feed counter
05A6   CD E2 3A        CALL   3AE2H          Carriage-Return + Line Feed
05A9   1111 F\         DJNZ   05A6H          ;to new page
05AB   18 12           JR     05BFH

05AD   CD B 3A         CALL   3AB6H          ;Print Characters
05B0   79              LD     A,C            ;Reload character
05B1   FE 0D           CP     0DH            {uar that an R?
05BJ   c8              RET    NZ             No, done
05B4   DD 34 04        INC    (1X+)          ;DCB + 1 line counter
05R7   DD 7E 04        LD     A,IIX+41       {at the beginning of a new page?
05BA   DD E 03         CP     (IX+3)         Line counter - lines/sexts)
05BD   79              LD     AC             Character again in A
05BE   C0              RET    NZ             ;no new page - finished
05BF   DD 36 04 00     LD     (IX+4),0       Line counter =
05CJ   C9              RET

05C4   DB 00           IN     A, (@)         Determine {Printer Status
05C6   Eb IH           AND    1              ;only BUSY checked
05C8   C9              RET
```

#$Hi #I # It# #k ####k##M#k#M4Hi

Delete 4-Byte Print Buffer

```
05C9   CS              PUSH   BC             BC + li rescue
05CA   IT              PUSH   HI
05CB   8               LD     B,4            ;counter= 4
05CD   21 D2 7A        LD     HL,7AD2H       Load buffer address
05D0   n               LD      (HL),A        ;A in buffer
05D1   23              INC    li.            ;Buffer Address+ 1
05D2   1 FC            DJNZ   05D0~          Counter -1 = 8? Yeah, done!
05D4   EI              POP    HI             ;Restore the register
05D5   C1              POP    BC
```

```
05D6   C9                RET
```

#k # ## t f t # k Mt #lt$

Part *of the* keyboard tray

Handles pressing a second key. before the first one was released (CRollover)
In flag 1 (7838H) the bits J and 4 are used, to the status of the two keyboard pushers Bl (7836H); *B2* 17837H).

bit4	bit3	Status
0	**0**	Bl and B2 are not pressed
0	1	Bl pressed, B2 not pressed
	0	Bl not pressed, B2 pressed
	1	Bl and B2 pressed

```
05D7   21.38.78          LD    11.7838H          ;Address flag 1
05D4   CB Sb             BIT   2.HL.)            ;Function flag set?
05DC   28 15             JR    Z.85F3H           ;no - 11Ladder at 85FJH
e5DE   57                LD    D,A                 ;Secure Key Code
05.DF  3A 3A 78          LD    A, (7834H)        ;Load time value
5E2    B7                OR    A                 ;=0 ?
05E3   288F              JR    Z,05F4H             Yes - continue on
05E5   3C                INC   A                 ; time value +1
85E    32 3A 78          LD    (7834H),A         to save back
li15E9 FE 2A             CP    42                Time expired?   (approximately
05EB   28 82             JR    Z.05EFH           <yes!
05ED   AF                XOR   A                 ;Delete
li15EE C9                RET                     ;and back
li15EF CB 9              RES   2, (fjl)          ;Delete Function Flag
05F1   AF                XOR   A                 ;Delete
05F2   C9                RET                     back

05FJ   57                LD    DA                ;Save character in D
05F4   21.38.78          LD    HL, 7838H         Address Flag 1
05F7   7E                LD    A,                Load in A
05F8   E 18              AND   010111i100B           ;Bits 3 and 4
05FA   28 8              JR    NZ, 007H          ;Bit 3 and/or Bit 4 set
05FC   C3 DE             SET   3                 Set Bit J
05FE   AF                XOR   A                 Delete iB2
05FF   32 J7 78          LD    !7837H),A
0602   7A                LD    A,D                ;Load the character
003    32 3 78           LD    (783/),A          ;and to Bt
```

```
06@6  C9              RET                 just one key pressed - ready'

                                Keeping Key

07    C               BIT   ,HL)          already two [boxes in the putter?
0609  20 2A           JP    NZ.0635H      Yes!
060B  3A 36.78        LD    A, 7836HJ     ;Load character from BI
060E  BA              CP    D             ;= Hit key?
060F  20.21           JR    NZ.0632H      No" a new
0611  ED 4B 42 78     LD    BC, i7842Hl   ;Load Row/Column Counter
0615  2A 44.78        LD    HL, (7844H)   ;Load Matrix Address
0618  7B              LD    A,E           ;Contents of the matrix line
601?  CD 35 2F        CALL  2F35H         ;Check remaining keys
061C  BA              CP    D             Like before?
061D  CA D7 2F        JP    Z,2FD7H       yes to character repetition
0620  FE 00           CP    0             No more?
0622  CA D7 2F        JP    Z,2FD7H       ijas for character repetition
0625  21.38.78        LD    HL.7838H      Address {F lag 1
0628  CB EN           SET   3, (HL)       ;Set both state bits 3+4
062A  CB E6           SET   4 HL)
062C  CB 96           RES   2 HL)         ;Reset Function Flag
062E  32 37 78        LD    !7837Hl,A     Trace in B2
0631  C9              RET                 ;and back
0632  7A              LD    A, D          ;new key code in A
0633  18F             JR    0625H         ;enter in B2

                        Two keys already registered

0635  34 3 78         LD    A, 783#)      ;Load character from Bl
0638  BA              CP    D             ;= new key code?
0639  28 08           JR    Z,0643H       Yes!
063.B YES 37 78       LD    A,(7837Hl     ;Load characters from B2
063E  BA              CP    D             $= new key code?
063F  28,112          JR    Z,0643H       {yes!
0641  AF              XOR   A             ;3 keys - Git
0642  C9              RET                 back with A =
0643  ED 4B 42 78     LD    .BC, (7842 Hl ;Load Row/Column Counter
0647  2A 44.78        LD    HL, (7844#)   ;Load Matrix Address
064A  7               LD    A,E           ;Load the contents of the catrix
04R   CD 35 2F        CALL  2F35H         ;Search Matrix IIN!iter
064E  BA              CP    D             Same key?
064F  28 05           JR    Z,0656H       Yes!
0651  FE O            CP    0             No more key?
0653  C2 D7 2F        JP    NZ,2FD7H      yes - for character repetition
```

```
0656  21.38.78      LD    HL.78JBH      Address Flag 1
0659  CB EN         SET   3, HL)        ;Set Bl Status
06511 CB A6         RES   +3 (H)        ;Delete B2 Status
065D  3A 36.78      LD    A,(783bHl     ;Load character from Bl
0660  BA            CP    D             -
  01  20.05         JR    NZ, 0668H     No
063   (J,f          XOR   A             Delete iB2
0664  32 37 78      LD    (7837Hl,A
067   C9            RET                 iback
06b8  3A 37.78      LD    A,(7837Hl     B2 to Bl
066M  32 36 78      LD    (783bHJ,A
06E   18 F3         JR    06b3H         ;B2 Delete
```

t#%t %%%%%%% %%%% %%% ~±3 #

BASIC - Initialisation Part 1

```
0674  O             NOP                 Appends well
0b75  00            NOP
07    21 D2 O       LD    HL,0bD2H      iR0/1 6D2 - 707 in
0679  11 80 78      LD    DE,78001-1    iRAM 7800 - 7835
067C  01 36 00      LD    BC, 31        transmit
0b7F  ED B0         LDIR
0b81  3D            DEC   A             all 12Bx
0b82  3D            DEC   A             {2727222
0b8J  20 FI         JR    NZ,0b7bH      Probably burn in!!!!!
0b85  6 27          LD    B.39          Delete the next 39 bytes
0687  12            LD    (EN),         ;(783b-785C)
0688  13            INC   EN
0689  lil FC        DJNZ  0687H
068   C3 75 00      JR    0075H         To BASIC initialisation T.    2
```

#%%#%%%-%%1%%~%~#~ %%~~$

BASI - Initialisation Part 3

Check to see if the RON cartridge is present

```
068E  21 00 40      LD    HL,40001-1    ;1. Option at 4000H
0691  CD A          CALL  06A4H         Check where
0694  21 00 60      LD    H, 6000H      i2, possibility at b000H
0697  CD M          CALL  0bAAH         check
069A  21 00 80      LD    HL,80001-1    ;3. Option at 8000H
069D  CD A          CALL  0bAAH.        check
```

```
8A0   FB          EI                ;no trav - interrupts
0641  C3 19 1A    JP    1A19H       ;to BASIC - Main Loop

0644  3E AA       LD    , MAAH      ;ROM tray Must be the

0Down BE          CP    HL)         iB!:last    AA 55 E7 18   start
06A7  23          INC   HL          {next byte
048   C0          RET   NZ          was already nowhere
06A9  2F          CPL               ;2. value (55)
06AA  BE          CP    (HU         equal?
06AB  23          INC   HL          ;next .byte
€064  C0          RET   NZ          unequal!
06AD  3E7         LD    A,0E7H      3. Value = E7
AF    BE          CP    (HU         {is that
06B8  23          INC   HL          next byte
06B1  C0          RET   NZ          ;no, not even
06B2  2F          CPL               ;4. value (18)
06R3  BE          CP    HL)         {Does this one agree?
06B   23          INC   Hl…         ;next .byte
0B5   C0          RET   NZ          no - no tray
0B    FB          EI                ;Enable Interrupts
0bB7  E9          JP    (HU         Mount ;ROM Bay

06CC  01 18 1A    LD    BC,1A18H    ;Load the main loop address

06CF  CJ AE 19    JP    19AW        ;BASIC variables and pointer init.
```

 # ### ! t # # ### k i t

The following .range from *bD2* to 707 **is**

transferred to RAN from 70011 to 7835

restart vectors

```
0bD2  C3 9b 1C    JP    1C9bH       ;RST 8H (compare l characters)
0D05  CJ 78 1D    JP    1D78H       ;RST 10H (next character)
0D8   C3 981C     JP    1C90H       ;RST 18H (HI.IDE. Comparison)
06DB  C3 D9 25    JP    25119H      ;RST 20H !Test data type)
66EN  C9 00 00    RET               ;RST 28H
06E1  C9 011 lt   RET               iRST JIH
OE4   FB          EI                3RST 38H (interrupt)
06E5  c9 00       RET
```

Keyboard - Device Control Block

```
66E7  01          DEFB  1           ;DCB type
06E8  F4 2E       DEFW  2EF4H       ;Address of driver
0bEA  00 08 08
```

```
61:D  41! 49          ENfll   'AI'
---
                      Screen - Device Control-l!lock
                      until aut does not use the cursor address.
06EF  0co             DEFB    0               ;DCB type (unknown>
FM    00 00           DEFW    0               ;SET, RESET, and so on. POINT.
06F2  00 70           DEFW    7000H           ;cursor address pointer
06F4  00 00 88

                      Printer - Device Control Block
06F7  O               DEFB    6               iDCB type
0F8   8D 05           DEFW    058DH           ;Driver Address
06FA  43              DEFB    67              ;Lines/Page +1
06FB  00              DEFB    0               ;Line counter
06FC  0o
06FD  50.52           DEFN    'PR'
06FF  C3 00 50        JP      5000IH          not used
0702  C7 00 00        RST     0               ;not used
0705  3E 08           LD      A,              ;strain at talschea DCB type
0707  C9              RET                     in the DCB call routine

                      FFF

                      Addition and subtraction with simple accuracy
                      Different points of strain correspond to the required
                      function.
                      Ring: X= Su111and Subtract
                            Hl... or V SuMand or lhnuend Aus9.< I
                      = Total or difference

                      X= X+ 0.5
0708 21 80 13         LD      HL,1380H        ;Address of constant 0.5

                      X= constant+ X
0701! CD C2 09        CALL 09C2H              ;Load constant to V{jump
070E 18               JR 0716H                to addition

                      X= constant - X
'710 CD C2 09         CALL 09C2H              ;constant to Y
```

```
                    X = Y - X
0713  CD 82 09      CALL  11982H        iX = -X
                    X = Y + X
071   78            LD    A, B          Y = 0? (Exp.   y = 0)
0717  117           OR    A
0718  approx        RET   or            Yes, done
0719  3A 24.79      LD    A, (7924H)    iX = 0? (Exp. X= i!l
071C  B7            OR    A
071D  CA B4 09      JP    Z,09.84H      There, done, I=Y
0720  90            SUB   B             Exp. I - Exp. Y in A
                                        -
0721  310C          JR    NC,072FH      ;Yes
0723  2F            CPL                 ;Negate Exp. Ditf
0724  3C            INC   A             ;X to Y
0725  EB            EX    DE,HL         ;LSB V secure
072   CD A4 9       CALL  09MH          iX aut Stack
0729  OR            EX    DE,HL         Restore iLSB V
872A  CD B4 09      CALL  9:B't         Transfer iV to X
072D  Cl            POP   JC            ;Load stack to V
072E  D1            POP   EN
072F  RD 19         CP    25            ;Exp.Ditt > Nantisse (24 bits)
1731  DI            RET   NC            ;No, X = X
1732  F5            PIJSH AF            Exp.Dift. secure
0733  CD DF 09      CALL  09DFH         ;Set sign bits= 1.
                                        ~A(7) = ! if same sign
                                        A!7) = 8 for unequal characters
1736  67            LD    H,A           ;Save Prefix Flag
1737  Fl            POP   NF            ;Return Exp. Difference
8738  CD D7 17      CALL  07D7H         ;V and push this difference right
173B  M4            OR    H             ;sign the same?
€e73  21 21 79      LD    HL.7921H      ;LSB X address in HL
87JF  F25417        JP    P, 0734#      ;No, subtract
                    addition of nantisses
0742  CD B7 87      CALL  87R7H         ;Add Nantissen
0745  D2 9 87       JP    NC, @79i      ;too loud? No=Jump
0748  23           INC   HL            ;Pointer to Exp. X
0749  34           INC   (L⊤)          Exp. X + 1
@74A  CA B2 17      JP    1,6722A       ;too loud? Yes=OV Error
874D  2E 11         LD    Ln1           ;Nantisse of X and 1 bit
874F  CD EI 87      CALL  87EBH         Move ;Right
1752  18.42         JR    87%~          {finished!
```

```
                    mantisses subtraction
0754  AF            XOR   A             Mant. Y - Mant. X to Kant. Y
0755  90            SUl\  1\            Down. Byte (created by pushing)
0756  47            LD    B,A           ;Result
0757  7E            ID    A, (HL)       ;LSB subtraction
1758  9B            SBC   A,E
0759  5F            LD    E,A
075A  23            INC   Hl            ;next byte
07511 7E            LD    A,(Hl)        ;subtract
075C  9A            SBC   AD
075D  57            LD    D,A
075E  23            INC   Hl            Subtract iMSB
075F  7E            LD    A, (HL)
0760  99            SBC   A,C
0761  4F            LD    CA            Unloud?
0762  DC C3 07      CALL  C,O7C3H       Yes, Mant. Negate Y
                                        invert ;sign flag

                    %#l% ##? tt # Mt } MM }

                        Normalise
0765  68            LD    L,            Erg. Tate. from CDEB to CDHL
07    63            LD    U
0767  AF            XOR   A             Sliding counter= 1
8768  47            ID    B,A
0789  79            LD    A,C           ;l'ISB Y: I?
076/t B7            OR    A
@76B  20 18         JR    NZ.0785H      ;no
076D  A             LD    C,D           ;Y 1111 Slide 1 Byte Left
076E. 54            LD    D,H           H to D
@7F   5             LD    H,L           il to H
0770  F             LD    L,A           L = 0
1771  7B            LD    A,B           ;Slider counter - B
1772  D 08          SUB   8
0774  FE E0         CP    8E0H          132 links? (number = %)
0776  20FO          JR    NZ.0768H      {no!
                    ----------------------------------------
                        Set real number= 1
0778  AF            XOR   M             {exponent in I =
0779  32 24 79      ID    (7924Hl,A     d.h, I = @
077C  C9            RET
                    ----------------------------------------
                        2. Part of normalisation
077D  15            DEC   B             ;Sliding counter - 1
```

```
077E  29          ADD   HiL+rf.     5CDHL emn bit left (HT. * 2)
077F  7A          LD    A,D         D * 2
0780  17          RLA
0781  57          LD    D,A
0782  79          LD    A; C        C * 2
0783  8F          ADC   A,A
0784  4F          LD    C,          ;highest bit  Y set?
0785  F2 7D 87    JP    P.077DH     No" continue
0788  78          LD    A, B        ;Slider counter to A
0789  SC          LD    EyH         ;CDHL back to CDEB
078A  45          LD    B,L
78,8  B7          0R    A           No postponement?
078C  28 08       JR    Z,0796H     1Yes
078E  21 24 79    LD    HL, 7924H   Address exponent
0791  86          ADD   A, (HL)     {Exp. I + Number of moves
0792  77          LD    (HU,A       = Exp. X. Underrun?
0793  30 E3       JR    NC,ltl778H  Yes! I= and back
0795  C8          RET   or          ;Number Moved= Exp.      X?   back!
0796  78          LD    A,B         ;Load LSJl Y
0797  21 24 79    LD    HL.7924H    ;X-exponent address
079A  B7          OR    A           3LSB Y7) = 0?
079.B FC A8 @7    CALL  11.lt17A8H  ;No - round Y
079E  46          LD    , HL )      Exp. } after Exp.  V
079F  23          INC   HL          ;Sign flag
0748  7E          LD    ; (H)       ;Load
07A1  E6 80       AND   BtHH        ;Hide sign
07A3  A9          XOR   C           ;Link to MS} Y (invert)
@7M4  4F          LD    C,A         and back to IISB Y
07A5  C3 BM 9     JP    09B4H       ;Y to X as result
```

###Mi }#kt Eil Mt IH#It#tHIIk

```
                  Round
1.17A8 1C         INC   E           ;LSB V+ 1
07A9  C0          RET   NZ          ;= 0? No-ready
07AA  14          INC   D           n,Byte Y +1
07AB  8           RET   NZ          {= 8? No-ready
07AC  0C          INC   C           IISB Y + 1
07AD  c0          RET   NZ          ;= ?, no-ready
07AE  0E 80       LD    C, 88i      Yes, IISB V = 88H
87.Bltl 34        INC   (li)        {exponent +1
7/81  C0          RET   NZ          i= 0? No-back
```

OVERFLOW Error

```
87R2  lE 0A          LD    E, 10          ;Error number in E
07l14 C3 A2 19       JP    1942H          to the error routine

fffff+ffffffffffffffffffffffffffffffffffffffffffffffffffffffl

Mantissen addition of simple accuracy

Ring: Mantisse X= SuMand
      Mantisse Y = sum
      HL = Address LSB X

Exp.: Nantisse V= Sult11e

07B7  7E             LD    A, (HL)        iLSll X in A
07B8  BJ             ADD   A,E            ;+ LSll Y
07B9  5F             LD    E,A            iSull1lle in LSB Y
07llA 23             INC   HL             ;X Address+ 1
071111 7E            LD    To {H.)        ;Add next byte
07BC  BA             ADC   AD
07BD  57             LD    D,A
07BE  23             INC   HL             ;lt. = ISB X
7BF   7E             LD    A,<HU          ;IISB X+ NSB Y
fll7C0 89            ADC   To C
07C1  4F             LD    C,A            ;in NSB Y
B7C2  c9             RET

t M MM kt i # #Hit#t#tHtt

   Negate Atlantic Y

07C3  21.25.79       LD    HL.7925H       invert ;sign flag
fl17C6 7E            LD    N, (H)
1117C7 2F            CPL
17C8  77             LD    (H)+
1117C9 AF            XOR   A              A =
117CA 6F             LD    L,A            L = lll
117CB 9lll           SUB   B              iLSB V= 0 - LSB Y
17CC  lt7            LD    B,A
fl17CD 7D            LD    A,L            {A =
17CE  9B             SBC   A,E            n.Byte Y = - n.Byte V
fl7CF 5F             LD    E,A
171)1 7D             LD    A,L            A =
07D1  9A4            SBC   A, D           n.Byte Y = - n.Byte Y
17D2  57             LD    D,A
07D3  7D             LD    A,L            {A=0
07D't 99             SBC   AC             3KB Y = @-IISB V
07D5  4F             LD    CA
```

```
07D   C9              RET

                      ±kt#Mi i# E } # % # % Mt t# fit t #

                      Move the number of simple precision to the right
                      Ring: Y = Number
                            A = number of shifts
                      Exp.: Y = Result
                            B = alc. low byte
0707  6e              LD    B,            ;LSB of result= 0
07D9  D8              SUB   8             iB or push 111more digits?
07DB  38 87           JR    C, 07E4#      No!
07DD  43              LD    B,E           Y one byte to the right
07EN  SA              LD    ED
07DF  51              LD    D,C
07Eo  0E 00           LD    C,0
07E2  18 FS           JR    '7D9H
07E4  C 99            ADD   A,9           ;number of shifts+ 1 in L
e7E   F               LD    L,A
07E7  AF              XOR   A             ;Delete Carry
07EB  2D              DEC   L             Slide counter - 1
07E9  approx          RET   or            ;= @? yes-ready
07EA  79              LD    AC            il'ISB V one bit to the right
07EB  1F              RRA
07EC  4F              LD    C,A
07ED  7A              LD    A, D          ;n.Byte Y one bit to the right
07EE  1F              RRA
17EF  57              LD    D,A
07F0  71              LD    A,E           n. .Bvte Y one bit to the right
07F1  1F              RRA
07F2  SF              LD    E,A
17F3  78              LD    A, B          ;LSB Y one bit to the right
@7F4  1F              RRA
07F5  47              LD    B,
07Fb  18 EF           JR    '7E7H         {more

                      Hfffall Falls

                      Constants

07F8  080 08 00 81                        = 1

                      for LOG - Function
```

```
07FC  1113                                  Number of Constants = 3
07FD  AA 56 19 80                           1= 0.598979
0801  Fl 22 76 80                           = 0. 961471
0805  45 AA 38 82                           1= 2.88539
```

 # t %i %###!%t % # kt # #Hi

LOG - Function

Computes the natural Logarith1111s
Eing: X= Argument
From9 X= Result

```
0809  CD 55 09     CALL  0955H
080C  B7           OR    A
080D  EA 4A 1E     JP    PE,           Yes, Function Code Error
0810  21 24 79     LD    HL.7924H      ;Exponent d. Arguments in A
0813  7E           LD    A, HU
0814  01 35 80     LD    BC,           Y = 0.707092
0817  11 F3 04     LD    EN,
081A  90           SUB   B             Offset Exp X in A
081B  F5           PUSH  AF            ;secure
081C  70           LD    (Hl.1.B       5Exp, X =
081D  D5           PUSH  EN            ;Y on stack
081E  CS           PUSH  C
```

X= !Arg - SQR(2)/21 /(Arg+ SQR!21/21

```
081F  CD 16 07     CALL  0716H         3I= X + 8.7087092
0822  Cl           POP   l!c           Load Y again with constant
0823  D1           PP    EN
0824  04           INC   B             5Exp. Y + 1  (V= SQR(2))l
0825  CD A2 08     CALL  08A2H         3X = SQR2)/X
0828  21 FB 07     LD    HL.07F8H      Constant 1 address       load
0821! CD 10 07     CALL  0710H         ;I = I- I
```

Calculate Series

```
082E  21 FC 07     LD    H, O7FCH      {Address of 1, series constant
0831  CD 94 14     CALL  149AH         Calculate {Series
0834  01 80 80     LD    BC,8080H      Y = -0.5
0837  11.80.00     LD    EN.0
083A  CD 16 07     CALL  0716H         1 = X - 0.5
083D  Fl           POP   AF            Exp, d. Arguments
083E  CD B89 0F    CALL  0F89H         I = I +A
```

X = X t LOG (2)

```
0841  01 31 80      LD                                  BC, 8031H
0844  11 18 72      LD                                  DE, 7218H

                     ********************************************************************************************

                     simple precision multiplication

                     I= X i Y

0847  CD 55 09      CALL                                0955H
084A  approx        RET                                 0r
084B  2E 00         LD                                  L.0
084D  CD 14 09      CALL                                0914H
08508 79            LD                                  A, C
0851  32 4F 79      LD                                  (794FH),A
0854  EB            EX                                  DE,HL
0855  22.50.79      LD                                  (7950Hl,A
0858  01 00 00      LD                                  B,0
0859  50            LD                                  D,B
085C  58            LD                                  E,B
085D  21 65 87      LD                                  HL.0765H

080   IT            PUSH                                HL
0861  21 9         LD                                  HL.0869H
086   E5            PUSH                                HL
0865  IT            PUSH                                HL
0B8   21 21 79      LD                                  HL.7921H
0869  7E            LD                                  A, (Hl)
8A4   23            INC                                 HL
086B  B7            OR                                  A
086C  28 24         JR                                  Z,0892H
086E  IT            PUSH                                HL
086F  2E 08         LD                                  0.8
0871  lF            RRA
0872  67            LD                                  HA
0873  79            LD                                  To C
0874  309 0B        JR                                  NC,0881H
0876  E5            PUSH                                HL
0877  2A            LD                                  HL, (79SBH)
087A  19            ADD                                 HL, DE
087B  EB            EX                                  DE,HL
087C  El            POP                                 Hl...
1187D 34 4F         LD                                  A, (794FH)
0880  89            ADC                                 AC
0881  lF            RRA
```

```
0882  4F          LD    C,A          il'tSB
0883  7A          LD    A,D          in. Byte
0884  lF          RRA
0885  57          LD    D,A
0886  7B          LD    A,E          n Byte
0887  lF          RRA
0888  5F          LD    E,A
0889  78          LD    +B           3LSR
088A  lF          RRA
088B  47          LD    ,A
088C  2D          DEC   L            bit counter - 1
088D  7C          LD    +H           Reload X-Byte
088E  20 th       JR    NZ,0871H     bit counter = 0, no-back
0890  El          POP   li.          yes - load X-byte address
0891  RET                            further

0892  43          LD    ,E           ;Result 1 byte to the right. B = E
0893  5A          LD    E,D          5E = D
0894  51          LO    D,€          D = C
0895  4F          LO    C,A          C = 0
0896  C9          RET
```

 # tt#It Mt t Mt# With Hit

simple-precision division

X = X / 18

```
0897  CD A4 89     CAU   09A4H       Save value in X to stack
889A  21 D8 0D0    LD    li,0DDBH    {Address Constant 1
889D  CD BI 09     CALL  89B1H       ii n X
08Ao  Cl           POP   BC          Eh. Load I value in Y
88Al  Dl           POP   EN
```

X = Y / X

```
08A2  CD 55 0%     CALL  0955#       Divisor = ?
88A5  CA 9A 19     JP    Z, 199AH    Yes, DIVISION BY ZERO - Error
08A8  2E FF        LD    L,0FFH      iFlag Expon, discarded. for division
88AA  CD 14 09     CALL  11914H      Exponents and signs disappeared.
Ir'I8AD 34         INC   (Hl.        ;Correct exponent result
I8AE  3            INC   (HL)        ; + 2   (8914 = Exp.Y - Exp,X - 1)
88AF  2B           DEC   Hl...       i li. on l'ISB X
08B0  7E           LD    A, (s)      iX in Divisions-UP (from 7848H)
88Bl  32.89.78     LD    (7889Hl,A     ;IB
88B4  2B           DEC   li.
```

-

```
88B5  7E              LD    , (H)          {n, Berge
08B6  32.85.78        LD    (7885H),A
08B9  2B              DEC   HL
08BA  7E              LD    , (HL)         ;LSB
88BB  32.81.78        LD    (7881H),A
08BE  41              LD    3.c            ;Y to B,HL (Div1dent)
8RF   EB              EX    DE, HL.
08c08 AF              XOR   A              iY = 0 (for quotient)
08C!  4F              LD    C,A
08C2  57              LD    D,
08C3  5F              LD    E,
08C4  32 ac 10        LD    (i788CH),A     MI5B Divisor = @{fr Slide)
08C7  E5              PUSH  HL             iDivident on Stack
08CB  CS              PUSH  BC
08C9  7D              LD    yL             ;LSR Load Div1dent
08CA  CD 80 78        CALL  7880H          iDividend Divisor
08CD  EN 00           SBC   A, 0           {/SR Divdent = transfer, Under Run7
08CF  3F              CCF                  Conplenish {Carry
08D0  30 07           JR    N, 08D09H      Yes - Subtr. back, in ratio
08D2  32 BC 78        LD    (788CH>,A      iNSB Divident in UP
08D5  FI              POP   AF             Remove Divident from Stack
08D6  Fl              POP   AF
0807  37              SCF                  ;Carry flag (1 in quotient)
0808  D2              DEFB  0D2H           iJP NC,              will never run.
                                           to skip the following, 2 POPs
08D09 D1              POP   BC             ;Get Divident v01 Stack
08DA  El              POP   HL             {= Undo Subtraction
0BDB  79              LD    To C           iltSB d. Quotients in A
08DC  JC              INC   A              ;Test Bit 7
08D0  3D              l1EC  A
08IIE 1F              RRA                  last bit for runc:ling in bit 7
08DF  FA 97 07        JP    N, 0797        ;was bit 7 at INC/l1EC=1, yes-done
IJBE2 17              RLA                  ;Quotient I bit to left
0BE3  7B              LD    E              Result bit (o o. 1)
IJBE4 17              RLA                  ;from dell carry bit
08E5  5F              LD    E,A
O8    7A              LD    A,D
0BE7  17              RLA
08E8  57              LD    D,A
08E9  79              LD    AC
fll8EA 17             RLA
08E3  4F              LD    C,A
08EC  29              ADD   HL, HL         iDivident • 2
08ED  78              LD    A,B
```

```
08EE  17              RLA
08EF  47              LD      BA
08F0  YES SC 78       LD      (A, 788CH)      ;= NS} Divident
08FJ  17              RLA
08F4  32 ac 7B        LD      (788CH),A
08F7  79              LD      A+ C            Is the score still 0?
08FB  B2              OR      D
08F9  33              OR      E
08FA  20 CB           JR      NZ,08C7H        No - next
0BFC  IT              PUSH    HL              ;Save Divident LSB
08FD  21 24 79        LD      HL, 7924        ;Address Quotient Exponent
0900  35              DEC     (HU             ;- 1
0981  El              POP     HL              iDivident LSB Reload
0982  20 CJ           JR      NZ,08C7H        Quotient Exp. unequal 0 - further
0904  CJ B2 07        JP      07B2H           Exponent = O, 0WERFLO - ErrOr
```

```
                        ****************************************

                        Processing the exponents and signs for
                        multiplication and division

                        Jump: Division, double accuracy
0907 3E FF              LD      A,0FFH          ;Set foreign exchange flag
0909 2E                 DEFB 2EH                iLD L,0AFH to skip the XOR

                        Jump: Multiplication, Double Precision
090A AF                 XOR     A               ;Set FYag for Multiplication
090B 21 2D 79           LD      HL.792DH        ;KSB Y
090E 4E                 LD      C, HiL)         {sign Y 1n C
090F 23                 INC     HL              ;HL.. Exponent X
0910 AU                 CA (HL)                 {Link to flag
0911 47                 LD      B,A             ;Mult: B=Exp,Y Div: B=-Exp.Y-1
0912 2E 00              LD      L.0             flag in L = 0


                        Jump: multiplication, simple accuracy (L=lill division,
                                simple accuracy (L=FF)
                        LD      A,B             ;Load Exponent V
                        0R      A               ;=? (i.e., Y= B)


                        JR      Z,0937H         Yeah, straight to the main prog.
                        LD      A,L             back ;Load flag
0914 78                                         Exp. Address X
0915 B7                                         ;Link to flag, i.e.
0916 28 1F                                      with ;Division. Exact, -Exp,X-
0918 7D                 ADD     A,B             1 ;otherwise unmodified.
0919 21 24 79           LD      B,A             ;+ Exponent V
091C AE                 RRA                     ;Sum    by    Exponent
                        XOR     B               V;Overflow?
                        LD      A,B
091D  80                JP      P.093H
091E  47                ADD     A, B0H          ;new exponent Y
091F  1F                LD      (HL),A          charge ;underflow/overf
0920 A                  JP                      low
0921 78                 Z,0890H                 ;Add Offset
0922 F2 3 09            CALL 09DFH              ;and as a new exponent spoke X. =
0925 C 8                D       (HL),A          Q? yes-to main prop on back ;handle
0927 77                 DEC     HL              sign
0928  CA 90  08         RET                     ;in Cache (7925H) ;Address
0923  CD  DF  09                                Exponent X
092E 77                 LD      HL,7924
092F 2B                 XOR     H (HL)
0930 C9

                        Overflow/Underflow Expansions
0931   CD 55  09        CALL 0955H              Test sign of X
```

```
0934  2F
0935 E1

0936  B7
0937 E1

0938 F2 78 07
09311  C3 B2 07
```

```
CPL
PO
P       H
        L
OR
POP     A
        PSW
J       0778H
P       07B2H
J
P
```

;Complement Result ;*Remove* Rebound from Stack;Return to Expression Analysis {Was It a Undersound? {still *an* Rsp address from the stack i.e. immediately to the main programme. ;Lower run, X=0, RET ;OVERFLOW Error

****

X= Multiplication of simple accuracy 1it 10

```
093E   CD .BF 09
0941 78
0942 B7
0943 C8
0944 C 82 0946
DA B2 07 0949 47
9944  CD  16  07
094D  21  24  79
0958 34
0951 ce
0952 C3 B2 07
```

```
CALL    09.BFH
LD      A,B
OR      A
RET     Z
ADD A,2
JP      C,07R2H
LD      B,A
CALL    0716H
LD      HL.7924H
INC (alto)
RET     NZ
JP      07B2H
```

Transferred iX to Y }
Value = Q ? {Ep. Y=)

Yeah, done
;Exp, Y + 2, i.e. Y= value + 4
i if overflow
OVERFLOIHrror ;exponent back in Y
X = X+Y, i.e. I= value + 5 ;Exponent X+ 1
i.e. X= value • 10
overflow? no-back
; yes - OVERFLOW Error

FFF

real number test
Ring: X= Number (simple o. double precision) Exp.:
111enn X< 0, A=FF CY=1 S=1
if X= 0, A=00 Z=1 P=1 if I >
6, At

```
8955 3A 24 79
8958 B7
0959 ca
095A 3A 23 79
095D FE
895E 2F
095F 17
09 9F
0961 C0
0962 LC
```

```
LD      A, 7924#)
OR      A
RET     0f
LD      A,
DEFB    ( 7923H)
CPL     lilfEH
RLA
SBC     +
RET
INC     NZ
        A
```

Exponent X Load ;=
07 (X = 0)
Yes!
Load i tlSB X
CP 2F - dummy command, elim. CPL
A complete (sep. indentation) ;sign X in carry push
iA = 0 - carry

{X} ? nemn - ready yes,
A = set 1

```
0963 C9              RET

                     ****************************************

                     8-bit number with sign 1n number easier


                     Convert Precision
                     Ring: A =
                     9.    X=       in one.enau1gket
@964  6 88           LD    B.88H          ;Exponent of result in B
096   11.80.00       LD    EN.0           Delete {for Normalisation
099   21 24 79       LD    HL 7924#       ;Address exponent in X
096C  4F             LD    C,             ;uazuwan!Number in C
096D  70             LD    (HL)+B         ;Exponent to X
096E  6 00           LD    0.0            B=lil for Normalisation
0970  23             INC   HL             ;Address sign byte in X       (NSB)
0971  3.80           LD    (HL),80lH      ;Indicator = '-' Set
0973  17             RLA                  Vorz, the uzu. Number to Carrg
0974  C3 b2 07       JP    0762H          ;for normalisation

                     %t#%%t #}}Mt kt k ME i N k %HHH



                     the blood value of a number
                     shall be Eing: X= Number
                     Exp.: X = Absolutes the number
                     CALL 11994H          iX > 0 ?
                     RET   P              {yes, done

                     ABS - Function



0977  CD 94 89       Invert real number in X
097A  Flil           LD    HL, 7923H      il'ISB X address
                     LD    A, (HI)        ;and Load
097B  E7             XOR   80H            ;Invert Prefix
                     LD    <HL>,A         ;l'ISB X write back
097C  FA SB          RET
097F  CA F6 0A       Invert number in X
                     RST   20~
0982  21 23 79       JP    1'1,0C5BH
                     JP    Z,0AF6H
111985 7E
0986  EE B8O
0988  77
0989  C9                                   ;Check Type X
                                            {Integer? yes - continue with
                                            C53 ;string? yes - TYPE l'IISI'IATCH
                                            Error

                     +drop

                     S6N - Function
                     Eing.: X=
                     Number
```

```
                                etc.< K=,if number =
                                       X= 1, 111enn number
                                       positive X= -1, if
                                       number negative
098A   CD 94                    CALL 0994H              !X Test
       09                      Convert A to 16-bit Integer (with sign)
098D   6F                      LD     L,A              !Number in L
098E   17                      RLA    Number< 0?
098F   9F                      SBC    A,A              yes, -1 in A and H
0998   67                      LD     H,A              No, 0 in A and H
0991   C3 9A 0A                JP     0A9A             !HI... Transfer to X

                                Test all nuclear types
0994   E7                      RST    20H              Check ;Type
0995   CA F6 0A                JP     Z,0AF6H          !String? yes - TYPE NISMATCH
0998   F2 55                   Error
       09                      JP     P,0955H          Ins. o. double precision

99:B   2A 21 79                Integer - Test Number
099E   7C                      LD     HL, (7921H)      ;Integer number in HL
099F   B5                      LD     A,H              = 0?
09A0   C8                      OR     L
09A1   7C                      RET    /                yes - ready
09A2   18 B. B.                LD     ,H               ;no - HSB in A
                               JR     095FH            {e ter bel @95FH

                                **************************************

                                Transport numbers of different types

                                from Xto stack {simple precision)
09AA   EB                      EX     DE,HL            Save HL to DE
0945   2A 21.79                LD     HL, 7921H)       LSB Xin HI.
09A8   E3                      EX     SP), HL.         swap with RET address on stack
949    IT                      PUSH   HL               iRET address back on stack
09AA   2A 23.79                LD     HL, (7923H1      MSB X+Exp. Xin HL
09AD   E3                      EX     (SP),HL          ;ait swap RET address to stack
09AE   IT                      PUSH   HL               ;RET address back to stack
@9AF   EB                      EX     DE,HL            {Restore Contents of HL
9:80   C9                      RET

                                Number of simple accuracy of RAM 1n k
                                Eing.: HL = Address of number i1 memory
09.B1  CD C2 09                CALL 09€2               {Number to Y Transfer
```

-

```
                        Number of simple exactlylgkent from i to X
091\4  EB       EX     DE,HL          {SR Y 1n Hi, Get Hi in DE
09B5   22 21 79  LD     i7921h>,HL     Transfer iHL to LSB Y
09BB   @         LD     + R            MI5R Y+Exp. Y in HL
299    69        LD     L,C
09BA   22 23 79  LD     (7923H)+Hi     as MS u, Exp, K save icontent of
09BD   EB        EX     DE,HL          HL 1111ederherstel len
09RE   C9        RET

                        Number of simple k to Y precision
09BF   21 21 79  LD     HL.7921H       ;Addressing LSB X
09C2   SE        LD     (E,iHL)        ;Load LSB
09C3   23        INC    Hl             ;next ß\jte
09C4   56        LD     D, HL)         ;Load
09C5   23        INC    HL             ;Load MSB
09C6   4E        LD     C,(HU
09C7   23        INC    HL             ;Exp, Load
09C8   46        LD     B, (Hll
09C9   23        INC    HL             Hi. behind the
09CA   C9        RET

                        Transfer the number of simple precision from X to RAM
09CB   11 21 79  LD     DE, 7921H      ;X address in DE
09CE   06 4      LD     B,4            ;Number of bytes f. No. accuracy
09D0   18 05     JR     09D7H

                        Number of each type from (HL) to (DE>
09D2   EB        EX     DE,H.          Swap {destination and source
                                       address
                        Number of each type (EN) transport to (HI) ;Load type of
09D3   3A AF 78  of LD A,7FH)          number
09D    47        LD     B,A            serves as a byte counter
09D7   1A        LD     A, (DE)        Load {Byte
0908   77        LD     (HL),A         ;and transfer to new
09D9   13        INC    EN             section ;Addresses+ 1
09DA   23        INC    Hl...
09DB   05        DEC    B              ; counter - 1
09DC   20 F9     JR     MZ, 09D7H      i> 0? yes - back
09EN   C9        RET                   ready

                        Character processing for real numbers
09DF   21 23 79  LD     HL.7923H       ;Addressing MSB X
09E2   7E        LD     A, (HL)        and load in A
09E3   07        RLCA   #sign in Bit @ van A
```

```
09E4  37          SCF             Carry = Set 1
09E5  IF          RRA             sign in Carry, #SR I7) = 1
09E   77          LD    (HU,A     ;in HSB X back
09E7  3F          CCF             Cape Cape Cape Cape Cape
09EB  1F          RRA             and A(7)
09E9  23          INC   HI        ;Address HL on sign flag
09A   23          INC   HL        ; (7925H)
09EB  77          LD    (Hll, A   {complete. Drop sign
09EC  79          LD    A,C       illSB Y in A
09ED  07          RLCA            Signs } in Mt 8 from A
09EE  37          SCF             Carry = Set 1
09EF  1F          RRA             MSB Y(7) = 1, Vorz.   Y in Carry
09F   4F          LD    CA        il'ISB Y back
09F1  1F          RRA             ;sign in A(7)
89F2  AU          XOR   (HL)      Link to Compl. Vorz#
                                  {A7) = 1 if forward. X = Ins.     y
09F3  C9          RET

                  Value Transport each type from Y to X

                  (78AFH = Type of value)
19F4  21 27 79    LD    HL.7927H  ;Y address in HL
19F7  11 D2 09    LD    DE,09D2H  ;Address of transport routine
09FA  18 lit6     JR    0402H

                  Transport value of each type from X to Y

                  (784FH = type of value)
09FC  21 27 79    LD    H, 7927H  ;V Address in HL
09FF  11 D3 09    LD    EN, 09D3  Transport routine address
0402  D5          PUSH  EN        Adr. transport routine on stack

                  Earn X-address in dependency on v01 type.

0403  11 21 79     ID    DE,7921H  ;X-Adr. f. Integer, Strings, and
                                  ;simple accuracy
ItAlit6  E7       RST   20H          Test igp
0487  D8          RET   C                 ;Double accuracy? no-ready
008   11 1D 79     LD    DE,791DH  X-Adr. t. double precision
I, WIJ  C9        RET

FFF

                  comparison routines

                  Comparison of numbers with simple accuracy

                  One: I = Number 1
```

```
                                Y =
                        Exp.:   K Y, A = 1
                                X= Y, A = 0,    Z = 1
                                X < Y, A = FF,    CY =   S = 1
0A0C  78                LD      A, B            Y = 07
0A0D  B7                OR      A
0A0E  CA 55 09          JP      Z,0955H         i3a - Test X and back
0A11  21 5E 09          LD      HL.095EH        {Adr. the test routine to stack
0A14  E5                PUSH    HL
0A15  CD 55 09          CALL    09551-i         $0?
0A18  79                LD      A, C            iHSB Y in A
0A19  approx            RET     0r              $Yes" sign of Y = result
0A1A  21 23 79          LD      HL.7923H        ;Load MSB X address
0A1D  AU                XOR     (HLJ)           ;sign X= sign V?
0A1E  79                LD      A,C             HSB Y in A
0A1F  FB                RET     M               ;no, -sign of Y = result
0A20  CD 26 0A          CALL    0421            ;Same sign comparison
0423  1F                RRA                     ;Carry 1n bit 7 of A
0A24  A9                XOR     C               if Y negative, A?) invert
0A25  C9                RET
0A26  23                INC     HL              Exp Address. K 1n H
0A27  78                LD      A+ B            Exp. Load Y
0A28  BE                CP      (HI...)         ;Comparison of the two exponents
0A29  C                 RH      NZ              Back if unequal
0A2A  2:S               DEC     HI              {/SB to SB Y Compare
0A2B  79                LD      A,C
0A2C  BE                CP      (HL)
0A2D  c0                RH      NZ              back; if unequal
0A2E  2B                DEC     HI              Nute I With n.Nyte Y
0A2F  7A                LD      A,D
04.3% BE                CP      (HL)
0A31  C0                RET     NZ              {back; if unequal
0A32  2                 DEC     HI              LSB! Compare to LSB Y
0A33  7B                LD      A,E
0A34  9                 SUB     (HI...)
0A35  C0                RET     NZ              back+ if unequal
0AJ6  El                POP     HL              ;X= Y, unstack RET addresses
0A37  El                POP     HL
0438  C9                RET                     back to main programme nit 1=1

integer comparison

 Ring: HI..= Number1   (Zl)
                EN =
                number2 [Z2]
```

```
                                  Exp. : Z1 > Z2, A = 1
                                         Z1 = Z2, A = 0, Z = 1
                                         Z1 < Z2,  A = FF, CY = 1, S = 1
01\39  7A          LD      To,D)          Excuse me?
0A3A   AC          XOR     H
0A3B   7C          LD      ,H             iMSB Z1 in A
0A3C   FA 5F 09    JP      M.095FH        {nemn, sign of Zi = result
0A3F   BA          CP      0              3/SB Z1 = NSR Z2 ?
044    C2 60 09    JP      NZ,            {no, Carry returns result1s
0A43   7D          ID      A,L            iLSB Z1 = LS8 12 7
0A44   93          SUB     E
0A45   C2 60 09    JP      NZ,0960H       ineim, carry yields result
0448   C9          RET

                                  Comparison of Double Precision

                                  X with constant (DE)
0A49   21 27 79    LD      HL.7927H       ;Address Y
0A4C   CD D3 09    CALL    09D3H          ;Transfer constant to Y

                                  Compare X to Y  (Y=7927 t.)

                                  Exp.: X> Y, A = 1
                                         X= Y, A = 0, Z = 1
                                               , A= FF, CY= 1, S = 1
0A4F   11 2E 79    LD      DE,792EH       ;Address Exponent Y
0A52   1A          LD      A, (DE)        Y = 0?
0A53   B7          OR      4
0A54   CA 55 09    JP      Z,0955H        Yes, I determines the result1s
0A57   21 5E 09    LD      HL,            ;Stack test routine address
0ASA   IT          PUSH    HL
0453   CD 55 09    CALL    0955H          33 = O?
0A5E   1B          DEC     EN             ;Address liSB Y
0A5F   !A          LD      A, (DE)        ;MSB Y in A and C
01\61  4f          LD      C+A
0A61   approx      RET     or             X = @, sign Y = Ergbm1s
0A62   21 23 79    LD      HL.7923H       ;Address 11SB X
0A65   AU          XOR     (H1)           ; sign X= sign y 7
MA6    79          LD      +
0A67   FB          RET     M              no, sign Y = result1s
068    13          lNC     EN             Address Exponent Y
0A69   23          INC     HL             ;Address exponent X
0A6A   06 8        LD      O.8            ;B 1\!,ltes compare
0A6C   1A          LD      A,iDE1           ;Load byte from
0A6D   96          SUB     HL)              ;- 1-
                                                byte
```

```
0AbE   C2 23 0A     JP     NZ,0A23H       {unequal, from carry result erm.
0A71   1B           DEC    OE             {drspInter 4,Y -f
0A72   2B           DEC    HL
0A73   05           DEC    B              $8 bytes compared?
0A74   20 F6        JR     NZ, 06CH       {nn, Next Byte
0A76   CI           POP    BC             ; Rt.icksprung address from1, stack
0A77   C9           RET                   {with A=@and l=1 return

                    'i with X vergl e 1 ct,
                    ^^
0A78   CD 4F 0A     CALL   0A4FH          ;o.a. Call comparison routine
0A7B   C2 5 9       JP     NZ, 095EH      if unequal, ResultIs Inverter.
llA7E  C9           RET

                    ************************************

                    CINT Function

                        Convert number to I-Bt integer
                    Elng.  X=Baseline
                    Exp.:  X=Integer
llA7F  E7           RST    20H            ;Test Type of Source
111A8ll 2A 21.79    LD     (OJ L 7921H)   ;X Address in Hl
0A83   FB           RET    M              Integer? Yes - fert19!
111A84 CA F6 0A     JP     Z,0AF6H        ;string? Yes - TYPE NISMATCH Error
0A87   04 B9 0A     CALL   NC,0AB9H       ;Dopp.Accurate.7 yes - first in
                                          Convert {single precision
0A8A   21B21117     LD     Hl,07B2H       ;OVERFLOW ErrOr - Address in Stack
111A8D IT           PUSH   HL
0A8E   34.24.79     LD     A, l7924Hi     {Abs,   I } 32767 ? (Exp.X 2 16)
0491   RD 90        CP     90H
0A93   30 0E        JR     NC,0AA3H       Yes!
0495   CD FB 0A     CALL   0AFBH          ;Integer X   EN
0A98   EB           EX     EN, Hl         further in Hl
499    D1           POP    EN             ;OV Error address from stack
0A9A   22 21 79     LD     (7921H) ,HL    ;HL transferred to X
0A9D   3E 1112      LD     A,2            ; Type = Integer
0A9F   32 AF 78     LD     !78AFH>,A      set
0AA2   C9           RET
04A43  01 80 90     LD     BC,9080H       ;- 32768 in Y !BCDEl
0AA6   11.00.00     LD     EN.0
0AA9   CD 0C 0A     CALL   0A0CH          X = -32768 ?
0AC    Co           RET    NZ             ;no, OVERFLOW Error
0AD    1            LD     H,C            Yes"  HL = -32768
```

```
0 €     LD      L,D
0 1     JR      0A99H           continue bel A99
ʌ o
        %## # # #i k 3 # # % k # i % # # % # MM # # ik k ± E

                CSN6 - Function

        Convert number to value of easier 6enau1
                    E1ng: K = Baseline
        etc.        X= value        in simple precision
0 E     RST     20H             Determine type of Baseline
0A E    RET     PO              Is simple accuracy'
04 C    JP      , @ACCH         Integer? yes - continue with 0ACCH
0 C     JP      Z,0AF6H         ;string? yes - TYPE l'IISMATCH
0 C     CALL    091\FH          ;X to Y
0 C     CALL    0AEFH           ;Type=" accuracy
0A 7    LD      ,B              ;X= 0?
0A B    OR      A
0 a     RET     0r              Yeah, done
0 C     CALL    09DFH           ;no, Separate sign
1 2     LD      HL.7920H        ;First, not to take over.       B
0 4     LD      .B,<HU          ;Provide for rounding
0 C     JP      0796H           ;Rounding and Normalising
ʌ ?
0 2     LD      HL,(7921Hl      ;Integer HL
ʌ ʌ
0 C     CALL    0AEFH           ;Type= simple precision
0 7     LD      A,H             {Parameters for Conversion
0 5     LD      D,L             Provide
0 1     LD      E,0
0 6     LD      B.90H           Set Exponent 1
0 C     JP      0969H           ;to the reprocessing routine
ʌ ?
        tfftfftftlHtfftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftf
        tftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftftf
        +ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft+ft
                CDBL Function

        Number in value of double accuracy u11Convert
                    Eing: X= Baseline
        Exp.:       X= Value in double precision
0 E     RST     20H             Determine Type of Number
0 1     RET     NC              {is already double precision
0 C     JP      Z,0AF6H         iString? TYPE MISl'IATCH Error
0 F     CALL    t, 0ACCH        {Integer? first in one.Exactly1gk.
0 2     LD          HL,0        the four low bytes
M2          LD  (791DH, HL      Delete in X
0A 2        LD  (791FH),HL
F0 ?
```

```
                                Type = double precision
0AEC   3E 08          LD      A, 8            {A = Type Code 8
0AEE   01             DEFB    1               LD RC,043E = Dummy Command
                                              wen ter be1 MAFIH

                                Type = simple exactly1gke1t
0AEF   3E 04          LD      A,4             {A = Type Code #
0AF1   C3 9F 0A       JP      0A9Fii          ;Save to type byte (78AF)

                                Verify that X contains a Strlng
0AF4   E7             RST     20H             Evaluate {type byte
0AF5   approx         RET     or              ;string? Yeah, done
0AF6   1E 18          LD      E, BH           ;Error Code f. TYPE MISNATCH Error
0AF8   C3 A2 19       JP      19A2H           ;to the error output routine

                                %i# k # i# # #k # #k k # # l # #k Hi # # k k
                                # # # # # # #

                                Ge111Single subprogramme for INT, FIX, CINT

0AFB   47             LD      B,A             ; if A = , back to Y = 0
0AFC   4F             LD      C,A
0AFD   57             LD      D,A
0AFE   5F             LD      E,A
0AFF   B7             OR      A
O      approx         RET     or              ok, = @
12:801 IT             PUSH    HL              Address of Exp.     Save X
12:802 CD BF 09       CALL    09BFH           ;X to V
12:805 CD DE 09       CALL    09DFH           ;Separate sign
0B08   AU             XOR     (tHL)           ;X negative?
6B9    67             LD      H,              ;sign in H(7)
0B0A   FC 1F 0B       CALL    1'1,0B1FH       X= neg, LSB X - 1
0B0D   3E 98          LD      A, 97H          ;Mantisses Length - Exponent
0B0F   90             SUB     B               ;= Number of right shifts
0B10   CD D7 07       CALL    07D7H           ;Perform Shifts
0B13   7C             LD      A,H             Was K negative?
0B14   17             RLA
0B815  DC AB 07       CALL    C,07ABH         ija, fixed number + f
0B18   O 00           LD      B,0             iLSB = 0
0B1A   DC C3 07       CALL    C.07C3H         if negative, I = -}
0B1D   E1             POP     HL              ;Load exponent address
0B1E   9              RET
```

```
0B1F  1B          DEC   EN          iLSB - 1
0B20  7A          LD    , D         = 0 7
0B21  A3          AND   E
0B22  3C          INC   A
0B23  C0          RET   NZ          ;no, done'
0B24  0B          DEC   BC          il'ISB - 1
0B25  C9          RET
```

% k Mt####}i##k%#k#kl#k# } f #k Mi M#$

FIX Function

Make integer TOM *of* character

```
Eng.: X = Baseline
Exp.:   = Function value
0B26  E7          RST   20H         Find Type
0B27  F8          RET   l'I         Already Integer? yes - ready
0B28  CD 55 09    CALL  0955H
0B2B  F2 37 0B    JP    P, 037H     Jas weiterbe1 @R37H
0B2E  CD 82 09    CALL  0982H       No, X = -}
0B31  CD 37 011   CALL  0B37H       ;Create Integer
0R34  C3 7B 09    JP    097BH       //= -X
```

###l#vi#M#i#i## # # k #t# l # % # # # # # # kt}

INT - Function

Calculates next lower integer

```
Ring: X= Baseline
Exp.:  X= Integer
0B37  E7          RST   20H         Determine type
0B38  F8          RET   M           Already Integer? yes - ready
0B39  301E        JR    NC,0B59H    Double accuracy? yes - jump
0BJB  28 B9       JR    Z,0AF6H     ;string? yes - TYPE l'IISMATCH
0B3D  CD 8E 0A    CALL  0A8EH       if possible, in Integer Ull!
0B4   21 24 79    LD    HL.7924H    ;Address exponent X
0B43  7E          LD    A, (Hl)     ;and load in A
0B44  FE 98       CP    98H         ;Exponent>= l'lantissenJangen?
0B46  3A 21.79    LD    A, (7921H)  Load iLSB from X
0B49  D0          RET   NC          Yeah, done – no decimal place.
0B4   7E          LD    A, (HL)     ;Load exponent X
0B4B  CD FB 0A    CALL  0AFl\H      ;Remove decimal
                                    and transferred to V
0B4E  36.98       LD    (HL), 9H    ;l'lantissenlänge in      Insert X
050   7B          LD    A,E         iLSB Y in A
0B51  F5          PUSH  AF          ;Stack backup
```


i

1

```
0B52  79          L[I   A,C          premature Y in carry
0B53  17          RLA
0B54  CD 62 07    CALL  0762H
0B57  FI          POP   AF           $formalise" when f, X=-X load
0B58  C9          RET                LSE i
```

```
                  Convert Double Precision to Integer
0B59  21 24 79    i[    Hl,7924H     Load {exponent address
01\SC 7E          LD      (HL)           Exponent 16^7 (X     (3278)
0B5D  FE 90       CP    90H
0B5F  DA 7F 0A    JP    C,0A7FH      $Jas continue De1 of the 'INT
01\62 20 14       J     NZ,          function ;E.xpanent > 16, continue
0,864 4F          LD    0B78I        {Exponent-78H
0R5   2B          DEC   HL           C;Address aut
01\66 7E          LD    , (HL)       MSJ:. ;X= -32768?
0B67  EE  80      XOR   80H          ; sign from k to 4K7);Byte
0B9   6           LD    R            count = l:i
0.B6B 2.B         DEC   HL           next byte
0B6C  1\6         OR    (HU          {if unequal, then
0B6D  05          DEC   B            mht, ;counter -1
0.B6E 20 FB       JR    NZ,0B6BH     when, ready
0.870 B7          OR    A            5A = ? {i.e. Y=-32768?) ;-
0B71  21 00 80    LD    HL,8000H     32768 in HL
0B74  CA 9A MA    JP    Z,0A9AH      Yeah, done!
077   79          LD    AC           ;Exponent Xback to
0B78  FE B8       CP    0B6H         A ;Exponent}= Mantissenlänge?
0B7A  D0          RET   NC           yes, done - no commas ;save
0B7B  F5          PUSH  AF           flags
0B7C  CD BF 09    CALL  09BFH        ;X to V ;Remove
0B7F  CD DF 09    CALL  09DFH        sign ;X = negative ?
0B82  AE          XOR   (HU          ;Address Save Exponent in HL
0B832B            DEC   HL           iExponent X= Plantissenlänge
0B84 36 B8        LD    HL),0E8H     iSignature A(7)
0B86  FS          PUSH  AF           ;X negative, LSB -1
0B87  FC A0 0B    CALL  1'1,0BA0H    Hl = Address l'ISB
0B8A  21 23 79    LD    HL, 7923H    X ;Hantissenlänge -
0.BSD 3E.BB       LD    A,0B8H       Exponent
0BSF  90          SUB   B            ;= Move number of right moves
0B98  6CD9 0D 0B93 CALL 0D69H        {sign back
F1                POP   AF           if X (B, Integer + 1
0B94  FC 20 0D    CALL  ,0020H       ;LSB for normalisation= 0
0B97  AF          XOR   A
0B98  32 1C 79    LD    !791CHI,A    {Flag tr Load Normalisation
0B9B  F1          POP   AF
```

```
0  D     RET          NC                 no normalisation - done
0  C     JP           0CD8H              Jump to Normal 1
0  2     LD           HL, 791DH          Address LS} {
0  7     LD           y (Hi)             Load iLS.B x
0  3     DEC          i-lL                ;- 1
0  B     0R           A                  {was before?
0  2     INC          HL                 inach B!ite 1n X
0  2     JR           Z,0BA3H            iJa+ continue
0  C     RET
```

%#3% # i k % t %%f # # fit % %3 # f # # i %}%}

Multiplication (for tatrix management)

```
                Eing: .BC = Factor
                  EN = factor
                 etc.: EN = Product
0  E     Pi.lSH       HL                 Save iHL
0  2     LD           HL,0               ;Set Result= 0
0  7     LD           A+ B               Factor= 0?
0  B     OR           C
0  2     JR           Z,0:SC4H           Yes, result1s = 0
0  3     LD           A, 1               ;counter for 16 passes
0  29    ADD          HL, HL             Ergbm1s # 2, Overflow ?
0B DA    JP           C.273DH            Yes, BAD SURSCRIPT Error
0  EB    EX           DE,HL              factor in DE + 2
0B 29    ADD          HL, HL
01 EB    EX           DE,HL
0B 30    JR           NC,0.BCIH          ;no spill, 111eiter
0  09    ADD          HL,BC              yes, add factor in R
0: DA    JP           C.273D0            ;overflow, BAD SUBSCRIPT error
0  3D    DEC          A                  ;counter - 1
0  20    JR           NZ, 0RB4H          ;not 0, new pass
0  EB    EX           DE,HL              ;Result in DE
0: E1    POP          HL                 Restore iHL
0  C9    RET
```

****

****

Integer subtract1on

```
                Eing.< DE = Minuend
                  HL = Subtract
          Exp.:   HL = Difference
                    (loud at lower/higher   in X ait one.Accuracy!
0 7       LD      ,H                              sign subtract in carry
```

```
11:SCB 17              RLA
0BC9   9F              SBC   A,A          ;R=-l, if subtraend @,otherwise
fllBCA 47              LD    B,
fllBCB CO 51 0C        CALL  0C51H        ; Substrate,end complement
lllBCE 79              LD    +€           ;A = 0
0CF    98              SBC   A,B          Complement Vrzechen-F lag
0BD0   18 03           JR    @B05H        ;jump to addition
```

i# i k t k % #k####i)k# k# k ### ik #k #k #i k # # k

integer addition

```
Ring: EN = sum
      HL = Summand
Exp.: HI = Total
       (Under/Overflow                  in X. Accuracy)
0BD2   7C              LD    +H           Prez. of 2, Summanden in Carry
fllBD3 17              RLA
0BD4   9F              SBC   —            ;B sign flag
0BD5   47              LD    B,A          {B=-l if sum.({ 0, otherwise B=fall
fllBDb E5              PUSH  HI           ;2, Summanden on stack
0BD7   7A              LD    A, D         {Vorz,      1. Summanden in Carry
0BD8   17              RLA
0D9    9F              SBC   +            ;A=-l if total.(@,otherwise A=@
0BDA   19              ADD   HL, DE       Create {Sum
0BDB   8B              ADC   A+ B         Over run? (if both negative and
                                          Result positive" or if both
illBDC 0F              RCA                positive and the result negative.
0BDD   AC              XOR   H
0BDE   F2 99 64        JP    P, 0499H     no overflow, HL in X and finished
```

Additions - excessive

```
0BE1   C5              PUSH  BC           ;Save sign flag
0BE2   EB              EX    DE, H        1. Summand in H.
0BE3   CD CF fllA      CALL  0ACFH        with one. Accuracy in X
0BEb   Fl              POP   AF           ;sign flag in A
fllBE7 El              POP   HL           ;2. Reload the scan
lllBEB CD A 89         CALL  9A4H         ;X to stack
fllBEB EB              EX    DE,HL        ;2. Sunand in DE
lllBEC CD B            CALL  0CBH         ;and with one. Accuracy in X
lllBEF C3 BF 0F        JP    0F8FH        ;X+ stack margin in X
```

t # % %% #%i% #%~%%1 %% #

Integer - l'ultiplication

```
                    Einq: EN= Factor
                          Hl. = factor
                    Exp.: HL = Product
                                    (if overflow with inf. accuracy in X)
0BF2  7C          LD    ,H        2. Factor = 0 ?
0BFJ  5           0R    L
0BF4  CA 9A 04    JP    l,0A9H    Yes, result = ,  finished
0BF7  E5          PUSH  Hl        2. Save Factor
0BF8  D5          PUSH  EN        il, save factor
0BF9  CD 45 0C    CALL  0C45H     ;Remove sign
                                      $COR of the two signs 1n B(7)
0BFC  c5          PUSH  BC        ;Save sign flag
0BFD  44          LD    MH        $2. Factor 1n BC
0BFE  (41)        LD    CL
0BFF  21.080.00   LD    HL,0      ;Set Result= 0
0C02  PER 10      LD    A, 16     ;counter= 16 passes
0C04  29          ADD   Hl, HL    Result1s +2
0C05  38.1F       JR    C,0C26H     excessive, special routine
0C07  EB          EX    DE,HL     il. factor f2
0C08  29          ADD   HL, Hl
0C09  EB          EX    DE,HL     ;too loud?
0C0A  30.04       JR    NC, 0C10H ;no, no addition
0C0C  89          ADD   HL,BC     Yes result + 2. Factor
0C0D  DA 26 0C    JP    C,0C26H   overflow special routine
0C10  YD          DEC   A         ;counter - 1
0C11  20F1        JR    NZ,0C04H            ;not 0, next pass
0c13  Cl          POP   BC        ;Load sign flag
0Cl4  D1          POP   EN        ;1. Get factor from stack
0C15  7C          LD    A,H       ;Result> 32767?
0Clb  B7          OR    A
0Cl7  FA 1F 0C    JP    l'l,0C1FH Ias overflow!
0C1A  D1          POP   EN        ;2. Get Stack Factor
0C1B  78          LD    A+B             {Result with sign flag
0C1C  C3 4 0      JP    0C4DH     correct

                    l'ultipliKat1ons - too loud
0ClF  EE 80       XOR   80H       ;Result= 32768?
0C21  BS          0R
0C22  28 13       JR    Z,0C37H   Yes!
0C24  EB          EX    DE,HL      ;1. Factor in
0C25  01          DEFB  01H       iLD BC,E1C1 = dummy command
0C2b  C1          POP   BC        ;Load sign flag
0C27  E1          POP   HL        1, Load factor in lil
0C28  CD CF 0A    CALL  0ACFH     I. factor with inc. Exact,    in X
```

```
2R EI
0CZ D A4 059
1i1C2F CD CF 0A
0C32 Cl
0C33 DI
004 C3 47 08

.37 7
0C38 B7
0C39 Cl
0C3A FA 9A 0A
0C3D D5
0C3E CD CF 0A
0C41 DI
0C42 C3 82 09
```

```
POP     ML          {2, fetch factor from Stacx; 1.
LL W?Al             Fanor from X aut Stack
CALL                {2. Fact with emnf. Exactly 19k. mn
0ACFH               X; 1. Stack 1n Y factor
POP     BC
POP     EN          I= Y +j
JP      0847H
LD      A,II        vrzechen-Fiag 1n ;Result1s
OR      A           should be negative se1n {Stack
POP     .BC         cleanlgen
IP      M,          {is negative, H (-3Z768) 1n ;
MA7AH               1. Stack Factor
PUSH DE             Hi (-3278) with emnf.Exactly1gk.1n }
CALL    0ACFH       1. Reload factor
POP     EN          ;X Complement, Ready
JP      0982H
```

```
0C45 7C
0C4b AA
0C47 47
0C48 CD 4C 0C
0C4B EB
0C4C 7C
0C4D B7
0C4E F2 9A 0A
0C51 AF
0C52 4F
0C53 95
0C54 6F
0C55 79
0C56 9C
0c57 67
0C58 C3 9A 0A
```

```
Link sign,
be1 negative factors complement
LD      ,H          {if sign equal,
XOR     D           B!7) = , uneven B7) = 1
LD      B,A
CALL    0C4Cli      {Make Absolute
EX      DE,HL       Values
LD      A,H         ;sign negative?
OR      A
JP      P,MA9AH     No, HL in X, done ;A =
XOR     A           0
LD      C,A         C = 0
SUB     L           0 - L in L
LD      L,A
LD      A,C         iA = 0
SBC AH              0 - Hin H
LD      HA
JP      0A9AH       ;HL transferred to
                    X
```

```
HFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Negative value of an integer form the
ring: X= Argument
Exp.: X= Function value
0C5B 2A 21 79
0C5E CD 51 0C
0C61 7C
0C2 EE 88
```

```
LD      HL, (7921H)     Transfer {Argument to HL
CALL 0C51H              ;0 - HL and X argument
LD      A,H             HL = 32768 ?
XOR     00H
```

```
0C64   B5          0R      L
0C65   C0          RET     NZ              {no, done!
0C66   EB          EX      DE,HL           Yes, H in mnf. accuracy surrounded,
0C67   CD EF 0A    CALL    0AEFH           Type = Mnf. Set Precision
0C6A   AF          K0R     A
0C611  6 98        LD      11.98H          ;Set exponent= 18
0C6D   C3 69 09    JP      0969H           ;continue at 0969H
```

%% t %$%% % $ 3 ±~ % 3% $% ± % 41~$ % 3.

double-precision subtraction

Ring: X= Minuend
Y = Subtract
Exp.: X = Di reference

```
0C70   21 2D 79    LD      HL., 792DH       Load Address SB Y
0C73   7E          LD      A, HU           ;Invert Character V
0C74   EE 80       XOR     B80H
0C76   77          LD      (HL)
```

%±i%%3%%% %#% %#p 3% $3%

double-precision addition

Eing. K = Sumand
Y= Sumand
Exp.: X = Su11111e

```
0C77   21 2E 79    LD      HL.7921H        Load Exponent Y
0C7A   7E          LD      +, (H)          Y= ?
IIIC7B B7          OR      A
0C7C   CS          RET     I               Yes, X= result
0C7D   47          LD      B,A             Exponent Y in B
0C7E   2B          DEC     HL              Address SR Y
0C7F   4E          LD      CR HL)          ; sign Y 1n C
0C80   11 24 79    LD      DE,7924H        Load Exponent X
0C83   1A          LD      A, {DE)
0C84   .B7         OR      A               {X = ?
IIICBS CA F4 09    JP      Z,09F4H         ijas Y after I as result
0C88   98          SUB     B               ;Exponent X>= Exponent Y?
0CB9   308 1       JR      NC,IIIClH       $Yes
0CBB   2F          CPL                     no, exponent diff, invert
ICSC   3C          INC     A
0CSD   FS          PUSH    AF              ;and secure to stack
                   Swap X and Y
0CBE   8E 08       LD      1.7             ;. load byte counter
11190  23          INC     HL              3Address Exponent Y   load
```

```
0C91    IT              PUSH    HL              ;and on stack
0C92    lA              LD      A,<DE)          il Swap Byte
0C93    4b              LD      B, (Hl)
0C94    77              LD      HL),A
0C95    78              LD      A,B
0C9b    12              LD      (EN),A
0C97    1ll             DEC     EN              ;Address pointer - 1
0C98    2B              DEC     HL
0C99    o               DEC     C               Ready?
0C94    20F             JR      NZ.092H         {no, next byte
0C9C    El              POP     HL              ;Load Exponent Y again
0C9D    4               LD      B (HL)          {Exponent Y in B
0C9E    2B              DEC     HL              ;Address 11SB Y in HL
0C9F    4E              LD      C, HL)          ;MSB Y to C
0CA     Fl              POP     AF              ;Load exponent difference
0CA1    FE 39           CP      39H             i>= Nantissene length + 1 7
0CA3    1)0             RET     NC              Yeah, done!
0CA4    FS              PUSH    AF              ;Exponent difference on stack
0CA5    CD DF 09        CALL    09DFH           Remove sign bits,
                                                ;Preliminary statement of the
0CA8    23              INC     HL              ;Zus. Right Shift Byte
111CA9  36 00           LD      HL),@           i (Delete 7926Hl
0CAB    47              LD      B,A             ;B sign flag
0CAC    Fl              POP     AF              ;Load exploit sequence
                                                ;= Moving counter
0CAD    21 2D 79        LD      HL.792DH        ;Address NSB Y
0CB9    CD 69 8D        CALL    0D9H            ;Y right
0CB3    34 2 79         LD      A, (792)        ;pushed out byte
0CB     32 1C 79        LD      (791CHl,A       ;to X
8€B9    78              LD      A+B             Both signs equal?
0CBA    B7              OR      A
CBRB    F2 CF OC        JP      P,0CCFH         ;no, subtraction

                addition of the nantisses

0CBE    CD 33 0D        CALL    0D33H           addition of royalties. Overflow?
0CC1    D2 1ME 0D       JP      NC, 008EH       {no, end
0CC4    EB              EX      DE,HL           $H = Address Exponent X
0Cc5    34              INC     (HU             Exponent X+ 1, too loud?
0cC     CA B2 07        JP      Z,07B2H         Yes, OVERFLOW Error
.c9     CD 90 D         CALL    0D90H           ;Move 1 bit Mantisse right
0CCC    C3 0E 00        JP      0D0EH           on to 0D0EH

                subtraction of the nantisses

0CCF    CD 45 8D        CALL    0D45            ;subtraction of rodents
```

```
0CD2  21.25.79          LD     HL, 7925H      ;Location
0CD5  DC 57 0D          CALL   C,0057H        Down? yes, l'!antisse X
                                              ;Complement

                  ************************************

                      Normalise

0CD8  AF                XOR    A              {shift counter =
0CD9  47                LD     .B,A
0CDA  3A 23.79          LD     A,             ;Load MSB X
0CDD  B7                OR     A              i= 0?
0CDE  20 lE             JR     NZ,0CFEH       ;no '
0CE0  21 1C 79          LD     HL.791CH       ;yes, move X by 1 byte left
0CE3  0E @8             LD     €1.8           {Byte counter
0CE5  5                 LD     D, HL.)        Load Byte
0CE6  77                LD     (Hll, A        ;last byte to this location
0CE7  7A                LD     A,D
0CE8  23                INC    HL             ;Increase URL
0CE9  0D                DEC    C              Ready?
0CEA  21 F9             JR     NZ,0CE5H       ;no, 111eiter
0CEC  78                LD     A,B            Moving Counter - 8
o'ED  D 08              SUB    8
0CEF  FE C0             CP     0C0H           ;40 shifts?          (X= 0)
0CF1  20 E6             JR     NZ,0CD9H       {nPiny Continue
0CF3  C3 78 07          JP     0778H          Yeah, K = @,done!
0CF6  05                DEC    B              {Scatter- 1
0CF7  21 1C 79          LD     HL.791CH       ;Load LSB X
0CFA  CD 97 0D          CALL   0D97H          ;X one bit to the left
0CFD  B7                OR     A              highest bit set?
0CFE  F2 F6 0C          JP     P,0CF6H        No Next
0001  78                LD     + R            ;Number of shifts= 111?
002   B7                OR
0D03  28 09             JR     1, 0D00EH      5Jas to end
005   21 24 79          LD     HL.7924H       ;Address Exponent X
8D08  6                 ADD    A, HL.)        ;new exponent= old exponent
                                              ; + Number, l Shifts
0DO9  77                LD     (HL)           back in X
0DA   D2 78 07          JP     NC, 0778H      {Underrun? yes, I=0, finished
000   approx            RET    or             {X= 8? Yeah, done!
0EN   3A 1C 79          LD     A,             See Bit of LSI! X= 0?
                                (791CH)
0D11  B7                OR     A
0D12  FC 20 lilO        CALL   M, 0D020H      ;no,    Round X
```

```
0D15   21.25.79          LO      HL.7925H        {Pre-draw F laq
0D18   7E                LO      + /HI\          {load and sign
0D19   E6 80             AND     80H             ;Hide
0D1B   2B                DEC     HL              ;Load NSB X
0D1C   2B                DEC     HL
0D1D   AU                X0R     (HL)            {Invert sign and mnt
                                                 {SR X Link
l!ID1E 77                LD      (HL)A           ;11S1\ back in X
0D1F   C9                RET
```

%#}} %## i&3 %i- ##k k # ± i # if # t # # % } k %%

Round

```
0020   21 1D 79          LD      HL, 7910H       ;Load LSB X
0D23   6 1:17 AM         LD      B7              ; length = 7 bytes
0D25   34                INC     (HL)            ;.Byte count + 1, overflow?
0D26   C0                RET     NZ              no, fert1g
0027   23                INC     HL              Yes, next byte
0D28   05                DEC     B               ;all 11antissenbytes?
0D29   20 FA             JR      NZ,             No; further
0D21\  34                INC     (HU             ;Carry through whole mantisses?
0D2C   CA B2 07          JP      Z,07B2H         ;yes, OVERFLOW Error
0D2F   2B                DEC     HL              ;MSB X= 80H
0D30   36.80             LD      (HL), 80H
0D32   c9                RET
```

±% #k t It It # ±}i# k # # E } % } }

Mantissen addition of double accuracy

Mantisse X= Mantisse X+ l'lantisse V

```
0D33   21 27 79          LD      HL.7927H        ;Address LSB Y
0D36   11 1D 79          LD      DE,791DH        ;Address LSB X
0039   O 87              LD      0.7             ;count= 7 bytes
0D3B   AF                XOR     A               Delete Carry
0D3C   1A                LD      A, (DE)         Load bytes from X
0D3D   8E                ADC     A,(HL1          ;.Add Byte From Y
!!IDJE 12                LD      (EN),A          {Save Sume to X
0D3F   13                INC     EN              ;Increase Location
0D40   23                INC     ed
0I41   D                 DEC     C               Ready?
02     20 FB             JR      NZ,             ;no, continue
0D4"   C9                RET
```

```
HHHHHHHHHHHHHHHHFhfffHFFFF
```

```
                        Mantissen - Subtraction of Double Exact1qket
                        Mantisse X= Mantisse X - Mantisse V
0D45   21 27 79         LD      HL, 7927H       iAddress LSB V
0048   11 1D 79         LD      DE, 791DH       iAddress LSB X
0D4B   IE "7            LD      C,7             i7 bytes as counter
0D4D   AF               XOR     A               Delete iCarry
0D4E   1A               LD      A, (DE)         ;Load byte from X
i!D4F  9E               SBC     A,(Hll          Subtract Bwte From V
0D50   12               LD      (EN),A          ;Save Difference in X
0D051  13               INC     EN              Address pointer + 1
0D52   23               INC     HL
0053   i!D              DEC     C               Ready?
0D054  20 F8            JR      NZ,004EH        no, go
056    C9               RET

%%%kii%& t #4}% %%% MM i# i %

                        Complement Mantisse of X
0057   7E               LD      A, (HL)         Complement {sign-F lag
0D58   2F               CPL
0059   77               LD      (HL),A
0D054  21 1C 79         LD      HL, 791CH       ; Load LSB X
005D   06 08            LD      B,8             Byte count = 8
0D5F   A                XOR     A               Delete Carry
0060   4F               LD      C,A             C = O
0061   79               LD      To C            {A=
0062   9f.              S.BC    A,I.L           ;Subtract byte from 0
0D3    77               LD      (Hll, A         ;and save back
0064   23               INC     HL              Address pointer + 1
0D5    05               DEC     B               Ready?
0066   20 F9            JR      NZ,0061H        ;no, continue
0D68   C9               RET

%% #i litt Mi##### l # tli

                        Move 8 bytes right
                        Eing. A = Number of bits to be moved
                                Hi.= Address MB d. to different range
                                C = Content MSB
0D9    71               LD      (Hl.l,C         ;Save MSB
0116A  E5               PUSH    Hl              ;MSB address on stack
8D6B   D 08             SUB     8               ;more than 8 shifts?
006D   38 IE            JR      C,0D7DH         ;no
```

```
0D6F  E1          POP   HL          ;Back address from stack
0D70  IT          PUSH  HL          and back to Stack
0D71  11 00 08    LD    DE,0800H    Counter = 8 ({D)
                                    ;Clear Cache              (E)
0D74  4E          1D    C, (HL)     ;Load byte to C
0D75  73          LD    (HL),E      ;last byte from cache
                                    to enter ;s
0D76  59          LD    E,          {bytes from 1n cache
0077  2B8         DEC   HI.         Address pointer-1
0078  15          DEC   D           .Byte counter -1
0079  20 F9       JR    NZ,         Ready? no-back
0D7B  18 EE       JR    8D6BH       ;Next Byte Shift
0D7D  C 89        ADD   A,9         ; bit shifts + 1
0D7F  57          LD    D,A         in D
0080  AF          XOR   A           Delete Carry
0081  E1          POP   HL          Stack address pointer     load
0D82  15          DEC   D           But a postponement?
0D83  approx      RET   or          No, done
0084  IT          PUSH  HI.         Rescue iAddress Cursor
0D085 1E 08       LD    E.8         Byte count = 8
0087  7E          1D    To (HL)     Load Byte
0DBB  1F          RRA               ;Move 1 bit right
0089  77          ID    (HL),A      ;and back to memory
8DBA  2           DEC   HI...       ;Address pointer -1
8D8B  1D          DEC   E           Counter -1
0D8C  20 F9       JR    NZ.0D87H    4 ready? no-back
0DBE  18 F0       JR    0D88H       ;Next bit shift
```

Mil #t MM It H Ei E } t i t Hi#ttl

X - Move the register one bit to the right

```
8D90  21 23 79    1D    HL.792JH    ;l!SB X
0093  16 01       1D    D,1         bit counter = 1
0D95  18 ED       JR    08D84H      continue at @DB4H
```

ffffffttffffHIIIIIIHFFFFhhHT

Move Storage Range 1 Bit Left

Ring: It.= Initial address of the range
Carry = bit to be copied

```
0D97  0E 08       1D    C,a         Byte count = 8
8D99  7E          LD    A, (s)      ;Load Byte
0D9A4 17          RI.A.             Move Left
```

```
0D9B  77          LD    (HU,A        and back to memory
0D9C  23          INC   HL           ;Address pointer+ 1
0D9D  fl1D        DEC   C            ;Byte counter - 1
0D9E  20 F9       JR    NZ.0D99H     Ready? no-back
flDA0 C9          RET
```

%%iM%%$%% d ℮ % # p % kt 3 } &t Wed #Wed

Duplicate Precision Multiplication

```
Eing:  x= factor
       Y = factor
Exp.:  X= Product
fl1DA1  CD 55 09   CALL  0955H          il, factor = III ?
  0DA4  CB         RET   or             Yeah, done
fl1DA5  CD 0A 09   CALL  111901\H       Exponent and sign disappeared,
  0DA8  CD 39 8E   CALL  0E39H         ;Mantisse 1,factor from X to 414A-
                                       ;4150. Delete X.
  0DAB  71         LD    (HL)+C       ;Delete LSB X
  0DAC  13         INC   EN           ;Address LSB 1. Factor
  0DAD  06 07      LD    B,7          Byte count = 7
  0DAF  lA         LD    A, (DE)      ;.Byte of 1, Load factor
  0DB   13         INC   EN           ;Address pointer 1, factor+ 1
fl1DB1  7          OR    A            ;=8?
  0DB2  D05        PU5H  EN           Stack Address Pointer
 0D113  28 17      JR    Z,0DCCH      Byte is 8!
  80M5  0E 08      LD    C,a          ;not 0, bit counter = 8
  0DB7  c5         PU5H  BC           Save bit counter
 0D118  1F         RRA                ;Set next bit?
  0DM9  47         LD    B,A          ;Transfer Byte to B
  0DTA  DC 33 O    CALL  C, 0D33H     ;yes, 2, add factor to X
  0DBD  CD 90 0D   CALL  0D901-i      Rotate X One Bit Right
  0DC0  78         LD    A,B          ;Byte from B back to A
  0DC1  C1         POP   BC           {Reload Byte Counter
  0DC2  0D         DEC   C            ;Byte finished?
  0DCJ  20F2       JR    NZ,0DB7H     ;no, next bit
  0DC5  D1         POP    EN          Reload {Address Pointer
  OC    05         DEC    B           all 7 bytes processed?
  0DC7  20 E       JR     NZ,0DAFH    ;no, next byte
  0DC9  C3 DB 0C   JP     0CD8H       for normalisation
  0DCC  21 23 79   LD     Hl.,7923H   Right-push result by 1 byte
  00Cf  CD 70 D    CALL   007111
  0DD2  18 Fl      JR     fl1DCSH     ;next clear
```

%%%%3% Mt # Mi I #' } #t #t Mi##

```
                                        Constant 10 idopp.          Ex
                                        Constant 10 (one.           Ex
*****************************************************************************
*****************************************************************************
Division by 1 with double accuracy

                    Eing.< I = vident
Exp.:               X = ratio
          LD        DE , @DDH             Load address of constant 10
          LD        HL.7927H              ; Load address from 'i
CALL                09D3H                 ;constant 10 1n Y
*****************************************************************************
*****************************************************************************
division with double accuracy

                    Eing: X= dividend
                    Y = divisor
Exp.:               X = ratio
          LD        A, (792EH)            {divisor = 9?
          OR        A
          JP        Z, 199AH              ;Yes, DIVISION BY ZERO - Error
CALL                0907H                 ;sign and exponent disappeared.
          INC       HL                    ;Exponent Correction (+2)
          INC       HL                    {(097 er9. Exp I - Ex Y - 1)
     CALL           0EJ9H                 ;Divident in range 414-4150.
                                          ;X for Clear Result
          LD        HL.7951H              See Byte of dividend = 0
          LD        (HL),C
          LD        B,                    ;Delete flag
          LD        DE,794AH              ;Address Dividend
          LD        LT.7927H              ;Address divisor
     CALL           0DB}                  iDivident Divisor in Divident
          LD        A, (DE)               ;NSB Load Divident
          SBC       To                    ;- Carry (C=8)
          CCF                             ;Invert carry, underflow?
          JR        ,@E12H                ;no, move 1 to result
          LD        DE, 7944H             ija" Undo Subtraction
          LD        H., 7927H             iDE=Divident, HL=Divisor
         CALL       0D39H                 iDivident + Divisor in D1vident
          XOR       A                     ;Delete Carry
         DEFB       0DAH                  4JP C,0412H          Dungs never
     LD             (EN),A                Save 3B Divident
          INC       B                     Set Flag
```

```
0E14  34 23 79      LD      A, (7923H)      ;MSB Load result
0E17  JC            INC     A               ;Bit 7 set?
0E18  YD            DEC     A
0E19  !F            RRA                     ;determined .bit   Round in A(7)
0E1A  FA 11 0D      JP      1'1,0D11H       ;finished, for rounding
0E1D  17            RLA                     {bit again push Carry
0E1E  21 LD 79      LD      HL, 791DH       ;LS.B Score (X)
0E21  0E 07         LD      C,7             {Bag counter = 7
0E23  CD 99 0D      CALL    0D99H           Rotate the result left,
                                            ;Insert bit,
 0E2  21 4A 79      LD      HL, 794AH       Address Divident
0E29  CD 97 0D      CALL    0D97H           ;Rotate 1 bit left
0E2C  78            LD      A, B            ;Flag set?
0E2D  .B7           OR      A
0E2E  20C9          JR      NZ,0DF9H        iYes" continue
0E30  21 24 79      LD      HL.7924H        ;no, exponent result - 1
0E.33 35            DEC     (HL)            Down?
0E34  20c3          JR      NZ,0DF9H        No, 111eiter
03    C3 B2 07      JP      07B2H           ;yes, OVERFLOW Error
```

%%4%% %%%%%4t %e Mi # H 4+ % t i t k

Multiplication and Division **Subprogramme**

double accuracy

```
8E.39  79           LD      A,C             ;l'ISB Y in memory
0E3A   32 2D 79     LD       (792DHl,A
0E3D   2B           DEC      HL             Ae1resse l'IS!! Load X
0E3E   11 50 79     LD       EN, 7950       ;Pointer to Hill Register
0E41   01 88 07     LD       BC,07001-i     ;X to Auxiliary Register
                                            ;X delete, byte count = 7
E44    7E           LD       A,(HL)         Load bytes from X
0E45   12           LD       (DEl,A         ;in auxiliary
E4     71           LD       (HL),C         ;Delete Byte in X
0E47   1.B          DEC      EN             ;Address -1
0E48   211          DEC      Hl...
IIIE49 05           DEC      B              {Byte count - 1
E4A    28 F8        JR       NZ,0E44H       {done? no-back
0E4    C9           RET
```

%~ M ii EH} # Mt M HK# Mil#

Duplicate replication 111it 10

Ring: x= factor

From9, Product

```
0E4D CD FC      CALL                    09FCH
0E50 EB         EX                      DE,HL
0E51 2B         DEC                     HL
0E52 7E         LD                      A, (HL)
0E53 B7         OR                      A
0E54 approx     RET                     /
0E55 C 02       ADD                     A,2
0E57 DA B2      JP                      C,07B2H
0ESA 77         LD                      (Hll, A
0E5B IT         PUSH                    HL
0ESC CD 77      CALL                    0C77H
0ESF El         POP                     HL
0E60 34         INC                     (HI)
0E61 C0         RET                     NZ
0E62 C3 B2      JP                      07B2H
     07
```

****

Convert string to double precision

Ring: HL = allresse of the string

```
0E65 CD 78      CALL                    0778H
0E68 CD EC      CALL                    0AECH
0E6M F6 NF      OR                      0AFH
```

#% HH% ME M k k Mi # # NM #Milk HE

String to Match Type

Eing: HL = String address

```
0Ec AF          XOR                     A

ED   EB         EX                      DE,HL
06E  01 FF      LD                      BC, MOFFH

0E71 60         LD                      H, B
0E72 68         LD                      L, B
1:73 CC 9A      CALL                    Z,IA9AH
076  EB         EX                      DE,HL
0E77 7E         LD                      A,<HU
1:78 FE 2D      CP                      2DH
0E7A FS         PWSH                    AF
0E7B CA 83      JP                      Z, l, ESJH
0E7E FE 2B      CP                      2BH
```

```
0E80  28 01          JR    Z,0E83H      yes, next character
0E82  2B             DEC   HL           ;no sign, pointer - 1
0E83  D7             RST   10H          {is a tremor
0E84  DA 29 WF       JP    C, F 29H     Yes!
0E87  FE 2E          CP    2EH          =' ° 7
0E89  CA E4 8E       JP    Z,0EE4H      {yes!
0EBC  FE 45          CP    45H          = 'E'?  (Exponent at one.   Gen.>
0EBE  28 14          JR    Z,0A         ia!
090   RD 25          CP    25H          = '7'2  (Number Considered Integer)
092   CA EE 0E       JP    Z,0EEH       3Yes!
0E95  RD 23          CP    23H          $= 732  (number as dopp,     ).
0E97  CA F5 E        JP    0EF5H        Yes'
0E9A  RD 21          CP    21H          ' 142}  (Number as one.      ).
0E9C  CA F E         JP    1, 0EF 6H    Yes!
E9F   FE 44          CP    44H          = 'D'?  !Exponent at dopp. Gen.)
0EA1  20.24          JR    NZ,0EC7H     {no!

                     Find the item

0EA3  B7             OR    A            Set Type Adjustment flag
0EA4  CD FB 0E       CALL  0EFBH        Number in one. o. Dopp. accuracy
0EA7  IT             PUSH  HL           ;Rescue Address
0EA8  21 BD 0E       LD    HL, OE BRDH  ;Load Reversal Address
0EAB  E3             EX    (SP),H       {i stack with adr. pointer
0AC   07             RST   1MH          next character
0EEAD 15             DEC   D            Exp. Vorz. Flag on   ,_,
0EAE  FE CE          CP    0CEH         ; "-7   (token>
0EB0  CB             RET   or           ija'
0EB1  FE 2D          CP    2DH          = '-12
0EB3  C8             RET   or           Yes!
0EB4  14             INC   D            Exp. Prez. Flag on'+'
0EB5  FE CD          CP    0CDH         i= '+'? IToken)
0EB7  approx         RET   or           3yes!
0EB8  FE 2B          CP    2BH          ;= 7412
0EBA  CB             RET   or           yes'
0EBB  2B             DEC   HL           no sign, Adr, pointer to.
8EBC  Fl             POP   AF           ;Return address vii stack
0EBD  D7             RST   10H          ;Load next character
0EBE  DA 94 gF       Jf'   C, OF94H     Shitter? yes-next at 0F94H
0EC1  14             INC   D            ;no, Exp. Preliminary Flag = ?
0EC2  20 03          JR    NZ,0EC7H     no'
0EC4  AF             XOR   A            Yes, Invert Exponent
0EC5  93             SUB   E
0EC6  SF             LD    E,A          and back 1n E
0EC7  IT             PUSH  Hl...        Save Address Cursor
```

```
0EC8   7B
i11EC9 90
0ECA FA 0A i11F
i11ECD FC 18 0F
0EDi11 20 FB
0ED2 El
0ED3 FI
0ED4 ES
i11ED5 CC 7B
09
i11ED8  El
0ED9    E7
0EDA    ES
0EDB ES
0EDC 21 90 08
0EDF ES
0EEi11 CD A3 0A

0EE3 C9
```

```
LD      A,E             ;Exponent-NacM:omastelien
SUB     B               {difference } 0 ?
CALL    P,0FMAH         Yes, Number+1k, Difference - 1
CALL 1, F 18H           Snemm Number/1, Difference + 1
JR      NZ,i11ECAH      {Repeat, Until Load Conference
POP     HL              = 0 Address pointer ;premature,
POP     AF              en-Flag lay {dresszelger u1eder
P\JSH HL                on Stack Preliminary flag='-?
CALL    Z, 097BH        Yes, I = -K {address pointer
POP     HL              back
RST     20H             Test type
RET     PE              double Exact1gk.2 yes-
PUSH HL                 ready ;address pointer on
LD      HL.0890H        stack ;return address on
PUSH HL                 stack
CALL    0AA3H

RET                     {emnf. Accuracy, if possible,
                        convert {to integer.
```

```
Process Dezi111aldot
RST     20H             Test type
INC     C               i','-Flag = 0? (111ar already
a','?)
JR      NZ,0EC7H        ;yes, done
CALL C,i11EFBH          ;one .accuracy!
                        Integer in one, accuracy UIIIMI,
                        continue at 08901-1

JP      i11E83H         next character
```

```
0EE4 E7
0EE5
0EEb 20 DF
0EEB DC F:s i1lE

0EEB C3 83 i1lE

i11EEE E7
i11EDF F2 97 19
i11EF2 23
i11EF3 18 D2


i11EFS B7


0F6 CD FB i1lE
i11EF9 18 F7
```

```
' X' found
RST     20H             Test type
JP      P, 1997H        no integer, SYNTAX - err
INC     HL              address pointer + 1
JR      0EC7H           Ready!

'It' found
OR      A               ;Set Type Adjustment Flag

'' found
CALL    0EFBH           iX in. o. Dopp. Exactly. uIIIII.
JR i11EF2H              continue at 0EF2H
```

#Mil Mt % i ik # t }# # i# # ME

Convert number to single or double precision Eing,:
X= Baseline

```
                                Z-Flag = I conversion to simple precision
                                Z-Flag = 0 Transformation to double precision
                        Exp.:   1 = Result1s
0EFB  IT                PUSH    Hi.             Save Register to Stack
0EFC  DS                PUSH    EN
0EFD  CS                PUSH    BC
0EFE  FS                PUSH    AF              ;Save flag
0EFF  C BI 0A           CALL    Z,0AB1H         ;Z flag=1, 1n            Gen.
0F082 F1                POP     AF              ;Reload flag
0F03  C4 DB 0A          CALL    NZ,0ADBH        ; Z flag=0, in dopp.   6. transform
OrO   C1                POP     BC              Restore Register
0F07  D1                POP     EN
0F08  E1                POP     H
0F09  C9                RET

FFFFTFFFFFFFFFFFFFFFFFFFFTFFFFFFFFFFFFFFFFFFFT

                        Real number ait 10 R1Ultiplicate

                        Ring: I = Baseline
                                Z flag = 0
                        Exp.:   X= Product
0F0A  C8                RET     0f              Z flag = 1 ?.   back
0F0B  FS                PUSH    AF              A to Stack
0F0C  E7                RST     20H             Test type
0F0D  FS                PUSH    AF              ;Remember type flag
0F0E  E4 3E 09          CALL    P0, 093EH       dopp, gene? => * 10
0F11  F1                PUSH    AF              Reload type flag
0F12  EC 4D BE          CALL    PE,0E4DH        Gen ? => + 1
0F 15 F1                POP     AF              Restore {A-Reg
F 1   3D                DEC     A               ;A - 1
0F17  C9                RET

                                fffffffffffffffffffffffffffft

                        Divide real number by 10

                        Ring.' I = Baseline
                        Exp.: X= Quotient
0F18  D5                PUSH    EN              ;Save Register
0F19  E5                POSH    HiL
0F1A  F5                PUSH    AF
0F1B  E7                RST     2H              Test gp
0F1C  F5                PUSH    AF              Save {Type Flag
0F1D  E4 97 08          CALL    P0.0897H        ;Tvpe=inf.Gen.? => /10
0F28  F1                POP     AF              ;Load Type Flag
```

```
0F21  EC DC 0D        CALL   0DDCH          ;Tvpe=dopp.Gen.? => /10
  0F  F1              POP    AF            Restore register contents
0F25  E1              POP    HL
0F26  D1              POP    EN
0F27  3C              INC    A             iA +1
0F28  c9              RET
```

%# k % ± # # k With # t ## ? k # k # # k # # # # t #

Process digit

```
 F29  D5              PUSH   EN            Exp. Prez. Flag u. Save Exponent
0F2A  78              LD     A,B           {decimal. +1 if ',' flag
0F2B  89              ADC    As C          {(Carry is set)
0F2C  47              LD     BA            in B
0F2D  C5              PUSH   BC            Nachk0111111ast. and save Flag
0F2E  E5              PUSH   HL            Save Address Cursor
0F2F  7E              LD     A, (H)        Load iDigit
F 30  D6 30           SUB    30H           ;Remove Zone Section
0F32  F5              PUSH   AF            ;Cleaned digit on stack
0F33  E7              RST    20H           Test type
IIF34 F2 5D 0F        JP     P,IIF5DH      1. o. Dopp. Accuracy!
```

Integer

```
0F37  2A 21.79        LD     HL, 7921H1    Load iValue from X
IIF3A 11 CD IIC       LD     DE,0CCDH      = 3277 ? (&d.h. 1 + X }=32778>
0FJD  DF              RST    IBH           EN 1it HI. compare
IIFJE 38.19           JR     NC,0F59H      yes in one. Accuracy.
0F40  54              LD     D,H           ;Number 1it 10 multiply
0F41  5D              LD     E,L
0F42  29              ADD    HL, HL        i* 2
0F43  29              ADD    HL, HL        {# 4
F44   19              ADD    HL, DE        + 5
0F45  29              ADD    HL, DE        i* 10
0F46  F1              POP    AF            ;Reload tremor
IIF47 4F              LD     CA            in BC (B =)
0F48  89              ADD    HL,BC         and add to number
0F49  7C              LD     A,H           ;new number> 32767?
IIF4A B7              OR     A
@F4B  FA 57 0F        JP     M, F57H       Yes, in one. accuracy !!!!f,
 0FE  22 21 79        LD     (7921HL,HL    ;new number back in X
0F51  E1              POP    ed            ;Reload address pointer
 052  C1              POP    BC            Aftercoast. +Flag Back
0F53  D1              POP    EN            ;Exp. Prez. Flag + Exponent Back
8F54  C3 83 0E        JP     0E83H         ;next character
```

```
0 7     LD          A,C                 ;Rescue tremor
0 F     PUSH        AF
0 C     CALL        0ACK                ;HL in. Precision etc.
0 3     SCF                             in.Ignore jump command
f 7
0 3
F 0     JR          NC,0F77H            Double accuracy? Yes jump!

        Simple precision number

0 0     LD          BC, 9474            ;constant 1Eb in Y
0 1     LD          DE,2400H
F C     CALL        0A0CH               ;number>= lEb?
0 F     JP          P, 8F74H            Yes, dopp, accurately. transform
0 C     CALL        i!193EH             1%
0 F     POP         AF                  ;Reload digit
0 C     CALL        0F89H               ;and add to number
0 1     JR          0F51H               ;back
r o

        Double Precision Number

0 C     CALL        0AE3H               ;number in dopp, inaccurate,        r
0 C     CALL        0E4DH               ; number + 10
0 C     CALL        09FCH               ;Transfer number to Y
0 F     POP         AF                  ;Load digit
0 C     CALL        094}                Transfer i to X
0 C     CALL        0AE3H               in doP, accuracy Bulldoze
0 C     CALL        0C77H               and add to number
0 1     JR          0F51H               back
r o
********************************************************************************
********************************************************************************
                        Add 8 Bit Integer to Number of Simple Accuracy

Eing. I = 1. Summand (united) Exact.)
                A= 2. Summand (8-bit signed integer)
        Sg.'    I = Sume
0 C     CALL        09AAH               1. Save Summand to Stack
0 C     CALL        0964H               ;2. Sumand with one. Gen. in X
0 C     POP         BC                  il, sand VOii stack in V
0 D     POP         EN
0 C     JP          07lbH               i SUMe
r 3

%%4% %%% ii # kt i t Wed #Mt#

Expansions - Process tremors

@ 7     LD          A, E                {exponent} 9 ?
r p
```

```
0F95   FE 0A          CP     1
0F97   3 09           JR     NC,          $Yes, overflow
0F99   07             RLCA                generate ;E>:ponent *
0F9A   07             RLCA                10
0F9B   83             ADD    A,E
0F9C   07             RLCA
0F9D   86             ADD    +
0F9E   D 30           SUB    30           {2 characters add zone
0FA0   SF             LD     E,A          part ;= new exponent
0FA1   FA             DEFB   0FAH         {J N,321EH Dummg, will never come
0FA2   IE 32          LD     E, 32        out9ef. Exponent = 3, calls
0FA4   C3 BD 0E       JP     0EBDH        overiaut herv. <Process next
                                          digit
```

****

```
       Print line number li..
       = Line number
       H                   {Line number on stack
```

To supplement an error message

```
0FA7   ES             ' IN ' HL,          ;Load text address
0FAB   212419         Eing.: 1924H        'IN' ;return text ;Reload
0FAB   CD A7 28       PUSH   2BA7H        line number
0FAE   E1             LD     HL
                      CALL
                      POP
lei lnumber Eing: H =
Line number
0FAF CD 9A 0A         CALL 0A9AH          ;Line number as integer 1n X
0FB2 AF               0R     A            Delete {Front Flag
0FB3 CD 34 10         CALL 134H           {save u. Initialise Putter
0FB6 B6               0R     (HL)         {X as an unsigned integer
0FB7 CD D9 0F         CALL 0FD9H          ;Convert to String
0FBA C3 A6 28         JP     28A6H        ;Output String
```

%%%±%~~%%%~~%%%%% % ti

```
Number in formatted string W11111Handle
Eing,: X= Number
       A = Format flag
           Bit 0 - Exponent output Bit
           1 - not used
           Bit 2 - also print the sign
           after number bit 3 - '+'
           Bit 4 - '$' before number
           Bit 5 - replace leading spaces with '*'. Bit -
           ',' to output thousands separation
```

```
                    Bit 7 - Format B = Number of
                  occurrences
C = Number of decimal places + 1 (for ',') (Hl.) · · · · =
formatted string ait X'00' closed. CALL 1034H ;Address
buffer start (7930Hl ;Format flag in 78D8H

lilFBE CD 34 10
lilFC1 E6 08 0FC3
28 082 0FC5 36
2 0FC7 EB
lilFC8 CD 94 lil9
lilFCll EB
lilFCC F2 D9 lilF
0FCF 3 2D
lilFD1 C5
lilFD2 ES
lilFD3 CD 71! lil9
lilFD6 E1
lilFD7 Cl
lilFD8 B4
lilFD9 23
0FDA 3 30 0FDC
3A D8 78 0FDF 57
0FE0 17
lilFE1 3A AF 78
@FE DA 9A 18
lilFE7 CA 92 18
0FEA FE 04 lilFEC
D2 3D 10

AND     8               ; '+' spend ? No!
JR      Z,0FC7H         j'+' in
LD      (HL),2RH        buffer ;buffer
EX      DE,HL           pointer in
CALL    0994H           DE ;number>= lil?
EX      DE,HL           ;Buffer pointer back in HL
JP      P,FD9H          yes'
LD      (Hl,2DH         ;-' 1n buffer
PUSH    BC              Stack length parameter ;Stack
PUSH    HL              buffer pointer ;Remove
CALL    lil97BH         sign ;Load buffer pointer ;Load
POP     HL              length parameter ;Reset null
POP     BC              flag ;Buffer pointer behind
OR      H               sign ; '0' in buffer
INC     Hl.             Format flag in D
LD      (Hll, '0'
LD      A,(78D8Hl       ;Do update? ;Load Type
LD      D,A             Yes!
RLA                     ; number= 0, ready
LD      A(78AFH)        ;t. o. Dopp. Accuracy? Yes!
JP      0.189A
JP      Z, llil92H
CP      4
JP      NC, 103DH

0FEF 01 00 00
lilFF2 CD 2F 13

Unconvert Integer to String
LD      BC,0            ;Delete parameters for 'u, ','
CALL    132FH           Create ;String

0FF5 21 38 79
FF8 46
8FF9 OE 20 lilFFB
3A 08 78 lilFFE 5F
OFFF E 20
1001 28 87
1083 78

Process format flag bits 2-5
LD      HL.79:ieH       ;buffer pointer to start
LD      B, (HU          ;Load sign
LD      C,' '           ;Load space after
LD      A,(78D8Hl       filler ;Forat flag
LD      E,A             in E
AND     20H             Fill in the 1t" (bit 5) ; no!
JR      Z,              ;sign= space?
LD      100AH
        A,B
```

```
104    B9              CP      C
1005   0E 2A           LD      C,2AH           Fill = '+'
1007   20 01           JR      NZ,100AH        No!
101119 41              LO      B, C            {sign = padding
100A   71              LD      (HL),           ;Putter character
100.B  D7              RST     10H               ;next character= end of line?
11110C 28 14           JR      Z,1022H         ;yes, do not fill
100E   FE 45           CP      45H               ;=Exp.Id f. United.Exactly,?
1010   2B 10           JR      Z, 1022H        Jas continues to fill
1012   FE 44           CP      44H                     i=Exp,Id t. dopp, right?
1014   28 0C           JR      , 1022          Ijas continues to fill
1016   FE 30           CP      30H             = '0' ?
1018   2B F0           JR      Z, 100AH        ;yes, refill
101A   FE 2C           CP      2CH             4=',' 7
101C   2B EC           JR      Z, 100AH        i.Jas continue
101E   FE 2E           CP      2EH             =,' 7
1020   20 03           JR      NZ,1025H        ;no, do not fill
1022   2B              OEC     HL              ;before ',', 'E' and. 'D' a   No.
1023   36 30           LD      HL),'0°
1025   7B              LD      A,E             Dollar sign before lahl?
1026   E6 10           AND     10H             ;(Format flag bit 4)
102B   28 03           JR      1,102H          {no!
102A   2B              DEC     HI.             ;Buffer pointer - 1
102B   36.24           LD      (HU,'$'         ;'$' in buffer
102D   7B              LD      A, E            ;sign behind the number?
102E   E 04            AND     4               ;(bit 2 of the format flag)
1030   C0              RET     NZ              i3a" back
1031   2B              DEC     HL              ;Puf ferze1ger before number
1032   70              LD      (Hl.),.B        ;Anticipate before the
1033   C9              RET

                       Initialise Buffer and Save Format Flag

1034   32 D8 78        LD      178DBH),A       ;Save format flag
1037   21 30 79        LD      H, 7930~        ;Address buffer start
103A   3 20            LD      (HI),' '        buffer space;
103C   C9              RET

                       Number of simple or double precision

                       unformatted to String.
103D   FE 05           CP      5               ;Find digits
                                               simple enauigk? Carry=1
103F   IT              PUSH    HI.             ;Buffer pointer to stack
1040   ENII0           S.BC    A,              Type - Carry in A
1042   17              RLA                     t 2 = Number of digits
```

```
                                        {one. = , double. = 16)
1043  57          LD    D,A             Number of digits mn D
1044  14          INC   D               j+1
1045  CD 01 12    CALL  1201H           Find the i10 exponent.
1048  01 00 03    LD    C, 0300M        {Parameters for '.' and',' set
104B  82          ADD   A,D
104C  FA 57 10    JP    M, 1057H        inein, exponent in putter
104F  14          INC   D               ;Number of digits + 21 D
1050  BA          CP    D               ;Exponent< Number of
1051  38.04       JR    NC,1057H        ;no, exponent in buffer
1053  3C          INC   A               yes, exponent +3 = decimal
1054  47          LD    B,A
1055  3.82        LD    On2             no exponent is issued
1057  D 02        SUB                   ;exponent-2 1n A
1059  egg         POP   HL              ;Reload buffer pointer
105A  F5          PUSH  AF              Exponent on Stack
105B  CD 91 12    CALL  1291H           ı .  and',' set
105E  36.30       LD    (HL), '0'       i '0' in putter
1860  CC C9 09    CALL  1.8909H         i',' set? yes, buffer pointer+ 1
103   CD AA 12    CALL  12A4H           ;Unuranium mantisses
1066  2B          DEC   HL              ;Buffer pointer -
1067  7E          LD    A, (Hl)         ;Load Characters
1068  FE 30       CP    'lll'           = '0°?
106A  28 FA       JR    Zilll66H        $yes" next
106C  FE 2E       CP    2EH             ;before the last zero '.'7
106E  C4 C9 09    CALL  NZ,             {no! Buffer hands + l
1071  Fl          POP   AF              Load exponent.= 0?
1072  28 lF       JR    1,1093H         ;yes, no exponent in
1074  F5          PUSH  AF              ;Exponent back on stack
1075  E7          RST   20H             ;Test type if one. Gene.    Carry=l
1876  3E 22       LD    A,22H           ' D' / 2 load
1078  8F          ADC   Y               ;Exp.Id = 'D' or 'E'
1079  77          LD    HL),            ;Add to buffer
107A  23          INC   HL              ;Buffer pointer+
U!7B  Fl          POP   AF              ;Load exponent.  "€?
107C  3 2M        LD    (HL),'+'        i'+' in putter
107E  F2 85 10    JP    P,1085H         Exponent> 0!
1081  36 2D       LD    (HL>, ._,       ;'-° in buffer
1083  2F          CPL                   Remove {sign
1084  3C          INC   A
1085  06 2F       LD    B,2FH           ;tremor= '0' - 1
1087  04          INC   B               Digit + i {returns 1.  Number)
1088  D MA        SUB   18              ;Exponent - 10 = Undernoisy?
108A  FR 30       JR    NC,             {no+ more
                                        1087u
```

```
108C   C6 3A          ADD    #, 3ri          Yes, last subtract1on back9an919
                                                    !O                  er
108E   23             INC    HL              {Buffering + 1
108F   78             LD     (HL),B          i. Figure 1n Buffer
1090   23             lNC    Hl              Bufferz1ger + I
1091   77             LD     HL),            ;2. Buffer tremor
1092   23             lNC    HL              {Buffer1ger + I
1093   300            LD     HL.),0          ;End Id 1n Putter
1095   EB             EX     EN, HL          ;Buffer end address in DE
1096   21 30 79       LD     HL.7930H        ;HL buffer start address
1099   C9             RET                    ;finished th

                      Format 3tr1ng witness

109A   23             INC    Hl              ;Buffer + 1
109.B  c5             PUSH   .SC             Stack length parameter
109C   FE O           CP     4                   ;simple o. Double accuracy?
109E   7A             LD     A,D             ;Format flag 1n A
109F   D2 09 11        JP     NC, 1 109       Yes!

                      Convert Integer to String

10A2   1F             RRA                    ;Exponent Output?       (bit 0)
10A3   DA A3 11        JP     c, 11A3k        Yes!
106    01 03 0b        LD     BC, 0603H       ;Parameters for'.' and','
                                              ;',' after 5th digit, ',' after  p.
10A9   CD 89 12        CALL   1289H           no ',' output?
10AC   D1              POP    EN              ;Load length parameter in DE
10AD   7A              LD     A,D             Incidents - 5 }= ?
10AU   D 05            SUB    5
10Be   F4 9 12         CALL   P, 1269H        ;corresponding, Number of zeroes
1083   CD 2F 13        CALL   132FH           ;Convert number to string
1B.1.6 7.B             LD     A,E             ;no recomposing points?
10/87  B7              OR     A
10.BS  CC 2F 09        CALL   Z.092FH         Yeah, '.' Delete to Buffer
10.BB  3D              DEC    A               {decimal places- 1 } ?
10.BC  F4 9 12         CALL   P, 1269H        ;equivalent number of zeroes
10.BF  E5              PUSH   HL              ;Pufter hand aut stack

                      Remaining Formatting

                      Create correct field length
10C    CD F5 F         CALL   0FF5H              Residual format requirements
                                              ;sign behind number?
10C3   E1              POP    HL              Load {buffer pointer
10C4   28.82           JR     Z,10CBH         ;no'
10C6   70              LD     (H)+B           ;Put sign after number
```

```
10C7  23          INC   HL          Buffer +1
10C8  36 00       LD    (HL),0      Highlight end of line mt '@'
10CA  21 2F 79    LD    HL, 792FH   ;Load address before putter
10CD  23          INC   HL          Putter address +1
10CE  3A F3 78    LD    A,          iLSB ',' Position
10D1  95          SUB   L           ;- LS.B Putter Address
10D02 92          SUB   D           ;- Vork011t111 = 0?
10D3  approx      RET   0r          Yeah, done
10D4  7E          LD    ;{HL.)      Load Characters
1005  RD 20        CP    20H         =° °2
10D7  28 F4       JR    Z, 10CDH    yes; further
1009  FE 2A        CP    2AH         = 'x 2
10DB  28 F0       JR    Z,10CDH     iYes"
10D0  2.B         DEC   HL          Buffer Pointer -I
10EN  E5          PUSH  HL          ;and on stack
10DF  FS          PUSH  AF          Zechen +Flag on Stack
10E0  1 DF 10     LD    BC,10DFH    ;Set Reversal Address
10E3  CS          PUSH  .BC
18E4  D7          RST   101-1       next character
10E5  FE 2D       CP    2DH         {= 7-7 3
10E7  C8          RET   0r          yes"
10E8  FE 2B       CP    2BH         ;= '+'?
10EA  C8          RET   0r          Yes,
 10B  RD 24       CP    24H         ;= '$° 2
10ED  C8          RET   0r          Continue
10E   C1          POP   BC          ;Remove Reversal Address
10EF  FE 30       CP    30H         = '0'?
10F1  20F         JR    NZ,1102H    no, field overflow
10F3  23          INC   HI          {Buffer pointer +1 {behind ','}
10F4  D7          RST   10H         ;next character. =digit?
10F5  30 0B       JR    NC, 1102H   ;no, field noise
10F7  2B          DEC   HI          {Buffer ze19er to '.'
10FB  01          ENfll 01          iLD.BC,772.B Duny-Betehl
10F9  2B          DEC   HI          ;Buffer pointer - 1
10FA  77          LD    (H),A       ;character in putter
10F.B F1          POP   AF          ;Get Characters From Stack
10FC  28 F.B      JR    Z, 10F9H    Last sign? no-to 1@F9H
1FE   C1          POP   .BC         ;Get buffer pointer from stack
10FF  C3 CE 10    JP    10CEH       continue at 1@EH

                  field noise
11112 F1          POP   AF          Fetch Character from Stack
11113 28 FD       JR    Z, 11112H   ;last character?
11115 E1          POP   HL          ;Load buffer pointer
```

```
1106  36.25          LD     (HL),25        i't.' for field overflow before
1108  C9             RET

                     Formatted string of numbers easier or

                     generate double precision
1109  E5             PUSH   HL             ;Buffer pointer to stack
110A  lF             RRA                   ;Exponent Output?
110B  DA AA 11       JP     C, llAAH       Yes!
110E  28 14          JR     Z, 1124H       with simple precision. => Jump
1110  11.84.13       LD     EN, 1384H      ;Address constant 1D16
1113  CD 49 0A       CALL   0A49H          ;lahl }: 1016?
1116  16.10          LD     D,1            ;Accuracy (16 digits)       in D
1118  FA 32 11       JP     M, 1132H       5Number 1DI6!

                     field overflow

111B  egg            POP    HL             ;Load buffer pointer
111C  Cl             POP    RC             Load {length par aneter
111D  CD BD 0F       CALL   0FBDH          ;Create unformatted string
1120  2B             DEC    HL             ;Buffer pointer - 1
1121  38.25          LD     (Hl),25H       i'7.' for field overflow before string
1123  C9             RET

                     Simple precision number

1124  01 8E B        LD     BC, OB6EH      ;Y= Set IE
1127  11 CA 1B       LD     EN, BCAH
112A  CD 0C 0A       CALL   0A0CH          Number )= 1E6 ?
112D  F2 lB 11       JP     P, 11 lBH      yes, field overflow
11/30 16.06          LD     D,             ;Accuracy (6 digits)       in D
1132  CD 55 09       CALL   0955H          Number = 8?
1135  C4 01 12       CALL   NZ, 1281H      No, Exp. - enauigk, +1       in A
1138  El             POP    Hl...          ;Load buffer pointer
1139  Cl             POP    BC             Load Length Parameters
113A  FA 57 11       JP     M, 1157H       decimal places? yes - jump

                     no decimal places

1130  C5             PUSH   BC             ;Stack length parameter
113E  5F             LD     E,A            Exp, - Accuracy +1 in E
113F  78             LD     A+B            {Incident fe] Dances in A
1140  92             SUB    D              ;- exponent
1141  93             SUB    E              ;- 1 }= 8?
1142  F4 69 12       CALL   P, 1269H       ;yes, equivalent number of zeros in
1145  CD 70 12       CALL   1270H          Paraeter f, ',' u. ','
1148  CD A 12        CALL   12AAH          Create ;String
114B  .B3            OR     E              {exponent-accuracy+1 } 8 ?
```

```
114(   C4 77 12        CALL   NZ,1277H      ;yes, equivalent number of zeros in
                                            5With',' and ','
114F   B3              OR     E             ;Exponent accuracy+!) 0?
1150   CA 91 12        CALL   NZ,1291H      yes, ',' and ',' in buffer
1153   Dl              POP    EN            Load Length Parameters
1154   C3 116 10       JP     10B6H         1. Run Formatting

                       decimal places present

1157   5F              LD     E,A           ;Exponent - Precision+ 1 to E
1158   79              LD     A,C           {decimal field length in A
1159   B7              OR     A             ;) 8 ?
United 16 OF           CALL   NZ,0F16H      Yes, I for '.'
115D   83              ADD    A,E           More than there?
11SE   FA 62 11        JP     I'l,11b2H     Yes!
1161   AF              XOR    A             ;no, number of digits overdue = 0
1162   CS              PUSH   BC            Stack length parameter
1163   FS              PUSH   AF            ;-Number of digits on stack
1164   FC 18 0F        CALL   1'1,0FlBH     Remove superfluous digits
1167   FA 11           JP     1'1, llb4H    {finished? no-back
116A   Cl              POP    BC            number of digits per stack
116.B  7B              LD     A,E           ;-Number of records to be issued
116C   90              SUB    B             ; decimal places in A
116D   Cl              POP    .BC           ;Load length parameter again
116E   SF              LD     E;            {E aftercomb
116F   82              ADD    A, D          + Accuracy } 0 ?
1170   78              LD     + R           {Incl. Field]Length in A
1171   FA 7F 11        JP     , 117F        Yes!
1174   92              SUB    D             ;Incident length - precision
1175   93              SUB    E             ;+ excl. Put it back in the box} 0?
117b   F4 9 12         CALL   P, 129H       corresponding Number of zeros mn
1179   CS              PUSH   BC            ;Length parameter on stack
117A   CD 7D 12        CALL   127DH         Parameter t. ',' u.    ',' find
117D   18 11           JR     1190H         {more at 1191:lH

                  To be discontinued only

117F   CD 69 12        CALL   1269H         ;t.Vorko111111paddle zeros in buffer
1182   79              LD     A,C           Aftermath]d]length in A
1183   CD 94 12        CALL   1294H         i'.' in Buffer
1186   4F              LD     C,A           ;decimal length} in C
11B7   AF              XOR    A             ;tats.Nachko111111Edit
11B8   92              SUB    D             ;- accuracy
1189   93              SUB    E             {= Number of zeros to insert
118A   CD 69 12        CALL   1269H         Add zeros to buffer
118D   CS              PUSH   BC            Save length parameters to stack
```

```
118E 47          LD    B,          Param. for '.' and ' delete
118F 4F          LD    C,
1190 CD A4 12    CALL  12A4H       {Str1ng 1n buffer
1193 Cl          POP   BC          Load length parameters from stack
1194 BI          OR    C           Number plate long } @?
1195 20 03       JR    NZ, 119AH   Yes!
1197 2A F3 78    LD    HL, (78F3Hi '.' - Load Address
119A 83          ADD   A,E         Number of decimal places - Number
                                   decimal places issued
119B 3D          DEC   A           ;- 1 fr '.'
119C FA 69 12    CALL  P,1269H     i> 0 ?Ex. number zeros issued.
119F 50          LD    D,B         {Incident Field Length
11A0 C3 BF 10    JP    10BFH       continue      10BFH
```

Formatted Exponent Output

Integer Swing

```
11A3 E5          PUSH  HL          stack pointer
11A4 D5          PUSH  EN          ;Format flag on stack
11AS CD C 0A     CALL  0ACK        Integer 1n single,enauigk,transform
11A8 D1          POP   EN          ;Reload format flag
11A9 AF          XOR   A           Flag t. Set precision
```

Incentive *f*, single and double accuracy

```
11AA CA B0 11    JP    ,11ROH      ;one.Accuracy?=> Jump
IIAD 1E 10       LD    E, 10       idopp.Accuracy = 16 digits
11AF 01          DEFB  01          iLD BC,061E  dummy instruction
11B0 1E %6       LD    It          {enf.Accuracy = digits
11B2 CD 55 09    CALL  0955H       ;Number= 0?
11BS 37          SCF               Yeah, put carry
11B6 C4 01 12    CALL  NZ, 1201H   ;no, exponent precision+1,C=0
11B9 Et          POP   HL          ;Load putter pointer
11BA Cl          POP   BC          Load Length Parameters
11BB FS          PIJSH AF          ;Exp. Accuracy+l u. Save Flag
UBC 79           LD    A,C         {Conflict panel dlength = ?
11.BD B7         OR    A
11.BE F5         PUSH  AF          decimal length on stack
11.BF C4 16 0F   CALL  NZ,1fF16H   No, trailing length - 1
11C2 80          ADD   A,.B        {Add Event Field Length
11C3 4F          LD    C,A         Total seed field length
11C 7A           LD    A+D         Test iForat Flag
11C5 E6 84       AND   4           ;sign behind number? (bit 2)
11C7 FE IU       CP    1           i3a, 8 to format flag
11C9 9F          SBC   + ,         ;otherwise -
11CA 57          LD    D,A
```

```
11CB  81          ADD   C          Total length-1 if sign
                                   Makes Behind Number
€11C  F           LD    C,A        ;in C
11CD  93          SUB   E          ;- Exp,-Exact+1 returns
11CE  F5          PUSH  AF         ;-Number of digits rounded away
11CF  c5          PUSH  BC         Stack length parameter
110   FC 18 F     CALL  M, WF 1    ;Job ldl!(Jround!
11D3  FA D0 11    JP    11,11D0H   {Loop to Number = 0
11D   Cl          POP   BC         ;Load length parameter
11D7  F1          POP   AF         ;Number of rounded digits
11D8  CS          PUSH  BC         Length paraeters back to stack
11D9  F5          PUSH  AF         ;Number 111egger. Make Stack
11DA  EN 11       JP    M, 11EH    ;Rounding out places? yes to 11DEH
11DD  AF          XOR   A          ;no digits rounded
11DE  2F          CPL              positive number
11DF  3C          INC   A          j+ 1
11Efll 88         ADD   A, B       + Template length
11E1  JC          INC   A          + 1
11E2  82          ADD   A,D        ;- 1 if sign before number
11E3  47          LD    ,A         {= Position of the','
11E4  0E 08       LD    C,fll      {parameter f, ',' = (no ',')
11E6  CD A4 12    CALL  12AAH      ;Putter String
11E9  F1          POP   AF         Total length - accuracy } 0 ?
11EA  F4 71  12   CALL  P,1271H   ;yes, equivalent number of zeros in
11ED  Cl          POP   BC         Reload length parameters
11EE  F1          POP   AF         {Marchkonma length = @?
11EF  CC 2F 09    CALL  1.092H     Yeah, '.' Delete in Putter
11F2  Fl          POP   AF         Number= 0?
11F3  38.03       JR    C, 11FH    Yes!
11F5  83          ADD   A,E        {exponents to be issued]n
11F6  90          SUB   B
11F7  92          SUB   D
11F8  or          PUSH  BC         ;Stack length parameter
11F9  CD 74 10    CALL  1il74H     ;Exponent in Buffer
11FC  EB          EX    EN, t.     ;HL buffer end address
11FD  D1          POP   EN         Length Parameters in DE
11FE  CJ BF       JP    10BFH      Continue at 10BFH

Number as often 1it 10 multiply or by 10
    divide, to exactly 6 or 1 digits.

1211  D5          PUSH  EN         Save {DE}
1282  AF          XOR   A          ;number of moves =
1283  FS          PUSH  AF         ;Number of moves on stack
1214  E7          RST   20H        Test Type
```

```
1205  E2 22 12        JP    P0, 1222H      {simple accuracy!
1208  34.24.79        LD    A,             lah1 )= 65536 ?
120B8 FE 91           CP    91H
1200  02 22 12        JP    NC,1222H       ija'
1210  11.64.13        LD    EN, 1364H      ;Address constant 1010
1213  21 27 79        LD    HL.7927H       ;Y
1216  CD DJ 09        CALL  09D3H          i1D10 to
1219  CD A1 D         CALL  0DA1H          ;Number 1D10
121C  Fl              POP   AF             Load number of moves
121D  D6 0A           SUB   10             ;-10
121F  FS              PUSH  AF             ;and again Stack
1228  18 E           JR    1208H          whom ter
1222  CD 4F 12        CALL  124FH          {Number X=1E !) or 1D1 (#)2
                                           Continue at 1244}
1225  E7              RST   20H            ;Test Type
1226  EA 34 12        JP    PE,1234H       dop. Accuracy? Yes,        1234H
1229  01 43 91        LO    B,7143~        ;constant 100000 in Y
122C  11 F9 4F        LD    DE,4FF9H
122F  CD 0C 0A        CALL  0A0CH          Number } 100000 ?
1232  18O             JR    123AH          Continue at 123/
1234  11 6C 13        LD    IJE,136CH      ;Address constant 1015
1237  CD 49 8A        CALL  0449           ;Number>= 1D15?
12311 F2 4C 12        JP    P.124CH        Yes!
123D  Fl              POP   AF             Load ;Shifts
123E  CD 8 OF         CALL  0F0BH          ;number/ 10, shifts+ 1
1241  FS              PUSH  AF             ;Shifts to stack
1242  18 th           JR    1225H          further
1244  Fl              POP   AF             Load ;Shifts
1245  CD 18 0F        CALL  0F18H          {number + 1, shifts - 1
1248  F5              PUSH  AF             ;Shifts to stack
1249  CD 4F 12        CALL  124FH          Number Y= 1E t!) or 1D1 (#) ?
                                           5yes; Continue at 124
124C  Fl              POP   AF             Load ;Shifts
1240  D1              POP   EN             Restore {IE
124E  c9              RET
124F  E7              RST   20H            Test Type
1250  EA SE 12        JP    PE, 125EH      Double accuracy!
1253  01 74 94        LD    BC, 9474       ;constant 1E6 in V
125   11 F8 23        LD    DE,23FBH
1259  CD 0C 0A        CALL  04CH           Number)= 1D ?
125C  18              JR    1264/i
125E  11.74.13        LD    EN, 1374H      ;Address constant 1D16
1261  CD 49 84        CALL  049%           Number >= 1016?
```

```
1264  E1              POP   HL              ;Load Reversal Address
1265  F2 44 12        JP    P,1244H         ijas after 1244H
1268  E9              JP    (H)             ;no,   normal return

                      Write zeros in putter

 129  117             OR    A               Number= 0?
126A  C8              RET   or                Yeah, done
126B  3D              DEC   A                 ;count - 1
126C  36.30           LD    HL,'0'          i'0' in buffer
126E  23              INC   HL              ;Buffer Address+ 1
126F  18 F9           JR    126AH           continue with 126AH

                      zeros in putter with '.' and ','

1271  20.04           JR    NZ,1277H        Number } @? Yes,   jump
1273  C8              RET   or              = 0? Ready?
1274  CD 91 12        CALL  1291H           i',' and','
1277  36.38           LD    (HU,'0'         j'0' in buffer
1279  23              INC   HL              ;Buffer Address+ 1
127A  30              DEC   A                 ;count - 1
127B  18 F6           JR    1273H           back

                      Parameters for'.'           and',','

1270  7B              LD    A, E            ;Number of moves in A
127E  82              ADD   To D            ;+ accuracy
127F  JC              INC   A               + 1
1280  47              LD    M,              _
1281  JC              INC   A               j+ 1
1282  D 03            SUB   3               ;Position of',','
1284  30 FC           JR    NC,1282H        - 3 to A negative
1286  C 05            ADD   A,5             j+ 5
1288  4F              LD    C,A             as ',' parameter in C
1289  3A D8 78        LD    A, (78I8%)      ;Load Format Flag
128C  E 40            AND   40H             ;',' wanted ?{bit 6)
12BE  CO              RET   NZ              Yeah,
128F  4F              LD    CA              Delete {no, ',' parameter
1290  C9              RET

                      '.'and',' set

1291  e5              DEC   B               ;Decimal - 1
1292  20.08           JR    NZ, 129CH       Reached decimal? No!
1294  36 2E           LD      '             '.' in Buffer
 129  22 F3 78        LD    (78FJH1,HL      ;Remember address of,'
1299  23              INC   HI.             ;Buffer pointer+ 1
129A  48              LD    C,.B            i",' = 0 (no','      (more)
```

```
129R  C9              RET                 {finished
129C  0D              DEC     C           ;','-Parameter- 1.     Night, next
129D  C0              RET     NZ          no" back
129E  €3 2            LD      (HU,','     i','in buffer
1240  23              INC     HL          ;Buffer + 1
12A1  0E 03           LD      , 3         ;','-Param = 3 f., next ','
12A3  C9              RET

                      Simple and double accuracy

                      to ASCII - string
12A4  DS              PUSH    EN          Secure ;DE
12A5  E7              RST     20H         Test type
12    E2 EA 12        JP      PO0,        Basic accuracy? continue with  12EAH
12A9  C5              PUSH    BC          Parameter f. '.' and ','   on Stacl:
12AA  IT              PUSH    HL          {Put ferze1ger on stack
12AB  CD FC 09        CALL    09FCH       iNumber in Y
12AU  21 7C 13        LD      H1,137CH    Address {constant @.5
12111 CD F7 09        CALL    09F7H       30.5 in
12B4  CD 77 0C        CALL    0C77H       ;number + 0.5 to X
12B7  AF              XOR     A           {Normalisation Flag (y) Delete
12BB  CD 7B 0B        CALL    8B7BH       Disconnect
12RB  Et              POP     HL          Load {buffer pointer
12BC  Cl              POP     BC          ;Parameters for '.' and ',' load
12BD  11 BC 13        LD      DE,138CH    Adr. {Fixed Constant 115-1D1.
12C0  3E 0A4          LD      A,1@        ;Numbers= 10
12C2  CD 91 12        CALL    1291H       '.' and ','
12C5  C5              PUSH    BC          Save {parameter for '.,' and','
12Cb  FS              PUSH    AF          Save Numbers
12C7  E5              PUSH    Hl...       ;Save buffer pointer
12CB  D5              PUSH    EN          ;Save constant address
12C9  O2F             LD      B,2FH       ; digit= '0' -1
12CB  O               INC     B           ;digit value + 1
€12c  El              POP     HL          ;constant address in H1
12CD  E5              PUSH    HI          ;and back to the stack
12CE  CD 48 0D        CALL    0D48H       ;Number - constant. Unloud?
12D1  30F             JR      NC,12CBH    {no, continue
1203  Et              POP     HI          ;Constant address in Hl.
1204  CD 3 0D         CALL    0D36H       ;Number+ Constant
1207  EB              EX      EN, Hl      ;constant address in DE (n.Konst.l
1208  Et              POP     HI          ;Load buffer pointer
12D9  70              LD      (Hl.1,B     ;Add digit to buffer
12DA  23              INC     Hl          ;Buffer pointer+ 1
12D08 Fl              POP     AF          ;Load number counter
12DC  Cl              POP     BC          {parameter f,  '.' and ',' load
```

```
120D  3D          IIEC  A          ;Numbers -1. 1@ Digits ore?
12EN  20 E2       JR    NZ,        {no, continue
12E0  c5          PUSH  11C        ;parameter f. '.'and',' back up
12th  E5          PUSH  HL.        ;Save buffer pointer
12E2  21 1D 79    LD    HL,        ;Rest ( 1D) with single gene, in X
12ES  CD 1 09     CALL  09B1H
12E8  €18         JR    12F6H      farther with simple accuracy
12EA  c5          PUSH  11C        Parameter t. '.' and ','     on stack
12EB  IT          PUSH  HL.        ;Putter on stack
12EC  CD 08 87    CALL  0708H      ;Number+ 0.5 for rounding
12EF  3C          INC   A          ;Delete Flag
12F0  CD FB MA    CALL  0AF11H     ;Integer of number in Y
12F3  CD B4 09    CALL  09114H     ;Enter number in X
12F   El          POP   HL.        ;Load buffer pointer
12F7  Cl          POP   11C        ; Parameters f..,' and ','   load
12F8  AF          XOR   A          Delete Repeat flag
12F9  11 D2 13    LD    EN,        Address {Constants IE5 and 1E4
12FC  3F          CCF              Invert Repetition Flag
12FD  CD 91 12    CALL  1291H      i',' and',' set
1300  c5          PUSH  BC         {parameter f, '.'and',' on stack
1301  F5          PUSH  AF         ;Repeat flag on stack
1302  IT          PUSH  HL.        ;Buffer pointer to stack
1303  D5          PUSH  EN         ;Constant pointer to stack
1304  CD 11F fU9  CALL  09BFH      ;Transfer number to Y
1387  El          POP   ed         ;Constant pointer in HL.
1308  O2F         LD    B, 2FH     Digit code = "- 1
130A  O           INC   B          !Number code + 1
138B  711         LD    A,E        ;Number - constant. Unloud?
130C  96          SUB   (Hl.)      iLSB
1300  SF          LD    E,A
 130  23          INC   HL         ;next tremor
130F  7A          LD    A, D
1310  9E          SBC   A, (Hl.
1311  57          LD    DA
1312  23          INC   HL         3/SB
1313  79          LD    AC
1314  9E          SBC   A, (HL)
1315  4F          LD    C,A
1316  211         DEC   HL.        Constant hands - 2
1317  2B          DEC   ed         ;to 1. Byte of constant
1318  3e F0       JR    NC,130AH   ;no undernoise, back
131A  CD 117 07   CALL  07B7H      ;Number+ Constant
 131  23          INC   ed         ;Address next constant
```

```
131E  CD B4 09        CALL   1119B4H        ;Transfer number to X
1321  EB              EX     DE,HL          ;constant-aoresse in DE
1322  El              POP    HI.            Load {buffer pointer
1323  70              LD     (HL),B         Transfer iDigit to Buffer
1324  23              INC    HL             Buffer + 1
1325  Fl              POP    AF             Load Repetition Flag
132b  c1              POP    BC             Parameter f, '.' and ',' load
1327  38.03           JR     C.12FCH        Two runs? no-back
1329  13              INC    EN             ;Pass Next Constant
132A  13              INC    EN
132B  3E 04           LD     A,4            {still 4 digits in Integer-ttode
132D  18O             JR     1335H          Edit i

                              Jump to Integer

132F  D5              PUSH   EN             ;Format flag on stack
1330  11 D8 13        LD     EN, 13D8H      ;Address constants 11000 to l
1333  3E 05           LD     A,5            {Numbers = 5
1335  CD 91 12        CALL   1291H          ;',' and ','
1338  CS              PUSH   BC             i Paraeter f. ' • ' and ', on stack
1339  F5              PUSH   AF             i2iffernumerator on stack
133A  E5              PUSH   HI.            ;Buffer pointer to stack
13311 EB              EX     DE, Hl.        Constant Address in Hl
133C  4E              LD     C, HL)         ;Load constant
133D  23              INC    HI.
133E  46              LD     , (H)
133F  C5              POSH   BC             ;and Save to Stack
1348  23              INC    HI             ;Address next constant
1341  E3              EX     (SP),H.        constant address
                                            Replace Stack
1342  EB              EX     EN, Hl         ;constant address in DE
 133  2A 21.79        LD     H, (7921H)     ;Load number in HL
1346  0 2F            LD     B, 2FH         Digit code = '0' - 1
1348  4               INC    B              ;digit code+ 1
1349  7D              LD     AL             {Number - constant (LSB)
134A  93              SUB    E
134B  f               LD     s
134C  7C              LD     /H             i (ICE)
134D  9A              SBC    A+ D
134E  67              LD     H,
13F   30 F7           JR     NC, 1348H      Lower run? no-back
1351  19              ADD    HL, DE         ;Number+ Constant
1352  22 21 79        LD     (7921HL,HL     Save the number in X
1355  D1              POP    EN             Load Constants Address
1356  Et              POP    th             ;Load buffer pointer
```

```
1357  7111              LD      (H),B          ;Transfer tremor to putter
1358  23                INC     HL             ;Buffer pointer+ 1
1359  Fl                POP     AF             ;Load timer
135A  Cl                POP     J.\C           Parall'll!ter t. '.' and ',' load
1351\ 3D                DEC     A              ;Numbers - 1
135C  2111 D7           JR      NZ, 1335H      ;all digits? no-back
135E  CD 91 12          CALL    1291H          ; ' • ' and ' ,'
1361  77                LD      (HU,A          Mark end of line ait X'80
1362  D1                POP     EN             ;DE restore
1363  C9                RET
```

#Mi} # M # % tMt #t } ## Et i i ###i #%%±

Constants

```
1364  00 80 00 1110 F9 82 15 A2        i= 1111 x 10 E9 (double precision)
13bC  FD FF 9F 31 A9 5F 63 B2          3= 1x18 E15 (double precision)
1374  FE FF 03 BF C9 1B 0E B           i= 1x10 Elb (double, precision)
137C  Oil Oil Oil Oil Oil Oil 0111 8il
                                       i138il-1383 = 0.5 (single precision)
138'+ 0i1 00 04 BF C9 1B 0E B          i= 1 x 10 E16 (double, precision)
138C  0fil 80 C A 7E 8D 03             1= 1000000000000008080
1393  00 40 7A 10 F3 SA ilO            {= 1000000000000000
139A  00 A 72 'lE 18 89 00             ;=  10000000000808
13A1  00 10 A5 D4 EB 00 080            ;=  10800il0il0000
13A8  08 E8 7 48 17 00 00              }= 101'10i10i100000
13AF  00 E4 01\ 54 02 Oil 011          ;=      110M01J000
131\b 00CA9A3B00001110                 {=      108000011100
13BD  1110 El FS 05 00 00 0e           ;=      1110i10i1000
13C4  Bil9b98000011101110              {=    111100001i!0
13CB  40 42 8F 00 80 ilO 011           {=          1000080
1302  A0 86 01                         {=          1001180
1305  10 27 1110                       {=          18000
13D8  10 27                            {=          1011100
13DA  EB 03                            {=          101110
13DC  64                               {=            100
13EN  0A 1110                          {=             10
1YE0  01 00                            {=
```

t##Mi M # ME M t t #

```
                    Sub-programme, for SQR and ATN
                    causes 1 'multiplication with -1
13E2 21.82.09       LD                                  HL.0982H
13E5 E3             EX                                  (SP),HL
13E6 E9             JP                                  (HLI)

                    HIf*****************************************************************************************

                                                                                                    SQR - Funct

                    Is the root of a number
                    A9.'                                X= Argument
                    Exp.:                               X= Function value
13E7 CD A @9        CALL                                09A4H
13EA 218013         LD                                  HL, 1380H
13ED CD Bl 09       CALL                                091\1H
13F0 18 03          JR                                  13F5H

                    ***********************************************************************************************************

                    Determine the power of a number
                    Eing. Base on the stack
                                                        X= exponent
                    u9, '                               X= Result
1JF2 CD Bl 0A       CALL                                0AB1H
13F5 Cl             POP                                 BC
13F6 D1             POP                                 EN
13F7 CD 55 09       CALL                                0955H
JFA  78             LD                                  A, B
13FB 28 3C          JR                                  1.1439H
1JFD F2 4 14        JP                                  P, 1404
1400 B7             OR                                  A
1401 CA 9A 19       JP                                  Z, 199AH
1404 B7             OR                                  A
1405 CA 79 87       JP                                  1.0779H
1408 D5             PUSH                                EN
1409 CS             PUSH                                BC
140A 79             LD                                  A,C
140B F6 7F          OR                                  7FH
1400 CD BF 09       CALL                                09BFH
1410 F2 21 14       JP                                  P,1421H
1413 D5             PUSH                                EN
1414 c5             PUSH                                BC
1415 CD 40 08       CALL                                BH
1418 Cl             POP                                 BC
```

```
1419  Dl              POP   EN
141A  F5              PUSH  AF            L5J! INT(Exponent) on Stack
1411! CD 0C 0A        CALL  0A0CH         INT(Exponent) = Exponent?
141E  EI              POP   HL            ;LSI! INT(exponent) in HL
141F  7C              LD    + u           iINT(exponent) odd?
1420  1F              RRA
1421  El              POP   Hi.           Transfer Base to X
1422  22 23 79        LD    (7923H),HL    3MSB
1425  E1              POP   HL
1426  22 21 79        LD    47921H),HL    iLSB
1429  DC E2 13        CALL  c, 13E2H      {Result + (-1)
142C  CC 82 09        CALL  1.0982H       ;Base= Base
142F  D5              PUSH  EN            ;Exponent on stack
143   c5              PUSH  BC
1431  CD 09 08        CALL  0809H         iLOG(Base) to X
1434  Cl              POP   BC            ;Load exponent in Y
1435  Dl              POP   EN
1436  CD 47 88        CALL  0847H         LOG(J!asis) * Exponent
```

EXP - Function

Exponential function *of a* number besti111111en
Ring: X= Argument
  Exp.: X= Function value

```
1439  CD A 9          CALL   09A44H          {Argument to stack
143C  01 38 81        LD     BC,B138H        ;constant 1.4427 in Y
143F  11 3B AA        LD     EN,0AA3BH
1442  CD 47 08        CALL   8847H           {Argument / L(2) in X
1445  3A 24.79        LD     A, (7924#4)     ;2 Exponent of Erg. > 136?
1448  RD 88           CP     BBH
144   D2 31 09        JP     NC, 0931H       Yes! continue at 931}
144D  CD 40 0B        CALL   0B40H           ;INT(exponent in A and X
1450  C6 88           ADD    M, 80~          ;Add Offset
1452  C6 02           ADD    A,2             ;Exponent> 126?
1454  DA 31 09        JP     C, 0931H        Yes! Continue at 8931H
1457  F5              PUSH   AF              ;Exponent (a.Offset on stack)
1458  21 FB 87        LD     HL.07FBH        ;INT(Arg,/L06(211 - 1 to X)
145.B CD 0B 07        CALL   870.BH
145E  CD 41 88        CALL   8841H           i* LOG (2)
1'161 Fl              POP    AF              Exponent d. erg. back
1462  Cl              POP    BC              ;Load argument again
1463  D1              POP    EN
1464  F5              PUSH   AF              Exponent back on stack
1465  CD 13 87        CAI.i  8713H           {x = LO(2)+1NT(Arg/LN2)-1)-Arg
1468  CD 82 09        CALL   0982H
146B  217914          LD     HL,1479H        ;Calculate Series
146E  CD A9 14        CALL   14A9H
1471  11 00 08        LD     EN.0            {0.5 + 2 M exponent in Y
1474  Cl              POP    BC
1475  4A              LD     C,
1476  C3 47 08        JP     8847H           with row result aultipliz.
```

Constants for Exponents - Series

```
1479  08                                     ;8 constants
147A  40 2E 94 74                            {= -1.4131 E-0
147E  71 4F 2E 77                            [=  1. 32988 E-63
1482  6E 62 88 7A                            i= -B.31136 E-13
1486  E 2A 7C                                {=  0,0416574
148A  51 AA AA 7E                            = -0.165
148E  FF 7F 7F                               {=  0.5.
1492  00 00 80 81                            = -1
```

149 00 08 00 B81 {= 1

%#%3%% % t pi t%}% t% it t#}

Series Calculation 1

K1 +K2 + Zw3 +K3 + Z#5
Eing: X= Number (Zl
HL = address of constants {K)
(points to number-bytes)
Exp.: X= series result

149A	CD AA 09	CALL	09A4H	;X to Stack
149D	11 32 0C	LD	DE,0C32H	;Return address to stack
140	D5	PUSH	EN	(causes Multipl. with Z at the
14A1	IT	PUSH	HL	;Constant Address on Stack
14A2	CD BF 09	CALL	09BFH	;Z to Y
14AS	CD 47 088	CALL	0847H	Z2 to I
14AB	El	POP	HL	;constant address in HL

% Mi %lt % ± #tt } M # % # } % %} %#t}

Series Calculation 2

Cl + K2 * Z + K3 * Ztt2 + K4 * Z+tJ
Ring: as series calculation 1
Exp.: as series calculation 1

14A9	CD A4 9	CALL	09AAH	z to stack
14AC	7E	LD	A, (HL)	Number of constants in A
14AD	23	INC	HL	Address of 1. Constants
14AU	CD B1 09	CALL	09B1H	il, constant in X
14B1	06	DEFB	O	;LD 3.0F1 dummy
14B2	Fl	POP	AF	;Load constant counter
14B3	Cl	POP	BC	;Z or Ztt2 (row 2 o. 1) in Y
14B4	Dl	POP	EN	
1435	3D	DEC	A	constant counter -1
14B	C8	RET	or	Ready!
14B7	D5	PUSH	EN	Y back on stack
14BB	c5	PUSH	BC	
14B9	F5	PUSH	AF	;constant counter on stack
14BA	IT	PUSH	Hi.	;Constant Address on Stack
14BB	CD 47 08	CALL	0847H	31 I (o) 72)
14BE	El	POP	Hi.	;Load constant address
14BF	CD c2 9	CALL	09C2H	next constant in Y
14C2	IT	PUSH	HL	;Constant Address on Stack
14C3	C 16 87	CALL	071bH	;Add constant to X
14C	El	POP	HL	;Reload constant address

```
14C7  18 E9          JR     14P.2H          further

                     #I # k%k#%#k # # # # k # #k ##i # k # #%#
                     # k}

                     RND - Function

                     Generate random number
                     Eing: X= Argument (0 or end of interval)
                     Exp.: X= random number
14C9  CD 7F 0A       CALL   0A7FH           ;Argument in Integer U1111J1
14CC  7C             LD     ,H              30 7
14CD  B7             0R     A
14CE  FA 4A LE       JP     M,1E4AH         ;Yes, FUNCTION CODE - Error
14D1  B5             OR     L               iArgument = 0 1
14D2  CA F 14        JP     Z, 14f0H        Yes, random number or %and 1 ore,
14D5  E5             PUSH   HL              ;Argument to stack
1406  CD F0 14       CALL   14f0H           ;Real random number in X
 14%  CD BF 09       CALL   09BFH           ;in V
14DC  EB             EX     DE,HL           ;Reload argument and
14DD  E3             EX     (SPl,HL         ;random number on stack
14EN  5              PUSH   BC
14DF  CD CF 0A       CALL   0ACFH           ;Argument time united. Gen. in X
14E2  Cl             POP    BC              ;Random again in Y
14E3  Dl             POP    EN
14E4  CD 47 08       CALL   0847H           {Random ± Argunent
14E7  21 FB 07       LD     HL.07F8H        + 1
14EA  CD 0B 07       CALL   070BH
14ED  C3 40 0B       JP     0R4MM           Erg,= INT(number * Arg,      +1)

                        New random number= old random number* 4253261 + 372837

14F0  21 90 78       LD     HL, 7898H       ;Multiplier Address
14F3  E5             PUSH   HL              ;on stack
14f4  11.080.00      LD     EN.8            Result reg. delete (CDE>
14F7  4B             LD     C,E
14F8  26 03          LD     H3              ;Byte count = 3
14FA  2E 08          LD     L+8             bit counter = 8
14FC  EB             EX     DE,HL           Result Register + 2
14FD  29             ADD    HL, HL          iLSB
14FE  EB             EX     DE,HL
14FF  79             LD     A+ C            1NSB
15%8  17             RLA
151   4F             LD     CA
15082 EJ             EX     (SP),HL           ;Multiplier Address in Hl
1503  7E             LD     A, (H)          ;Byte of the multiplier in A
1504  e7             RLCA                      ;highest bit in carry
```

```
1505  77              LD      (HL),A          ;and save back
150   E3              EX      (SP),H.         ;Multiplier Address on stack
1507  D2 16 15        JP      NC, 1516        Bit not set, no addition
150A  IT              PUSH    HI.             {counter on stack
150B  2A AA 78        LD      HL, !78AAH1     Add Last Random Number
150E  19              ADD     H.,DE           1L.SB8
150F  EB              EX      DE,HL
1510  3A AC 78        LD      A, (78ACH)      il1SB
1513  89              ADC     +C
1514  4F              LD      ,A
1515  El              POP     HL              {Reload Counter
1516  2D              DEC     l               ;bit counter - 1
1517  C2 FC 14        JP      NZ.14FCH        Byte processed? no-back
151A  E3              EX      (SP),H          ;lt.II tiplikator address in HL
151B  23              INC     HI.             + l
151C  E3              EX      (SP),HL         ;and back to the stack
151D  25              DEC     H               Counter - 1
151E  C2 FA 14        JP      NZ, 14FAH       Ready? no-back
1521  El              POP     HI...           ;Correct stack
1522  21 65 Be        LD      H, 0R065H       ;Result+372837 = new random number
1525  19              ADD     HL, DE          ;LSB
1526  22 AA 78        LD      !78AAHJ, H1
1529  CD EF OA        CALL    0AEFH           ;Type= simple precision
152C  3E 85           LD      A,5             1NSB
152E  89              ADC     As C
152F  32 AC 78        LD      (78ACH),A
1532  EB              EX      DE,HI...        Transfer to Y
153J  0 80            LD      B.80H           Exp. Y = 6, thus 0 and 1 respectively
1535  21.25.79        LD      HL.7925H        ;Set sign flag
1538  70              LD      (HL),B          ;Result= positive
1539  2B              DEC     HI              Exponent I = Exponent Y
153A  70              LD      (HL),B
153B  4F              LD      C               il'ISB in C
153C  0 00            LD      3.0             Delete ilSB
153E  C3 65 07        JP      075H            ;for normalisation

FFF

COS - Function

Determine the cosinus of an angle
Eing.t I = Argument ia Radian
Exp.: X= Function value
1541  21 SB 15        LD      HL, 158BH       ;Address constant PI/2
1544  CDU 07          CALL    070BH           ;Pl/2 to add rgunent
```

SIN - Function

Determine the sine of an angle
Eing. I = Radical argument
Exp.: X= FunctionSllert

```
1547  CD A4 09        CALL   09A4H          Argument to Stack
154A  01 49 83        LD     R.8349H       ;Constant 2PI in Y
154D  11 DB F         LD     DE,0FDBH
1550  CD B 09         CALL   @9BH          Transfer i2PI to X
1553  C1              POP    BC            ; Argument in Y
1554  D1              POP    EN
1555  CD A2 08        CALL   08A2H         iX =Argument/ 2PI
1558  CD A 9          CALL   0944}         ;Argument /2PI to Stack
155B  CD 4 8B         CALL   0BH           iINT (Arg/2Pll in X
155E  C1              POP    C             Arg/2PI voa Stack in Y
155F  D1              POP    EN
1560  CD 13 f117      CALL   fl1713H       ;X = Arg/2PI - INT(Arg/2PI}
```

Interval (0 · 1) at interval (-0.25 · · · 0.25)

```
1563  218F 15         LD     HL, 158FH     ;Address constant 0.25
1566  CD 1107         CALL   0710H         0.25 - X in K
159   CD 55 9         CALL   1955H         ;X )= ?
15bC  37              SCF                  Flag f. Multipl. Delete with (-1)
156   F2 77 15        JP     P,1577H       Yes!
1570  CD 08 07        CALL   0708H         0.5 + X in X
1573  CD 55 09        CALL   0955H         ;X )= 0?
157   B7              OR     A             Flas f. Multipl. with (-1)
1577  F5              PUSH   AF            Flag on stack
1578  F4 82 09        CALL   P.0982H       Yes! X = -X
157B  218F 15         LD     HL.158FH      Address constant .25
157E  CD 0B 117       CALL   11711BH       0.25 + X in X
1581  F1              POP    AF            Load flag
1582  D4 82 119       CALL   NC, 0982M     Carry = 8? yes - X= -X
1585  219315          LD     HL.1593H      ;Series Constants
1588  C3 94 14        JP     149AH         Calculate Series
```

% Mi#Mt Mt#t Htti tWith

Constants

```
158B  DB  0F  49  81
158F  0fil 10 118 7F
```

i=
1,57118
5= 0.25

```
                    for sine series calculation
1593  115                           ;count= 5
1594  BA D7 1E 8                    {=  39,71117
1598  6 2 99 87                     = -76.575
159C  58 34 23 87                   {=  81st 6022
1548  Eli SD A5 86                  = -41.3417
15M   DA 8F 49 83                   {=  6,28319
```

HFFFFFFFFFFFFFFFFFFFFFFhhhfff

TAN - Function

Calculates the tangent of an angle
Ring: X= Argument in radians
From9.: K Function value

```
15AB  CD A4 09    CALL  0944}       ;Argument to stack
154   CD 47 15    CALL  1547H       ;Sin (Arg)
15AE  Cl          POP   BC          ;Argument in Y
15AF  El          POP   HL
15B9  CD A4 09    CALL  09A4H       ;Sin (Arg) on stack
15B3  EB          EX    DE,HL
15B4  CD B4 09    CALL  09BH        Transfer {Argument to I
15B7  CD 41 15    CALL  1541H       ;Cos !Arg)
3:BA  C3 A0 l!B   JP    08Al!H      ;Tan(Arg) = Sin(Arg.l / Cos(Arg,l)
```

%%#t M#1 } # t # ## tE #kE

ATN - Function

Arcus-Tangens calculation
Eing: X= Argument
etc. ! = Angle **i** radian

```
15BD  CD 55 09    CALL  0955H       Argument ({ ?
15C0  FC E2 13    CALL  M, 13E2H    Yes, result (-1)
15C3  FC 82 09    CALL  1'1.0982H   ;Abs(argument in X
15C6  3A 24.79    LD    A,(7924Hl   {Argument 1 ?
15C9  FE 81       CP    81H
15CB  38 l!C      JR    C, 15D9H    Yes!
15CD  01 08 81    LD    B, 8100H    No! Y = 1
15D0  51          LD    D,C
15D1  59          LD    E€
15D2  CD A2 88    CALL  0842H       ;X= 1 / Argument
15D5  21 10 07    LD    HL.0710H    Sprungadr. to 0711 on Stack
15D8  IT          PUSH  Hl.
15D9  21E315      LD    HL, 15E3H   ;Addressing Constants for Row
15DC  CD 9A 14    CALL  149AH       ;Calculate Series
```

```
15DF  21 SB 15          LD      HL.158BH        ;Load URL from P1/2
15E2  C9                RET                     continue at 71@H
```

%%#%fWed}#%%i%t#%&#%t%#}

Constants for the Arcus-Tangens series

```
15E3  09                                        Number= 9
15E4  4A D7 JB 711                              {=  2.86623 E-03
15E8  02 6E 84 7B                               ;= -0.0161657
15EC  FE C1 2F 7C                               ;=  0,04299
15F0  74 31 9A 7D                               i= −0.0752896
15F4  84 3D SA 7D                               {=  0. 106563
15FB  C8 7F 91 7E                               i= -0.142089
15FC  E4 BB 4C 7E                               ;=  0. 19993
16800 6 AA 7F                                   5= -0. 333331
1604  00 00 0 81                                {=  1
```

%~%%% %%%%~%#fit It ± %% %#%

Functions Bounce Table

!Tokens D7 to FA)

```
1608  8A 09             DEFW    098AH           iD7 = SGN
160A  37B               DEFW    0B37H           iDB = INT
160C  77 09             DEFW    0977H           1D9 = ABS
 10E  D4 27             DEFW    27D4H           {DA = FRE
1610  EF 2A             DEFW    2AEFH           ;DB = INP
1612  FS 27             DEFW    27F5H           iDC = POS
 114  E7 13             DEFW    13E7H           iDD = SQR
1616  C9 14             DEFW    14C9H           iDE = RHO
1618  09 08             DEFW    0809H           iDF = LOG
161A  39.14             DEFW    1439H           4E = EIP
161C  41 15             DEFW    1541H           iE1 = COS
161E  47 15             DEFW    1547H           iE2 = SIN
1620  A8 15             DEFW    15ABH           1E3 = TAN
1622  BD 15             DEFW    15BDH            E4 = ATN
1624   AA2C             DEFW    2CAAH           ES = PEEK
1626  52.79             DEFW    7952H           iE6 = CVI
1628  58.79             DEFW    7958H           iE7 = CV5
162A  5E 79             DEFW    795EH             4E = CWD
162C  61.79             DEFW    7961H           iE9 = EOF
162E  4.79              DEFW    7964H            EA = LOC
1638   7.79             DEFW    7967H            EB = LOF
lb32  6A 79             DEFW    79bAH           ; EC = I'IKI$
```

```
1634  6D 79            DEFW  79DH            ED = FMD$
163   70.79            DEFW  7970H           EE = HKD$
1638  7F 0A            DEFW  0A7FH           iEF = CINT
163A  .B1 0A           DEFW  0A.B1H          F0 = CSN6
163C  DB 0A            DEFW  0ADBH           File = CD.BL
163E  2 WB             DEFW  8B21            F2 = FIX
1640  03 2A            DEFW  2A0JH           F3 = LEN
1642  36.28            DEFW  2836H           1F4 = STR$
1644  C5 2A            DEFW  2AC5H           F5 = VAL
14    filf 2A          DEFW  2AfilfH         F6 = ASC
1648  1F 2A            DEFW  2A1FH           3F7 = CHR$
164A  61 2A            DEFW  2Ab1H           F8 = LEFT$
164C  91 2A            DEFW  2A91H           iF9 = RI6HT$
164E  9A 2A            DEFW  2A9AH           {FA = MIDS

                       %%%%%% %~1% E # i t t#Et i # Hi}

                       BASIC Keywords Table

                       (ascending sorted by token)

150   CS               DEFB  8fllH+'E'        8111 = END
1651  4E 44            DEFM  'ND'
1653  C                DEFB  SIIIH+'F'       81 = FOR
1654  4F 52            DEFM  'OR'
1656  D2               DEFB  80H+'R'         ; 82 = RESET
1657  45 53 45 54      DEFl'I  'ESET'
165B  D3               DEFB  SIIIH+'S'       83 = SET
165C  45.54            DEFl'I  'ET'
165E  CJ               DEFB  80H+'C'         84 = CLS
165F  4.53             DEFN  'LS'
1661  81               DEFB  81H             85 = CND      (power encoded)
12    00 8             DEFB  0.0
1664  81               DEFB  81H               ;86 = RANDOf1 (not encoded)
1665  08 08 00 08 00   DEFB  0.0.0.0.0
166   CE               DEFB  SIIIH+'N'       87 = NEXT
166.B 45.58.54         DEFl1  'EXT'
166E  C4               DEFB  8111H+'D'       i88 = DATA
166f  41.54.41         DEFM  'ATA'
1672  C9               DEFB  BilH+'I'        189 = INPUT
1673  4 50 55 54       DEFl1  'NPUT'
1677  C4               DEFB  SIIIH+'D'       SA = D111
1678  49 4D            DEFM  '111'
167A  D2               DEFB  SIIIH+'R'       48B = READ
167B  45.41.44         DEFM  'EEAS'
```

```
167E  CC               DEFB  B0H+'L'        €18 = LET
167F  45.54            DEFl1 'ET'
1681  C7               DEFB  B0H+'G'        i8D = GOTO
1682  4F 54 4F         DEFl1 'OTO'
1685  D2               DEFB  80H+'          ;SE= RUN
1686  55 4E           DEFl1 'UN'
1688  C9               DEFB  80H+'          8F = IF
1689  46               DEFM  'F'
168A  D2               DEFB  8MH+'          ;90 = RESTORE
168B  45 53 54 4F      DEFM  'ESTORE'
      52.45
1691  C7               DEFB  B0H+'G'        i91 = GOSUB
1692  4F 53 55 42      DEFM  'OSUB'
189   D2               DEFB  80H+'R'        ;92 = RETURN
1697  45 54 55 52 4E   DEFl1 'ETURN'
169C  D2               DEFB  B0H+'R'        {93 = RE
169D  45 4D            DEFl1 'El'I'
169F  D3               DEFB  B0H+'S'        i94 = STOP
 16   54 4F 50         DEFl'I  'TOP'
16A3  CS               DEFB  80H+'E'        ;95 = ELSE
16A4  €4.53.45         DEFl'1  'LSE'
16A7  C3               DEFB  80H+'C'        i96 = COPY
16A8  4F 58 59         DEFl1 'OPY'
16AB  CJ               DEFB  80H+'C'        97 = COLOUR
16AC  4F 4C 4F 52      DEFll 'OLOR'
16B0  D6               DEFB  80H+'V'        498 = VERIFY
16Bl  45 52 49 46 59   DEFl'l 'ERIFY'
16B   81               DEFB  81H            199 = DEFINT    (not encoded)
16B7  00 00 00 00 00   DEFB  0.0.0.0.0
16BC  81               DEFB  81H            i9A = DEFSN6    (not encoded)
16BD  00 00 00 00 00   DEFB  0.0.0.0.0
16c2  81               DEFB  81H            19 = DEFDBL     (not encoded)
16C3  00 00 00 00 80   DEFB  8.0.0.0.0
16C8  CJ               DEFB  80H+'C'        €19 = CRUN
16C9  52 55 4E         DEFN  'RUN'
16CC  CD               DEFB  80H+'N'        9D = l'IODE
16CD  4F 44 45         DEFN  'ODE'
16D0  D3               DEFB  8MM~+'S'       39E = SOUND
16D1  4F 55 4E 44      DEFN  'OUND'
16D05 81               DEFB  81H            ; 9F = RESUI'E  (not encoded)
 1    00 00 08 00 00   DEFB  0.0.0.0.0
16DB  Cf               DEFB  80H+'0'        1A = OUT
16DC  55.54            DEFN  'UT'
16EN  81               DEFB  81H            A1 = ON          (not coded)
```

```
16DF  00              DEFB  O
1Eo   81              DEFB  81H          iA2 = OPEN      (not encoded)
16E1  00 00 00        DEFB  8.8.8
16E4  81              DEFB  81H          iA3 = FIELD     (not encoded)
1bE5  00 08 00 00     DEFB  6.0.0.0
16E9  81              DEFB  81H          ;A4 = GET       (not encoded)
1bEA  00 00           DEFB  0.0
1bEC  81              DEFB  81H          ;AS= PUT        (not encoded)
1bED  00 80           DEFB  0.0
1bEF  81              DEFB  81H          {A = CLOSE      (not encoded)
16F   00 00 00 O      DEFB  0.0.0.0
16F4  81              DEFB  81H          ;A7 = LOAD      (not encoded)
16F5  080 @0 00       DEFB  0.0
16F8  81              DEFB  81H          iAB = 11ERGE    (not encoded)
16F9  00 00 00 00     DEFB  0.0.0.0
16FD  81              DEFB  81H          iA9 = NA11E     (not encoded)
16FE  00 08 00        DEFB  0.0.0
1701  81              DEFB  81H          iAA = KILL      (not encoded)
1702  00 08 00        DEFB  0.0.0
1705  81              DEFB  81H          iAB = LSET      (not encoded)
170   00 08 08        DEFB  0.0.0
1709  81              DEFB  81H          ;AC= RSET       (not encoded)
170A  00 00 00        DEFB  0.0.0
170D  81              DEFB  81H          iAD = SAVE      (not encoded)
170E  00e Oe          DEFB  0.0.0
1711  81              DEFB  81H          AE = SYSTEM     (not encoded)
1712  00 08 00 00 08  DEFB  0.0.0.0.0
1717  CC              DEFB  80H+'L'      iAF = LPRINT
1718  50 52 49 4E 54  DEFl'I  'PRINT'
171D  81              DEFB  81H          B0 = DEF        (not encoded)
171E  00 00           DEFB  0.0
1720  D0              DEFB  80H+'P'      iB1 = POKE
1721  4F 4B 45        DEFl'I  'OKE'
1724  D0              DEFB  80H+'P'      B2 = PRINT
1725  52 49 4E 54     DEFl'I  'RINT'
1729  C3              DEFB  80H+'C'      BJ = CONT
172A  4F 4E 54        DEFI'I  'ONT'
1720  CC              DEFB  80H+'L'      B4 = LIST
172E  49 53 54        DEFl'I  'IST'
1731  CC              DEFB  80H+'L'      BS = LUST
1732  4C 49.53.54     DEFl'I  'LIST'
1736  81              DEFB  81H          Bo = DELETE     (not encoded)
1737  00 00 00 O 00   DEFB  8.0.0.0.0
173C  81              DEFB  81H          iB7 = CAR       (not encoded)
```

```
173D  00 00 00          DEFB   0.0.20
1740  C3                DEFB   80H+'C'          BB = CLEAR
1741  4C 45 41 52       DEFM   'LEAR'
1745  C3                DEFB   80H+'C'          B9 = CLOAD
1746  4C 4F 41 44       DEFM   'LOAD'
174A  C3                DEFB   80H+'C'          ;BA= CSAVE
174B  53.41.56.45       DEFM   'SAVE'
174F  CE                DEFB   80H+'N'          {BB = NE]
1750  45.57             DEFM   'EW'
1752  D4                DEFB   80H+'T'          ;BC = TAB(
1753  41 42 28          DEFM   'AR'
1756  D4                DEFB   80H+'T'          Ibid = Tue
1757  4F                DEFM   '(I)'
1758  81                DEFB   81H              ;:BE= FN        (not encoded)
1759  00                DEFB   8
175A  D5                DEFB   80H+'U'          i:BF = USING
175B  53 49 4E 47       DEFH   'SING'
175F  81                DEFB   81H              ;C0 = VARPTR    (not encoded)
170   00 00 00 00 00    DEFB   0.08.0.0.0
1765  D5                DEFB   80H+'U'          ;Cl= USR
1766  53 52            DEFH   'SR'
17b8  81                DEFB   81H              1C2 = ERL       (not encoded)
1769  00 00             DEFB   0.0
17bB  81                DEFB   81H              ;C3 = ERR       (not encoded)
€176  00 00             DEFB   0.0
17bE  81                DEFB   81H                 ;C4: STRING$ (not encoded)
17bF  00 00 00 00       DEFB   0,0,00,0
      00 00
1775  81                DEFB   81H              1C5 = INSTR     (not encoded)
1776  00 00 00 00       DEFB   8.0.0.0
177A  D0                DEFB   80H+'P'          C6 = POINT
177B  4F 49 4E 54       DEFM   'OINT'
177F  81                DEFB   81H              iC7 = TIIU        not coded)
1780  00 00 80 00       DEFB   0.08.0.0
1784  81                DEFB   81H              1C8 = NEM      (power encoded)
1785  00 O              DEFB   0.0
1787  C9                DEFB   80H+'I'          19 = INKEY$
1788  4E 4B 45 59 24    DEFH   'NKEYf'
178D  4                 DEFB   80H+'T'          CA = THEN
178E  48 45 4E          DEFM   'HEN'
1791  CE                DEFB   80H+'N'          CB = NOT
1792  4F 54             DEFM   'OT'
1794  D3                DEFB   80H+'S'          : CC = STEP
1795  54.45.50          DEF11  'TEP'
```

```
1798 AB              DEFB  80H+'+'        CD = +
1799 AD              DEFB  80H+'-'        iCE = -
179A A               DEFB  BM}+ +         CF = *
179B AF              DEFB  B0H+'/'        iD0 = /
179C EN              DEFB  80H+5EH            {DI = Up arrow {potentiate)
179D cI              DEFB  80H+'A'        iD2 = AND
179E 4E 44           DEFM  'ND'
17A0 CF              DEFB  80H+'O'        1D3 = 0R
17A1 52              ENfl1 'R'
17A2 BE              DEFB  8MH+)>         iD4 = >
17AJ BD              DEFB  BMH+'          D5 = =
17A4 C               DEFB  80H+"'"        {D =
17A5 DJ              DEFB  80H+'S'        D7 = 5SGN
17A6 47 4E           DEFM  'GN'
17A8 C9              DEFB  8MH+'I'        ;DB= INT
17A9 4E 54           DEFM  'NT'
17AB Cl              DEFB  80H+'A'        ;D9 = ABS
17AC 42.53           DEFM  'BS'
17AU 81              DEFB  81H            iDA = FRE       (not encoded)
17AF 00 00           DEFB  0.0
17B1 C9              DEFB  80H+'          ;DB= INP
17B2 4E 50           DEFM  'NP'
17B4 81              DEFB  81H            iDC = POS       (not encoded)
17B5 00 00           DEFB  0.0
17B7 DJ              DEFB  80H+'S'        iDD = SQR
17B8 51.52           DEFM  'QR'
17BA D2              DEFB  80H+'R'        iDE = RND
17BB 4E 44           DEFM  'ND'
17BD CC              DEFB  80H+'L'        $DF = L0G
17BE 4f 47           DEFM  '06'
17C0 C5              DEFB  80H+'E'        JE = EXP
17C1 58.50           DEFM  'XP'
17C3 CJ              DEFB  80H+'C'        Egg = COS
17C4 4F 53           DEFM  'OS'
17C6 DJ              DEFB  80H+'S'        iE2 = SIN
17C7 49 4E           DEFM  'IN'
17C9 D4              DEFB  8MM+'T'        4E3 = TAN
17CA 41 4E           DEFM  'AN'
17CC Cl              DEFB  80H+'A'        ;E4 = ATN
17CD 54 4E           DEFM  'TN'
17CF D0              DEFB  80H+'P'        E5 = PEEK
17D00 45.45.43       DEFM  'EEK'
1703 81              DEFB  81H           ;E6 = CVI      (not encoded)
17D4 00 00           DEFB  0.0
```

```
17D6  81              DEFB  BlH          E7 = CVS      (not encoded)
17D7  00 00           DEFB  0.0
17D9  1               DEFB  81H          ;ES= CVD      (Makes encoded)
17DA  00 O            DEFB  0.0
17DC  81              DEFB  81H          iE9 = EOF     (n1cht encoded)
17DD  00 %0           DEFB  0.0
17DF  81              DEFB  81H          iEA = LOC     (not encoded)
17E0  00 00           DEFB  0.0
17E2  81              DEFB  BlH          ;EB = LOF     (not encoded)
17E3  00 00           DEFB  0.0
17E5  81              DEFB  81H          ;EC = MKIS    (not encoded)
17E6  00 00 00        DEFB  0.0.0
17E9  81              DEFB  81H          ;ED= MKS$     (not encoded)
17EA  00 00 00        DEFB  0.0.0
17ED  81              DEFB  81H          EE = MKDS     (not encoded)
17EE  00 00 O         DEFB  0.0.0
17F1  81              DEFB  81H          ;EF = CINT    (not encoded)
17F2  00 00 00        DEFll 0.0.0
17F5  81              DEFB  BlH          FO = CSNG     (not encoded)
17Fb  O00e8           DEFB  0.0.0
17F9  81              DEFB  81H          ;F1 = CDBL    (not encoded)
17FA  00 00 00        DEFB  0.0.0
17FD  81              DEFB  81H          ;F2 = FIX     (not encoded)
17FE  00 00           DEFB  0.0
1800  CC              DEFB  80H+'L'      iF3 = LEN
1801  45 4E           DEFH   'EN'
1803  D3              DEFB  81-i+'S'       F4 = STR$
180Jt 54.52.24        DEFH   'TR$'
1807  D6              DEFB  80H+'V'      F5 = VAL
1808  41 4C           DEFH  'AL'
1884  C1              DEFB  80H+'A'      F6 = ASC
180.B 53.43           DEFH  'SC'
180D  C3              DEFB  80H+'C'       F7 = CHR$S
180E  48 52 24        DEFH  'HR$'
1811  CC              DEFB  80H+'L'      {F8 = LEFT$
1812  45 46 54 24     DEFN  'EFT$'
1816  D2              IIEFB 80H+'R'      iF9 = RI6HT$
1817  49 47 48 54 24  DEFN  'I6HT$'
181C  CD              DEFB  80H+'H'      ; FA = MID$
1B10  49 44 24        DEFN  '1Ds°
1820  A7              IIEFB BH+27H       iFB = '
1821  80              DEFB  80H          End of table
```

Jump table for commands lToken 80 - BB)

1822	Af. 1D	DEFW	1DAEH	180 = END
1824	A1 1C	DEFW	1CA1H	81 = FOR
1826	38.01	DEFW	0138H	; 82 = RESET
1828	35.01	DEFW	0135H	i83 = SET
182A	C9 01	DEFW	@1C9	;84 = LS
182C	73.79	DEFW	7973H	i85 = Cl'ID
182E	D3 01	DEFW	01D3H	{ = RANDON
1838	B6 22	DEFW	22B6H	187 = NEXT
1832	05 IF	DEFW	1F05H	88 = DATA
1834	9A4 21	DEFW	219AH	89 = INPUT
1836	8 2	DEFW	2608H	BA = Dil'I
1838	EF 21	DEFW	21EFH	;SB= READ
183A	21 IF	DEFW	1F21H	i8C = LET
183C	C2 1E	DEFW	1EC2H	18 = 6GOTO
18JE	A3 IE	DEFW	IEAJH	BE = RUN
1840	39.20	DEFW	2039H	SF = lF
1842	91 1D	DEFW	1091H	i90 = RESTORE
1844	BlE	DEFW	lEBlH	i91 = GOSUB
1846	EN lE	DEFW	lEDEH	i92 = RETURN
1848	07 1F	DEFW	1F07H	;93 = REN
184A	A9 1D	DEFW	1DA9H	i94 = STOP
184C	07 lF	DEFW	1F07H	395 = ELSE
184E	12.39	DEFW	3912H	9 = CÖPY
1850	9D 38	DEFW	389DH	97 = COLOUR
1852	38.37	DEFW	3738H	i9B = VERIFY
1854	03 lE	DEFW	1E03H	i99 = DEFINT
1856	o1E	DEFW	1EOH	;9A = DEFSNG
1858	09 1E	DEFW	1E09H	i9B = DEFDBL
185A	2E 37	DEFW	372EH	€9 = CRN
1B5C	63 2E	DEFW	2E63H	39D = FASHION
185E	FS 2B	DEFW	2BFSH	i9E = SOUND
180	AF LF	DEFW	iFAFH	;9F = RESUDE
1862	FB 2A	DEFW	2AFBH	1A4 = OUT
1864	6C 1F	DEFW	lFbCH	{A1 = ON
16	79.79	DEFW	7979H	iA2 = OPEN
1868	7C 79	DEFW	797CH	iA3 = FIELD
1864	7F 79	DEFW	797FH	AA = GET
186C	82.79	DEFW	7982H	A5 = PUT
186E	85.79	DEFW	7985H	TA = CL0SE
1B70	88.79	DEFW	7988H	iA7 = LOAD

```
1872  8B 79          DEFW   798BH          {A8 = MERGE
1874  SE 79          DEFW   798EH          ;A? = NAME
1876  91 79          DEFW   7991H          {AA = KILL
1878  97 79          DEFW   7997H          ;AB= LSET
187A  9A 79          DEFW   799AH          {A = RSET
187C  A0 79          DEFW   7940H          {AD = SAVE
187E  00 00           DEF   e              AE = SYSTEM
1880  67.20          DEFW   2067H          AF = LPRINT
1882  5B 79          DEFW   795BH          ;BO = DEF
1884  Bl 2C          DEFW   2Cil1H         4RI = PONE
1886  6F 20          DEFW   206FH          ;B2 = PRINT
1888  E4 1D          DEFW   1DE4H          PJ = CONT
188A  2E 2B          DEFW   2B2EH          {B = LIST
188C  29 2B          DEFW   2B29H          1B5 = LIST
188E  C6 2B          DEFW   2RCH           {B = DELETE
1890  8 20           DEFW   2008H          B7 = CAR
1892  7A 1E          DEFW   1E7AH          iB8 = CLEAR
1894  56 36          DEFW   3656H          ;B9 = CLOAD
1896  A9 34          DEFW   34A9H          ;BA = CSAVE
1898  49 1B          DEFW   1B49H          iBB = NEW
```

i ## # # # # ## # k # # k } i ## kt k ki # #

Operator priority codes

The 1it operator with the higher code has priority

```
189A  79             DEFB   79H           i +
189B  79             DEFB   79H           i -
189C  7C             DEFB   7CH           . *
189D  7C             DEFB   7CH           j/
189E  7F             DEFB   7FH           i ** (potentiate)
189F  50             OEFB   50H           i AND
1BA0  46             DEFB   46H           i OR
```

 # tl # t% With Mt %} t l ± %} %

Type Adjustment Jump Table

```
18Al  DB 0A          DEFW   04DH          illlrMilanage in double accuracy
18A3  00 00          DEFW   8             ;unused
18A5  7F 0A          DEFW   0A7FH         {wooded in integer
18A7  F4             DEFW   AF4H          Test Type on String
                                          ;TYPE HISMATCH - Error if not!
18A9  BI 0A          DEFW   0AB1H         simple precision conversion
```

#%% %%%% %% #%3} 4±% $%MM%%#i Mt ~

Basic Data Type and Comparison Jump Table

Double Precision

```
18AB  77 0C     DEFW   0C77H          iAddition
18AD  70 0c     DEFW   0C70H          ;subtraction
18AF  A1 0D     DEFW   0DA1H          il'll replication
18.Bl ES 0D     DEFW   0DESH          ;Division
18/83 78 84     DEFW   0A78H          iPotentise
```

Simple accuracy

```
18B5  16 07     DEFW   071H           ;Addition
18/87 13 07     DEFW   0713H          ;subtraction
18B9  47 08     DEFW   0847H          il'lultiplication
18.BB A2 08     DEFW   08A2H          ;Division
18.BD €0A4      DEFW   040CH          ;potentiation
```

Integer

```
18.BF D2 8B     DEFW   0BD2H          ;Addition
18C1  C7 0.B    DEFW   0.BC7H         ; subtraction
18C3  F2 0B     DEFW   0.BF2H         iMul replication
18CS  90 24     DEFW   2490H          ;Division
18C7  398A      DEFW   0439H          ;potentiation
```

Hit Mt it et t t} % %t MM~

Error abbreviations

ascending sorted by error codes
(who does not use the 1m LASER 110-31)

```
18C9  E 4       DEFN   'NF'           ;NEXT WITHOUT FOR
18C.B 53 4E     DEFN   'SN'           ;SYNTAX ERROR
18CD  52.47     DEFl't 'RG'           ;RETURN WITHOUT GOSU.B
18CF  4F 44     DEFN   '0D'           ;OUT OF DATA
18Dl  4 43      DEFI't 'FC'           ;ILLEGAL FUNCTION CALL
18D3  4F 56     DEFl't 'OV'           ;OVERFLOW
18D05 4F 4D     DEFN   'Ol't          i OUT OF l'tEl'IORY
18D7  €55.4     DEFl'I 'UL'           ilJNDEF INED LINE
189   42.53     DEFtl  'BS'           iSUBSCRIPT OUT OF RANGE
1BD.B 44.44     DEFl'I 'DD'           iREDil'IENSIONED ARRAY
1800  2F 30     DEFI'l '/0'           ;DIVISION BY ZERO
1BDF  49 44     DEFl'I 'ID'           ILLEGAi.. DIRECT OPERATION
1BE1  54 4D     DEFtl  'Tl'I'         iTYPE puriNATCH
1BE3  4F 53     DEFtl  'OS'           ;OUT OF STRING SPACE
1BE5  €4.53     DEFN   'LS'           ;STRING TOO LONG
18E7  53.54     DEFtl  'ST'           iSTRING FORMUL.A TOO
                                      COl'tPLEX
```

```
1BE9  43 4E     DEFM   'CN'          ;CAN'T CONTINUE
1BEB  4E 52     DEFM   'NR'          ;NO RESUl'IE
1BEO  52.57     OEFM   'RW'          ;RE5\JME WITHOUT ERROR
1BEF  55.45     OEFH   'UE'          ;UNPRINTABLE ERROR
18F1  4D 4F     OEFM   '11(1)'       ;MISSING OPERAND
18F3  44        DEFM   'FO'          ;BAD FILE DATA
18F5  4C 33     OEFM   'L3'          ;O I SK BAS I C COMNAND
```

% i% b &} i%%%%%% % % %i Wed k t }% #ti

data and subprogrammes **in the** BASIC

Initialisation to RAM.

Subprogramme for Division

```
18F7  1)6 00    SUB    0             ;Subtraction 22 - 21
18F9  6F        LD     L,A           is dified before every call 1110
18FA  7C        LD     A,H
18FB  EN00      BC     A,0
18FD  67        LD     H,A
18FE  78        LD     A,B
18FF  EN 00     SBC    A,0
1901  47        LD     B,A
1902  3E 00     LD     A,
19%4  C9        RET
```

System Data

```
1905  4A LE     DEFW   1E4AH         ;USR Startup Address
                                     ;preloaded with FUNCTION CODE Err.
1907  D 4                            ;RND multiplier
```

hter programme for]NP

```
190A  DB 08     IN     A, ()         ;Load Input Port in A
190C  C9        RET
```

OUT Subprogramme

```
190D  D3 08     OUT    (0),A         ;Output A-Reg via port
190F  C9        RET
```

System Data

```
1918  1111      DEFB   8             {IMEY$ Cache
1911  080       DEFB   8             Last error code for ERR
1912  00        DEFB   0             ;Printhead Position
1913  00        DEFB   0             Output F lag
1914  48        DEFB   64            ;Line Length on Ship'
```

```
1915 30          DEF.8  48          Last tab position on screen
1916 00          DEF.8  0           ;unused
1917 4C 7B       DEFW   7B4CH       Begin of String Pane
1919 FE FF       DEFW   0FFFEH      Current Line Numbers
191B E9 7A       DEFW   7AE9H       {programme text beginning
```

4%1~%%%~ %% %% Wed t #k & # #It

Texts

```
191D 20.45.52.52 DEFN   'ERROR'
     4F 52 0o
1924 20 49 4E 20 DEFN   'IN'
     00
1929 52 45 41 44 DEFM   'READY'
     59 0D 00
1938 42 52 45 41 DEFN   'BREAK'
     4B 80
```

%%4%%%%%%%%%% It% tiF # ME

Subprogramme for FOR/NEXT and GOSUB/RETURN

Retrieves data from stack

```
193  21 04 00    LD     HL.4        ;Stackpointer + 4 in Hl.
1939 39          ADD    HL,SP       j(2) Reverse Rear Rear.
193A 7E          LD     To (H)      ;Load flag
193B 23          INC    HL
193C FE 81       CP     81H         ;Data from FOR loop?
193E C8          RET    HZ          No, done
19JF 4E          LD     C, (HLJ     yes Load Running Variables
1940 23          INC    Hl
1941 6           LD     B, {H.)
1942 23          INC    HL
1943 IT          PUSH   Hl          Stack Address Pointer
1944 9           LD     L,C         Volume Variables Address in Hl
1945 6           LD     H,B
 194 7A          LD     A, D        Run variable specified?
1947 3           OR     E
1948 EB          EX     DE,HL       ;no, 1it address in DE back
1949 28 02       JR     Z.194DH
194.B E.B      . EX     DE,HL
194C DF          RST    lBH         yes, = sound variable found?
194D 81 GE 9     LD     .BC,14      i14 in BC
1958 El          POP    HL          Reload Location Pointers
1951 approx      RET    or          Yeah, done
```

```
1952  09              ADD    HL,BC          Pointer to next stack data
1953  18 ES           JR     193AH          ;same thing again

                      #%#####k### # kt i # ki#i}i ##i ########% %

                      Space for the programme line to insert
                      or clear variable
                      Ring: DE= Source block start address
                             BC = final address of the source block
                             HL =Destination
1955  CD 6C 19        CALL   196CH          is H still in the open air, he?
                                            ;no, OUT OF NENORY - Error
1958  CS              PUSH   BC             Exchange iHL and BC
1959  E3              EX     (SP),H
195A  Cl              POP    BC
1958  DF              RST    18H            ;Reached the beginning of the
195C  7E              LD     A,<HLl         ! Convert Byte
195D  02              LD     (BC),A
195E  annrnx          RET    0r             Yeah, done!
195F  0R              DEC    BC             Address pointer - 1
1960  2B              DEC    HL
1961  18 FB           JR     195BH          next byte

                      FFF

                      Test whether 2•C bytes are free
                      If not, OUT OF NEl10RY Error
1963  E5              PUSH   HL             ill. on stack
 194  2A FD 78        LD     HL, (78FDHl    initial adr, of the free feeder
1967  06 0            LD     B,             ;B=0
1969  09              ADD    HL,BC          ;C 2 Add eel to hl
196A  99              ADD    H,BC
196B  YES             DEFB   0E5H           LD A, OE5H  Dunny instruction
196C  E5              PUSH   HI             ;test whether HL is still in free
196D  JE Ci           LD     A, 0           ;H. } FFC} ?
196F  95              SUB    L
1978  F               LD     L,A
1971  JE FF           LD     A,
1973  9C              SBC    A,H
1974  38.04           JR     C,             Yes, OUT OF IIEl10RV Error
1976  67              LD     H,A            iHL + 4A >= SP?
1977  39              ADD    HL,SP
1978  El              POP    HL             ill. restore
1979  Profit          RET    C              ;no, back
```

****

Processing and output of error messages OUT

OF MEl'IORY - Error

```
LD      E,0CH           ;Error Code in E
JR      19A2H           ;to l'message output
LD      Hl,(78A2Hl      Load-Line Number
LD      A,H             in direct death? <=FFFFl
AND     L
INC     A
JR      Z.198EH         ;no, jump in END
LD      A, (78F2H)      {Trap-Flag set ?
OR      A
LD      E, 22H          ;NO RESUNE - Load Error Code
JR      NZ,19A2H        Ijas on the l'lelight
JP      DlCH            {leap in END
```

Implicit End

SYNTAX ERROR in DATA row

```
LD      Hl, (78DAHl     ;last DATA line
LD      (78A2Hl,HL)     ;as current line number
```

SYNTAX ERROR

```
LD      E.02            ; Error code in E
DEFB 01                 LD BC,141EH Dung-Betehl
```

DIVISION BY ZERO

```
LD      E,1-'IH         ;Error Code in E
DEFB 01                 LD BC, BOIEH Dug Command
```

NEXT WITHOUT FOR

```
LD E,                   ;Error Code in E
DEFB 01                 LD BC,241EH Duy
                        Command
```

RESUNE WITHOUT ERROR

```
LD      E,24#
                        ;Error Code in E
```

```
197A  1E 0C
197C  18 24
197E  2A A2 78
1981  7C
1982  A5
1983  JC
1984  28 08
1986  YES F2
1989  B7
198A  1E 22
198C  20.14
198E  CJ C1 lD
1991  2A DA 78
1994  22 A2 78
1997  1E 02
1999  1
1994  1E 14
€199  01
199D  1E 00
199F  01
19A40 1E 24
```

Output Error Message

Eing.: Error Code in E

```
19A2  2A A2 78          lD      Hl, (78A2Hl     ;Load current time lennllllM!r
19A5 22 EA 78           LD      (78EAH),HL      ;Save as Error Line
```

```
19A8  22 EC 78        LD     (78H),HL
19AB  01 B4 19        LD     BC, 19RH        ;Load Resume Address
19AE  2A E8 78        LD     HL, (78E8Hl     ;Load stack start address
19Bl  C3 9A lp        JP     1B9AH           Jump in NEW, Stack initialis.
 19B  Cl              POP    BC              Correct stack
19B5  7B              LD     A, E            ;A and C error code
 19B  4B              LD     ,E
19B7  32 9A 78        LD     7894H),A        Save ;es
19BA  2A E6 78        LD     HL, 78E6#)      Load {Programme pointer
19BD  22 EE 78        LD     (78EEH) , HL    Save as Error
19C0  EB              EX     DE,HL           and in
19C1  2A EA 78        LD     HL.,78EAH)      ;Line number= FFFF?
19C4  7C              LD     A,H             i (= Direct Model
19CS  A5              AND    L
 19C  JC              INC    A
19C7  28 07           JR     Z. 19D0H        ija; no break pair am.
19C9  22 F5 78        LD     (78F5Hl,HL      ;Save Error ZNr for CONT
19CC  EB              EX     DE,HL           ;Line Address in Hl
19CD  22 F7 78        LD     (78F7Hl,Hl      5 Save as CONT Pointer
19D00 2A4 F0 78       LD     HL, (78FMH,     ;Load URL of an error routine
1903  7C              LD     A, H            5= 0?
19D4  B5              OR     L
19D5  EB              EX     EN, HL          EN
 19D  21 F2 78        LD     H, 78F 2M       ;TRAP flag Load Address
19D9  28 08           JR     Z, 19EJH        no error routine (TRAP)
190B  A6              AND    (Hll            {open error TRAP
                                             i (without RESUNEl?
190C  208.05          JR     NZ, 19E3H       yes, no error handling.
19EN  35              DEC    (Hll            ;TRAP flag
19DF  EB              EX     DE,HL           ;The error routine address in ll.
19E0  C3 31D          JP     1036H           Continue the programme there
19E3  AF              XOR    A               Delete iTRAP Flag
19E4  77              LO     (HL),A
19E5  59              LO     E,C             ;error code back in E
19    CD F9 20        CALL   20F9H           ;if created, issue CR
19E9  21 EC 3C        LD     HL, JCECH       {Address of error messages
19EC  CD A 79         CALL   794             {RA Expansion Output
19EF  57              LD     D,A             D= 0
190   3E JF           LD     A,JFH           ; '?' spend
19F2  CD 24 03        CALL   032AH
19F5  CD D4 3C        CALL   3CD4H           ;Report Error
19F8  00 8 00         DEFB   0,0,0,0,0       ;6 x N0P
      080 00 80
19FE  21 1D 19        LD     HL.191DH        ;Text 'ERROR'
```

```
1A111  IT              PUSH   HL              and on Stack
 1A02  2A EA 78        LD     HL, (78EAH)     ;Load Error Line Number
 1A05  E3              EX     (SP),HL         ;with text adr, swap to stack
 1A06  CD A7 28        CALL   28A7H           Print i'ERROR
 1A09  El              POP    HL              ;Error line number voii stack
 1A0A  11 FE FF        LD     DE, OFFER       = 65534 2
 1A0D  DF              RST    18H
 1A0E  CA 74 O         JP     Z,0674H         Yes New System Initialisation
 1A11  7C              LD     A, H            = 65535? !FFFFl
 1A12  AS              AND    L               i <Direct Mode)
 1A1J  JC              INC    A
 1A14  C4 A7 0F        CALL   NZ,0FA7H        No, 'IN     Output row
 1A17  YES             DEFB   3EH             LD A, OCIH  dummy instruction
```

 # %# # # k E k#%#%###M}**# # ### fit**

BASIC - Main Loop

Jump either at 1A18 or 1A19

```
1A18  Cl              POP    BC              Correct stack
1A19  CD SB 03        CALL   038BH           ;Output flag on screen when
                                             ;required, CR on printer off,
1A1C  CD AC 79        CALL   79ACH           < RAN expansion output
1A1F  00 00 00        DEFB   0.0.0           {3 x NOP
1A22  CD F9 211       CALL   20F9H           CR on screen when ref.
1A25  21.29.19        LD     HL, 1929H       ;Text 'READY'
1A28  CD A7 28        CALL   28A7H           and spend
1A2B  YES 9A 78       LD     A, (7894/06)    ;without meaning
1A2E  D 62            SUB    2               i  .
1A311 00 00 00        DEFB   0, 0,0          3 NOP
1A33  FF 21           LD     HL, IIFFFFH     Current line number = Set FFFF
 1A3  22 A2 78        LD     (7842H),H
1A39  YES El 78       LD     A, 7EIH)        ;Auto Function Enabled?
1A3C  B7              OR     A
lAJD  28 3A           JR     Z,1A79H         no, normal input
```

%%%3%%3%#%~%#%# fit } % i MM

Programme Input under AUTO Function

```
1A3F  2A E2 78        LD     HL, 78E2H>      ;load next AUTO line1111111er
1A42  IT              PUSH   Hl...           ;and on the stack
1A4J  CD AF 0F        CALL   0FAFH           {Print Line Number
```

```
1A46  3E 20          LD    A,' .        then a space
1A48  C ZA 03        CALL  032AH
1A4.B D1             POP   EN           ; Line number in DE
1A4C  D05            PUSH  EN           and back on Strong
1A4D  CD 2C JB       CALL  I.B2CH       Find {part 1m programme text
1A50  DC 53 2E       CALL  C.2E53H      ;output' line
IA53  O              NOP
1A54  CD E3 03       CALL  0JE3H        {Read line from keyboard
1A57  D1             POP   EN           Load {AUTO Line Number
1A58  30O            JR    NC,1A60H     ike1n BREAK, normal continue
1A5A  AF             KÜR   A            Delete {AUTO flag
lASB  32 El 78       LD    (78E1Hl,A
lASE  18 B9          JR    1A19H        return to main loop
1A60  2A E4 78       LD    (OJ 478E41)  Load Auto Increment
1A63  19             ADD   HL, DE       ;Add to AUTO Line Number
1A64  38 F4          JR    C, 1A5AH     ;overly loud, exit AUTO mode
1A66  D5             PUSH  EN           ;Auto Line Number on Stack
1A67  11 F9 FF       LD    DE,0FFF9     {new CAR line number? 65528?
1A6A  DF             RST   !BH          ;HL/DE Comparison
1A6B  D1             POP   EN           {Auto Line Number Reload.
1A6C  3EC            JR    NC, IA5AH    i> 65528! Leave Auto Mode
1A6E  22 E2 78       LD    (78E2H),HI.. ;Remember new car line number
1A71  00 00          DEFB  0, 0         2 x NoP
1A73  21 E7 79       LD    Hl.,79E7H    Input/Output Buffer - 1 Addressing
1A76  C3 81 1A       JP    1A81H        Analyse and Retain iLine
```

%kt# i# k # } } # # t # %# #k # k #} k %o i %###

Normal programme input without CAR

```
1A79  00 80          DEFB  0.0          i2 x NOP
1A7B  CD EJ 03       CALL  03E3H        ;Read line from keyboard
1A7E  DA 33 1A       JP    , 1A33H      iBREAI<! zIIrMain Loop Start
1A81  D7             RST   10H          1. Find Character {$'
1A82  JC             INC    A           {=End of Line (0@)7
1A83  3D             DEC    A
1A84  CA 33 1A       JP    Z,1A33H      iYes" to main loops - Start
1AB7  FS             PUSH  AF           Save flag {y=1 if digit)
1A88  CD 5A 1E       CALL  lESAH        Decode {Line Number
!ABB  2B             DEC   Hl...        ;Buffer Address Back  (behind ZNr)
lASC  7E             LD    A, HL)       ;Load Characters
1A8D  RD 28          CP                 -
1ABF  28 FA          JR    Z, IABBH     Yes, continue back
1A91  23             INC   Hl...        Buffer pointer to 1.2          ZNr
                                        characters n
```

```
1A92  7E          LD    A, (H)      ;Load Characters
1A93  RD 20       CP                {= spaces?
1A95  C. C9 09    CALL  Z.09C9H     Yeah, one, white. pass
1A98  D5          PUSH  EN          {Line number on stack
1A99  CD C0 1B    CALL  1.BC0H      ;Generate Intercode
                                    {(HL=Beginning-1, BC=Length+5)
1A9C  DI          POP   EN          Reload line number
1A9D  Fl          POP   AF          ;Load flag 1
1A9E  22 E 78     LD    (78EH) , HL ;Beginning of the Intercode! as
                                    Save {current programme pointer
1AA1  CD B2 79    CALL  79B2H       {RAN-Erwe1 Term Output
1AA4  D2 SA 1D    JP    NC, 1D5H    Run Command
                                    il, character was not a digit
1AA7  D5          PUSH  EN          {Line number on the stack
1AA8  CS          PUSH  .BC         Line length on the stack
1AA9  AF          XOR   A           ;Clear RESUME/RETURN flag
1AAA  32 DD 78    LD    (78DDH), A
Alad  D7          RST   10H         {Line blank?
1AAE  B7          0R    A           Jan Zero Flag = 1
1AAF  F5          PUSH  AF          ;Save flag to stack
1AB   EB          EX    DE,HL       {Nostalgia of TR-80 Editor
L     22 EC 78    LD    (78ECH1,HL
1AB   E.B         EX    DE,HL
1AB5  CD 2C 1B    CALL  1B2CH       Find Row in Programme Text
1A.B8 CS          PUSH  .BC         ;Address pointer on it in the stack
1A.B9 DC E4 2.B   CALL  C, 2RE4H    ;if found, delete
1ABC  D1          POP   EN          Line Address in DE
1ABD  F1          POP   AF          ;Reload flags
1ABE  D5          PUSH  EN          ;Row address back to stack
1ABF  28 27       JR    Z, 1AE8H    back to start with blank
1AC1  D1          POP   EN          ;Reload Row Address
1AC2  2A F9 78    LD    HL, (78F9H1 Load {Programming Address
1AC5  E3          EX    (SP1,HL     ;Swap to stack with line length
1AC   C1          POP   RC          ;Prograuend address in BC
1AC7  89          ADD   H,BC        ;End Address+Line Length
1ACB  IT          PUSH  HL          ;=new end address. To Stack
1AC9  CD 55 19    CALL  1955H       ;Create new row space
1ACC  E1          POP   HI.         {new programmes Load d-Address
1ACD  22 F9 78    LD    (78F9H1,HL  ;and save
1AD111 EB         EX    EN, H       ;Line Address in HL
1AD1  74          LD    (HU,H       type any row pointer.
1AD2  D1          POP   EN          ;Reload Line Number
1AD3  IT          PUSH  HL          ;Line address on stack
1AD4  23          INC   HL          Row pointer to Munernfeld
```

```
1AD5  23          INC   HL
1AD6  73          LD    (HU,E        Enter {Line Number in Row
1AD7  23          INC   HL
1AD8  72          LD    HL),D
1AD9  23          INC   HL           {Row pointer to 1.Text byte
1ADA  EB          EX    DE,HL
1ADB  2A A7 78    LD    HL, (78A7H)  ;Input/Output Buffer Start Address
1ADE  EB          EX    DE,HL        EN
1ADF  1B          DEC   EN           ;- 2 = Begin of intermediate code
1AE0  1B          DEC   EN
1AE1  1A          LD    A,           Intercode in programme text
1AE2  77          LD    (HL)/        carry over
1AE3  23          INC   HL           ;Address pointer+ 1
1AE4  13          INC   EN
1AE5  B7          0R    A            ;End of Line? (08)
1AE6  20 F9       JR    NZ,1AE1H     ;no, transfer next .byte
1AE8  01          POP   EN           ;Load line start address
1AE9  CD FC 1A    CALL  1AFCH        from Row Addresses
1AEC  CD B5 79    CALL  79B5H        { RA! Extension Output
1AEF  CD 50 1B    CALL  1B5DH        Variables table and other
                                     ;Delete programme data
1AF2  CD B8 78    CALL  78B8H        RA} extension output
1AF5  C3 33 IA    JP    1AJJH        to start the main loop
```

%#ikt MM}Mi H}kt M

Renew Row Pointer to Full Text

```
1AF8  2A A 7B     LD    H, (7844H)   }programme text beginning in DE
1AFB  EB          EX    DE,HL
```

##With# With tt# i % It litt

Partly Renew Row Pointer

Ring: DE = the line address of the line, from
the row pointer should be renewed.

```
1AFC  62          LD    H, D         ;Initial Line Address in HI.
1AFD  B           LD    L,E
1AFE  7E          LD    (H)          {newline = @?
1AFF  23          INC   HL           {Prodigal?)
18    BM          OR    (HL.)
11101 approx      RET   or           Yeah, done
1B82  23          INC   HL           {pointer and leilennuner pass
11103 23          INC   HL
1MM   23          INC   HL
```

```
1B05  AF          XOR   A            A =
1BM6  .BE         CP    (HU          ;compare to byte from row
1B07  23          INC   HL           {Row pointer + {
1B08  20 FC       JR    NZ, 1B06H    ;no end of line, back
1BOA  EB          EX    DE,HI.       ;Line start address in HL
                                     ;DE= Next Line Address
1B0B  73          LD    (Hl),E       ;Addresses. Row as Row
1B0C  23          INC   HL           to save ;s
1B0D  72          LD    (Hl.l,D
1B0E  18EC        JR    lAFCH        ;next line
```

% %i% # With k i Mt t t Wed With Hi

Analyse arguments for LIST command

Eing. Zer flag = 1 if no argument is given
HI. = programme text address
Exp.: BC = Address of 1. output row
stack = 2. Line number

```
11110  11 00 08   LD    EN, 0        1, Line number = Set
1:BtJ  D5         PI/SH EN           ;and on stack
1B14   28.89      JR    Z, 1:B1FH    no arguments, next
1:B16  01         POP   EN           {@Remove from Stack
1:B17  CD 4F 1E   CALL  1E4FH        I, decode line number
1:BtA  D5         PI/SH EN           and pack on stack
1B1B   28 0B      JR    Z,1B28H      {no more characters!
                                     $2. Line number = 1.   set
1B1D   CF         RST   8            follows a '- 2
1B1E   CE         DEFB  CEH          ;token for '-'
1:B1F  11 FA FF   LD    EN, OFFF     2. Line = Set 5539
1B22   C.lt 4F 1E CALL  NZ, 1E4FH    more signs? Yes,
                                     2. Decode Line No.
1B25   C2 97 19   JP    NZ, 1997     ;more characters?
                                     Yes, SYNTAX ERROR
1B28   E          EX    DE,HL        2. Line number!r in 11.
1:B29  Dl         POP   EN           1. Line 1n EN
1B2A   EJ         EX    (SP),HL      2. Row Numbers on Stack with
                                     Swap Return Address,
1B2B   E5         PI/SH HI.          ;Return address back to stack
```

%i#k ## ## t # iH MM }# #litt

Find Row in Programme Text

Ring: DE = Number of row

Exp.: Line present: Carry = 1 Z flag = 1

```
                                        BC = Line Address
                                        HL = address of n,  lenle
        Row not present:
                                        Carry = 0, Z flag =
                                        BC = Line address of n. Line
                                        HL = Address of the supern. Line
        not found and reached Progral11111end:
                                        Carry = , Z flag = 1
                                        BC,HL = Programme address Z
 1B2C  2A4 78        LD    HL,(78A4HI      ;Progrill!111Load attachment
 11!2f  44           LD    B,H             Line address in BC
 1,830  40           LO    C,L
 1B31  7E            LD    A,(H)           End of programme ?
 1B32  23            INC   HL              ;(Row = 0000)
 1B33  II6           OR    (HL)
 1B34  2B            DEC   HL
 1B35  CB            RET   or              Yes, done!
 11\36  23           INC   HL              Programme pointer to line number
 1B37  23            INC   HL
 11\38  7E           LD    Ay(H)           Line number in hl. load
 139  23             INC   HL
 1B3A  6             LD    H,(HI)
 1B36  6F            LD    L,A
 1B3C  DF            RST   18H             ;HI.IDE. Comparison
                                           i= Looked Line?
 11\3D  60           LD    H,B             ;Load line start address
 1B3E  69            LD    L+C
 1B3F  7E            LD    A, (H)          ;Load Row
 1B'Iiil  23         INC   HI...
 1B41  b             LD    H, (HI..)
1:8'12  6F           LD    s
1:8'13  3F           CCF                   Invert Carry Flag
  1B4  C8            RET   or              ;Lookup line? yes-ready
1:8'15  3F           CCF                   Carry Flag back
 1B4  D0             RET   NC              ;Line111ter > Lookup Line
1:8'17  18 E         JR    1B2FH           ;Examine next row
 7M
```

HHfHffHHFFfHHHfHfHfHH

NEW - Command

Reset all variables and pointers
(the String Range Definition is retained)

```
 1B49  Co            RET   NZ'             ! Parueter? ja-SYNTAX ERROR
 IB'IA  CD C9 81     CALL  01C9H           ;Clear Image
```

```
1BD   2A AA 78     LD     HL, (78A4Hl     Start of programme text in HL
1B50  CD F8 1D     CALL   1DF8H           ;Call TROFF
1B53  32 EI 78     LD     (78E1H), A      ;Delete Auto Mode
15    77           LD     (HL),A          Row pointer = @o to programme
1B57  23           INC    HL              ;textantang (delete programme)
1B58  77           LD     (HU,A
1B59  23           INC    HL              {pointer behind ▼▼
1B5A  22 F9 78     LD     (78F9Hl,HL      Save {as Programming Address
1B5D  2A Alt 78    LD     HL, (78A4H)     } Load programme start address
1.Bb0 2B           DEC    HL              j- 1
1.Bb1 22 DF 78     LD     178DFHl, HL     to the programme gate t
                   Type Code Table = Set Simple Precision
1B4   O1A          LD     B, 1Al          {counter = 2
1B    21 01 79     LD     H, 7901H        ;Start Table Address
1.Bb9 36 04        LD     (H),4           ;Code f. a.Gen. enter
1.BbB 23           INC    HI.             next byte
1BC   10 FB        DJNZ   1M9H            {counter - 1. Ready?
1.B6E AF           XOR    A               yes, delete TRAP flag
1B6F  32 F2 78     LD     (78F2),A
1B72  6F           LD     L+              H = 8
1B73  67           LD     H,A
1B74  22 F0 78     LD     (7F0H), hi.     {URL of an error routine = @
1B77  22 F7 78     LD     17BF7Hl,HL      ;CONT address pointer = 0
1B7A  2A Bl 78     LD     HL,(78B1Hl      ;BASIC-RAN Load End Address
1B7D  22 D6 78     LD     (78D6HL,HL      as a string area pointer.
                                          ;deletes all string variables
1B88  CD 91 1D     CALL   1D91H           Call iRESTORE
1B83  2A F9 78     LD     HL, 17BF9HJ     Load {Prgr to End Address
1B86  22 FB 78     LD     17BFBl,HL       ▬
1B89  22 FD 78     LD     17BFDHl,HL      ;= Natrix table end address
1BSC  CD BB 79     CALL   79BBH           {RA!! Extension Output
1BSF  C1           POP    BC              ;Load Reversal Address
1B98  24 A 78      LD     HL,(78A0Hl      {Address of the String Range
1B93  2B           DEC    HL              ;- 2
1B94  2B           DEC    HL
1B95  22 ES 78     LD     1 78E8MH), HL   Save as stack header
1B98  23           INC    HL              ; + 2
1B99  23           INC    HL
1B9A  F9           LD     SP,HL           ;Transfer to stack pointer
1B9B  21 M5 78     LD     HL.78B5H        Cache f. Delete strings,
1B9E  22 B3 78     LD     (78B3),HL       ; ( Start address in pointer 1
1BA1  CD SB 13     CALL   038BH           ;Output flag on Screen,CR on
                                          ;Print printer if ertorderl.
```

```
1BA4  CD 69 21      CALL   2169H          ;End Querv
1BA7  AF            XOR    A              {A =
1BAB  67            LD     H,A            HL = 0
1BA9  6f            LD     L,A
1BAA  32 DC 78      LD     (7BDCH),A      Unlock Indication
1BAD  IT            PUSH   HL             ;0 on stack as final detection
1.BAE  CS           PUSH   .BC            Skip back, back to stack
1.BAF  2A DF 78     LD     HL, (78DFH)    Programme Continue Pointer
1.8.82 C9           RET
```

 # #}% 4%% %8%% 4%±%%%% #± %}

Emit question mark and read a line

```
L.B.83  3E 3F       LD     '?'            ;Print question mark
1.BBS   CD 2A4 03   CALL   032AH
1BB8    3E 20       LD     A,' '          Print spaces
1.BBA   CD 2A 03    CALL   B832Al
1BBD    C3 34 05    JP     0534/05        ;Read a line
```

#i# With It Mil # MM} E

Analyse Row and Generate Intercode

Ring: th = Text start address (completed with 08)
Exp.: BC = Length of intercode+ 5
th = Address before Intercode
(= Input/Output Buffer - 3)

```
1BC0  AF            XOR    A              Delete iDATA Flag
1BCl  32 B8 78      LD     (78RMH),A
1BC   4F            LD     C,A            Character counter =
1.BC5 EJI           EX     DE,HL
1BC   2A A7 78      LD     HI,!7BA7Hl     ;Input / Output buffer address
1BC9  2B            DEC    HI.            ;- 2
1:BCA 2B            DEC    HI.
1BCB  EB            EX     DE,HL          ;in
1BCC  7E            LD     AR (HL)        Load characters from text line
1BCD  RD 20         CP     ' '            i= Spaces?
1BCF  CA SB 1C      JP     Z,1C5BH        Yes! transmit directly
1BD2  47            LD     ,A             ;Character in B !as separator)
1BD3  RD 22         CP     31             ;=Quotation mark?
1BD5  CA n 1C       JP     Z.1C77H        Yes! Transfer String
1BD8  8/87          OR     M              ;End of Line?
1BD9  CA 70 1C      JP     Z,1C7DH        Yes! finished
1BDC  34 B 78       LD     A, ( 7B8ROH)   Load iDATA Flag
IBDF  B7            OR     A              {set?
```

```
1BE0  7E          LD    ,{HL.)      ;Load Characters
1BE1  C2 SB 1C    JP    NZ, 1C5BH   Yes! transmit directly
1BEA  FE 3F       CP    • 2°        {= Question mark ?
1BE6  3E B2       LD    A, @R2      Load iPRINT Token
1BE8  CA SB 1C    JP    Z, 1CSBH    Yes! to transfer to intermediate
1BEB  7E          LD    A+ HL)      ;Load the character again
1BEC  RD 38       CP    '0'         Character < '0' ?
1BEE  38 05       JR    C, 1BE5H    yes, check for keywords
1BF0  FE 3C       CP    O           ; Character<'(' ?
1BF2  DA SB 1C    JR    C,1C5BH     ;yes, take over directly

                  Check text for valid BASIC key location

BFS   D5          PUSH  EN          ;Intercode pointer to stack
1BF6  11 4F 1     LD    DE,164FH    Start address of keywords
1BF9  CS          PUSH  BC          ;Character counter on stack
BFA   01 3D 1C    LD    BC,1C3DH    ;Set Reversal Address
lBFD  C5          PIJSH BC
1BFE  O 7F        LD    B,7FH       ;Set token counter = 7F
1c00  7E          LD    To (HL)     ;Load characters from text
1C01  FE 6I       CP    61H         ;lowercase?
1C03  38.87       JR    C, 1CCH     No!
1C05  FE 7B       CP    7BH
1C07  30.03       JR    NC, 1 C0CH  No!
1C09  Eb SF       AND   SFH         to uppercase
1C0B  77          LD    (HU,A       ;Character back to text
1C0C  4E          LD    C. (It)     1. Load Characters
1C0D  EB          EX    DE,HL       {keyword indicator in HL
1C0E  23          INC   HL          ;Find Next Keyword
1C0F  B6          OR    (HL)        {Start of keyword?
1C10  F2 0E 1     JP    P, 1C0EH    No, continue
1C13  8           INC   B           ; token counter + 1
1C14  7E          LD    A, (s)      1. Keyword character
1ClS  E 7F        AND   7FH         Delete Bit 7
1C17  approx      RET   or          ;End of key1110rt table
1C18  B9          CP    C           J= Text characters?
1C19  20 F3       JR    NZ, 1C0EH   ;no, next keyword
1ClB  EB          EX    DE,HL       Swap Location
1ClC  E5          PUSH  Hl...       ;Buffer pointer to stack
1C1D  13          INC   EN          Key value pointer + 1
tClE  lA          LD    A, (DE)     {n, keyword character
1C1F  B7          OR    A           {new keyword?
1c20  FA 39 1C    JP    H,1C39H     yes, keyword detected
1C23  4F          LD    CA          ;Character in C
1c24  78          LD    A, B        Token = 60TO?
```

```
1C25  FE BD     CP    BDH
1C27  20,082    JR    NZ,1C2BH      {n1m, continue
1C29  D7        RST   10H           ;yes, space allowed
1C2A  2B        DEC   Hl...         {Buffer pointer one character back
1C2B  23        INC   HL            ;Next character buffer pointer
1C2C  7E        LD    A, HL)        ;Character from text Jaden
1C2D  FE 61     CP    1H            lowercase?
1C2F  38.02     JR    C,1C33H       {no!
1C31  E 5F      AND   SFH           Convert to uppercase
1(33) B9        CP    C             {= Keyword character?
1C34  2B E7     JR    Z,1C1DH       Yes continue
1C3   El        POP   HL            {no, buffer pointer back
1C37  1B D3     JR    1C0CH         ;Try Next Keyword

                Determined Token

1C39  4B        LD    CM            ;token in C
1C3A  Fl        POP   AF            Clean up the stack
1C3B  EB        EX    DE,HL         ;Swap Location
1C3C  C9        RET

                Token or Text in Intercode

1C3D  EB        EX    DE,HL         {H = Buffer Pointer
1C3E  79        LD    A,C           ;Character or token in A
1C3F  Cl        POP   BC            ;Load character counter
1C40  Dl        POP   EN            Load {Intercode Pointer
1C41  EB        EX    DE,HL         Swap Location
1C42  RD 95     CP    95H
1C44  36 3      LD    HL),':°       ':° in Intercode
1C4   20 02     JR    NZ, 1CH       No, 'ignore
1C4B  0C        INC   C             ;Yes, character counter+ 1
1C49  23        INC   HL            Intercode pointer behind ':'
1C4A  FE FB     CP    0FBH          • • • - token?
1C4C  20 0      JR    NZ,lCSAH      No!
1C4E  3 34      LD    (H),':'       o
1CS0  23        INC   Hl...         ;Intercode Pointer+ 1
1CS1  O 93      LD    3.93H         ;REN token in Z111
1CS3  70        LD    (HL),B
1CS4  23        INC   Hl...         ;Intercode Pointer+ 1
1035  EB        EX    DE,HL         ;Swap Location
1C56  0c        IN    C             Character counter + 2
1C57  %c        INC   C
1C5B  18 1D     JR    1C77H         Remaining text from putter
                                    to transfer to intermediate code
lCSA  EB        EX    DE,HL         Swap Location
```

```
1C5B  23          INC   HI...          ;Buffer pointer+ 1
1C5C  12          LD    (EN),          ;Token or character in intermediate
1C5D  13          INC   EN             Intercode Pointer + 1
1C5E  0c          INC   C              ;Character counter+ 1
1C5F  D 3A        SUB   ":°            {= ':' 7
1C61  28 04       JR    Z.1C67H        Yes, delete DATA flag
1C63  FE 4E       CP    EH             iDATA Token? (88-YES)
1C55  20 03       JR    NZ, 1C6        {no!
1C67  32 BI 78    LD    (78B0H),A      yes, set DWTA flag
1C6A  D 59        SUB   59H            REN token? (93 - YES)
1C6C  C2 CC 1B    JP    NZ, 1BCCH      no, back
1C6F  47          LD    MA             i0 as separator in B
1C70  7E          LD    AR {HL)        ;Text to separator or line
                                       ;end unchanged in intermediate code
1C71  B7          Oll   A              ;End of Line?
1C72  28 09       JR    Z,1C7DH        Yeah, done
1C7't B8          CP    B              ;separator?      (in ı = · )
1C75  28 E4       JR    Z, 1C5BH       Yes, back
1C77  23          INC   HI...          ;buffer pointer+ l
1C78  12          LD    EN) ,A         ;Characters in Intercode
1C79  0c          INC   C              ;Character counter+ 1
1C7A  13          INC   EN             ;Intercode Pointer+ 1
1C7B  18 F3       JR    1C70H          next character
1C7D  21.85.080   LD    HL.5           iHL = 5
1C80  44          LD    B,H            RB = 8
1C81  @9          ADD   HL,BC          ;Character counter+ 5
1C82  44          LD    B,H            in BC
1C83  4D          LD    C,L
1C84  2A A7 78    LD    HI, (78A7H)    Anfar.  d, I/O Buffer
1C87  2B          DEC   HL             - 3
1C88  2B          DEC   HL
1C89  2B          DEC   HL             ;Intercode
1CBA  12          LD    (EN),          End of Intercode with 3 zeros
1CBB  13          INC   EN             Select {
1CBC  12          LD    (EN),          ;(end identifier for direct
1C8D  13          INC   EN
1CBE  12          LD    (EN) ,A
lCBF  C9          RET                  that's it
```

#%Hit+ it Mt With i ! 11t

Restart 18

Comparison of HL and DE

```
                        Eing: HL,DE = 1 unsigned bit integer
                        Exp.: HL > DE: Z=0, Cy=0
                              Hl = EN Z=1, Cy=0, A=0
                              HL DE: Z=8, Cy=1
1C90  7C          LD    A, H          iMSB HL = 11SB DE?
1C91  92          SUB   D
1C92  C0          RET   NZ            No, done
1C93  7D          LD    L.            L.SB HL = LSB DE?
1C94  93          SUB   E
1C95  C9          RET

                  %#%%% %  %%f %lt # # %# t ii}ME } d

                        Restart 8

                        syntax checking
                        Eing. H = address of the byte to be checked
                              RST 8 test byte - Command
                        Exp.: HL = character after the check byte, llll!nn equal.
                              SYNTAX ERROR in case of inequality.
1c96  7E          LD    A, (Hl)       ;Load character from pointer
1C97  E3          EX    (SP),H        Swap Pointer with Reverse Reverse
1C98  BE          CP    (Hll          ;= dea, follow the call, characters?
1C99  23          INC   HL            Return address + f
1C9A  E3          EX    (SPl,HL       ;swap with pointer again
1C9B  CA 78 1D    JP    Z, lD7BH      ;same, continue with RST 10
1C9E  C3 97 19    JP    1997H         ;Unequal, SYNTAX ERROR

                        ## #Mi ttt# #M kt

                        FOR - On.eisung
1CA1  3E          LD    , 64}         Lock Indexing
1CA3  32 DC 78    LD    (78DCH), A
1CA6  CD 21 1F    CALL  1F21H         Start value in Run variable
1CA9  E3          EX    (SPl,Hl       {Programme pointer to stack
lCAA  CD 36 19    CALL  1936H         loop with same run
                                      ;variables already on the stack?
lCAD  D1          POP   EN            {Prog hand in DE
lCAE  20.05       JR    NZ, 1CB5H     No!
1CBe  @9          ADD   HL,BC         ;yes, by stack correction all
                                      ;Grind to Delete There
1CB1  F9          LD    SP,HL         Reset {stack pointer
1CB2  22 E8 78    LD    (78E8Hl,Hl    Save {and New Starting Value
1CB5  EB          EX    EN, Hl        {H Image Pointer
1CB6  0E 88       LD    1.7           ;at least 16 bytes free?
```

```
1CB8  CD 63 19     CALL  193H          ;no, OUT OF MEMORY - Error
1CRB  IT           PUSH  HL            Programme Pointer on Stack
1CBC  CD 05 1F     CALL  1F05H         Find {next Statement
1CBF  E3           EX    (SP),HL       ;Programme pointer to n. Statement
                                       on stack, load old pointer
1CC0  IT           PUSH  HL            ;and also back to the stack
1CC1  2A A42.78    LD    HL, (78A2H)   Load Line Number
1CC4  E3           EX    (SP), HL      ;Switch to stack with pointer
!CCS  CF           RST   8             ;Follows a '@ - Token ?
1CC6  RD           DEFB  0BDH
1CC7  E7           RST   20H           Test Tgp of the Volume Variables
1CC8  CA F MA      JP    Z,0AF6H       iString? Yes, TYPE MISMATCH Error
1CCB  D2 F 0A      JP    NC, AF1       dopp.en.? ja TYPE NISNATCH - Err
1CCE  FS           PUSH  AF            ;Save Type Flag
                                       ;{FF = Integer, @1 = Eint.Accurate)
1CCF  CD 37 23     CALL  2337H         ;Compute Encoded Expression
1CD2  F1           POP   AF            ;Load Type Flag
1CD3  IT           PUSH  HL            ;Programme pointer to stack
1CD4  F2 EC 1C     JP    P,1CECH       ;t. Accuracy
1CD7  CD 7F 0A     CALL  0A7FH         ;Integer, End Value UIIMllland
1CDA  E3           EX    (SP),H.       ; HL programme pointer
                                       ;End!Selects the stack
lCDB  11 01 00     LD    EN, 1         ; Increased = 1
lCDE  7E           LD    , (HL)        ;Load next character
1CDF  FE CC        CP    0CH           i= STEP token?
lCE1  CC 01 2B     CALL  Z,2B01H       yes, value increase and
                                       to Integer to DE)
1CE4  D5           PUSH  EN            Increase to the stack
!CES  IT           PUSH  HL            Rescue { Programme Slider
lCEb  EB           EX    DE,HL         {Increased in Hi
1CE7  CD 9E 09     CALL  099EH         ;Test increase value
1CEA  18 22        JR    100EH         Next on 1D0E
lCEC  CD Bl 0A     CALL  0AB1H         ;End value in a single gene. roll-up
lCEF  CD BF 09     CALL  09BFH         Transfer to Y
1CF2  egg          POP   HL            Reload {programme pointer
1CF3  CS           PUSH  C             ;End value on stack
1CF4  D5           PUSH  EN
1CF5  01 00 81     LD    BC 810MH      Increase value = 1 in Y
1CFB  51           LD    D,C
1CF9  54           LD    E,D
1CFA  7E           LD    A, (Hl)       ;Load next character
lCFB  FE CC        CP    0CCH          i= STEP token?
lCFD  3.01         LD    A,1           Set {positive increase flag
```

```
1CFF  20 0E          JR    NZ, 1DOFH     {nen:
1D01  CD 38 23       CALL  2338H         Evaluate {Increased
1D0   E5             PUSH  HL            ;Programme pointer to stack
1D005 CD 111 0A      CALL  0AB1H         {Increased in units.        wall,
1008  CD BF 09       CALL  09BFH         and enter 1n Y
10011 CD 55 09       CALL  0955H         Test {increase value (=1    Enn
                                         ;positive, A=FF  if negative)
1D0E  El             POP   HL            Load programme pointer
1D0F  CS             PUSH  llC           Increased to stack
1D10  DS             PUSH  EN
iD11  4f             LD    C,A           {increase flag 1n C
1D12  E?             RST   20H           Test increase value type
1D13  47             LD    11.A          ;Type flag in B
                                         ; (01 = single gene. , FF= Integeri
1D14  5              PUSH  llC           gp flag u. erh flag aut stack
1015  IT             PUSH  HL            {Programme pointer on stack
1D16  2A DF 78       LD    HL, (78DFHl   ;Address of the 1n HL volume
1D19  E3             EX    (SP),HL.      ;swap with prog pointer to stack
1D1A  6 81           LD    B.81H         ;FOR token (81) 1n B
1D1C  CS             PUSH  llC           ;as marking on the stack
1D1D  33             INC   SP            ;Remove LSB
```

#%%%% % t% t% 3 t 3 } %i }}3 % ii4

programme execution

HL. Must point to ":" of end of line

```
DlE   CD 58 83       CALL  0358H         ;Remove keyboard
1D21  117            OR    A             ;new key pressed?
1022  C4 A0 1D       CALL  NZ, 1DA0H     yes, analyse
1D25  22 E6 78       LD    !78E6HL,HL    Save {programme pointer
1028  ED 73 ES 78    LO    !78E8Hl,SP    ;Save stack pointer
1D2C  7E             LD    A, HI)        ;Load Characters
1D2D  FE 3A          CP    ';            ;:'? (Multiple statements in line
102F  28 29          JR    Z, 1D5AH      Yes!
1D31  117            OR    A             ;End of Line?
1032  C2 97 19       JP    NZ,1997H      No, SYNTAX ERROR
1035  23             INC   HL            {programme end?
1036  ?E             LD    To (HL)       i<row pointer = 0000)
1D37  23             INC   HL
1038  B              OR    (HI...)
1039  CA 7E 19       JP    Z.197EH       yes, implicit end
1D3C  23             INC   HL            {Programme pointer on line number
1D3D  SE             LO    E HL)         ;Load line warning in DE
103E  23             INC   HL
```

```
1D3F  5          LO    D, (HL)
1D40  EB         EX    DE,HL          No. in HL, prog. zeiger in OE
1D41  22 A2 78   LD    (78A2H),HL     Line number = current ZNr
1D44  3A 1B 79   LD    A4, (791BH)    ;Tractor on?
1D47  1.7        OR    A              (TRON)
1048  28 0F      JR    Z, 1DS9H       {no!
1D4A  D5         PUSH  OE             {Programme pointer to Stark
1041! 3E 3C      LD    A,3CH          i' >' output
1D4D  CO 2A 03   CALL  032AH
1D50  CD AF 0F   CALL  0FAFH          ;Print Line Number
1D53  3E 3E      LD    A,3EH          ;''
1D55  CD 2A 03   CALL  032AH
1058  01         POP   EN             ; Reload the programme pointer
1059  EB         EX    DE,HL          HL programme pointer
1D5A  D7         RST   10H            ;Address next character
1051! 11 1E 1D   LO    EN, NLeh       ;Return address to stack
1D5E  D5         PUSH  EN
105F  approx     RET   or             End of statement
1D    D 80       SUB   80H            iToken?
1D62  DA 21 1F   JP    C,1F21H        ;no, no LET assignment
1D65  FE 3C      CP    3CH            Instruction token?
1D7   D2 E7 2A   JP    NC,2AE7H       No!
1DoA  e7         RLCA                 ;Token * 2 in BC
1DB   4F         LD    C,A
1D6C  6 00       LD    B,0
lDoE  EB         EX    DE,HL          ;Programme pointer in DE
1D6F  21 22 18   LD    HLi1822H       ;Start of jump table
1D72  09         ADD   HL,BC          ;+ 2 * Token = Pointer to Sprungadr.
1073  4E         LD    C HL)          ;Load explosion address
1D74  23           INC   Hl...
1075  4            LO    3, HL)
1D76  CS         PUSH  BC             ;and on the stack
1D77  E\         EX    DE,HL          {Programme pointer back in H.
```

#Mt killt}"t #_i_k ### E #ME#With#

Restart 10

Find Next Character in Proi;,ra11111text
089, A (LF) and 28 {' ') who passed the
Eing. H = Programme pointer
Exp.: A = Character
Carry = 1 if digit
Z-Flag = 1, all line or statement ends

```
1D78  23         INC   HL             { Programme pointer +
                                                          1
```

```
1079  7E              LD    A, (HL)         Load Cards
1D7A  FE YES          CP    :               ':' 3
1D7C  D0              RET   NC              3yes!
1D7D  RD 20           CP                    Empty?
1D7F  CA 78 1D        JP    , 1D78H         yes, next character
1D82  FE OB           CP    0BH             3€ 0Bra?
1084  30.05           JR    NC, 1D8BH       No!
1D86  FE 09           CP    09H             ;> 09H?      (closes 09 u,  0A off)
1088  D2 78 1D        JP    NC,1D78H        yes, next character
1D8B  FE 30           CP    '0'             ;digit?
1D8D  3F              CCF                   $yes, carry = 1
1D8E  JC              INC   A               ;End of Line?
1D8F  3D              DEC   A
1090  C9              RET                   ;finished
```

%#ii #~%3%%%%%% %%%%~1$%~% 3%%~

REST0RE Statement

Reset the DATA pointer

```
1091  EB              EX    DE,HL           { programme pointer in DE
1092  2A4 78          LD    HL, !7BA4HJ      ;Load Prograa Startup Address
1095  2B              DEC   HL              i - 1
  1D  22 FF 78        LD    (78FFH>,HL      ;as DATA pointer
1099  EB              EX    DE,HL             Programme pointer back in H
1D9A  C9              RET                   finished
```

#Mt litt# 4 #Hit#tt %%#~

Keystroke During the Run

or at LIST

```
1D9R  CD 58 03        CALL  0358H           ;Press?
1D9E  B7              OR    A
1D9F  C8              RET   /               No!
1DA8  00 80 00 00 00  DEFB  0.0.0.8         15 x~NOP
1DA5  32.99.78        LD    (7899),A                Characters in IMKEY$ Cache
1DA8  3D              DEC   A               BREAK?
1DA9  Co              RET   NZ                    No, done!
```

%i} 3 iWith tti#Ei

BREAK *interruption of* programmes

```
lDAA  JC              INC   A                       iA = Set 1 (BREAK ID)
lDAB  C3 B 1D         JP    1DB4H           {continue at END
```

END Statement

Stop Executing

```
1DAE  C0          RET   NZ            {follow parameters? yes, mistake
1DAF  FS          PIJSH AF            iEND flag !A=01 on stack
1DB0  CC B.B 79   CALL  1, 79BBH      ;RA Expansion Out5gang
1D.83 Fl          POP   AF            ;Load End Flag again
1D.84 22 E6 78    LD    178E6HL,HL       Save current programme pointer
10/87 21 BS 78    LD    HL.78.BSH           Cache, delete strings.
1DBA  22 3 78     LD    (7BB3Hl,HL      ;<pointer to top)
10.BD 21          DEFB  21H             ;LD HL,0FFF6H  Dmmy instruction
```

BREAK response in INPUT statement

```
1DBE  F FF        OR    QFFH            ; END=Flag = FF !BREAK in INPUTl
1DC0  Cl          POP   BC              {Return address from stack
1DCl  2A A2 78    LD    HL, (78A2Hl)    Load current line number
1DC4  IT          PUSH  HL              ;on stack
1DC5  FS          PUSH  AF              iEND flag on stack
1DC   7D          LD    A; L            {Line number = FFFF 7
1DC7  A           AND   H               ; <Direct Hode)
1DCB  3c          INC   A
1DC9  28 09       JR    Z,1DD4H         Yes!
1DCB  22 FS 78    LD    478F5H),H            ;no, as CONT Line No. save
1DCE  24 E 78     LD    HI… (78E6H)     current programme pointer
10D1  22 F7 78    LD    (7BF7HJ,HL      Save as CONT Pointer
10D4  CD 8R 03    CALL  038BH           ;Output Flag    Up    Monita.      CR
Up
                                            Emit {Drurker if required.
1DD7  CD F9 2     CALL  20F9H                ;CR on screen if required
1DDA  F1          POP   AF              Load iEND flag
1DDB  21.30.19    LD    HL, 1930H       ;Text 'BREAK'
LT    C2 %61A     JP    NZ, 1AH         if not END and not Direct
                                        il'lode, 'BREAK IN LINE'
1DE1  C3 1B 1A    JP    1A18H           back to main loop
```

%%%%%4%%% % %i ##### #%% H#HE

CONT Statement

Continue programme execution after BREAK or Error

```
1DE4  2A F7 78    LD    HI… (78F7Hl     ;CONT - Load the programme pointer
1DE7  7C          LD    A,H             = 080?
1DEB  BS          OR    L               {no sequel possible
1DE9  1E 20       LD    E, 201H         ;Error Code CANT CONTINUE
```

```
1DEB  CA A2 19    JP    1, 1942H        yes, output error message
1DEE  EB          EX    DE, HL          { programme pointer in DE
1DEF  2A FS 78    LD    Ht.,<78F5H      {CANT line number load
1DF2  22 A2 78    LD    I (78A2H1,      as current line number spe1ch.
1DF5  EB          EX    HL DE, HL       {Programme pointer again 1n
1DF   C9          RET                   H ;Continue running
```

%%i# ## ###}#%#f%#}###}k%ikk # # %%kt

```
                  TRN - Tractor on statement
                  DEFB 3EH              Set iLD A,0AFH to TRON A<>0

                  TROFF Statement
                  Disable Trace
                  XOR   A               Set to TROFF A = 0
                  LD    (791BH,A        Save 5as TRACE flag
                  RET

                  POP   AF              not used
1DF7 JE           POP   HI.
                  RET

1DF8  AF
1DF9  321B 79
1DFC  C9
1DFD  FI
1DFE  El
1DFF  C9
```

4%%kt#t%tt t # i MM # t #t t }cd

```
                  DEFSTR - Define String
                  Variable Statement
1EBfil 1E         LD    E.3             Tgpcode = String in E
03 1E02 01        .DEFB 01              ;LD BC,021EH Duwuny command

                  DEFINT - instruction
                  Define Integer Variable
1E03 LE 02        LD    E.2             ; igpcode = Integer in E
1E05 01           .DEFB 01              ;LD BC,041EH lluamy Command

                  DEFSNG Statement
                  Define variable simple precision
1E86 1E 04        LD    E.4             Type Code = inf. Precision in E
1EIII 01          DEFB IU               iLD BC,081EH Duny Command

                  DEFDRL Statement
                  Define Variable Double Precision
1E09 1E 08        LD    E.8             igpcode = dpp. Precision in E
```

```
                                common routine
1E0B  CD 3D 1E      CALL   1E3DH        ;n. Text character= letter?
1E0E  01 97 19      LD     BC, 1997H    Address iSN Error Routine
1E11  CS            PUSH   BC           ;and pack on stack
1E12  Profit        RET    C            no letter, SYNTAX ERROR from9.
1E13  D 41          SUB    41H          ;Determine position in alphabet
1E15  4F            LD     C,           transferred to B and C
1E1   47            LD     B,A
1E17  D7            RST    10H          ;Load next character
1E1B  FE CE         CP     0CEH         ;= '-' tokens
1E1A  208.09        JR     NZ,1E25H     {no!
1E1C  D7            RST    10H          ;Load next character
1E1D  CD 3D 1E      CALL   1E3DH        ;=letter?
1E28  Profit        RET    C            ;no, output SYNTAX ERROR
1E21  D6 41         SUB    41H          ;Determine position in alphabet
1E23  47            LD     ,A           as upper value in B
1E24  D7            RST    10           ;Address next character
1E25  78            LD     A, B         2. Letter { 1. Letter?
1E2   91            SUB    C
1E27  Profit        RET    C            yes, SYNTAX - ERROR
1E2B  3C            INC    A            ; Difference+ 1 = Counter
1E29  E3            EX     (SP)/H.      ;Programme pointer to stack
                                        ;Delete SN routine address
1E2A  21 01 79      LD     HL, 7901     ;Address type code table
1E2D  O 8           LD     3.0          Offset t. 1. Letters in BC
1E2F  09            ADD    HL, B        ;+Top = 1st character in Tab.
1E38  73            LD     (Hll, E      Enter type code in table
1E31  23            INC    HL           ;table address+ 1
1E32  3D            DEC    A            ;counter - 1
1E33  28 FB         JR     NZ,1E30H     Ready? no-next character
1E35  El            POP    th           Load {Programme pointer
1E3   7E            LD     A, (H)       ;Load characters from programme
1E37  FE 2C         CP     ','          Are more parameters following?
1E39  8             RET    NZ           {no, ready
1E3A  D7            RST    10H          ;Load next character
1E3B  18 CE         JR     1EBH         enter further definitions
```

%%4%%### ##%k 1k # Mt#

Tests if character is letter

Ring: **H = Address of** character to be examined

etc. Cy = - letter, Cy = 1 - no letter

```
lEJD  7E            LD     A, (1.       ;Load Characters
1E3E  FE 41         CP     'A'          3A?
```

```
1E40  D8              RET   C             ;ves, no letter
1E41  FE 5B           CP    SBH           <= 1  yes, carry = 1
1E43  3F              CCF                 Invert Carry
1E44  C9              RET
```

%%%%%% i # } } %± i %t #

Evaluate expression and integer value

< 32768.
Eing: HL = Text Address - 1
Exp.: EN= Result

```
1E45  D7              RST   10H           ;Address next character
1E46  CD 02 2B        CALL  2B02H         Evaluate {expression
1E49  F0              RET   p             ;> 32767? No, done
```

%± ttltit% t Eilt pi }} #%# tt

FUNCTION CODE Error

```
1E4A  1E B            LD    E.8           ;Error Code in E
1E4C  CJ A2 19        JP    19A2H         Feh lernel Run
```

##M't## kt #With kt Mt Mt

String in number Ulllignde1n (< 65530 >

Eing: II.= Address of the string
Exp.: EN = Number

```
1E4F  7E              LD    A, (left)     ;Load character from string
1E58  FE 2E           CP    2EH           3='.° 2
1E52  EB              EX    DE,1i.        ;String pointer in DE
1E5J  2A EC 78        LD    HL, 78ECH)    ;'.' line number in H
1E56  EB              EX    DE, H         Swap hands and '.' ZMr
1E57  CA 78 1D        JP    Z.1D78H       Yeah, done
1ESA  2B              DEC   1i.           ;String pointer - 1
1E5B  11 ill          LD    EN,0          ; number= set 8
1ESE  D7              RST   drop          ;Load next character
1E5F  D               RET   NC            ;no digit, done
1E    IT              PUSH  1i.           String pointer to stack
1E61  F5              PUSH  AF            ;Character on stack
1E62  219819          LD    1i., 1998H    ;Number> 1998H?
1E65  DF              RST   18H           (i.e. number8 > 5529)
1E66  DA 97 19        JP    C, 1997H      Yes, SYNTAX ERROR
1E69  62              LD    HD            Number in Hl. 1,111Load
1E6A  6               LD    L,E
1E6B  19              ADD   HL, DE        Number + 2
```

```
1E6C  29          ADD   HL, HL          * 4
1ED   19          ADD   HL, DE          * 5
1E    29          ADD   HL, Hl    i     * 10
ELO   Fl          POP   AF              ;Load the character again
1E71! D 30        SUB   38              ;Remove Zone Section
1E72  5F          LD    E,A             ;in DE
1E73  16.00       LD    D,0
1E75  19          ADD   HL, DE          Add to 10f:Add Number
1E7   EB          EX    EN, Hl          ;Number transferred to DE
1E77  El          POP   HI.             ;Load String
1E7B  18 E4       JR    1E5EH           ;next Zi fter
```

M} ## #H} # i tHHH # #i

CLEAR Statement

Delete Variable and Define String Range

```
1E7A  CA 61 LB    JP    Z, 1B61H        No parameters? Jump in NEW
1E7D  CD 46 1E    CALL  1E46H           Evaluate Expression
1E80  28          DEC   HI…            ;Programme pointer- 1
1EB1  D7          RST   10H             Address next character
1E82  Co          RET   NZ              ;Instructor? No, Error
1E83  E5          PUSH  HI…             {Programme pointer on stack
1E84  2A Bl 78    LD    HL, (78B1Hl     ;BASIC-RA!'! Load End Address
1E87  7D          LD    A,L             - CLEAR statement argument
1E88  93          SUB   E               i= Start of string range - 1
1E89  5F          LD    E,A
1E8A  7C          LD    +H
1EBB  9A4         SBC   AD
1EBC  57          LD    D,
1EBD  DA 74 19    JP    C, 197AH        UNDERflow, OUT OF IENORY Error
1E90  24 F9 78    LD    HL, (78F9H)     iStartCard. of the variable table
1E93  081 28 Olil LD    BC,0            + 4
1E96  09          ADD   HL,BC
1E97  DF          RST   18H             i< new string range address - 1?
1E98  D2 7A 19    JP    NC,197AH        ;no, OUT OF L'IENORY - Error
1E9B  EB          EX    DE,HL           ;New String Area Board - 1
1E9C  22 AB 78    LD    (78ABH),HL      to save ;s
1E9F  El          POP   HI…             Load programme pointer
1EAB  C3 61 1B    JP    1:BolH          Continue with NEW
```

%%# ii%# it#ttt#MM#Mitlit}

RUN Statement

Start **the programme**

```
1EAJ  CA 5D 1B        JP     Z, 1.B5DH      ;no line number? continue at NE
1EA6  CDC7 79         CALL   79C7H          ;RAM expansion out
1EA9  CD Part 1.B     CALL   1.B61H         ;Delete Variable
1EAC  01 1E 1D        LD     .BC,1D1EH      ;Load Reversal Address
1EAF  18 10           JR     1EC1H          continue at öTo
```

%k#Hit}i%i#%3%}%i#MM

GOSUB Statement

Run Sub-Programme

```
1E.B1 0E 03           LD     6.3              test,  if there are still 6. bytes
1E.B3 CD 63 19        CALL   1963H
1E.B6 C1              POP    .BC            {Delete Return Address
1E.B7 E5              PUSH   HI             ;Programme Show. for RETURN aut Stack
1E.BB IT              PUSH   HL             ;Prog pointer on stack
1E.B9 2A A2 78        LD     HL, (78A2H>    with current line number
1EBC  E3              EX     (SP>,H1        ;swap
1E.BD PER 91          LD     A, 91H         ;91 as flag for GOSUB
1E.BF F5              PUSH   AF             ;to the stack
1EC0  33              INC    SP             Remove iLSB
1EC1  C5              PUSH   BC             Reverse Rec. back on stack
```

#iWith #kt #Mt#i Mt # With

GTÖ **Statement**

unconditional jump

```
1EC2  CD 5A IE        CALL   1E5AH          Find {Jump Line Number
1EC5  CD 07 1F        CALL   1F07H          Find End of Line
1ECB  E5              PUSH   ed             {Programme pointer to stack
1EC9  2A A2 78        LD     HL, (7842H)    ;current line111111er in HI.
1ECC  DF              RST    18H            {= Jump line number ?
1ECD  E1              POP    HL             ;Load Project Pointer
1ECE  23              INC    HI             ;to the beginning of the next line
1ECF  DC 2F 1B        CALL   C,1B2FH        yes, jump line from this line
                                            Search
1ED2  D4 2C 1B        CALL   NC,1B2CH       3no, jump line from programme
                                            Start Find
1ED5  6               LD     H,.B           {address of the jump line in HL
1ED6  69              LD     L,C
1ED7  2B              DEC    HI.            { Programme pointer before jump line
1EDS  D8              RET    C              {line present? Yes, continue there
```

in the event of

```
                                UNDEFINED STATEMENT Error
     1ED9  1E 8E                LD    E,0EH          Error Code in E
     lEDB  C3 A2 19             JP    19A2H          {Print Error Message

%%%%%%%%%~%% % ##tt e #i%Hi$4 ~%±

                                  RETURN Statement

                                return of a subcarrier to
ELEVATION  c0                   RET   NZ             iParameter? yes, mistake
     lEDF  FF 16                LD    D,0FFH         Retrieve VOii Stack
      oil  CD 36 19             CALL  193H           ;(FOR - skip data)
     lE't  F9                   LD    SP,HL          Re-initialise iStack
     1EE5  22 E8 78             LD    (78E8H),HL
     1EE8  FE 91                CP    91H            Data from a GOSUB call?
     lEEA  1E M                 LD    E,             iCode door RETURN WITHOUT GOSUB
     lEEC  C2 A2 19             JP    NZ,19A2H       ;no, output error message
     lEDF  El                   POP   HL             Load Line Nuner VOii Stack
     1EF0  22.42.78             LD    (78A2H), HL    ;save as current line.
     1EF3  23                   INC   HL             Direct code?
     1EFA  7C                   LD    A,H            ;(=FFFF)
   1EF5    B5                 OR    L
   1EF6    28 07              JR    NZ, 1EFFH      {no!
   1EF8    3A DD 78           ID    A, (78DDH)     Set iRESUNE/RETURN flag?
   1EF.B   7                  OR    A
   1EFC    C2 18 1A           JP    NZ,1A18H       Go back to main loop
   1EFF    21 lE 1D           LD    HL, 1D1EH      Load Return Address
   1Fi12   E3                 EX    (SP>,HL        Exchange {with programme pointer
   1Fil3   3E                 DEFB  3EH            3LD A, OEIH dummy instruction
   1F84    El                 POP   HL            Load {Programme
                                                   pointer

ftfffffffffffffffffffffffltllllNI

                                  DWT Statement

                                Find End of Statement
   1Fil5   IU 3A 0E           LD    C, OE3A        Separator = ':'      in C
   1FG8    088                NOP

HFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF

                                  ELSE Statement

                                Find Row End
   1Fil7   0E 08              ID    0.0            Separator!= 0111in C
                                                   ;Warning: 1Fil7-1F"8 redefined
```

```
1F09  06 0o          LD    B.0          ;separator? = in R
1F0B  79             LD    A+ C         ;Replace separator 1 and 2
1F0C  48             LD    ,B
1F0D  47             LD    B,A
IF0E  7E             LD    A, (HU       ;Load Characters
1F0F  B7             OR    A            ;=End of Line?
1F10  approx         RET   or           Yes, done
drop  BB             CP    B            ;= separator2?
1F12  approx         RET   or           Yes, done
1F13  23             INC   HL           Programme pointer + 1
1F14  RD 22          CP    '            i= Quotes?
1F16  28 F3          JR    Z, 1F0BH     ;Yes, swap separator
                                        i.e. only find line ends)
1F18  D6 8F          SUB   BFH          ; IF - Token?
1F1A  20F2           JR    NZ, 1FEH     No, go on
1F1C  B8             CP    B            if not in Str1ng or after
                                        ELSE, Carry = Set 1
1F1D  BA             ADC   A,D          ;Nest Counter + 1
1F1E  57             LD    D,A
1F1F  18 ED          JR    1F0EH        further
```

%%lt # # t # M% #i } # # #k t#

LET - Anve1 sung

Mr t assigned

```
1F21  CD 0D 26       CALL  260DH        ;Find variable in table
1F24  CF             RST   8            Does the sign follow'='?
1F25  D5             DEFB  S
1F26  EB             EX    DE,HL        {Variable Table Address
1F27  22 DF 78       LD    (78DFH),HL   ;Remember Variable door
1F2A  EB             EX    DE,HL
1F2B  DS             PUSH  EN           and pack on the stack
1F2C  E7             RST   2H           Test type
1F2D  FS             PUSH  AF           Type flag on stack
1F2E  CD 37 23       CALL  2337H        Evaluate Expression
1F31  F1             POP   AF           Load TgpFlag
1F32  E3             EX    (SP),HL      ; Stack Page
                                        Load URL to Variable Table
1F33  CS 03          ADD   3            Calculate TypeCode
1F35  CD 19 28       CALL  2819H        Result of expression in
                                        Convert Right Type
1F38  CD 03 04       CALL  083H         ;X address in DE
1F3B  IT             PUSH  HL           ;Adr, the variable tab. on stack
1F3C  20.28          JR    NZ.1F66H     ;Jump, ll!lln not string
```

```
                        String allocation
1F3E  2A 21.79          LD      HL, (7921H)     ;Load string pointer from X-Reg
1F41  IT                PUSH    Hl...           ;and on stack
1F42  23                INC     Hl...           ;Load string address
1F4J  5E                LD      E, (HL.)          ;in
1F44  23                INC     Hl...
1F45  56                LD      D, (HL)
1F46  2A A 78           LD      HL, (7844H)     String not in programme text; or
                                                i string area ?
1F49  DF                RST     !BH
1F4A  30                JR      NC, 1F5AH       yes, string in string range
1F4C  2A A 78           LD      HL, (7840H)     {String in programme text?
1F4F  DF                RST     18H
1F50  Dl                POP     EN              ;Load String
1F51  38 @F             JR      NC, 1F2H        Yeah, string not in string range!
1F5J  2A F9 78          LD      HL, (78F9H)     shows string pointer on Var.Tab.?
1F56  DF                RST     1BH
1F57  30.09             JR      NC, IF62H       {no, string not in the string.
1F59  YES               DEFB    JEH             LD A, QDIH  dummy instruction
155A  D1                POP     EN              ;Load String
1F5B  CD F5 29          CALL    29F5H           Delete iString in Cache
1F5E  EB                EX      DE, HL.         String pointer in HL
1F5F  CD 43 28          CALL    2843H           ;String in String Pane
1F62  CD FS 29          CALL    29F5H           ;String i111 Clear Cache
1F65  E3                EX      (SP),H          Pointer to Cache
                                                ;Load Stack, Var.Tab Address
1F66  CD D3 09          CALL    09D3H           ;value of X in variable table
1F69  Dl                POP     EN              ;stack  clean up
1F6A  El                POP     HL              ;Load the programme pointer
1F6B  C9                RET

Hit FFF

                        ON - Statement
                        jumping-strip branching
1F6C  FE 9E             CP      9EH             Is there an ERROR token?
1F6E  20 25             JR      NZ.1F95H          No!
                        ON ERROR
1F70  D7                RST     10H             ;Address next character
1F71  CF                RST     B               Is it a GOTO token?
1F72  8D                DEFll   BDH             i (BD = GOTO token)
1F73  CD SA lE          CALL    1E5AH           ;Decode Line111111er
```

```
1F7b 7A          LD    A, D          i: l'J?
1F77 33          OR    ~             i <disable error handling)
1F78 28 I'J9     JR    Z.1F83H       Yes!
1F7A CD 2A 1B    CALL  1B2AH         Find Row in Programme Text
1F7D SI'J        LD    D,B           Line address in DE
1F7E 59          LD    E,€
1F7F E1          POP   HL            Load {Programme pointer
1F88 D2 D9 1E    JP    NC, 1ED9H     Line not present!
                                     iUNOEFINED STATEMENT Error
1F83 EB          EX    DE,HL         Error routine address
1F84 22 Fl'J 78  LD    (78Fl'JHl, HL save
1F87 EB          EX    EN, HL
1F88 Profit      RET   C             Line no >l'J. done!
1FB89 34 F2 78   LD    A, 178F2H)    has an error already occurred?
1FBC B7          OR    A
1F8D approx      RET   or            No, done
1FBE 34.94.78    LD    , (7894H)     Error Code in E
1F91 SF          LD    E,A
1F92 C3 AB 19    JP    19ABH         for error handling

                 ON GOTO - ON GOSUB

1F95 CD 1C 2B    CALL  2B1CH         Evaluate Expression, Integer
                                     Value < 25) in E
1F98 7E          LD    (H)           Load characters from programme text
1F99 47          LD    B,A           in B
1F9A FE 91       CP    91H           ;= 60SUB token?
1F9C 28 I'JJ     JR    Z,1FA1H       Yes!
1F9E CF          RST   8             Is it a GOTO token?
1F9F 80          DEFB  BDH           ;(8D = GOTO token
1FA0 2B          DEC   HL            {Programme pointer
1FA1 4B          LD    CE            {Leap Variable in C
1FA2 O           DEC   C             iVariable - 1 =8?
1FA3 78          LD    A, B          {token in A ,, jump
1FA4 CA b0 10    JP    Z, 160%       yes, jump m n, line no
1FA7 CD SB 1E    CALL  1ESBH         ;Decode line number
1FAA FE 2C       CP    ,             A coma follows?
1FAC C8          RET   NZ            {no, programme is the next
                                     {Continue Statement
1FAD 18 F3       JR    1FA2H         next line number
```

#MM#Mt# li % t Mt#Mi Hit tt}

RESUME Statement

step back from *the* error handling

```
1fAF  11 f2 78      LD    DE, 78F2H      Address iTRAP Flaq
1FB2  1A            LD    A, (EN         Error occurred?
1FB3  B7            OR    A
1FB4  CA A 19       JP    Z, 1940H       No, RESUME WITHOUT ERROR
1FB7  3C            INC   A              {A =
1FB8  32 9A 78      LD    (7B894H),A     ;Delete Error Code
1FBB  12            LD    (EN),A         Delete iTRAP Flag
1FBC  7E            LD    A+ H.)         ;Load Characters
1FBD  FE 87         CP    87H            i= HEXT token ?
1FBF  28 0c         JR    Z, lFCDH       Yeah! RESUME NEXT
1FCl  CD 5A 1E      CALL  1E5AH          ;Decode line
1FC4  C             RET   HZ             more signs? yes error
1FC5  7A            LD    A,D            {Line number = A?
1FC   B3            OR    E
1FC7  C2 C5 1E      JP    HZ,1EC5H       in, continue with GOTO
1FCA  3C            INC   A              A = 1
1FCB  18 02         JR    1FCFH

                    RES\Jl'E NEXT
1FCD  D7            RST   purple         {next character in programme text
1FCE  C0            RET   NZ             no end of line, error
1FCF  2A EE 78      LD    HL, (78EEH>    Pointer to incorrect row
1FD2  EB            EX    DE,HL          EN
1FD3  2A EA 78      LD    HL, (78EAH)    Load Error Line Number
1FD6  22 A2 78      LD    178A2H>,HL     ;as current line number,he entry.
1FD9  EB            EX    DE,HL          Pointer back in H
1FDA  Co            RET   HZ             3 RESUME 8? yes-ready
1FDB  7E            LD    + rDI '        ;End of Line?
1FDC  B7            OR    A
1FDD  20.04         JR    NZ,1FE3H       no, next statement in line
1FDF  23            INC   HL             {Programme pointer on 1, statement
1FE0  23            INC   HL             of the next line
1FE1  23            IHC   HL             ;(behind pointer and line number)
1FE2  23            INC   HL
1FE3  23            INC   Hl...
1FE4  7A            LD    A, D           Direct-l'lode?
1FE5  A3            AND   E              i1LineWells!1' = FFFF)
1FEb  3C            INC   A
1FE7  C2 85 IF      JP    NZ, 1F05H      No, next statement, done
1FEA  3A DD 78      LD    A,78DDH)       ;RETURN/RES\Jl'E flag set?
1FED  3D            DEC   A
1FEE  CA BE 1D      JP    Z, 1DREH       Yes, finish programme execution.
1FF1  C3 05 lF      JP    1F05H          3 Find next statement, done
```

```
                ********************************************************************************
                ERROR Statement
                generated error
1 C     CALL        2B1CH               {Analyse Error Code
                                        ;ert {256) in A
1 C     RET         NZ                  ;more characters? yes error
1 E     OR          A                   ;Error Code= 0?
1 C     JP          Z, lEAAH            Yes, FUNCTION CODE - Error
1 3     DEC         A                   {Find Internal Error Code
1 8     ADD         ,A
1 S     LD          E,A                 ;and place in E
1 F     CP          2DH                 32D 7
2 3     JR          C, 2005H            Yes!
0 8
                ********************************************************************************
                ********************************************************************************
                UNPRINTABLE ERROR
2 1     LD          E.26H               ;Error Code in E
2 C     JP          19A2H               Error routine
0 3
                ********************************************************************************
                ********************
                AUTO Statement
                Auto0111atic Line Numbering
2 1     LD          EN, 10              Initial and increase value = 1
2 D     PUSH        EN                  to the stack
2 2     JR          Z,2025H             ;no 111more characters entered!
2 C     CALL        1E4FH               Decode initial value
2 E     EX          DE,HL               Initial value in H, Prg pointer DE
2 E     EX          (SP),HL             {Start value on stack
                                        ;10 in HL
2 2     JR          Z,2026H             no more characters entered!
2 E     EX          DE,HL               {Programme pointer in H
2 C     RST         8                   {Following a coma?
2 2     DEFB
2 E     EX          DE,HL               ;Progreauzeiger 111Again in DE
2 2     LD          HL,(78E4Hl          ;Load old increment
2 E     EX          DE,HL               ;HL Prog hand
2 2     JR          Z,2025H             no more signs after coma!
2 C     CALL        1E5AH               Decode {increase value
2 C     JP          NZ, 1997H           ;End of Line? no-SYNTAX ERROR
2 E     EX          DE,HL               ;Increase value in H
2 7     LD          M, H                = 8?
0 6
```

```
2027  B5          0R    L
2028  CA 4A LE    JP    Z,1E4AH          Yes, FUNCTION CüDE Error
 282  22 EA 78    LD    (78E4Hl,HL       Save increase value
202E  32 El 78    LD    !78E1Hi,A        ;Set Auto Flag
2031  El          POP   HL               Load {Start Value
2032  2           LD    78E.2H), HL.     ;and save
2035  Cl          PüP   BC               ;Get Rebound Address from Stack
2036  C3 J3 LA    JP    1A33H            to the main loop
```

if # MY# k # lt # # i # i l #it%# #f% lt

IF - Statement

condition query

```
20839  CD 37 23    CALL  2337H            Evaluate {condition expression
20JC   7E          LD    A, (H)           ;Load Characters
203D   FE 2C       CP    ' ı              = Coma?
203F   CC 78 1D    CALL  Z, 1D78H         ;yes, next character
2042   FE CA       CP    0CAH             ▬
2044   CC 78 1D    CALL  Z.1D78H          ;yes, next character
2047   2B          DEC   Hl…              {Programme pointer f
2048   IT          PUSH  HL               ;and on the stack
2049   CD 94 09    CALL  0994#4           {Result1s =}? (not fulfilled!)
204C   El          PüP   HL               Reload the programme pointer
2040   28 07       JR    1, 205H          ija to ELSE - Execution
```

THEN

```
204F   07          RST   10H              ;next character
2050   DA C2 1E    JP    C,1EC2H          ;digit? Run yes jump
2053   C3 SF 1D    JP    1D5FH            neim" next statement,
```

ELSE

```
205    16 01       LD    D,1              {Shift counter = f
2058   CD 05 1F    CALL  1F05H            Find next statement,
                                          if IF, verse, counter + 1
2085   B7          0R    A                ;End of Line?
205C   approx      RET   or               {finished, know ELSE
205D   07          RST   10H              ;next character
205E   FE 95       CP    95H              = aSE token?
2060   20F        JR    NZ,2058H         ;no, continue searching
2862   15          DEC   D                real ELSE?
                                          ;(nesting counter - 1          =)
2063   20F3        JR    NZ,2058H         ;no, search 111tab
2865   18 ES       JR    204FH            {yes; like THEN
```

LPRINT - Amue1 sung

Print Output

```
2 3   LD          ris 1              ;Output Flag = Printer
2 €   LD          1789CHi,A
2 C   JP          209BH              continue at PRINT
0 3
```

****

****

PRINT Statement

Output to screen

```
2 C   CALL        79CAH              3RA!-EreI ter ung5au5gang
2 F   CP          '3°                PRINT 3 2
2 2   JR          NZ.208FH           No!
0 0
2 C   CALL        2B01H              ;Evaluate position expression
0 D                                  Value (&32768) 1n DE, MSB in A
2 F   CP          2                  Position} 511?
2 D   JP          NC,1E4AH           Yes, FUNCTION CODE - Error
2 I   PUSH        HL                 {Programme pointer on stack
2 2   LD          HL,7000H           ;Load screen start address
2 1   ADD         HL, DE             ;Add Position
2 2   LD          (7820HL,HL         ;Save as new cursor address
2 7   LD          A,E                ;Position of the cursor in line erm.
2 E   AND         1FH                i= 5 last bits d, cursor address
2 3   LD          (7B8A1)9           ;Save as new cursor position
2 E   POP         HL…                ;Load the programme pointer
2 CF  RST         8                  A comma follows?
2 2   DEFB        '
2 R   CP          'II'               ;cassette output?
2 2   JR          NZ,21119BH         No, continue
2 C   CALL        3B58H              ;Write header on cassette
2 3   LD          A, 88H             ;Output flag on cassette
21 3  LD          (789CHJ,A
2 2   DEC         HL…                {Programme pointer- 1
2 D   RST         18H                } Next character. Instructor end?
2 C   CALL        1, 2FEH            ;yes, output CR
2 C   JP          Z,2169H            ;and finished
2 F   CP          0BFH               i= USIN6 - Token?
20 C  JP          Z,2CBDH            yes, formatted output
20 F  CP          IBCH               = TAB token?
2 C   JP          Z,2137H            Yes!
2 I   PUSH        HL…                {Programme pointer on stack
0 F
```

```
20AU  FE 2C          CP      ' ',          {Kma?
20B0  CA 08 21       JP      Z,2108H       Yes, to the next TAB position
20B3  FE 3B          CP      ';'           Semicolon?
20B5  CA 0C 3B       JP      Z,3B0CH       wait until all characters are output.
                                           ;Continued at 2164H
20B8  C1             POP     R             Load programme pointer
20B9  CD 37 23       CALL    2337H         ;Evaluate Expression
201\C IT             PUSH    HL            ;Programme pointer to stack
20BD  E7             RST     20H           ;Test data type
201\E 28.32          JR      1, 20F 2H     String? Yeah, jump
20C0  CD BD 0F       CALL    0FBDH         No. Convert values to String
20C3  CD 65 28       CALL    2865H         String in Cache and X
20C6  CD CD 79       CALL    79CDH         ;RAM expansion out
20C9  2A 21.79       LD      HL, (7921H)   ;Load string pointer from X
 2Cc  JA 9C 78       LD      A, (789CH)    ;Load Output Flag
20CF  B7             OR      A             and test
2D0   FA E9 20       JP      11.20E9H      ;Cassette? yes-no formatting
20D3  28 08          JR      Z,20DDH       ;screen? yes jump
2005  34 9B 78       LD      A, (789BH)    ;Load PHD Location
2000  86             AD      A, (HL)       ;+ String Length
 2D9  FE 84          CP      84H           ;> Line Length (132)?
20DB  18 09          JR      20E6H         ;continue at 20E6H
28DD  34 9D 78       LD      A, (7890H)    ;Load Screen Line Length
                                           ;(initialised at 64.)
20E0  47             LD      B,A           ;in B
20E1  3A From 78     LD      A, (78A6H)    ;Load cursor position in line
20E4  86             ADD     A, {HL.)      }+ String Length
28E5  B8             CP      B             j} Line length (64) ?
20E6  DA FE 20       CALL    NC,20FEH      Yes, issue Carrage Return
20E9  CD AA 28       CALL    28AAH         ;Output String
20EC  3E 20          LD      A4,' '        ;then a space
20EE  CD 2A 03       CALL    032AH
20F1  B7             0R      A             Z=0, this will n. Command overspr.
20F2  CC AA 28       CALL    Z.28AAH       ;Print String
20F5  Et             POP     HI...         ; Load programme pointer
20F6  C3 9 20        JP      209BH         {more!

Check whether the cursor is at the beginning of the line.

20F9  CD 1C JB       CALL    3B1CH         ;Load cursor position
20FC  B7             CIR A                 5=8?
20FD  CB             RET     0r            {yes   back

      issue carriage return

20FE  3E 6D          LD      A, 8DH        i Load CR Code
```

```
2100  CD 2A 03        CALL    032AH           and spend
2103  CD D0 79        CALL    79D0H           RA} extension5sau5gang
2106  AF              COLOUR  A               A + Reset Flags
2107  C9              RET

                        ',' evaluate
2108  CD D3 79        CALL    79D3H           { RA!! Extension Output
210.B 3A 9C 78        LD      (A, 4789CH)     ;Load Output Flag
210E  8/87            OR      A               Screen or printer?
210F  F2 19 21        JP      P.2119H         Yes!
2112  3E 2C           LD      ' ,'            Record iK011111a on cassette
2114  CD 2A 03        CALL    032AH
2117  18 4.B          JR      2164H           Continue at 214H
2119  28 0.B          JR      Z,2123H         {Screen? yes - jump
211B  3A 9.B 78       LD      A, 1789.BH1     ;Header position< last tab position?
211E  FE 70           CP      70H             i (: 112)
2120  C3 2B 21        JP      212BH           Continue at 212BH
2123  3A 9E 78        LD      A, (789EH)      Load last tab position      (48)
2126  47              LD      R,A             ;in .B
2127  3A AE 7A        LD      A, (78AH4)      ;Load cursor position (unbuffered)
212A  BB              CP      .B              ;< last tab position ?
212.B D4 FE 20        CALL    N, 20FEH        no, issue carriage return
212E  30.34           JR      NC, Z164H       and further
2130  D 10           SU.B    16              ;cursor position - 16 to< 0
2132  30FC            JR      NC,2130H
2134  2F              CPL                     i= Number of digits -1
2135  18 23           JR      215AH           Print spaces

                        Evaluate TAB
2137  CD 1.B 2.B      CALL    2B1.BH          Evaluate Expression
                                              ;all. Value (256) in A
213A  E6 7F           AND     7FH             ;Delete Bit 7 (Aax 127)
213C  5F              LD      E,A             {in E
2130  CF              RST     8               Follows a ')' ?
213E  29              DEFB    ')'
21JF  2               DEC     HL              {Programme pointer- 1
2140  E5              PUSH    HI.             ;and on the stack
2141  CD D3 79        CALL    79D3H           ;RAN expansion output
2144  3A 9C 78        LD      A, (789CH)      Load Output-F
2147  B7              OR      A               ;and test
2148  FA 4A 1E        JP      t, 1EAA         ;Cassette? FUNCTION CODE Error
214B  CA 53 21        JP      ,2153H          Screen? yes jump
214E  3A 9.B 78       LD      A, (789.BH)     ;Load PHD Position
2151  18 03           JR      2156H           Continue on Screen
```

```
2153  3A From 78        LD    A, (78ABH)      ;Load cursor position
215b  2F                CPL                   Form {1 complement
2157  83                ADD   A,E             ;+ Tab Value
2158  30 0A             JR    NC,21b4H        ;already reached or exceeded
215A  JC                INC   A               + 1
215B  47                LD    ,A              ;= Number of spaces to insert
215C  PER 20            LD    A,'             ;Print spaces
215E  CD 2A 03          CALL  032AH
21b1  05                DEC   B               ;counter - 1
2162  20 FA             JR    NZ.215EH        i=0? no - next space

                        Next PRINT - subexpression

 216  El                POP   HL              ;Load the programme pointer
21b5  D7                RST   10H             ;Address next character
21bb  C3 A0 20          JP    20A0H           ;and back

                        final query

21b9  3A 9C 78          LD    A, (789CH)      {Output Load Flag
21bC  00 00 00          DEFB  0.0.0.0         i4 x NOP
2170  AF                XOR   4               Output flag on
2171  32 9C 78          LD    (789CH),A       {Set Image Scale
2174  CD BE 79          CALL  79BEH           {RAM Expansion Output

                        %%%% t i ti MM # t Mr }

                        Text Definition

2178  3F 52 45 44 4F    DEFM  '?REO'
217D  O 08              DEFW  000DH

                        % t%%%%%%%%%%%% i # t}} i %%Hi Et%

                        Error parsing data

217F  3A EN 78          LD    A, (78DEH)      ;DATA flag set?
2182  B7                OR    A
2183  C2 91 19          JP    NZ, 1991H       ;yes, SYNTAX ERROR in DATA statement
218b  3A A9 78          LD    A, (7BA49H)     Input Cassette?
2189  B7                OR    A
218A  LE 2A             LD    E.2AH           Error Code in E
218C  CA A2 19          JP    1, 1942H        Yes, BAD FILE DATA - Error
218F  C1                POP   BC              keystroke, load buffer pointer,
2190  21.78.21          LD    LT...2178H      ;Text '?REDO'
2193  CD A7 28          CALL  2847H           and spend
21%6  2A E6 78          LD    HL, (7E6)       {Current forecast1ger in HL
2199  9                 RET                   Resume Input
```

```
                ************************************************************************

                INPUT Statement

                Import Data
2  CD           CALL          2828H                {Direct Command?
                                                    yes, ILLEGAL DIRECT OPERATION
2  7E           LD            To HL.)              ;Load Characters
2  CD           CALL          79D6H                {RA-Eruei terun9saus9ang
2  D6           SUB           '#'                  Read Cassette
2  32           LD            (78A9H1,A            ;INPUT flag difference          (0
2  7E           LD            To (Hi.)             ;Load Characters
2  20           JR            NZ,21C9H             No cassette!
1  2

                Cassette Scan

2  CD           CALL          3I168H               ;Find File on Cassette
2  IT           PUSH          HL                   ;Programme pointer to Stark
2  06           LD            B,0FAH               max. 250 1 oak
2  2A           LD            HL,(78A7H1           ;Address I/O buffer
2  CD           CALL          3B88H                to read a byte
2  77           LD            H)+A                 ;Transfer to buffer
2  23           INC           HL                   ;Buffering + f
2  FE           CP            liIDH                End of sentence?
2  28           JR            Z,21BDH              Yes!
2  11           DJNZ          21B2H                iCounter - 1 = 0?
2  2B           DEC           HL                   Yes, end of sentence Mark with
2  36           LD            (HL),@
2  00           DEFB          0.0.0                3 x NOP
2  2A           LD            HL, (78A7H1          ;Address buffer start
2  2B           DEC           HL                   {Buffer pointer! Byte Before
2  18           JR            21DAB                Continue at 21EBH
1  22

                Keyboard Scan

2  01           LD            BC.21DBH             Set {Return Address
€  C5           PUSH          BC
2  RD           CP            ,.                   with previous text output?
2  Clil         RET           NZ                   ;no, .continue at 21DBH
2  CD           CALL          2866H                ;Cache text u.                  X
2  CF           RST           8                    Is a semicolon following?
2 J             DEFB          1
2  IT           PUSH          li.                  {Programme pointer to stack
2  CD           CALL          28AAH                ;Print Text
2 E             POP           li.                  ;Load the programme pointer
2  C9                RET                           continue at 21DBH
1
```

```
21DB   IT          PUSH   HL          ;Programme pointer to stack
21DC   CD B3 1B    CALL   1BB3H       ;'2° print and one line 1n
                                      Import the input/output buffer
21DF   C1          POP    BC          {Programme pointer in B
21E0   DA BE 1D    JP     C, 1DBEH    BREAK? yes - jump
21E3   23          INC    HL          {buffer pointer      1.
21E4   7E          LD     A, (H1)     {Load Cards
21E5   B7          OR     A           ;text?
21E6   2B          DEC    H1          ;Buffer pointer back in     Character
21E7   CS          PUSH   BC          }Stack Programme Pointer
21E8   CA 04 1F    JP     , 1F0H      no text, INPUT instruction. pass
21EB   32c         LD     (H11,','    ;Set Koama to first character
21ED   18.05       JR     21F4H       Continue at 21F4}

                   RED Statement

                   Read data from the programme text
21EF   IT          PUSH   HI.         programme pointer to stack
21F0   2A FF 78    LD     HL, (78FFH) ;DATA - Pointer in H1
21F3   F6 Af       0R     OAFH        ;DATA - Set Flag
21F4   Af          XOR    A           ;DATA - Delete Flag
                                      ;Warning: Redefinition of 21F4H
21FS   32 EN 78    LD     (78DEH),A   ;DATA - Save flag
21F8   E3          EX     (SP),HL     ;Buffer/DATA pointer to stack
                                      ;Load programme pointer
21F9   18 02       JR     21FDH       Continue at 21FDH

                   Next Variable

21FB   CF          RST    8           Coma next?
21FC   2C          DEFB   ',
21FD   CD 0D 2     CALL   260DH       ;Find variable in var. table
                                      ;Var.Tab.Address in DE
2208   E3          EX     (SP),HL     ;Programme Pointer to Stack
                                      Load buffer pointer
221111 D5          PUSH   EN          ;Var.Tab. Address on stack
2202   7E          LD     A, (H11     ;Load characters from the buffer
2203   FE 2C       CP     ',          {= Comma?
2205   28 2        JR     Z.222DH     Yes, continue

                   Buffer empty (no',')

2207   YES EN 78   LD     A,(78DEH1   ;DATA - Flag set?
2204   B7          OR     A
220B   c2 9 22     JP     NZ.2296/i   ija, find next DATA statement
2211JE YES A9 78   LD     A, (78A9H)  ;Input from cassette?
```

```
2211  B7           OR     A
2212  1E 06        LD     E.6          ;Error code in E
2214  CA A2 19     JP     Z, 1942H     yes, OUT OF DATA - err
2217  3E 3F        LD     '?'          ;Keyboard: '?' spend
2219  CD 2A 03     CALL   032AH
221C  CD B3 1B     CALL   1BB3H        Re-enter with '2
221F  Dl           POP    EN           ;Load Var. Table Address
2228  Cl           POP    BC           ;Programme pointer in BC
2221  DA BE lD     JP     C, 1DREH     BREAK? yes - jump
2224  23           INC    HI.          ;Buffer Pointer   1. Character
2225  7E           LD     , (HL)       Load i
2226  B7           OR     4            ; End of line?
2227  2B           DEC    HI.          ;buffer pointer before 1,
2228  C5           PUSH   BC           {Programmer on Stack
2229  CA 04 LF     JP     Z,1F04H      Yes, skip the remaining input,
                                       without changing variable values
222C  D5           PUSH   EN           iVar,Tab address back on stack

                   Decode Input

2220  CD DC 79     CALL   79DCH        { RAM expander
2230  E7           RST    20H          ;Test variable type
2231  F5           PUSH   AF           Save Type Flag
2232  208 19       JR     NZ.224DH     {numerically? Yeah, jump

                   Retain String

2234  D7           RST    10H          ;Next character buffer pointer
2235  57           LD     D,A          as a separator in D and B
2236  47           LD     B,A
2237  RD 22        CP     ,.           ;Quotes?
2239  28 85        JR     1, 2240H     ;yes, use ''' as separator
223B  16 3A        LD     1 .          ;no ':' and',' as separator
223F  2B           DEC    H            ;Buffer I Byte Back
2240  CD 69 28     CALL   2869H        ;String in cache and X

                   Save New Variable Value

2243  Fl           POP    AF           Type Load
2244  EB           EX     DE,HL        ;buffer pointer in DE
2245  21 SA 22     LD     HL.225AH     {return address in hl
2248  EJ           EX     (SP),HL      ;with Var.Tab.-Adr on stack swap.
2249  D5           PUSH   EN           ;Buffer pointer to stack
224A  CJ 33 1F     JP     1f33H        {leap in LET and then 225AH

                   Number in X taken

224D  D7           RST    10~          ;Address next character
```

```
224E  F1              POP   AF          ;Load Type Flag
224F  F5              PUSH  Af          and back to Stack
2251! 1.1 43 22       LO    BC, 2243H   ;Return address to stack
2253  C5              PUSH  BC
2254  DA bC l!E       JP    C, ECH      Integer and one.Accuracy?
                                        ija, convert string, then 2243/
2257  D2 5 0E         JP    NC, L!EbSH  Dope. Accuracy? um.+ then Z243
225A  23              DEC   Hl...       {Buffer Pointer I
225B  07              RST   10H         ;next character. 0B or':'      ?
225C  28 l5           JR    1.2263H     Yeah, line end!
225E  FE 2C           CP    ,           Coma?
220   C2 7F 21        JP    NZ.217FH    ;no, error
2263  E3              EX    (SP),HL     {Programme pointer m. buffer
                                        Exchange to de!t Stack
224 2                 DEC   Hl...       {Programme pointer- !
225   07              RST   10H         ; next character,= End of
22    C2 FB 21        JP    NZ,21FBH    3 no, next, n, variables

                      no other variables

2269  D1              POP   EN          ;buffer pointer in DE
226A  00 00 08 0 08   DEFB  0.0.0.0.0   5 NOP
 22F  3A EN 78        LO    A,(78DEHl   Load iDATA Flag
2272  B7              OR    A           1 set?
2273  EB              EX    DE,Hl...    ;HL buffer pointer,Progr,pointer-DE
2274  C2 9b 1D        JP    NZ, 1D9bH   ;Buffer pointer as DATA pointer sp.
                                        Programme Pointer in Hl...,
2277  D5              PUSH  EN          ;Programme pointer to stack
2278  CD DF 79        CALL  79DFH       ; RAM expansion output
227B  B               OR    (Hl...)     ;End of line in buffer?
227C  21.86.22        LO    HL.228bH    ;Text '?EXTRA IGNORED' address
227F  C4 A7 28        CALL  NZ,28A7H    nin, output text
2282  El              POP   Hl...       Load {Programme pointer
228J  C3 69 21        JP    21b9H       ;Output flag on screen,finished

2286  3F 45 58 54     DEFl't '?EXTRA
      52 41 28 49              IGNORED'
      47 4E 4F 52
      45.44
2294  0D 8            DEFW  000DH

                      Find Next DATA Statement

2296  CD l!5 1F       CALL  1Fl!SH      Find {End of Statement
2299  '7              OR    A           i= End of Line?
229A  2112            JR    NZ,22AW     {no!
```

```
229C  23          INC   HL          Yes, programme end?
229D  7E          LD    A, (HL)     (Row pointer = W00)
229E  23          INC   HL
229F  B6          OR    (HL)
22A   1E          LD    E.6         ;Error Code in E
22A2  CA A2 19    JP    Z,19A2H     Yes, OUT OF DATA Error
22A5  23          INC   tL          ;Load Line Number
22A6  SE          LD    E, (HU
22A7  23          INC   HL
22AB  5           LD    D, (HU
22A9  EB          EX    DE,HL       1n H
22M4  22 DA 78    LD    (78DAH1,HL  and save as DATA-ZNr
22AD  EB          EX    DE,HL       lei lnumner neither in DE
22NE  D7          RST   10H         next character from programme text
22AF  FE 8B       CP    BBH         ;DATA - Token?
22B1  20 E3       JR    NZ, 22#     {no, continue searching
22B3  C3 2D 22    JP    222D        ;Read more
```

4 % # MM

NEXI - **Anei su9**

FOR—NEXT loop repetition

```
22B6  11 0 08     LD    EN, 0       ;Var.Tab Address = Set
                                    {(for NEIT without variable)
22B9  C4 0D 26    CALL  NZ.260DH    more signs? yes - variable
                                    Search, Var.Tab Address in DE
22BC  22 DF 78    LD    (7BDFH),HL  { Save Prgr more in more detail
22BF  CD 36 19    CALL  1936H       ;next stack, or loop
                                    search with correct run variables
22C2  C2 9D 11    JP    NZ, 199DH   ;not found, NEXT WITHOUT FOR
22C5  F9          LD    SP,HL       by stack correction all daza.
22c   22 E8 78    LD    (78EBH),HL  ;nested loops removed.
22C9  D5          PUSH  EN          ;Var.Tab.-Adr the loudvar. on St.
22CA  7E          LD    A, H.)      ;Load increase flag
22CJ  23          INC   HL          {stack pointer + 1
22cc  FS          PUSH  AF          ;Rise flag on stack
22CD  D5          PUSH  EN          ;Var.Tab Address on Stack
22CE  7E          LD    ed M.)      ;Load Type Flag
22CF  23          INC   HL          stack pointer + 1
22Di  B7          OR    A           —
2201  FA EA 22     JP    K.22EAH     {no! - jump
```

Simple Accuracy Volume

```
2204  CD M1 89    CALL  19B1H       Increased in I
```

```
22D7  E3          EX    (SP) ,HL     iVar.Tab,Load Address
                                     stack pointer
22D8  IT          PUSH  HL           Var.Tab. Address back to stack
22D9  CD 0B 07    CALL  870RH        Run Variable+Increase Value
22DC  El          POP   HL          Var! Load Tab Address
22DD  CD CB 09    CALL  09CBH        Save new value d, Laufvar
22E0  El          POP   HL          Load stack pointer
22E1  CD c2 09    CALL  09C2H        ;Load to Y
22E4  IT          PUSH  Hl           ;stack pointer to stack
22ES  CD 8C 0A    CALL  0A0CH        See the end value in the run
22E8  18 29       JR    2313H        we1 ter at 2313%

                  Integer as Run Variable

22EA  23          INC   HL           ;2 Above.  Skip stack levels
22EB  23          INC   HL
22EC  23          INC   Hl
22ED  23          INC   HL
22EE  4E          LD    c, (HL)      Increase value in BC
22EF  23          INC   HL           {stack pointer + 1
22F0  4           LD    B, (HL)      (i111SB)
22Fl  23          INC   HL           Cursor + 1
22F2  EJ          EX    (SP),HL      Stackze1ger on the stack
                                     Var, Tab. number of the run    country
22F3  5           LD    E, HL)       ;Load value of the run
22F4  23          INC   Hl
22FS  5           LD    D, (Hl.
22F6  IT          PU5H  HL           Var.Tab,-dr + 1 to the stack
22F7  9           LD    L,C          Increase in Hl.
22F8  B           LD    H,B
22F9  CD 02 0B    CALL  0RD2H        Expiry Var.+Increase Var. in   X
22FC  YES AF 78   LD    A, (78AFH)   Type in X = one.
22FF  FE 4        CP    4            ;!overflow)
2301  CA B2 07    JP    1.07B2H      Yes, OVERFLOW Error
2304  EB          EX    DE,HL        ;new run variable in DE
2395  El          POP   HL           Var.Tab. -Adr + I Load
2306  72          LD    HL),D        Add {and New Value
2307  2B          DEC   HL
2308  73          LD    (HL),E
23089 El          POP   Hl…          Load {stack pointer
230A  D5          PUSH  EN           {new value of Laufvar,   on stack
230B  5           LD    E, {H)       Load {End Value
230C  23          INC   Hl…
230D  5           LD    D, {HL)
230E  23          INC   HL           {stack pointer + 1
```

```
230F   E3              EX     (sP),HL        stack pointer
                                              ;Load new value of run variables
2310   CD 39 0A        CALL   0A39H          Run Varable vs Endured
2313   El              POP    HL             Load {Stackziger
2314   Cl              POP    BC             ;Load increase flag
2315   90              SUB    B              ;Comparison result with increase
                                              ;Link Flag
2316   CD C2 09        CALL   09C2H          Line number and start pointer
                                              Load #in DE and B
2319   28 09           JR     Z,2324H        ;Loop complete? yes - jump
231B   EB              EX     DE,HL          {Line number in H
231C   22 A2 78        LD     (78A2Hl,HL)    as current ZNr. save
231F   69              LD     n              Loop Beginning Pointer in HL
2320   8               LD     H,B
2321   CJ 1A 1D        JP     DlAH           ;Cycle through again

                       Schlei te terminated
2324   F9              LD     SP, HL         by stack correction loop
2325   22 E8 78        LD     (78EBH>,HL     ;Remove from stack
2328   2A DF 78        LD     HL, (78DFH)    Load {Programme pointer
232B   7E              LD     As HL)         ;Load Characters
232C   FE 2C           CP     ','            A coma follows?
232E   C2 lE 1D        JP     1D1EH          ;no, next command
2331   D7              RST    10H            ;Address next character
2332   CD B9 22        CALL   22B9H          Edit next outer loop

                       ti With} Mttt}i % %t%Mt %f litt } #

                       Evaluate pressure

                       Ring: H = Start address in programme text
                       Exp.: X= Result
2335   CF              RST    B              ;the expression begins with
2336   28              DEFB   '('            Klaner? no - SYNTAX ERROR
2337   2B              DEC    ed             {Programme pointer - I
2338   16 8            LD     O,             ;Prio code d. last operands= 0
233A   D5              PUSH   EN             ;Prio code on the stack
2331!  O @ı            LD     , 1            ;at least 2 more bytes free?
233D   CD 63 19        CALL   1963H          ;no, OUT OF MEMORY - Errar
2340   CD 9F 24        CALL   249FH          ;Analyse operands and in X
2343   22 F3 78        LD     !7BFJHl,HL     Save {programme pointer
  23   2A FY 78        LD     LT, (78FJH)    Load programme pointer
2349   Cl              POP    BC             Pri. Load Code in B
```

```
234A  7E          LD    A, (HL)          {next character from Arabic
234B  1 00        LD    D,0              ; Operator code= i set
234D  D6 D4       SUB   D4H              Comparison operator? {  ) = ({)
234F  38.13       JR    C,2364#          No!
2351  FE 03       CP    3                <Token D4, D5 and D)
2353  30 8F       JR    NC,2364H         {no!
2355  FE 01       CP    1                ;for ')' Set Carry
2357  17          RLA                    Move Bit Left
                                         j( > - 1 !  = - 2 1 ( - )
2358  AA          XOR   D                corresponding, Bit ia operator
2359  BA          CP    D                Was previously set sclloo?
235A  57          LD    D,A              i.e. the same operator 2x>
235B  DA 97 19    JP    C,1997H          yes - SYNTAX ERROR
235E  22 D8 78    LD    178D8H),HL       { Save Image Pointer
2361  D7          RST   UIH              ;Load next character
2362  18 E9       JR    234DH            ;and investigate
Z34   7A          LD    A, D             jOperator-Codl! > 1?
2345  B7          OR    A                (comparison operator found)
23    C2 EC 2J    JP    NZ,2JECH         Yes!
23%   7E          LD    A, (s)           ;Load Characters
 234  22 D8 78    LD    (78D8%) ,H       Save { programme pointer
236D  D6 CD       SUB   IICDH            one of the other operators?
236F  Profit      RET   C                ;+ - f / ff AMD OR?
23711 RD 17       CP    7
2372  DI          RET   NC               No!
2373  5F          LD    E,A              ;Operator code in E
2374  3A AF 78    LD    A, (7FH)         Reads in X in String ?
2377  D6 OJ       SUB   3
2379  B3          OR    E                ;and'+' - Operator?
237A  CA 8F 29    JP    Z.298FH          yes, string linking
237D  21 9A 18    LD    H, 1894H         Table of prices codes adress.
2381  19          AD    H, DE            + Opera atorde
2381  78          LD    M,B              {last priority in A
2382  56          LD    D, (H)           new Prio. from table in D
2383  BA          CP    D                last prio. >= IIIIUII Pri0. ?
2384  D0          RET   NC               yes, perform last operation
2385  CS          PUSH  BC               No, last prio. on stack
2386  1 4 23      LD    BC,2346H         Adr. t. n. Stack Operands
2389  C5          PVSH  JC
238A  7A          LD    A, D             {New priority n A
238B  FE 7F       CP    7FH              ;= 7FH !Operator= ff)  ?
238D  CA D4 23    JP    1.234H           $Jas Continues at Z
2398  FE 51       CP    51               {Operator r oter 0R?
```

```
2392  DA E1 23        JP     , 231H          yes; continue be1 23E1H

                        operands for +, -, +and i to the stack

2395  21 21 79        LD     HL.7921H        ;X address in HL
2398  B7              OR     A               Delete Carry
2399  JA AF 78        LD     A,              Load iT!:fp Code
239C  3D              DEC    A               ;Type Code - 3
239D  30              DEC    A               = string 7
239E  3D              DEC    A
239F  CA F6 0A        JP     l,0AF#          ; yes, TYPE MISl'IATCH Error
23A2  4E              LD     C, HL)          {Load Operands
23A3  23              INC    HL              iX Address + 1
234   4               LD     B, (HL)         n. Wat
23A5  CS              PUSH   BC              and aut Stack
234   CS 23           JP     M.23CSH         {Done at Integer!
23A9  23              INC    HL              otherwise Z load more bytes
234   4E              LD     C, HL)
23AB  23              INC    Hl...
23AC  46              LD     B, HL)
23AD  CS              PIJSH  BC              and on the stack
23AE  F5              PUSH   AF              ;Type flag on the stack
23AF  B7              OR     A               -
23B8  E2 C4 23        JP     P0.23C4H        Yeah, done
23B3  Fl              POP    A               Load type flag again
23B4  23              INC    HL              i X Address + 1
23B5  38.03           JR     C,23BAH         Ouch Stack? yes - jump
23B7  21 1D 79        LD     H... 791DH      Load iLSB X Address
23BA  E               LD     C, H)           i2 Load more bytes
23RB  23              INC    HL
11:BC 4               LD     B, (H.)
11:BD 23              INC    HL
11:BE CS              PUSH   BC              42 bytes on the stack
11:BF 4E              LD     C, I..)         and load another 2 bytes
23CI  23              INC    HL
23C1  4               LD     B, (HI.)
23C2  CS              PUSH   BC              on the stack
23C3  06              DEFB   06H             1LD 3,FIH  Dung instruction
                                             skips the POP
23C4  Fl              POP    AF              Load type flag (eint.Gen.>
2305  C 83            ADD    A3              ;Calculate Type Code
2JC7  4B              LD     C,E             Operator code in C
23C8  47              LD     MA              ;Type Code in 11
23C9  C5              PUSH   JC              ;pack on dl!TI stack
2JCA  1 8 24          LD     JC,2416H        lAdresw for the
```

```
23CD  CS          PUSH  C            ;Operations on the stack
23CE  2A D8 78    LD    HL, (7808H)  { Load Pogr Pointer
23D1  CJ YES 23   JP    233AH        ;next operand

                  Operands for potentiation on the stack

23D4  CD BI 0A    CALL  0AB1H        X surrounded in simple precision.
23D7  CD AA 09    CALL  @944}        ;X to the stack
23DA  01 F2 13    LD    BC, 13F2H    power calculation address
23DD  16 7F       LD    D,7FH        ;new prime code = 7F
23DF  18EC        JR    23CDH

                  Operands for AND and OR on the stack

23E1  DS          PUSH  EN           ;Operator Code on Stack
23E2  CD 7F BA    CALL  0A47FH       Convert operand to integer
23E5  D1          POP   EN           Load Operator Code
23E6  IT          PUSH  HL           ;Operand to the stack
23E7  01 E9 25    LD    BC, 25E9H    !Address for 'AND' and 'OR'
23EA  18 E1       JR    23CDH

                  Process comparator operators

23EC  78          LD    AB           had last operator higher
23ED  RD 6        CP    64}          or equal priority?
23EF  D0          RET   NC           yes, calculate last operation
23F0  C5          PUSH  BC           last priority on stack
23F1  D5          PUSH  EN           ;Operator Code on Stack
23F2  11 4 6%     LD    EN, 6404#    ;Priority in D, comparison code in
2JF5  21 B8 25    LD    HL,25:BBH    ;Edit URL
23F8  IT          PUSH  HL           ;Stack Comparison Result
2JF9  E7          RST   20H          ;Test data type
2JFA  c2 95 23    JP    NZ, 2395#    numerical? yes Operands on Stack
2JFD  2A 21.79    LD    HL, (7921H}  5tring! String pointer to stack
2400  E5          PUSH  HL
2401  01 8C 25    LD    BC.258CH     ;URL Load String Comparison
2404  18 C7       JR    23CDH        and on the stack

                  Run Operations

2406  C1          POP   BC           ;Load Operator Code and Type
2407  79          LD    A,           ;Save Operator Code
2408  32 B0 78    LD    (78BMH), A
240B  78          LD    A, B         Type in A
240C  RD 88       CP    8            1. operand dopp,      ?
240E  28 28       JR    Z,2438H      Yes!
 241  YES AF 78    LD    A, (78AFH}   ;Type of 2. Load operands
2413  FE 08       CP    8            = Double accuracy?
```

```
2415  CA o 24          JP     Z,2460H        1yes'
2418  57               LD     D,A            ;Type of 2. Operands in D
2419  78               LD     A,B            ;Type 1. Operand in
241A  FE 04            CP     4              {= simple detail?
241C  CA 72 24         JP     Z,2472H        Yes!
241F  7A               LD     A,D            ;Type 2. Load Operand
2420  FE 083           CP     3              ;and test
2422  CA F6 0A         JP     Z, 0AF6H       ;String' - TYPE MISMATCH
2425  D2 7C 24         JP     NC,247CH       ;one, accuracy!     - jump

                       Integer - Execute operations

2428  21 BF 18         LD     Hl…18BFH       ;Start the Integer jump table
242B  6 00             LD     B,0            ;Add Operator Code 2x
242D  89               ADD    HL,BC
242E  69               ADD    HL,BC
242F  4E               LD     C, (HL)        ;Load explosion
2438  23               INC    Hl…
2431  4                LD     B, (Hl.
2432  D1               POP    EN             ;1. Get Operand from Stack
2433  2A 21.79         LD     Hl, (7921Hl.   2. Load operand from X
243   c5               PUSH   BC             ;Jump to stack
2437  C9               RET                   ;and jump to routine

                       1, double-precision operand

2438  CD DB 0A         CALL   0ADBH          Convert 2 operand to
243B  CD FC O9         CALL   09FCH          and transfer to Y
243E  El               POP    HL             ;1. Operand voii Stack in X
243F  22 LF 79         LD     (791FHJ,HL     First the 4 Niederu. bytes
2442  El               POP    HL
2443  22 LD 79         LD     (791DHl,HL
2446  Cl               POP    BC             ;dive. 3 bytes and the exponent
2447  DI               POP    EN
2448  CD B 09          CALL   {09B4}         also in X
244B  CD DB 0A         CALL   0ADBH          1. Operand in dopp.Gen.
244E  21.18            LD     HL, IBABH      ;Load the jump table start
2451  3A B0 78         LD     A, (7BB0Hl     Load ;Operator Code
2454  87               RLCA                  l* 2
2455  C5               PUSH   BC             ;BC secure
2456  F                LD     C,A            Operator code + 2 in B
2457  6 00             LD     B,0
2459  09               ADD    H,BC           to StartPoint. the jump tab,  Add.
245A  Cl               POP    BC             ;B Restore
245B  7E               LD     A, (HL)        ;Load explosion address
245C  23               INC    HL
```

```
245D  6b6          LD    H, (H)
245E  6F           LD    L,A
245F  E9           JP    (HL)            Run Routine

                   2. Operand = double precision

2460  CS           PUSH  BC              ;Operator code and type on stack
2461  CD FC 09     CALL  09FCH           E. Transfer operand to Y
2464  Fl           POP   AF              ;Type of 1. operand
2465  32 AF 78     LD    (7BA4FH),A      in type-rate
2468  FE 04        CP    4
246A  28 DA        JR    Z,2446H         5yes" continue at
246C  El           POP   HL              No, Integer in HL
246D  22 21 79     LD    (7921HL,HL      and in X
278   18 D9        JR    24411H          Continue at 244BH

                   1. Operand= simple precision

2472  CD BI A      CALL  0AB1H           {2, Convert operand to inf.Gen.
2475  Cl           POP   BC              {1. Operand by Stark in Y
2476  Dl           POP   EN
2477  21 5 18      LD    HL, 18B5H       ;Start address of the jump table
247A  18 D5        JR    2451H           Continue at 2451H

                   2. Operand= simple precision

247C  El           POP   HL              1. Operand (Integer)  in HL
247D  CD A4 09     CALL  09A4H           {2. Operand on Stack
2480  CD CF 0A     CALL  0ACFH           I, convert operand to inf.Gen.
2483  CD F 09      CALL  09BFH           and transferred to V
2486  El           POP   HL              ;2. Stack Operand in X
2487  22 23 79     LD    (7923H), HL.    Exp. + 1SB)
248A  El           POP   HI…
248B  22 21 79     LD    47921H)/HL      ;2 bytes LSB)
248E  18 E7        JR    2477H           Continue at 2477H
```

%%Mi% t #Ei t ## k #I # t3 Wed

```
                   Integer - Division

                   Eing: DE= dividend
                         HL = divisor
                   us9.' X    = ratio         (in simple precision)
2490  IT           PUSH  HL              ;Divisor on stack
2491  EB           EX    DE,HL           ;Dividend in HL
2492  CD CF A      CALL  0ACFH           ;and thus one.Gen. in X
2495  El           POP   HL              ;Load Divisor
2496  CD A4 09     CALL  09A4H           ;Dividend from X to stack
```

```
2499  CD CF 0A        CALL   0ACFH          ;Divisor 1111single aene. in X
249C  C3 TO 8         JP     0840H         to division with enf.     Exactly.

%%%%%%%%%%%%# # ~ k %###i %%%% kk

                      Expression Analysis Operands

                      Ring: HL = Address of operand iw, text
                      Exp.:  X= Result
249F  D7              RST    10H           ;Address next character
2448  1E 28           LD     E.28H         ;Error Code in E
24A2  CA A2 19        JP     Z, 19A2H      Statement End, MISSINS OPERAND Error
2445  DA 6C OE        JP     , OE6CH       Shake! Calculate value and X
24A8  CD 3D 1E        CALL   1E3DH         ;letter?
24AB  D2 40 25        JP     NC, 2540H     i.Yas expression variable value
24AU  FE CD           CP     0CDH          j't' - sign?
24B0  28 ED           JR     Z.249FH       yes, ignore
24B2  FE 2E           CP                   ; character = '.' ?
24B4  CA 6C OE        JP     L,0ECH        yes" Number by X, finished
24B7  FE CE           CP     CEH           ;'- sign?
24B89 CA 32 25        JP     Z,2532H       Evaluate {yes
24BC  RD 22           CP     19            {quotation mark?
24BE  CA ob 28        JP     Z,28bbH       ia. string constant in X
24C1  FE CB           CP     0CBH          ; NOT - token?
24C3  CA C4 25        JP     €1.24€4H      ;yes, execute
24C   RD 2           CP     't'           ;= " ?
24c8  CA 94.79        JP     1.7994#       ija; to RA! extension5aus9ang
24C.B FE C3           CP     0C3H          i= ERR token?
2/tCD 20 0A           JR     NZ,24D9H      ;no, 111eiter

                      ERR - Function

                      returns the last error code
24CF  D7              RST    10H           ;Address next character
2400  YES 9A 78       LD     A, (789AH)    ;Load last error code
24D3  E5              PUSH   HL            ; programme pointer to stack
24D4  CD F8 27        CALL   27FBH         ;Error code as an integer in X
24D7  El              POP    Hl...         {Reload programme pointer
2408  C9              RET                  ;finished
24D9  FE C2           CP     0C2H          i= ERL token?
24D.B 20.84           JR     NZ,24E7H      i no, continue

                      ERL - Function

                      returns the last error line
24DD  D7              RST    18H           ;Address next character
```

```
24EN  IT          PUSH  Hl          {Programme pointer on stack
24DF  2A EA 78    LD    HL, (78EAH) Load last error line number
24E2  €CD 6       CALL  @Cl         to one,gene, and to X
24E5  El          POP   Hl          Reload {programme pointer
24E6  C9          RET               finished
24E7  FE C0       CP    0C0k        i= VARPTR token?
24E9  20.14       JR    NZ.24FFH    No, continue
```

VARPTR Function

Detect variable address in the variable table

```
24EB  D7          RST   10H         Address Next Draw
24EC  CF          RST   8           Follows a ' (' ?
24ED  28          DEFB  '('
24EE  CD 0D 26    CALL  281H        ;Find the variable address
24F1  CF          RST   8           locked with')' ?
24F2  29          DEFB  ')'
24F3  IT          PUSH  Hl          Programme Pointer on Stack
24F4  EB          EX    DE,HL       ;Var.Tab Address in HL
24F5  7C          LD    ,H          {=8 ?
24F6  B5          OR    L           ;(variable not in table)
24F7  CA 4A LE    JP    Z, 1E4AH    Yes, FNCTION CODE Error
24FA  CD 9A 0A4   CALL  49/         Var.Tab.Address as Integer in X
24FD  El          POP   Hl          Reload the programme pointer
24FE  C9          RET               ;finished
24FF  FE Cl       CP    0C1H        i= USR token?
2501  CA FE 27    JP    Z.27FEH     Yes!
2504  FE CS       CP    C5H         i= INSTRt token?
2506  CA 9D 79    JP    Z.799DH     Yes" to the RA} extension output
2509  approx      CP    0C8H        ;= 11E11 tokens?
258B  CA C9 27    JP    Z,27C9H     Yes!
25E   FE C7       CP    C7H         ;= TIME token?
2510  CA 76 79    JP    Z,7976H     yes to the RA' extension au5gang
2513  FE C6       CP    0C6H        ;= POINT token?
2515  CA 32 01    JP    1.0132H     Yes!
2518  FE C9       CP    09H         i= INKEYt token?
251A  CA 9D 01    JP    1.019DM     Yes!
251D  FE C4       CP    0C4H        ;= STRINGt token?
251F  CA 2F 2A    JP    Z,2A2FH     3Yes!
2522  FE BE       CP    0BEH        ;= FN token?
2524  CA 55 79    JP    1.7955H     5yes" to the RAN extension output
2527  D6          SUB   0D7H        ;Function token?
2529  D2 4E 25    JP    NC,254EH    Yes!
```

```
                    (1 ammer expression
252C CD 35 23       CALL    2335H           inein, in klaR'llllern

252F  CF            RST     8               completed mnt )' 7
2530  29            DEFB    'T'
2531 C9             RET                     ;finishe
                                            d

                    • -° - Evaluate sign
2532 16 7           LD      D, 7DH          '-' sign priority
253-4 CD 3A 23      CALL    233AH           ;Clear Priority Expression
2537 2A F3 78       LD      HL, (78F3H)     Load {Programme pointer
253A E5             PUSH    HL              ;and on stack
253B CD 7B 09       CALL    097BH           i Result        (-1)
253E El             POP     H               Reload {programme pointer
253F C9             RET     ;finished

                    Expression Variable Value
2540 CD 0D 2        CALL    260DH           ;Variable in variable table erm.
                                            Ivar.Tab,-Adr. in DE. If not ;present, X=
                                            0, immediately back ;Prograad pointer to
2543  ES                                    stack
2544 EB             PUSH    HL              Var. Tab Address in HL
2545 22 21 79       EX      DE,HL           ;and in X as a string pointer ;string
2548 E7             LD      17921HL,        variable?
2549 C4 F7 09       HL                      ;no, load variable value in X
254C El             RST     20H             iProgramme pointer;done
C9 2540             CALL    NZ, 09F7H
                    POP     HL              Funk t in-
                    RET                     grunente
                                            evaluate and
25E 06 00           start function routines
2550 07             LD      B,0             B=0
2551 4F             RLCA    {A= (token-D7H) + 2
2552 CS             LD      C,A             ;in BC as table offset
2553 D7             PUSH    BC              ;Save the stack
2554.79             RST     10H             ;Address next character
2555 FE 41          LD      A,C             iLSB Tab Offset> 41H?
2557 38 1           CP      41H             ;(HID$, RIHTS · LEFTS)
2559 CD 35 23       JR      C,256FH         {no!
255C CF             CALL    2335H           ; 1. Evaluate Argument
255D 2C             RST     8               {is followed by a koa?
255E CD F4 0A       DEF:B   ','
2561 EB             CALL    0AF4H
                    EX      DE,HL           ;1. Argument no string? Tlt-terror;
                                            Programme pointer in DE
```

```
2562  2A 21.79      LD    HL, (7921H)    ;Load String
2565  E3            EX    (SPl,HL        Load Tab.0ffset, Stringz.aut Stk,
2566  IT            PUSH  HI.            ;Tab.Offset also back to Stack
2567  EB            EX    DE,HL          ; Programme pointer back in H
258   CD 1C 2B      CALL  2BlCH          2. Analyse Argument
                                         {6anzz.ert (25) in EN
25B   EB            EX    DE,HL          {2. Ar9, 1n HL, prog pointer in DE
€256  E3            EX    (SPl,HL        ;Load Offset, 2nd Arg on stack
256D  18 14         JR    2583H          Continue at 2583
256F  CD 2C 25      CALL  252CH          Evaluate {Argument
2572  EJ            EX    (SP>,HL        ;Load Offset, Prog.z.aut Stack
2573  7D            LD    A,L            ;LSB Tab.Offset< 0CH?
2574  FE 0C         CP    0CH            1 (S, INT, ABS, FRE,     POS)
2576  38.07         JR    C,257FH        Yes!
2578  FE 1B         CP    1BH            ;LSB Tab.Offset< lBH?
                                         ;(SQR,RND,LOG,EXP,COS,SIN,TAN,ATN>
257A  IT            PUSH  HL             ;Tab. Offset to Stack
257B  DC Bl 0A      CALL  C, 04M1H       ;yes, single-precision argument
257E  El            PP    HL             ;Reload Tab. Offset
257F  11 JE 25      LD    DE,253EH       ;Set Reversal Address
2582  05            PUSH  EN
2583  01 08 16      LD    BC, 168H       Start address of the jump table
2586  @9           ADD   HL,BC          ;+ Table Offset
2587  4E            LD    C, HL)         {Load Jump Address
2588  23            INC   HL
2589  b             LD    H, H)
258A  9             LD    L,C
258B  E9            JP    (HL)           ;Run Routine
```

##I # #i#kMMk k M tk Mi k###k}M## #k # i#Mi

String Comparison

```
258C  CD D7 29      CALL  29D7H          2. Cache Strins
                                         ;and Remove Strings
258F  7E            LD    (H)            Length 2. String in A
2590  23            INC   HL             {Address Z. String in BC
2591  4E            LD    C. (HI\
2592  23            INC   HI.
2593  4b            LD    B,<HLJ
2594  Dl            POP   EN             ;Address 1. String in DE
2595  CS            PUSH  BC             ;Address 2. String to Stack
259   FS            PUSH  AF             Length 2. String to Stack
2597  CD DE 29      CALL  29DEH          1. Cache Ctrl
```

```
                                        ;and Remove Strenqth Area
259A  Dl          POP   EN             ;Length 2. String by D
259B  SE          LD    E, (HL)        Length 1. String to E
259C  23          INC   HL
259   4E          LD    C, (HL)        ;Address 1. String in BC
259E  23          INC   HL
259F  4           LD    B, (HL)
25A0  El          POP   HL             Address Z, Str1ng 1n HL
25A1  7B          LD    A,E            {both strings empty?
25A2  B2          R     D
25A3  C8          RET   0r             ijas back with A=@ (same)
25A4  7A          LD    A,D            #Length 2. Str1n9 = 0?
2545  D6 01       SUB   1
25A7  Profit      RET   C              Yes, back mut A=FF, Cy=1,    5=1
                                       ;i.e. String 1} String 2
25AB  AF          XOR   A              {2. String = empty?
25A9  BB          CP    E
25AA  3C          INC   A
25AB  D0          RET   NC             iYes" back with A=1, l=@, y=@
                                       i.e. Strin9 1 {Strin9 2
25AC  15          DEC   D              for the string lengths - 1
25AD  1D          DEC   E
25AE  0A          LD    A, (BC)        ;Character from 1. Load
25AF  BE          CP    (H)            ;ait character from 2, string
25B0  23          lNC   HL             ;String + 1
25.Bl 03          INC   BC
25B2  28 ED       JR    Z,25A1H        both characters equal, next
25B4  3F          CCF                  ;Complement Carry
25B5  C3 O 09     JP    09MH           ;Ready flag

#%11~Mi%t%Mt~± i # Hi } Hi E

Result of comparison with the comparison operator

Add to Total
Ring: 1. 0p. } 2, 0p. - A=FF, g=l
        1. 0p= 2. 0p.- A=0      1. Op. < 2. 0p.- A=1
25B8  3C          INC   A              ;Comparison result+ 1
25B9  8F          ADC   A,A            $32 + carry
25BA  C1          POP   BC             ;Load YerEqual Operator Code
                                       iB!01=1 - >, B(11=1 - =, B(21=1 - <
25BB  AB          AND   3              ;a bit matches?
25.BC Co FF       ADD   A,0FFH         Yes, Carg = I
25BE  9F          SBC   A,A            ;and A=FF
25BF  CD 8D 09    CALL  098DH          A as integer in X
```

```
25C2  18 12          JR     25DH          ;continue at 25DbH

                     }% # # k # k # % %# %i } i t %###

                     Run NOT

25C4  16 SA          LD     D,5AH         {Not Priority 1n D
25C6  CD 3A 23       CALL   233AH         Evaluate expression with priority
25C9  CD 7F 0A       CALL   0A7FH         Convert Result to Integer
25CC  7D             LD     A,L           and invert
25CD  2F             CPL                  ; LSB)
25CE  6f             LD     L,A
25CF  7C             LD     A,H
25D0  2F             CPL                  ;(1SR)
25D1  61             LD     H,A
25D2  22 21 79       LD     (7921H) ,HL   ;in X
25D5  cI             POP    BC            ;last priority     load
2506  C3 46 23       JP     2346H         ;continue at 234bH

                     %#%it% t #%t# l i # %i%t% # M% %t#

                     Restart 20

                     tests I register data type
                     Eing: 78AF = type code
                     Exp.: A = type code -3
                             Integer: S=1, P=1,  Cg=1
                            emnt.6: g=1
                            dopp.g.: p=1
                            String: Z=1,    C=1, p=1
25D9  JA AF 78       LD     A, (78AFH>    Load Type Code
25DC  FE 08          CP     8             {double accuracy?
25EN  30.05          JR     NC,25E5H      Yes!
25E0  D6 03          SUB    3             {TgP Code - 3
25E2  B7             OR     A             ;Set Flags
25E3  37             SCF                  Carry=1 and Others
25E4  C9             RET
25E5  D 03           SUB    3             {type-ade - 3
25E7  B7             OR     A             Set Flags
25EB  C9             RET

                     %MM## i MM # #t # % t t # HE #

                     Run AND and OR

25E9  c5             PVSH   BC            ;last priority on stack
25EA  CD 7F 0A       CALL   0A7FH         2. Convert operand to integer
```

```
25ED  F1          POP   AF          ;Load priority in AF
25EE  D1          POP   EN          !! Load Operand
25EF  01 FA 27    LD    RC, 27FA    Set bounce address
25F2  CS          PUSH  BC
25FJ  FE 4        CP    46H         iAND?
25F5  20.06       JR    NZ.25FDH    Yes!
25F7  7B          LD    A,E         ;OR-Link both operands
25F8  B5          OR    L           3L.SB)
25F9  6F          LD    L,A
25FA  7C          LD    A,H
25FB  B2          0R    D           {(0SB)
25FC  C9          RET
25FD  7B          LD    A,E         {both operands UN}
25FE  AS          AND   L           4(LSB)
25FF  6F          LD    L+
200   7C          LD    A,H
2601  A2          AND   D           ;(+SB)
2602  C9          RET
```

Mil#} Mi #k it} i # t i # l # kt

Provide **More** Arguments for DIM

```
2603  2B          DEC   HL          { Prgr ammziger- 1
2604  D7          RST   10H         ;Address next character
2605  approx      RET   or          {Instructor? yes-back
268   CF          RST   8           {Following a comma?
2607  2C          DEFB  ' '
                                    1
```

FFF

DIN - Statement

Configure matrices

```
208   01 03 26    LD    BC,2603H    ;Return address f.        n,
260B  CS          PIJSH BC
260C  F6 AF       OR    AF          ;D111 flag
```

FFF

Find and set up variable in table if not available

Ring: H = Variablennane address

Exp.: DE= Address in variables table

```
260   AF          XOR   A           ;D111 Clear Flag
                                    Warning: 260D Redefined
```

```
20E   32 AE 78          LD     (78AWl,A       Save iDIM flag

                        Determine Name

2611  46               LD     B, (Hl)        il,sign of the Var.Name in B
2612  CD 3D !E         CALL   !E3DH          Rest but?
2615  DA 97 19         JP     C,1997H        5no, SYNTAX ERROR
2618  AF               XOR    A              ;C (2. Characters) Delete
2619  4F               LD     C,A
261A  D7               RST    10H            ;Load next character
261B  38 05            JR     C,2622H        Number? yes jump
261D  CD 3D 1E         CALL   1E3DH          ;letter?
26208 38 89            JR     ,262RH         ;no, name only 1 letter
2622  4F               LD     C,A            ;2. Characters in
2623  D7               RST    10H            ;Load next character
2624  38 FD            JR     C,2623H        ;digit? yes-pass
262   CD 3D 1E         CALL   1E3DH          ;letter?
2629  30 FB            JR     NC, 2623~      ;yes, pass

                        Find Type

262:S 11.52.2          LD     DE,2652H       ;Set return address
262E  D5               PUSH   EN
262F  16.82            LD     D,2            ;Type Code =
2631  RD 25            CD     '%'            ;n. Character= 'Y.'
2633  addrox           RET    or             Yes, done
23    14               INC    D              ; Type Code =
2635  RD 24            CP     '$'            ;n. Character='$'  ?
2637  CB               RET    or             ija  finished
2638  00 00 00 00 0%0  DEFB   0.0.0.0.0.0.0.0.0  ;9xNOP
      00 00 80 0e

                        Remove type code from table

2641  7B               LD     + D            Position of 1.    in
2642  D 1              SUB    'A'            Find Alphabet
2644  E6 7F            AND    7FH            ;Delete Bit 7
26    5F               LD     E,A            ;as table offset in DE
2647  16 0e            LD     D,0
2649  E5               PUSH   HL             Delete {Programme pointer
264   21 01 79         LD     HL.7901H       ;Top of type code table adress.
264D  19               ADD    HL, DE         ;+ Offset
264E  56               LD     D,!Hll         ;Load type code from table
2b4F  El               POP    HL             Reload {programme pointer
2650  2B               DEC    HL             ;-1, because no explicit type

2651  C9               RET                   Continue at 2652H
```

```
2652  7A              LD    A, D          Transfer Typcade to Type Byte
2653  32 AF 78        LD    784H),A

                      Find variable in Var1ables table

2656  D7              RST   10H           Address Next Draw
2657  3A DC 78        LD    A,(78DCH)     Independence blocked?
265A  R7              OR    A             ;(for run variable)
2658  C2 64 26        JP    NZ,2664H      Yes'
265E  7E              LD    A, {HtL)      Load Cards
265F  D6 28           SUB   '('           = '(' ?
2661  CA E9 26        JP    1.26E9H       3yes; indexed variable
2664  AF              XOR   A             Unset Indirect Lock
2665  32 DC 78        LD    (78DCH>,A
26b8  IT              PUSH  HL            {Programme pointer on stack
2669  D5              PUSH  EN            ;Type Code on Stack
26bA  2A F9 78        LD    HL, (78F9)    Load {beginning of variable table
26D   EB              EX    DE,HL
26E   2A FB 78        LD    HL, (78FBHJ)  Load {end of variable table
2671  DF              RST   18H           iAddresses equal?
2672  EI              POP   HL            ;Type Code in H
2673  28 19           JR    Z.268EH       yes, variable not found
2675  1A              LD    A, (DE)       ;Load type from variable table
276   6F              LD    L,A           ;in L
2677  BC              CP    H             ;=Type of variable sought?
2678  13              INC   EN            ; Address Var. Table + 1
2679  20 0B           JR    NZ.2686H      ;no, next variable
267B  1A              LD    A, (DE)       ;2. Load Characters from Table
267C  B9              CP    C             = 2. Characters of the variables?
267D  28 07           JR    NZ.2686H      ;no, next variable
267F  13              INC   EN            ; Address Var. Table + 1
280   IA              LD    A, (DE)       i 1. Load Characters from Table
2681  B8              CP    B             —
2682  CA CC ?         JP    Z,26CCH       Yes, variable found!
2685  3E              DEFB  13H           3LD A, 13H  Dung instruction
288   13              INC   EN            Address Var.Table to 1. Character
2687  13              INC   EN            ;Address Var. Table to Value
2688  IT              PUSH  HL            Lookup variable type on stack
289   26.00           LD    H,            ;Address of the var.table
268B  19              ADD   HL, DE        ;+ Length of type= next Eintr.
268C  18DF            JR    26DH          continue

                      Variable not included in variable table

268E  7C              LD    A,H           ;Type in A
268F  E1              POP   HI.           Load programme pointer
```

```
2690  E3          EX    (SP1,Hl     ;swap with return address
291   F5          PUSH  AF          {type aut
292   D5          PUSH  EN          ;Var.Tab end address. on stack
2693  11 F1 24    LD    DE,24F1H    ;Jump address= 24F1H?
269   DF          RST   18H         ;(from VARPTR)
2697  28 3        JR    ,2CFH       ija" next be1 2CFH
2699  11 43 25    LD    DE,2543H    ;Return address= 2543H?
€269  DF          RST   18H         ;(from Expression Analysis)
269   D1          POP   EN          ;Var.Tab, -Reload End Address
29E   28.35       JR    Z,26D5H     ija+ Continue at 26DSH

                  Configure New Variable

26A   Fl          POP   AF          Load type
2641  E3          EX    (SP1,HL     ;Return address to stack
                                    ;Load the programme pointer
2bA2  IT          PUSH  Hl          ;Programme pointer to stack
2bA3  CS          PUSH  BC          ;Variable number! to the stack
Z     4F          LD    C,A         ;Type in C
2bAS  6 00        LD    B,0         B=e, i.e. BC enth. Length of value
26A7  CS          PUSH  BC          Length on the stack
26AB  03          INC   BC          + 3
26A9  03          INC   BC          ;= total length of Var.Tab entry
26AA  3           INC   BC
264B  2A FD 78    LD    HL,(78FDHl  Initial adr. of free memory
2NE   IT          PUSH  HI          to the stack
2bAF  09          ADD   HL,BC       ;+ Total length Var.Tab entry
260   C1          POP   BC          ;Load value length
261   IT          PUSH  HL          ;new Anf.Adr.fr.Memory aut Stack
26B2  CD 55 19    CALL  195SH       ;Move Natrix table to
                                    ;Space for the new variable to sh
26B5  E1          POP   HL          ;Ant.Adr., Load Free Memory
2M6   22 FD 78    LD    (78FDHl,HL  ;and Save
26B9  6           LD    H,B         ;new ref., matrix tab.     in HI.
2BA   69          LD    L,
26BB  22 FB 78    LD    (78FBHl,HL  ;and Save
26BE  2B          DEC   HL          Delete new Var.Tab entry
26F   3.00        LD    HL),0       ;!DE= Var.Tab of variables)
26c1  DF          RST   18H         Ready?
26C2  20 FA       JR    NZ,26BEH    ;no, next byte
26C4  D1          POP   EN          ;Load Type in E
26C5  73          LD    (Hll, E     ;enter in variable table
2     23          INC   HL          ;table address+ 1
26C7  D1          POP   EN          ;Name v0111 Get Stack
26CB  73          LD    (Hll, E     ;2. Characters in Var.
                                    Table
```

```
2C9   23          INC   HL          ;table address+ 1
2bCA  72          LD    HL),D       {I. 1 n Var. Table
26CB  OR          EX    DE,HL       ;Table address in DE
2bCC  13          INC   EN          ; + l = 1. Value Entry Address
2bCD  El          POP   HL          ;Load the programme
2bCE  C9          RET               ;finished

                  for VARPTR variable not in table

2CF   57          LD    D,A         ;Var.Tab Address 1n DE= 0
2600  5F          LD    E,A
26D1  FI          POP   AF          Correct {stack
26D2  Fl          POP   AF
2D03  E3          EX    (SP),HL     {Return address to stack
                                    ;Load the programme
26D4  C9          RET               ;back in VARPTR routine

                  variable not in table for expression analysis

2D05  32 24 79    LD    (7924H) ,A  I = for Dopp.          Exactly.
26D8  Cl          POP   .8C         Correct {stack
26D9  67          LD    H,A         iHL = for Integer
2DA   6F          LD    n
2bD.8 22 21 79    LD    (7921H),HL  ;and also enter 1n
26EN  E7          RST   20H         Identify iTYP
2DF   20O         JR    NZ,2bE7H    ;string? no jump
2bE1  21.28.19    LD    HL.1928H    {String9er on Empty String
2E    22 21 79    LD    (7921H) ,HL ;In X
2bE7  Et          POP   HL          Load {Programme pointer
2bEB  C9          RET               ;back to Expression Analysis
                                    (2 steps)
```

#%} # E ' # # ## i k kt i #

Matrix - Management

```
26E9  E5          PUSH  HI          {Programmes1ger on Stark
2EA   2A PE.7B    LD    HL, (78AH)  ;Load DIM flag and type
26ED  E3          EX    (SP>,HL     with Prog, pointer swap
2bEE  57          LD    DA          DIY =
2bEF  D5          P\JSH EN          ;DIN counter on stack
26F08 CS          PUSH  BC          ;Variable name on stack
2bF1  CD 45 1E    CALL  1E45H       ;Evaluate indexing.
                                    Result (3278) in DE
2F    Cl          POP   BC          ;Load Variable Name
2F5   Fl          POP   Af          ;DIN counter in A
2F    EB          EX    DE,HI.      Index Value 1n H
```

```
26F7  E3          EX    (SP),HL     ;swap to stack with DIM flag
26F8  E5          PUSH  HL          iDIM flag and type on stack
26F9  EB          EX    DE,HL       {programme pointer 1n H
26FA  3C          INC   A           iDIM counter + 1
2FR   57          LD    D,          and in D
26FC  7E          LD    A, (HL)     ;Load Characters
26FD  FE 2C       CP    ¦           Is there a comma ?
26FF  28 EE       JR    Z.26EFH     Next index value
2701  CF          RST   8           {follows a ')' 3
2702  29          DEFB  ')'
2703  22 F3 78    LD    (78F3H),H   ;Save programme pointer
2706  El          POP   HL          Load iDIM flag and type
2707  22 AE 78    LD    (78AWl,HL   ;and Save
270A  D5          PUSH  EN          ;DIN counter on stack
270B  2A FB 78    LD    HL, (78FBH) ;Top of the matrix table adress.
270E  3E          DEFB  3EH         LD A,19H       dummy
270F  19          ADD   HL, DE      ;Add matrix length to tab pointer.
2710  EB          EX    DE,HL       Address of the tatrixtab, in HL
2711  2A FD 78    LD    HL, (78FDHl Request. Load the free memory
2714  EB          EX    DE,HL       ;Exchange Addresses
2715  DF          RST   lBH         ;Addresses same?
2716  3A AF 78    LD    A, (78AFHl  Load type
2719  28 27       JR    1.2742H     Yes, matrix not found'
271B  BE          CP    (HL.)       Type = with table entry?
271C  23          !NC   HL          ;table address+ 1
271D  20.08       JR    NZ,2727H    {no, next table entry
271F  7E          LD    (H)         ;2. Character of name from table
2720  B9          C     C           {= 2. Signs of the requested tatr1x?
2721  23          INC   HL          ;table address+ 1
2722  20.04       JR    NZ.272BH    ;no, next table entry
2724  7E          LD    (HL)        ;1. Character of name from table
2725  B8          CP    B           {= 1 corpses of the searched matrix?
2726  3E          DEFB  3EH         iLD A,23H      dummy
2727  23          INC   HL          ;table address+ 1
2728  23          INC   HL          ;table address+ 1
2729  SE          LD    E, (HL)     Load {Matrix Length
272A  23          INC   HL          ;table address+ 1
272B  5           LD    D, HL)
272C  23          INC   HL          ;table address+ 1
272D  20 E0       JR    NZ, 27FH    1, equal unequal! n.    Table Entry

                  Matrix found

272F  3A AE 78    LD    A, 74EH)    ;DIN flag set 7
```

```
2732  B7              0R      A
2733  1E 12           L       E, 1ZH          ;Error Code in E
2735  C2 A2 19        JP      NZ,19A2H        ;yes, REDIMENS!ONED ARRAY - Error
2738  Fl              POP     AF              ;Load DIM Payer
2739  96              S0B     HL)             {=Number of dimensions in gef.tatrl?
273A  CA 95 27        JP      Z,2795H         Yes, continue at 2795/
273D  1E 10           LD      E,10            ;no, SUBSCIPT OUT OF RANGE - Err
273F  C3 A2 19        JP      19A2H           ;to the error output routine

                              New tatr1x inr1
2742  77              LD      (Hi),4          Save type
2743  23              INC     HL              ;table address+ 1
2744  5F              LD      E,              ;Length of an item     (=Type in DE
2745  16.00           LD      D,0
2747  Fl              POP     AF              ;Load DIM counter
2748  71              LD      (HL+            Z2. Zechen Matrixname in Tabelie
2749  23              INC     HL              ;table address+ 1
274A  70              LD      HL),B           1, lend matrix name in table
274B  23              INC     HL              {table lenze1ger + 1
274C  4F              LD      C,A             DIII-count in C
2740  CD 63 19        CALL    1963H           {still 2+D1M counter bytes free?
                                              ; no, @UT OF MENORY - Error
2750  23              INC     HL              ;table address+ 2
2751  23              INC     HL              ;behind length entry
2752  22 DB 78        LD      (78D8H1,HL      ;Save table address
2755  71              LD      (HL),C          ;DIM counter in table
2756  23              INC     HL              ;table address+ 1
2757  3A AE 78        LD      A,(78AH1        ;Push DIM flag into carry
275A  17              RLA
275B  79              LD      A+ C            ;DIM counter in A
275C  01 0B 00        LD      1\C, 11         ;Dimension=11 (default)
275F  30 02           JR      NC, 2763H       {no DIN, jump
2761  C1              POP     BC              ;Dimension from stack in 1\C
2782  03              INC     .BC             + I for 0-index
2763  71              LD      (Hli, C         ;Add to Table
2764  23              INC     HL              ;table address+ 1
2765  70              LD      (HL),B
276   23              INC     HL              ;table address+ 1
2767  F5              PUSH    AF              ;DIN counter on stack
2768  CD AA 0B        CALL    0BAAH           {last tatr. value length * dimension
                                              at start = length of a value
27B   F1              POP     AF              ;DIii counter Jaden
276C  3D              DEC     A               ;- 1
2760  20 ED           JR      NZ.275CH        Other dimensions? yes jump
```

```
 27F F5              PUSH   AF              ;Carry DIM flaq on stack
2770 42              LD     MD              {Matrix Value Length in B€
2771 4B              LD     C,E
2772 EB              EX     EN, Hl          ;to table address
                                            Now to 1. value byte)
2773 19              ADD    HL, DE          ;Add
2774 38 C7           JR     C, 273D0        ;excessive, SUBSCRIPT OUT OF RANGE
2776 CD 19           CALL   199CH           Sufficient free Spelher?
                                            ;no, OUT OF MEMORY - Error
2779 22 FD 78        LD     (78FDH) ,HL     ;new free memory address
277C 2B              DEC    Hl              ;Delete Matrix Values
2770 3 08            LD     HL),0           by writing @0
277F DF              RST    18H             {the e-Begin Address increased?
2780 20 FA           JR     NZ.277CH        ;no, next byte
2782 03              INC    BC              ;l'latrix value length +1 (f. DIM
2783 57              LD     D,A             $D = 0
2784 2A D8 78        LD     HL, (78D8H)     ;Load DIM counter address
2787 5E              LD     E, (HL)         ;Dil'I payer in E
27B8 EB              EX     EN, Hl          and mn HL
2789 29              ADD    HL, HL          iDIH counter * 2
278A 89              ADD    HL,BC           ;+ matrix length
278B EM              EX     DE,HL           ;= l'latrix lenqth (in DE>
278C 2B              DEC    HL              ;table address - 2
278D 2B              DEC    Hl              {= Length Field
278E 73              LD     <HL>,E          ;Insert matrix length into table
278F 23              INC    Hl              ;table address+ 1
2790 72              LD     (HL,D           { #15B)
2791 23              INC    HL              ;table address+ 1
2792 F1              POP    AF              ;Load DIM flag
2793 38.30           JR     C,27C5H         DH1? yes-ready

                     Determine the URL of a Matrix Element

2795 47              LD     B,A             ; Set Matrix Offset= 0
27%6 4F              LD     C,              51n RC
2797 7E              LD     A HL)           ;Number of dimensions in A
2798 23              INC    HL              ;Table pointer to 1. Dimension
2799 I               DEFB   11i             LD D, OEIH   dummy instruction
279A E1              POP    HL              ;Load table address
279' 5E              LD     E, (H)          ;Load Dimension
279C 23              INC    HL              ;table + 1
279D 5               LD     D, HL)          l (0lSB)
279E 23              INC    HL              ;Table + 1
279F E3              EX     (SP),H          ;Table pointer to stack
                                            iLoad Index
```

```
PUSH    AF          ;DIM counter on the stack
RST     !SH         Index }= Dimension ?
JP      NC, 273DH   yes, SURSCRIPT OUT OF RANGE
CALL    @BAAH       ;tatriy-@ffset +Dimension
ADD     HL, DE      ;+Index= new matrix offset
DEC     A           Processed all dimensions?
LD      B,H         {new tatri offset in RC
LD      C,L
JR      NZ, 279A4   ;more Dillll!nsions, back
LD      A, (78AFH)  ;Load type (=value length)
LD      B,H         ;New Matrix Offset
LD      ,L
ADD     HL, HL      Matrix offset t 2
SUB     4           ;String or Integer?
JR      C,27BDH     Yes!
ADD     HL, HL      {matrix offset + 4
JR      Z,27C2H     {mnf.Accuracy? yes jump
AOD     HL, HL      ;Matrix Offset + 8 {dopp.en. )
OR      A           {Number of dopp. Accuracy?
JP      P0.27C2H    yes'
AOO     HL,BC       {String, 3 + tatr1x offset in H
POP     BC          ;Load value start address
AOO     HL,BC       ;Add Matrix Offset
EX      DE,HL       ;=Element address, in DE.
LD      HL, 178F3H) ;Programme pointer
RET                 finished
```

##\#EI ' # k # l # # # k #k #l # i # # # ##

I'IEI1 - Function

Determine Free Memory Size

```
XOR     A           Type byte = (no string!)
PIJSH   HL          ;Programme pointer to stack
LD      (78AFH>,A
CALL    27D4H       Call iFRE
POP     HL          Load programme pointer
RST     10H         ;Address next character
RET                 {finished
```

****

****

FRE - Function

Size of free memory or free string area

identify

```
27D4  2A FD 78        LD    HL,          Beginninq adr. of free memory
27D7  Eli             EX    EN, HL           EN
27D8  21 00 00        LD    HL,0         ;HL stack pointer
27DC  E7              RST   2ilH         ;Test Type. String?
27DD  20 0D           JR    NZ.27ECH         No!
27DF  CD DA 29        CALL  29DAH        Cache Argument and
                                         Delete {Stringscope
27E2  CD E6 28        CALL  28E6H        ;Sort Strength Area, Over-
                                         Remove liquid strings
27E5  2A A0 78        LD    HL,          String Area Beginning - 1
27E8  EB              EX    DE,HL            EN
27E9  2A D6 78        LD    HL,          last free byte of the stringber,
                            (78DH)
27EC  7D              LD    L.           ;difference between HL and DE
27ED  93              SUB   E            5= free space; or
27EE  6F              LD    L,A          ides free string
27EF  7C              LD    A, H         ; (01SB)
27F0  9A              SBC   A, D
27F1  67              LD    HA
27F2  CJ 66 0C        JP    0C66H         Hl with one gene.  in X, ready
```

%%#ti# Mt i# ##} kt #M ##### k #i &M

POS Function

determines the cursor position

```
27F5  3A A6 78        LD    A, (78A6H)     ;Load cursor position
```

Number as Integer (no sign) in X

```
27F8  6F              LD    L,A            Lahl in L
27F9  AF              XOR   A              iA = M
27FA  67              LD    H,A            iH = 0
27FB  CJ 9A 0A        JP    0A9AH          Hl as Integer 1n #
```

%#%} #) i # # %} % # t #% t k % %~%#3##

USR Function

Invoke a Machine Programme Rout in
The **parameter** is the argument in X, the type in A
and in the case of strings d1e, give the address of the Str1ngs

```
27FE  CD A9 79        CALL  79A9H          ;RAN expansion output
2801  07              RST   10H            ;next character in the programme
2882  CD 2C 2S        CALL  252CH          ;Evaluate argument (in X)
2805  E5              PUSH  HI             ;Programme pointer to stack
2806  21.90.08        LD    HL, 0890H      ;Return address to stack
```

```
2       PUSH        HL
2             LD    , (784FH)             Load Tgp of Argument
2       PUSH        AF                    to the stack
2             CP    3                     = Str1ng?
2       CALL        Z.29DAH               Yes, last stream from luischensp.
                                          and remove Str i ngbere1 eh
2       POP         AF                    Reload type
2             EX    DE,HL                 {String Address 1n EN
2             LD    HL, (788EH1           tasrh-rout1ne iStartaoresse
2             JP    (HI.)                 {Run Routine
```

****

****

Value in desired type umede]n

Per.< A = Type
= Initial
Exp.: } = Result **in Posted** Type

```
2       PUSH        HL                    Get on the stack
2           AND     7                     Type=dopp.Gen, 0 as Tab.Offset
2            LD     H, 181H               {Type Conversion Jump Table
2            LD     C,A                   ;Tab.Offset (=T!:IP, except for dopp,)
2            LD     0.0                   in .BC
2           ADD     HL,.BC                to jump table start
2      CALL         2586H                 ;add again, load address
                                          and start
2           POP     HL                    Restore iHL
2           RET                           ;finished
```

#t MM i MM i Mil M #

Check whether execution is in DIRECT-Norus.

111enn yes, ILLEGAL DIRECT OPERATION Error

```
2       PUSH        HL                    ;Progra111111pointer to the stack
2       LD          HL, (78A2H1)          Current line111Load the
2            INC    HL                    {=FFFF ?
2       LD          A,H                   ; (=Direct needle
2       OR          L
2           POP     HI…                   Load {Prog hand
2           RET     NZ                    ;no, back
2          LD         E, 16H              ;Error Code in E
2          JP      19A2H                 ;ILLEGALLY ISSUE OPERATION DIRECTLY
```

%#ti ##litt Et#i ## I HE

```
                        STR$ - Funktion
                        Change number to string
2836  CD BD 0f          CALL   0FBDH          5Number  to transform into
2839  CD 65 28          CALL   2865H          ;Cache String
                                              ;and take X
283C  CD DA 29          CALL   29DAH          Delete String from Zwischenlager.
28JF  01 2B 2A          LD     BC,2A2BH       ;Set Reversal Address
2842  CS                PUSH   BC
2843  7E                LD     A, (HL)        {Str1 length in A
2844  23                INC    HL             {Ctrl + 1
2845  IT                PUSH   HL             ;Stringzeiger to the stack
284   CD BF 28          CALL   28:SFH         {Place for String in String Range
                                              {Reserve
2849  E1                POP    HL             ;Load Stringzeiger
284A  4E                LD     C, HL.)        ;Load Stringadr
284:S 23                INC    Hl…            3(in BC)
284C  46                LD     , (HL)
284D  CD SA 28          CALL   285AH          {address in string range in prel.
                                              ; Z111Slot More Transferred
2850  IT                PUSH   Hl…            ;Cache Address on Stack
2851  6F                LD     L,A            ;String length in L
2852  CE 29 CD          CALL   29CEH          ;String in String Pane
2855  D1                POP    EN             ;Load Cache Location
2856  C9                RET                   ;X cache,                finished
```

MM # k # kM## k # # # kt ik #i##

Identify address in string area and provisionally
Save Cache
Ring.< A = String Length
Exp.: DE = String address in the string area
Hl = Preliminary cache address
String Length+Stringad. in the previous cache

```
2857  eo BF 28          CALL   28BFH          ;Reserve space in the string area
2854  21 D3 78          LD     HL.78DJH       ;Adr. des vorl. Cache
2850  IT                PUSH   HL             to the stack
285E  77                LD     (Hl.),A        ;Insert String Length
285F  23                INC    HL             Adr, Cache + 1
280   73                LD     (HL>,E         ;Insert String Address
2861  23                INC    HL
2862  72                LD     (HL),D
2863  EI                POP    HL             ;Load URL of cache
2864  C9                RET
```

```
2865 2B
2866 86 22
2868 50
2889 E5
2864 0E FF
286C 23
286D 7E
286E 0C
286F B7
2870 28 O
2872 BA
2873 28 03
2875 B8
2876 20 F4 2878
FE 22 287A CC 7B
ID 2B7D E3

287E 23
287F EB
2880.79
2881  CD  SA  28
2884  11  D3  78
2887 3E
2888 D5
2889 2A 113 78
288C 22 21 79
288F 3E 3 2891
32 AF 78 2894 C
D3 09

2897 I1 D6 78
2B9A DF
289B 22 113 78
289E
289F 7E
28Alil  C0
```

```
**********************************

String constant in between,enspe1cher and X take over
Eing.' HL = Pointer to string art aunt
etc.' String length and string address in
      luisspencher, cache address in }

DE    HL          String - 1
C     B, 22H      ; separator 1; '"
LDD   D, B        ;= Separator 2  ;String1ger
PUSH  HL          - I on stack ;Character
LD    C,0FFH      counter = -f ;String pointer
INC   HL          + 1
LD    , (HL)      ;Load
INC   C           Characters ;Charac
OR    A           ter count+ 1
JR    Z,2878H     ; End of line?
CP    D           ;Yes, Stringende
JR    1.2878H     ;= separator 2 yes,
CP    B           striking
JR    NZ,286CH    i= separator 1?
CP                ;no, next character
CALL  Z, 1D78H    borrow last '++7
EX    (SP), HL    ;yes, next character ;string
                  pointer - 1 load, {derz.
INC   HL          String pointer to
EX    DE,HL       Stark ;String pointer + 1
LD    AC          in EN
CALL  285AH       ;String length in A
LD    DE, 78D3H   ;String in vorl. Cache Adr. of the
DEFB  3EH         vorl, cache {LD A,AD5H dummy
POP   EN          command ;Load string pointer into
LD    HL, ( 78B3H) DE ;Mollentane intersp,adr. Load
LD    (7921HL,HL  in X.
LD    A,3         Type = Set String
LD    (784H),A
CALL  09D3H

LD    DE, 78D6H   ;Preliminary cache to ;next cache
                  location ;Cache full?
RST   18H
LD    (78B3H),HL  ;Next Cache Location. Aerk. Load
POP   HL          {Programme pointer
LD    AHL)        ;Load next character
RET   NZ          Zu.Sp. not full, fert1g
```

```
                STRING FORNULA TOO COMPLEX Error
26a 1E          LD          E, 1EH              ;Error Code in E
28A C3          JP          19A2H               ;Report Error
3   2
                ********************************************************************************
                ************************
                Print String
                Ring: HL ~ String Address
                            th au er ng du '" or completed
28A 23          INC         HL                  ; String address + I
28A CD          CALL        2865H               {String in Cache + X
2BA CD          CALL        29DAH               ;Delete cache string,
28A CD          CALL        09C4H               Stringadr. in RC, length in D
28B 14          INC         D                   String + 1
28  15          DEC         D                   ;all characters printed?
2BB CB          RET         or                  Yeah, done
28B 0A          LD          A, (BC)             ;Load Characters
28B CD          CALL        032AH               ;and spend
28B FE          CP          0DH                 Carriage Return?
2BB CC          CALL        Z,2103H             Jas via RA! output 79DM0H to.
2BB 03          INC         BC                  Stringadress + 1
2BB 18F         JR          28B1H               ;next character
D   2
                ********************************************************************************
                ************************
                Reserve space for a string in Stringbere1ch
                Ring: A = string9length
                etc. ! DE = String area address
28B B7          OR          A                   Delete iPACh Flag
28C 0E          DEFB        0EH                 iLD C,0F1H     dummy instruction
28C F1          POP         AF                  ;Load PACK flag from stack
28C F5          PUSH        AF                  ;to the stack
2BC 2A          LD          HL,(78A0Hi          ;Begin of string range - 1
28C EB          EX          DE,HL               5in
28C 2A          LD          HL., (7D)           Pointer to 1. free byte in litre
28C 2F          CPL                             Complement String Length
28C 4F          LD          C                   and in BC
28C O           LD          B,0FFH
28C 09          ADD         HL,BC               String Pane - Length
28C 23          INC         HL                  ;+ 1 (result correction▶
280 DF          RST         18H                 4Beginning of the string area - 1?
28D 38.         JR          C, 28DAH            $yes, pack string area
28D 22          LD          (78D6H),Hl          ;save new string area pointer,
3   D
```

```
28     23              iNC     HL              :+ 1 = string address in bar, bereh
28D7   EB              EX      DE,HL           Transfer {1n DE
28D8   Fl              POP     AF              1-bit 1eder          load
28D9   C9              RET                     i finished
```

%i# k i% k # # % i #k#}% # } i# #% i# # 1 k ### # #k#

Packaging Strength Area

```
28DA   FI              POP     AF              Load iPACK Flag
28DB   1E 1A           LD      E, Alpha        ;Error Code Rn E
28DD   CA A2 19        JP      0,1942H         already packed,     OUT OF STRING SPC
28E0   BF              CP      A               Set iPACK flag
2BE1   FS              PUSH    AF              and on the stack
28E2   01 Cl 28        LD      BC,28C1H        ;Set Reversal Address
28E5   CS              PUSH    BC              ;retry after packing,
                                               ;if enough space available.
28E6   2A to 78        LD      Hl, (78ll1Hl    {String Area Code1ger = RAM Endpoint.
28E9   22.78           LD      (78DbH),HL
2BEC   21 00 00        LD      HL,0            Highest Highlight =
2BEF   IT              PUSH    HL              ;to the stack
28F0   2A A0 78        LD      HL, (78A0Hl     highest string = Ant Stringber.
28F3   IT              PUSH    HL              to the stack      1
```

Find Highest String in Z111 Cache Strings

```
2BF4   21 B5 78        LD      HL,7BBSH        ;Start Cache in HL
28F7   EB              EX      DE,HL
2BF8   2A4 378         LD      Hi, (78R3H)     {Adr, next free lui
2BFB   EB              EX      DE,HL           Pe1cherplatz in DE
2BFC   DF              RST     18H             Cache cleared?
2BFD   01 F7 28        LD      RC, 28F7        Return address for
                                               {next Cache Path. load
2900   C2 44 29        JP      NZ.294AH        ;no, highest string actualis.
```

Find Maximum String in Simple Variables

```
2903   2A F9 78        LD      HL, (78F9Hl     ;request, the var table in HL
29     EB              EX      DE,HL           ;Endadr. of the Var. table in DE
2907   2A Case 78      LD      H., (78F BH)
290A   EB              EX      DE,HL
290B   DF              RST     18H             {End of table of variables?
€290   28 13           JR      Z,2921H         Yes, check matrices
290E   7E              LD      A (H)           ;Load type from var. table
290F   23              INC     ed              ;Var.Tab Address to Value
2918   23              INC     HL
2911   23              INC     Hl...
```

```
2912  FE 03          CP    3            ;String variable?
2914  20.04          JR    NZ.291AH     3 no!
291b  CD 4B 29       CALL  294BH        ;Update highest string
2919  AF             XOR   A            {A = Bh to prevent pointers from
291A  SF             LD    E,A          Transfer Type 1n DE
291B  16 00          LD    D,0
```

```
291D  19              ADD   HL, DE          Address Var, Tab. + Type (Length)
291E  18 E6           JR    2906H           ;Next Entry

                      Find the highest string in tatrix table

2920  C1              POP   C               ;Correct stack
2921  EB              EX    DE,HL
2922  2A FD 78        LD    HI., (78FDHl    ;Endadr. Matrix Table in DE
2925  EB              EX    DE,HL
2926  DF              RST   18H             End reached?
2927  CA B 29         JP    1.29BH          ;yes, highest string next
                                            ;higher part i111 string area,
292A  7E              LD    A, (HI)         ;Load the matrix type
292B  23              INC   HL              ;table address+ 1
292C  CD C2 09        CALL  09C2H           ~Matrix length in BC
                                            ;Table address on Dil1 counter
292F  E5              PUSH  HI.             ;table address is the stack
2930  09              ADD   H,BC            {+tatrix length = beginning of
2931  RD 03           CP    3               String ttatriy?
2933  20EB            JR    NZ, 2920%       ;no, next matrix
2935  22 DB 78        LD    (78D8H),HI.     Adr. the n, Save Matrix
2938  E1              POP   HI.             ;Load address DIN counter
2939  4E              LD    C, (HL)         Load ;DIN counter
293A  0 00            LD    B,0
293C  @9              ADD   HL,BC           Add to Address DIt Counter 2
293D  09              ADD   HL, BC
293E  23              INC   HL              ;+ 1 = Value address of matrix
293F  EB              EX    DE,HL
2940  2A DB 78        LD    HL,(78D8H)      Adr. of n. l'latrix in DE
2943  EB              EX    DE,HL
2944  DF              RST   18H             ;Matrix fully edited?
2945  28 DA           JR    Z,2921H         5yes, next tatrix
2947  01 3F 29        LD    BC,293FH        ;no,Load return address

                      Compare string with the highest string to date, and
                      if i111 string range is higher, replace it.

294A  CS              PUSH  BC              Return address to stack
294B  AF              XOR   A               ; String length = ?
294C  B               OR    (HL)
294D  23              INC   HL              ;String pointer to string address
294E  SE              LD    E, (H)          ;Load string address
294F  23              INC   HL
2950  5               LD    D, HL.)
2951  23              INC   HL              {String + I
2952  CS              RET   0r              {String Length = , done!
```

```
2953  44          LD    M,H          {Ctrl pointer in BC
2954  4D          LD    CL
2955  2A D6 78    LD    HI,(78D6Hl)  ;String address > String pointer?
2958  Df          RST   18H          i!String already resorted)
2959  60          LD    H, B         String pointer back in HI.
295A  69          LD    L,C
295B  Profit      RET   C            {Yes, done'
295C  egg         POP   HL           Return address in HL
295D  E3          EX    iSPl, Hl     ;Load the highest string address
                                     ;Return address back to stack
295E  Df          RST   18H          String to be examined higher than
                                     Highest string?
295F  E3          EX    !SPl,HL      ;URL of the highest string on
                                     ;Stack, Load Return Address
2960  IT          PUSH  HL           Return address to stack
2961  60          LD    H,B          ;String pointer back in HL
2962  9           LD    L,C
293   DO          RET   NC           No, done!
2964  Cl          POP   BC           ;Return address in BC
2965  F1          POP   AF           ;Top URL and pointer
2966  F1          POP   AF           ;Get Strings From Stack
2967  IT          PIJSH HI...        pointer and address of the
2968  DS          PIJSH EN           ;Strings tested as new
                                     ;highest string on stack
299   CS          PUSH  BC           {Return address on stack
2964  C9          RET                ;finished

                  Sort Highest String

296B  D1          POP   EN           ;Load the highest string address
296C  E1          POP   HL           Pointer of the highest string in HI.
296D  7D          LD    A,L          {pointer = 8 ?
296E  B4          OR    H            ;(all strings sorted)
296F  approx      RET   0r           Yeah, done!
2970  2B          DEC   HI...        {String pointer to string address
2971  46          LD    B, HL.)      Load {String Address
2972  2           DEC   HI...
2973  4E          LD    C, (HL)
2974  IT          PUSH  HI...        String pointer to the stack
2975  2B          DEC   HI...        ;Load String Length
2976  6E          LD    Ls (H)       In HI.
2977  26 00       LD    H,0
2979  69          ADD   H,BC         j+ String Address
297A  50          LD    D,B          ;String address in DE
297B  59          LD    E,C
```

```
€297 2B          DEC   HL          HL = Outbound
297D 44          LD    RH          {1n BC
297E 4D          LD    C,L
297F 2A D6 78    LD    HL, (78D6Hl Load 2enger to Strings
2982 CD 58 19    CALL  1958H       ;string below the string-
                                   Save Area-Legers
2985 EI          POP   HL          {Stringzelger Load Again
2986 71          LD    (HU,C       Save {new string address
2987 23          INC   HL
2988 70          LD    (HU,B
2989 69          LD    L,C         and mn HL
298A 60          LD    H, B
298B 2B          DEC   HL          ;- 1 = new string area pointer
298C CJ E9 28    JP    28E9H       Save $and continue
```

Mil k #}## #k#}## } # } f

String Link

```
298F CS          PUSH  JC          ;Last priority on stack
2990 E5          PU5H  HL          {Programme pointer to stack
2991 2A 21.79    LD    Hl, (7921HJ) 1. String pointer in Hl.
2994 E3          EX    (SPJ,HL     ;Load the programme pointer
                                   1. String pointer to stack
2995 CD 9F 24    CALL  249FH       E. operands besti111111e11
2998 E3          EX    (SPl,HL     { programme pointer on stack
                                   1. Load String Pointer
2999 CD F4 0A    CALL  0AF4H       2. Operand string? no - TN Error
299C 7E          LD    A (HL)      ;1, Load String Length
299D E5          PUSH  HL          ;1. String pointer to the stack
299E 2A 21.79    LD    Hl.,(7921Hl. 2. Load String Pointer
29Al E5          PUSH  HL          ;and also on the stack
29A2 6           ADD   (liL)       ;Add Strings
29AJ 1E 1C       LD    E,1CH       ;Error Code in E
29A5 DA A2 19    JP    C. 19A2H    12,256? ja-STRING TOO LONG Error
29AB CD 57 28    CALL  2857H       ;Link String
                                   ;shade, in prep.intermediate,one
29AB Dl          POP   EN          ;2, Load again string pointer
29AC CD DE 29    CALL  29DEH       2. String from Intermediate
294F E3          EX    (SP),HL.    ;2. Stack Stringzeiger
                                   il, load string pointer
29B0 CD DD 29    CALL  29DDH       1. Remove String from
29B3 E5          PU5H  HL          1. String pointer to the stack
29B4 2A D4 78    LD    Hl... 1 (78D4H) Stringadr. from vorl.Zw.speicher 1.
29B7 EB          EX    DE,HL       ;in
```

```
29BB  CD 29      CALL  29CH        1 . String in String Pane.
29BB  CD C6 29   CALL  29C6H       ;2. String in String Pane.
29BE  21 49 23   LD    HL.2349H    ;Load Reversal Address
29C1  E3         EX    (SPl,HL     to swap with programme pointer
29C2  E5         PUSH  HL          {programme back on stack
29C3  C3 84 28   JP    2884H       ;Prel. Cache in X and
                                   ;Cache
```

#itf %tti t kt It ME # kt Mit}

Move String to String Pane

```
29C   El         POP   HL          ;Load Reversal Address
29C7  E3         EX    (SP),HL.    ;Load string pointer,
                                   ;Return address to stack
29CB  7E         LD    A, (HL.     ;Load String Length
29C9  23         INC   HL          4String + I
29CA  4E         LD    C, (H)      Load String Address
29CB  23         INC   HL
29CC  4          LD    B, HL.)
29CD  6F         LD    n           ;String length in L
29CE  2C         INC   L           ; + 1
29CF  2D         DEC   L           String length - 1,=8?
27D0  approx     RET   or          Yeah, done!
29Dl  M          LD    A, (BC)     ;A character in the string area
29D2  12         LD    (EN),
29D3  03         INC   BC          {Ctrl + 1
29D4  13         INC   EN          ;Address String Area + 1
29D5  18 FB      JR    29CFH       ;next byte
```

#%1%#%~~# i~~1~111#

Remove String from Cache and String Pane

```
29D7  CD Fit BA  CALL  AF 4        no string in X? ja-TYPE NISl'IATCH
29DA  2A 21.79   LD    H,(7921H)   ;Load string pointer from X
29DD  EB         EX    DE,HL       ;in
29EN  CD F5 29   CALL  29F5H       ;String at the top of the memory?
                                   Yes - remove!
29El  EB         EX    DE,HL       ;HL string pointer
29E2  Co         RET   NZ          ;Not removed? done!
```

Remove String from String Pane

```
29E3  05          PUSH  EN          ; Ctrl-to the stack
29E4  50           LD   D,B         ;String address in
29E5  59           LD   E,C
29E6  1B          DEC   EN          j- 1
29E7  4E           LD   C,<HU       {String Length in C
29EB  2A D 78      LD   HL, (78DbH) ;HL padding pointer
29EB  DF          RST   18H         i= String Address - 1 ?
29EC  20.05        JR   NZ.29F3H    {no, done!
29EE  47           LD   B,A         3B = 0
29EF  09          ADD   HL,BC       ;String Pane + Length
29F0  22 D 78      LD   (78D6H),HL  {= New String Area Pointer
29FJ  El          POP   HL          ;Load Stringzei9er
29F4  C9          REET
```

#% 4%%%%% %Ei#%pi f lt} f %&}HE

Remove String from Cache

```
29F5  2A BJ 78     LD   Hl., (78BJH) ;Load Next Cache
29F8  2B          DEC   HL          ;- 1
29F9  4b           LD   B, HL)      Load the address of the last
29FA  2B          DEC   HL
29FB  4E           LD   C, HL)
29FC  2B          DEC   HL          Pointer to start of last.
29FD  DF          RST   18H         = String pointer
29FE  c0          RET   NZ          No!
29FF  22 BJ 78    LD    (78B3H),HL  Save {new pointer,
                                    last entry deleted
2A82  C9          RET
```

%# Mt#Eilt t tikt} Mi tFH#±#

LEN - Function

Gets the length of a string

```
2A03  111 FB 27   ID    BC,27F8H    ;Set Reversal Address
2A06  CS          PUSH  BC
2A07  CD D7 29    CALL  29D7H       ;Argument String from Intermediate
                                    ;Remove memory + string area.
2AA   AF          XOR   A           $D = @
2Af/JB 57         LD    D,A
2Af/JC 7E         LD    , (H)       Load string length
2Af/JD B7         OR    A           ;and test
2A0E  C9          RET               Continue at 27FIIH
```

fffil-

fallflfffflfffflflflflflflflflflflflflflllfff

```
                        AS - Function
                        Identify the ASCII code of 1, String Character
2A0F 01 F 27    LD      BC,27F8H        Set {Return Address
2A12 C5         PUSH    BC
2A13 CD 07 2A   CALL    2A07H           Strin9length = ?
2A16 CA 4A LE   JP      Z, 1EAH         Yes, FUNCTION CODE Error
2A19 23         INC     HL              ;String pointer to string address
2A1A 5E         LD      It (HL.)        ;Load string address
2A1B 23         INC     HL
2A1C 5          LD      D, HL)
2A1D 1A         LD      A, (EN)         Load iL characters
2A1E C9         RET                     ;continue at 27F8}

%%# t #i t # i # # #H } # i #k #±#

                        CHR$ - Function
                        Created from del! argument    (ASCII code) a 1-byte string
2A1F 3.01       LD      A,1             String length = 1
2A21 CD 57 28   CALL    2857H           ;Reserve space in the string area
                                        ;and preloaded cache
2A24 CD 1F 2B   CALL    2B1FH          6anzz.ert of the argument in E
2A27 2A D4 78   LD      HL, (78D4#)     ;Stringadr, from vorl.Zw.speicher
2A2A 73         LD      (HL),E          ;Save character there
2A2B C1         POP     BC              ;Remove bounce address
2A2C C3 84 28   JP      2884H           Prel. Cache to X and
                                        ;to Cache

%~%%~~% # Mil % % # %~

                        STRING$ - Function
                        creates a string of n equal characters
2A2F D7         RST     10H             ;next character
2A30 CF         RST     8               {follows a '( ?
2A31 28         DEFB    '('
2A32 CD 1 2B    CALL    2B1CH           Evaluate string length and E
2A35 D5         PUSH    EN              {Stringlength on Stack
2A3  CF         RST     8               A comma follows?
2A37 2C         DEFB    ','
2A38 CD 37 23   CALL    2337H           Evaluate {character expression
2N3B CF         RST     8               ;follows a) 7
2A3C 29         DEFB    ')'
2A3D E3         EX      (SP1, HI...     ;Load string length in L
                                        {Pro9Render on Stack
```

```
2A3E  E5           PUSH  HL          ;Strlngiangen down on the Stark
2A3F  E7           RST   20H         ; Character Expression= 5tring
2A40  28 5         JR    Z,2A47H     Yes!
2A42  IF2B CD      CALL  2B1FH       {nelms 9anzz. Value in
2A45  18 03        JR    2A4AH       continue be1 ZAH
2A47  CD 13 2A     CALL  2A13H       ;1. String character in A
2A4A  D!           POP   EN          {Stringlength 1n E
2A4B  F5           PUSH  AF          ;Character 2x on city
2A4C  FS           PUSH  AF
2A4D  7B           LD    A,E         ;Reserve space 1m
2A4E  CD 57 28     CALL  2857H       {String in Prel. Cache
2A51  5F           LD    E,A         ;String length in E
2A52  Fl           POP   AF          Reload the Drake
2A53  1C           INC   E           String length = 0 ?
2A54  1D           DEC   E
2A55  28 D4        JR    Z,2A2BH     ija vorl.Zu.sp. in memory + I
2A57  2A D4 78     LD    HI, (78D4Hl ;Load string address
2A5A  77           LD    (Hll, A     ;Transmit Characters
2A5B  23           INC   Hl          ;String Address + 1
2A5C  1D           DEC   E           {String Length- 1,=
2A5D  20 FB        JR    NZ,2A5AH    ;no, next character!
2A5F  18 CA        JR    2A2B        {vorl.Zw.sp, in memory &.      X
```

Mi Hi k # # k Mi k # } # # #M i # #E

LEFT$ - Function

to separate the left part of a string

```
2A61  CD DF 2A     CALL  2ADFH       {Programme pointer in  Test 1,
                                     2. Argument in B
2A64  AF           XCIR  A           Left Offset = {f.RIGHT$ u.MID$)
2A65  E3           EX    (SP),HL     ;Programme Pointer to Stack
                                     Load String Pointer
2   4F             LD    CA          Left offset in C
247   3E           DEFB  3EH         LD A, 05H  lully instruction
248   E5           PVSH  HI...       ;Stack string pointer
                                     ;(USIN6 stack correction)
2A9   E5           PUSH  HI...       ;Stack string pointer
2AA   7E           LD    A (H)       Load 5String Length
2AB   B8           CP    B           12th Argument?
2AhC  38.82        JR    C,2A70H     yes, Erg.stringl=stringlenght
2A6E  78           LD    A,B         no, string length = 2nd argument
2AhF  11           DEFB  11H         ilD DE,BNEH  duy instruction
2A70  0E 0o        LD                iLeft offset = 0
2A72  C5           PUSH  BC          left—Offset on stack
```

```
2A7 C     CALL     28:BFH       {Place f.Result1s in Stringber.res.
2A7 C     POP      BC           ;Left Oftset Reload
2A7 E     POP      Hl.          ;Load String
2A7 I     PUSH     HL           and back on the stack
2A7 2     INC      HL           ;String pointer to string address
2A7 ⊿     LD       B, (HL)      ;Load string address in HL
24  2     INC      Hl.
2A7 b     LD       H, HL)       i!11SBl
2A7 b     LD       L+M          {(L.SR)
2A7 0     LD       B,0          ;Add Left-Oftset to Stringadr.
248 8     ADD      HL,BC
2A8 4     LD       B,H          ;Result string address in BC
2A8 4     LD       C,L
2AB C     CALL     285AH        ;Result string in vorl.Zw.speich.
2A8 6     LD       L            Erg.String Length in L
2A8 C     CALL     29CEH        ;String in string region
2AB D     POP      EN           ;String pointer in DE
2A8 C     CALL     29DEH        ;Argument String from Zw
                                and Delete Strings
2AB C     JP       2884H        ;Vorl.Zw.sppager in memory +X
```

#ttiMt%Mt %M} ##t Mt iM Wed tE

RIGHT$ - Function

to separate the right part of a string

```
2A9 C     CALL     2ADFH        ; programme pointer in H,      ')'
                                12th Argument in B
2A9 D     POP      EN           ;Load String
249 D     PUSH     EN           ;and back to stack
249 1     LD       A, (DE)      ;Load String Length
2A9 9     SUB      B            ;- 2. Argument= Left Oftset
2A9 1     JR       2Ab5H        continue at LEFT$
```

HFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF FFFFFFFFFFFFFFFF

KIDS - Function

to separate the middle part of a string

```
2A9 E     EX       DE,HL        }Programme pointer in H
2A9 7     LD       A, (HU       ;Load Characters
2A9 C     CALL     2AE2H        2. Argument in B
2A9 0     INC      B            2. Argument = ?
2   0     DEC      B
2M1 C     JP       Z, 1E4AH     Yes, FUNCTION CODE Error
2M E      PU5H     BC           ;2. Argument to the stack
```

```
2AA5  L FF          LD    E,0FFH       3. Argument = 255 (h=Reststring)
2AA7  RD 29         CP    ')'          ;')' to complete the parameters?
2AA9  28 05         JR    Z,2A1\0H     $Jan only? arguments
2AAB  CF            RST   8            {no, a comma follows?
 2AC  2C            DEF1\ ','
 2AD  CD 1C 21\     CALL  21\1CH       3. Evaluate Argument   (in El
2A1\0  CF           RST   8            Now ')' must follow
2A1\1  29           DEFB  ')'
2A1\2  Q!           POP   AF           ;2. Load argument in A
2A1\3  E3           EX    (SP1,HL      {Programme pointer on stack
                                       ;Load String
 2AB  01 69 24      LD    BC,2A69H     ;Set Reversal Address
2A1\7  CS           PUSH  BC           ; (Leap in LEFT$)
2A1\8  3D           DEC   A            ;2. Arg. - 1 = Left offset
2A1\9  BE           CP    <HL1         {String length { 2. Argument?
2A1\A  6 00         LD    B,0          Result String Length = @
2A1\C  D0           RET   NC           ;yes, result string is empty
2A1\D  4F           LD    CA           ;Left offset in C
2A1\E  7E           LD    .(HL)        Load {String Length
 2ABF  91           SUB   C            ;- Left Offset
 2AC0  BB           CP    E            € 3. Argument?
2AC1   47           LD    B,A          String Length = Difference
 2AC2  Profit       RET   C            ijas ges.Reststring=Result String
 2AC3  43           LD    1\,E         No, three. Argument = Erg.
 2AC4  C9           RET                continue at LEFT$
```

tffiHttffiHttttttttttttttffHfftfftfftHfftfftHfftfft

MP. - Function

Convert string to number

```
2AC5  CD 07 2A4     CALL  2A07H        {String length argument = 7
2AC8  CA FB 27      JP    Z,27F8H      yes, as an integer in k, ready
2ACB  SF            LD    E,A          ;Strin9length in E
2ACC  23            INC   HI           {Ctrl + 1
2ACD  7E            LD    A+ (HL)      ;HL string address
2ACE  23            INC   HL
2ACF  66            LD    H, L
2A110 6F            LD    s
2AD1  IT            PUSH  HL.          ;String address on the stack
2AD2  19            ADD   HL, DE       ;+ String Length
2AD3  46            LD    M, HL)       ;1. Next String Character
2AD4  72            LD    (HU,D        to @ (end of line)
2AD5  E3            EX    (SP),HL      Stringadr. n.String to the stack
                                       Load string address akt.String
```

```
2AD6  C5          PUSH  BC          i1. save characters from strings
2AD7  7E          LD    A, (HL)     1. Load String
2AD8  CD 65       CALL  OE5H        {Transform String to Number (I)
2ADB  C1          POP   BC          ;1. Load Characters from n.Strings
2ADC  E1          POP   HL          ;Load String Address
2ADD  78          LD    (HL),B      ;1. Character in next string to.
2ADE  C9          RET

                  ************************************

                  Subprogramme for LEFT$, RIGNT$ and ID$
2ADF  EB          EX    DE, HL      ;LEFT$ and RIGHT$
                                    programme      in HL
2AE0  CF          RST   8           icheck that:   ')' completed
2AE1  29          DEFB  ')'
2AE2  C1          POP   BC          iInstruction
                                    ;Load Reversal Address
2AE3  D1          POP   EN          ;2. Load argument in E
2AE4  C5          PUSH  BC          ;Backjump address aut stack
2AE5  43          LD    B,E         2. Argument in R
2AE6  C9          RET

                  %%%i##M##########################$#

                  Function tokens on the left side of an assignment
2AE7 FE 7A 2AE9   CP    7AH         i= MIO$ - Token
C2 97 19 2AEC     JP    NZ, 1997H   ;no, SYNTAX ERROR
C3 D9 79          JP    79D9H       Yes, to the RAM-er..at the
terungsausgang

                  ****************************************

                  INP - Function
                  Read Data from Input Port
2AEF CD 1F 2B     CALL ZBIF         Whole. Value of the In argument
24F2 32 9  78     LD (7894H),       {1n RAt Subprogramme as PORT#
2AF5 CD 93 78     CALL 789JH        ;Run RAM Subprogramme
2AF8 CJ FB 27     JP 27F8H          {A Content as Result in X

                  %%## %## %k#MM#i%'#ki%MEM

                  UT Statement
                  Output Port Data
2AFB CD 0E 2B     CALL! ZB@EH       {Analyse both arguments and {port
                                    number in RAM subroutine
```

```
2AFE C3 96 78          JP      789bH          ;Run RAM Subprogramme

                       ****************************************

                       Evaluate expression and convert result to integer
                       Eing.' Hl Print Address in Programme
                       Exp.: EN= Result
                              Flags: S= 1 Result1s {@
                                     / = 1 Result< 256
2B01   D7              RST     10H            ;Address next character
202    CD 37 23        CALL    2337H          ;Evaluate Expression
205    IT              PUSH    HL             programme pointer to stack
2B     CD 7F 0A        CALL    0A7FH          Convert Ergeonis to Integer
29     EB              EX      DE,HL          {result in DE
2BOA   El              POP     HL             {Programme      load
2RE    7A              LD      A,D            ;Set Flags
2B0C   37              OR      A
2B0D   C9              RET


                       Analyse 2 arguments for OUT
2RE    CD 1C 2B        CALL    2B1CH          ;Apply Port Nu1111111er (in Al
2B11   32.94.78        LD      (7894Hl,A      {in INP and OUT Subprogramme on
2B14   32.97.78        LD      (7897Hl,A
2B17   CF              RST     8              Coma next?
2B1B   2C              DEFB    ,
2B19   18 01           JR      2B1CH          ;Analyse Value    1<256) and A
                       #%##i% i # }} # # # #kt# i Kt f#k3}# # i%

                       #l % i% ##Mt t # # i # # MM#

                       Evaluate expression, Result in Integer u1Jt11Jandeln
2B1B   D7              RST     10H            ((256) ;Address next character
2B1C   CD 37 23        CALL     2337H         Evaluate expression
2B1F   CD 05 2B        CALL 2B05H             {Convert result to integer (DE) i>
2B22   C2 4A 1E        JP      NZ, 1E4AH      256 = FUNCTION CODE - Error
2825   2B              DEC     Hl.            {Programme pointer- 1
2326   D7              RST     10H            ;Address next character;Result
2B27   7B              LD      A,E            in A
2B28   C9              RET

                       ****************************************

                       LLIST Statement
```

```
                    Pr oar on to list on printer
2R29  3E 1          LD    A,1           Print Output Flag
2B2B  32 9C 78      LD    (789CH) ,A

                    %%%# # # {k ##i f ####ii# # M # i # M #t

                     LIST Statement

                    List programme on screen
2B2E  C1            POP   BC            ;Return address from stack
2B2F  CD 10 1B      CALL  11MH          analyse both arguments
                                        il.Lineadr.in BC, 2.Zeil.nr=Stack
2R32  C5            PUSH  BC            i 1. Row Address on Stack
21133 CD 25 JB      CALL  3B25H         ;Interrupt/Cancel List?
2B36  22 A2 78      LD    (78A2H),HL    Set iDirect Command (Znr=FFFF>
2B39  E1            POP   HL            ;1st line address in HL
2B3A  D1            POP   EN            2.Line number in DE
2B3B  4E            LD    C IHL)        Load Row Pointer
2BJC  23            INC   HL
2B3D  4             LD    B, (H1)
2B3E  23            INC   HL            {Line Address on Line Number
2113F 78            LD    A,B           End of programme ? {pointer=0@)
2B0   B1            OR    C
2M1   CA 19 1A      JP    Z, 1A19H      Yes back to main loop
2B    CD DF 79      CALL  79DFH         iRAN extension output
2B47  CD 9B 1D      CALL  1D9B          ;Deselect key stroke
2B4A  C5            PUSH  BC            {address of next row on stack
2B4B  4E            LD    C, (HL)       {Load Line Number
2B4C  23            INC   HL
2M4D  4             LD    B, HL)
2BJ+E 23            INC   HL            Programme pointer to line text
2BJ+F C5            PUSH  BC            {Line number on stack
2B50  E3            EX    (SP),HL       Programme pointer to stack
                                        Load line number in H
2B51  EB            EX    DE,HL         End-ZNr. in HL, Line number in DE
2B52  DF            RST   18H           Line number } End line number?
2B53  C1            POP   .BC           ;BC programming pointer
2B54  DA 18 LA      JP    C, IAIBH      Yeah, done
2B57  E3            EX    (SP>,H1       ;Load address next line.
                                        ;End Line11111er to the stack
2B58  E5            PUSH  HL            ;Address next line on stack
2B59  C5            PUSH  BC            ;Programme pointer to stack
2B54  EB            EX    DE,HL         Line in HI.
2M5B8 22 EC 78      LD    !78ECH1,HL    to save as '.,'
2BSE  CD AF WF      CLL   0FAFH         spend lei lennuner
```

```
2,861  3E 20          LD     ,'°          line number blank
2,863  EI             POP    HL           Load {Programme pointer
2:S64  CD 2A 03       CALL   032AH        Print spaces
2,867  CD 7E 2:S      CALL   2B7EH        ;Intermediate text ore
2:S6A  2A A7 78       LD     HL, (78A7H)  ;Addressing Input/Output Putter
 2R6D  CD 75 2:S      CALL   2:S75H       ;Output text of the line
 2B70  CD FE 20       CALL   20FEH        ;issue carriage return
2:S73  BE 18          JR     2B33H        ;next line
BM
```

% # # % i# # %## # k i #i Mi k # # MM

Text String output (completed with **00**)

```
 2B75  7E             LD     ,{HL)        ; Load Characters
2:S76  S7             OR     A            ;=Extender?
 2B77  approx         RET    or           Yeah, done!
2:S78  CD 2A 03       CALL   032AH        {Print Characters
 2B7B  23             INC    HL           Text address + 1
 2B7C  18 F7          JR     2B75H        ;next character
```

%#i%# #ti # t # l i ##l # lt± # i

Generate text from intermediate code

Comes from the programme line into the input/output buffer
.

```
 2B7E  E5             PUSH   HL           programme pointer to the stack
 2B7F  2A A7 78       LD     HL… (78A7Hl  ;Load buffer pointer in BC
 2B82  44             LD     B,H
 2B83  4D             LD     CL
 2B84  El             POP    HL           Load {Programme pointer
 2B85  FF 16          LD     D,0FFH       {max loans = 255
 2M87  18 03          JR     2B8CH
 2B89  03             INC    BC           ;Buffer pointer+ 1
 2B8A  15             DEC    D            ;Character counter - 1
 2B8B  approx         RET    or           Buffer voll? yes-ready
2B8C   7E             LD     A, (H)       ;Load characters from programme
2B8D   .B7            OR     A            ;End of Line?
2B8E   23             INC    HL           Pr gr indicator + 1
2B8F   02             LD     (BC),A       ;Transfer character to buffer
2B90   CB             RET    or           ;End of line, done!
2B91   C3 9D 2E       JR     2E9DH        Continue at 2E9DH <Backpack)
2B94   FE FB          CP     @FBH         =' - Tokem?
296    20 18          JR     NZ, 2BAI     No!
2898   0B             DEC    BC           ;':REN' delete from buffer
2399   0B             DEC    Bc           ; (Buffer pointer - 4)
```

```
2B94  9B          DEC     BC
29B   0B          DEC     BC
2B9C  14          INC     D           Character counter + 4
2B9D  14          INC     D
289E  14          INC     D
2B9F  14          INC     D
2BA   FE 95       CP      95H         ELSE token?
2BA2  CC 24 0B    CALL    1.0R24H     Yes, remove '' before that
2B45  D 7F        SUB     7FH         ;Token - 7F = Nunaer des
2BA7  E5          PUSH    HI.         {Programme pointer to stack
2BA8  5F          LD      E,A         Keyword number in E
2BA9  21.50.16    LD      H, 150H     ;Address key1110rt table
2MAC  7E          LD      A, H)       ;Load characters from table
2BAD  B7          OR      A           Start of a new keyword?
2BAE  23          INC     HL          ;Table + 1
2BAF  F2 AC 2B    JP      P,2BACH     No, continue searching
2BB2  1D          DEC     E           ;yes, keyword search?
2BB3  28 F7       JR      NZ,2BACH    No, next keyword search.
2BJ!5 E 7F        AND     7FH         ;in !.Sign Delete Bit 7
2BB7  02          LD      (BC),A      ;Transfer character to putter
2BB8  03          INC     BC          ;Buffer pointer+ 1
2BB9  15          DEC     D           ;Character counter - 1
2:BBA CA D8 28    JP      Z,2BDBH     ;Putter full, ready
2BBD  7E          LD      A, (HL)     ;next key1110rt character
2:BE  23          INC     HL          ;table + 1
2BBF  B7          OR      A4          new keyword?
2BC0  F2 B7 2B    JP      P,2BB7H     No, carry over
2BC3  E1          POP     HI.         {Reload Programme Pointer
2BCA  18 Co       JR      2BBCH       ;next character

                  -
                  lfIHl:ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

                  DELETE Command

                  Delete programme lines
2BC   CD UI 1B    CALL    1B10H       {Analyse both arguments
                                      ;1.Row-in BC, 2.ZNr.on stack
2BC9  D1          POP     EN          2. Line number in DE
2:BCA CS          PUSH    BC          . Row Address on Stack
2BCB  CS          PUSH    BC          {(2 Ral)
2:BCC CD 2C 1B    CALL    1B2CH       ;2. Get Row Location
2BCF  30.05       JR      NC, 2RH     ;not before, FUNCTION CODE - Error
2BD1  54          LD      D,H         ;2.  Line Address in DE
2BD2  5D          LD      E,L
2BD3  EJ          EX      (SP>,HL     ;1.  Load Line Address
```

```
                                        52nd Row Address will Stack
 2BD4  E5              PUSH   HL         1. Zeiienaoresse 1111eder on stack
 2BD5  DF              RST    18H        1. Row address (= 2. Two lenders.
 2BD6  D2 4A 1E        JP     NC,1E4AH   ;no, FUNCTION CODE - Error
211D9  212919          LD     HL, 1929H  ;Text 'READY'
 2BDC  CD A7 28        CALL   28A7H      and spend
 2BDF  C1              POP    BC         ;1. Load Line Address
 2BE0  21 ES           LD     HL, AEBH   ;Rücl: load jump address
 2BE3  E3              EX     sP),HL     m1t Z. Zeulenadr. swap
 2BE4  EB              EX     DE,HL      ;2. Line 1n DE
 2BE5  2A F9 78        LD     HL <7BF9H) {programme text end in HL
 2BEB  1A              LD     A, (DE)    ;Load characters from hint range
 2BE9  02              LD     (C),A      ;and forward
 2BEA  03              INC    BC         ; programme pointer +1
 2BEB  13              INC    EN
 2BEC  DF              RST    18H        { End of programme reached ?
 2BED  20 F9           JR     NZ,2BE8H   ;no, next character
 2BEF  60              LD     H,B        last destination=new Prague.end
211F0  69              LD     L+
 2BF1  22 F9 78        LD     (78F9Hl,HL save
 2RF4  C9              RET

%#k # k I E ## I # # # } # E #I± ##

                      SOUND Command

 2BF5  CD lC 2B        CALL   2111CH     1.Analyse Parameters (Note)
 2BF8  RD 20           CP     32         Note > 31 ?
 211FA D2 4A 1E        JP     NC, 1EH    yes, FUNCTION CODE Error
 2BFD  32 D2 7A        LD     (7AD2H) ,A Save {Node Value
 2000  CF              RST    8          Coma next?
 2C01  2C              DEFB   1 1
                              1
 2C02  CD 1C 211       CALL   2B1CH      Analyse {2.Parameter (Length)
 2C05  117             OR     A          Length = 0 ?
 2C06  CA 4A 1E        JP     Z, 1E4AH   Yes, FUNCTION CODE Error
 20089 FE 0A           CP     18         ;Lenqth > 9?
 2C0B  D2 4A 1E        JP     NC,1E4AH   ;yes, FUNCTION CODE - Error
 2C0E  F3              DI                Disable Interrupts
 2C0F  E5              PUSH   HL         { programme pointer to the stack
 2CHl  3D              DEC    A          ;Length - 1
2C11   FS              PUSH   AF         to the stack
 2C12  3A D2 7A        LD     A, (7AD2H) ;Load note value
2C15   117             OR     A          = 0 ?
2C16   28 48           JR     1, 2058~   Yes, pause
```

```
2C18  3D              DEC   A             ;Node value - 1
2C19  CB 27           SLA   A             i* 2
2C1B  4F              LD    ,A            in BC = Offset for frequency table
2C1C  AF              XOR   A
2C1D  47              LD    B,A
2C1E  F1              POP   AF            Load Length
2C1F  21 CF 02        LD    HL.02CFH      ;Address frequency table
2C22  89              ADD   HL,BC         ;+ offset for note
2C23  SE              LD    E, (HL)       Load {frequency value from table
2C24  23              INC   HL
2C2S  5               LD    D, (HL)
2c2   D5              PUSH  EN            ;Frequency value of the stack
2C27  21.61.083       LD    HL.0361H      ;Table of time base values adress.
2C2A  CB 39           SRL   C             ;frequency offset/ 2
2C2C  09              ADD   HL,BC         ;+Tables Start Address
2C2D  SE              LD    E, (Hl)       ;=Basic time value for 1/8 note
2C2E  16 88           LD    D,0
2C30  21 21 83        LD    H, 0321H      ;Table of 11 multipliers for
                                          {Address Toner Length
2C33  4F              LD    ,A            ;Sound length! in BC (=Tab.Offset)
2C34  09              ADD   HL,BC         ;+Tables Start Address
2C35  4b              LD    B, H)         ;Load multiplier from table
€2.36 DS              PUSH  EN            Time base value with BI
2C37  E1              POP   HL            ;Result in HL
2C38  19              ADD   HL, DE
2C39  10 FD           DJNZ  2C38H
2C3B  IT              PUSH  HL            ;Broadcast sound time to BC
2C3C  Cl              POP   BC            5as Cycle Counter
2C3D  E1              POP   HL            ;Load Frequency Value
2C3E  CD FB YES       CALL  3AFH          ;BREAK key pressed?
2C41  YES 3B 78       LD    A, (783RH)    ;Output latch - load byte
2C44  57              LD    DA            in D
2C45  CD 69 34        CALL  3469H         i Emit sound
248   0B              DEC   BC            ;Cycle counter - 1
2C49  79              LD    A,C           = 0 ?
2C4A  RO              OR    B
2C4B  20 FI           JR    NZ,2C3EH      No, keep the sound
2C4D  El              POP   HL            ;Load the programme pointer
2C4E  FB              EI                  ;Enable Interrupts
2C4F  7E              LD    A; (H)        ;Load Characters
2C50  23              INC   HL            { programme pointer + f
2C51  FE 3B           CP    ı             Follows a '3' 7
2C53  CA FS 2B        JP    Z,2BF5H       ;yes, play next note
2C5b  2B              DEC   HL            ; Programme pointer - 1
```

```
2C57  C9              RET             ;finished
2C58  F1              POP     AF      ;Length - 1    load
2C59  4F              LD      C,A     in BC
2C5A  AF              XOR     A
2C5B  47              LD      B,A
2C5C  21.21.03        LD      HL.0321H ;Table of Multip!ikat.adressier.
2C5F  09              ADD     HL,BC   ;+ Length-1
2C60  46              LD      B, (HU  ;Load multiplier from table
2Cbl  21 3 19         LD      HL, 6454H ;Load base value for 93.75 ms
 2C5  IT              PUSH    HL      {Basic value with B+! Multiply
2C65  D1              POP     EN      i= Pause counter
ZC    19              ADD     HL, DE
 2c7  10 FD           DJNZ    2C6#
 2c9  CD F8 3A        CALL    3AF8H   ;BREAK key pressed?
2cC   2B              DEC     HL      i counter - 1
 2CD  7D              LD      A,L     1= 0?
2C6E  B               OR      H
 2CF  20 F8           JR      NL, 2C9H {no'
2C71  18 DA           JR      2C-4DH  Yeah, break finished!
```

%%%## f # # t M# MM Wed # # f kt

Print Graphics Characters on a Printer

```
2C73  CS              PUSH    BC      Save {BC to Stack
2C7-  47              LD      B,A     ;Character in
2C75  3E 8            LD      8       Toggle {Printer in Graphics Mode
2C77  CD BA 3A        CALL    3ABAH   by X'08 output
2C7A  78              LD      A, B    ;Draw again in A
2C7B  E6 0F           AND     0FH     ;Delete top half byte
2C70  IT              PUSH    HL      Save HL to stack
2C7E  CB 27           SLA     A       Character + 2
2C80  4F              LD      C,A     ;as table offset in BC
2C81  AF              XOR     A
2C82  47              LD      B,A
2C83  21 AF 02        LD      HL, 2AFH ;Address beginning chart
2c86  09              ADD     HL,BC   ;+ character offset
2C87  7E              LD      A, (HL) 1. Load Table Value in B
2C88  47              LD      B,A
2C89  23              lNC     HL      ;table address+ 1
2C8A  7E              LD      , HL )  ;2. Load Table Value in C
2C8B  4F              LD      CA
2C8C  78              LD      A,B     1. Table value in A
2C8D  CD BA 3A        CALL    3ABAH   ;Value three times on the printer.
```

```
2C90  CD BA 3A        CALL   3ABAH
2C93  CD BA 3A        CALL   3ABAH
2c9   79              LD     A,              ;2. Table value in A
2C97  CD BA 3A        CALL   3A1!AH          ;value three times on the printer,
2C9A  CD BA 3A        CALL   3ABAH
2C9D  CD BA 3A        CALL   3ABAH
2CA0  E1              POP    Hl              ;HL and BC recover
2CA1  C1              POP    BC
2CA2  3E 0F           LD     A, 0FH          ;Printer back in text mode
2CA4  CD BA 3A        CALL   3ABAH           by output from Y'@F°
2CA7  C9              RET                    ; finished
```

#k##### #t # #k ####k k # # #i%##k#k k#k i k # # k % i #l#

PEEK - Function

Load the contents of a location

```
2CAA  CD 7F OA        CALL   0A7FH           {Convert Argument to Integer (H.)
2CAD  7E              LDL    A, (HL)         Load the contents of the location
2CAE  C3 FB 27        JP     27F8H           as result in X
```

%k#####k# # # i # ##### # ########################

POKE Statement

Write value to location

```
2CBI  CD 02 2B        CALL   2B02H           ;Analyse address and in DE
2CB4  D5              PUSH   EN              ;to the stack
2CB5  CF              RST    8               Is a comma flying?
2CR6  2C              DEFB   ',
2CB7  CD 1C 2B        CALL   2B1CH           Evaluate value (256) and A
2CBA  D1              POP    EN              Load {Address
2CBB  12              LD     (EN),A          Save value to this address
2CBC  C9              RET
```

#######k###k# k ##k##k k #k #k # #k#± # k#

US ING Statement

Formatted Output

```
2CBD  CD 38 23        CALL   2338H           ;Format string out
2CC0  CD F4 0A        CALL   F4H             No string? TYPE MISMATCH Error
2CC3  CF              RST    8               Is there a semicolon ?
2CC4  30              DEFB   ';
2CC5  E1i             EX     DE,HL           ;Programme pointer in DE
2CCb  2A 21.79        LD     HL, (7921H)     Load pointer to format string
2CC9  18 08           JR     2CD3H
```

```
                        other output pins with the same format-3tr1nq
2CCB   3A EN 78         LD      A, (78DEH)      ;Load next character
2CCE   B7               OR      A               {= Instructor?
2CCF   28 0C            JR      Z,2CDDH         Yes, FUNCTION CODE - Error
2CD1   DI               POP     EN              Load Formatstring Ziger
2CD2   EB               EX      DE,HL           {1n Hl
2CD3   IT               PUSH    HL              ;Formatstring pointer to stack
2CD4   AF               XOR     A               Delete last character
2CD5   32 EN 78         LD      478DEH),A
2CD8   BA               CP      D               ;Z flag deletes, en and set Cy
2CD9   FS               PUSH    AF              {Result flags to the stack
2CDA   D5               PUSH    EN              ;Programme pointer to stack
2CDB   46               LD      B, (HL)         ;Load String
2CDC   B0               OR      B               = 0 ?
2CDD   CA 4A LE         JP      Z,1E4AH         Yes, FUNCTION CODE Error
2CE0   23               INC     HL              ;Format string pointer + 1
2CE1   4E               LD      C, HL)          ;HL string address
2CE2   23               INC     HL
2CE3   b                LD      H, (HU
2CE4   69               LD      L,C
2CE5   18 1C            JR      2D0JH

                        'I°  - Find field length
2CE7   58               LD      E,B             String length in E
2CE8   IT               PUSH    HL              ;Stack Stringpointer
2CE9   0E 02            LD      C,2             ;Number of characters=2 (for limit)
2CEB   7E               LD      A, (Hl)         ;Load Characters
2CEC   23               INC     HL              ;String + 1
2CED   RD 25            CP      O               !O
2CEF   CA 17 2E         JP      Z.2E17H         Yes, output formatted string
2CF2   RD 20            CP      20H             Space?
2CF4   20 03            JR      NZ,2CF9H        No, no, no!
2CF6   €                INC     C               ;Number of
2CF7   1iH2             DJNZ    2CEBH           {String length- 1 } 0 ? yes"back
2CF9   El               POP     HL              Reload String Pointer
2CFA   43               LD      B,E             1String length again in B
2CFB   3E 25            LD      A, '7°'         j' %'spend

                        Find the beginning of a string or number field
2CFD   CD 49 2E         CALL    2E49H           ;'+' outside Mummernfeld from9.
2D00   CD 2A4 03        CALL    032AH           ;Print Characters
2D3    AF               XOR     A               {A = W
2D04   5F               LD      E,A             Field length =
```

```
2D05  57          LD    D,A          5Format flag = 0
2006  CD 49 2E    CALL  2E49H        ;'+' out of Number Field
2009  57          LD    DA           ;Format flag in D
2D0A  7E          LD    A,(Hl        ;Load character from string
2D011 23          INC   HL           String + 1
2D0C  RD 21       CP                 ;Exclamation mark?
2D0E  CA 14 2E    JP    Z,2E14H      Jas 1, print string characters
2011  RD 23       CP    3            ;Numbers?
2013  28 37       JR    1, 2D4CH     Yes, analyse number field
 215  05          DEC   to           ;String Length - 1
2D16  CA FE 2D    JP    Z,2DFEH      = 8 ? 3a, Stringende!
2019  FE2         CP    '+'          = 4° 2
2D1B  3E 08       LD    A, 8         Format flag = 8
2D1D  28 E7       JR    Z,2D06H      Yes, jump
2D1F  211         DEC   Hl...        ;Reload character
 220  7E          LD    A, (HL)
2D21  23          INC   Hl...
2D22  FE 2E       CP                 ;Point?
2D24  28 40       JR    Z.2D66H      ;yes, decimal places bestil1111en
2D26  RD 25       CP                 = "1' 2
2028  28BD        JR    Z,2CE7H      yes, format string
2D2A  BE          CP    (HL)         ;= next character?
2D2B  28 D0       JR    NZ,2CFDH     No, go on
2D2D  RD 24       CP    $'           i2 dollar sign?
2D2F  28 14       JR    Z,2D45H      {Yes, Set Format Flag
 231  FE 2A       CP    '*'          ;2 stars?
2033  20C8        JR    NZ,2CFDH     ;no, continue
2D35  78          LD    A,B          ;last character still in format
 236  FE 02       CP    2
2D38  23          INC   Hl.          ;string pointer to next character
2D39  38.03       JR    C,2D3EH      3 no!
2D311 7E          LD    A+ (H)       ;Load Characters
2D3C  RD 24       CP    '$'          Dollar sign?
2DJE  3E 20       LD    A, 20H       ;bit 5 of format flag for      'x' = f
2140  20 07       JR    NZ.2D49H     No!
2042  05          DEC   B            ;String Length - 1
2043  1C          INC   E            ;Number field length + 1
2144  FE          DEFl1 0EFH         iCP 0AFH     Dung instruction
2045  AF          XOR   A            ;Clear Format Flag
2146  C6 10       ADD   A, 1H        illit 4 of the Format flag     '$° = 1
2048  23          INC   Hl.          String + 1
2049  1C          INC   E            {Number Field Length + 1
2D44  82          ADD   A, D         ;Link Format Flag to Last
 2D   57          LD    D,A          ;and in D
```

```
2D4C  1C          INC   E             ;Nurr.Mernfeldlength + 1
2D4D  O 00        LD    C,0           ; Number of decimal places = 0
2D4F  05          DEC   B             ;String Length - 1
2D50  28.47       JR    Z.2D99H       ;= @? yes, format string evaluated
2D52  7E          LD    A, (HLU)      ;Load Characters
2D53  23          INC   HL            {Ctrl + 1
2D54  FE 2E       CP                  ;Point?
2D56  28 18       JR    Z,2D70H       Yes, Find Dependencies
2D58  RD 23       CP    '#'           {Number plates?
2D5A  28 F0       JR    Z,2D4CH       $yes, further evaluate number
2D5C  FE 2C       CP    ,             ;comma?
2D5E  20 LA       JR    NZ,2D7AH      ;no, number field parameter
2160  7A          LD    A, D          {bit of format flag for',' = f
2D61  F 40        OR    40H
263   57          LD    D,A
2D64  18 E6       JR    2D4CH         ;continue at 2D4CH

                  Determine the number of post-account locations

2D    7E          LD    A, (HL)       ;Load Characters
267   RD 23       CP    !O            {number plate?
269   EACH 2E     LD    '.            .
2D6B  28.90       JR    NZ,2CFDH      No,'.' spend
2D6D  0E 01       LD    , 1           ;Counter of decimal places: 1
2Dr   23          INC   HL            String + 1
2p070 0C          INC   C             ; counter t.NachkOHastelle +
2071  05          DEC   B             {String Length - 1
2D72  28 25       JR    Z.2D99H       {= 0 ? Yes, long-distance!
2D74  7E          LD    A (H)         ;Load Characters
275   23          INC   HL            {String pointer + 1
2D7   RD 23       CP    '#'           Number plates?
2D78  28F         JR    Z,2D70H       Yes!

                  door] d-parameter

2D7A  D5          PUSH  EN            ;Format flag on the stack
2D7B  11 97 2D    LD    DE,2D97H      Set bounce address
2D7E  D5          PUSH  EN
2D7F  54          LD    D,H           ;String pointer in DE
2080  5D          LD    E,L
2081  FE SB       CP    5BH           last.Character = 'Up arrow'   ?
2D83  Co          RET   NZ            ineim, continue at 2097H
2084  BE          CP    (HI)          also the next 3 characters?
2D85  CO          RET   NZ            {no!
2086  23          INC   HI.
2087  BE          CP    (HI)
```

```
2D88  co      RET   NZ          No!
2D89  23      INC   Hl
2DBA  BE      CP    (Hll
2DBJ! Co      RET   NZ          No!
2DBC  23      INC   Hl
2DBD  78      LD    A,B         {Stringlength €4 ?
2D8E  D M4    Say!  4
2D90  D8      RET   C           Yeah, ignore the four arrows
2D91  D1      POP   EN          ;VOIR Stack return address
2092  D1      POP   EN          Load {Format Flag
2D93  47      LD    B,A         1String Length - 4
2D94  14      INC   D           ;Rit 1 of the format flag for
                                ;Set exponent output
2095  23      INC   Hl          ;String + 1
2D9   CA      DEFB  0CAH        ;JP        dumay instruction
2097  EB      EX    DE,HL       String pointer in HL again
2098  D1      POP   EN          ;Load format flag
2099  7A      LD    A,D         ;Format flag in A
2D9A  21!     DEC   Hl…         {String-pointer- 1
2D09B 1C      INC   E           Number Field Length + 1
2D9C  E 8     AND   8           i'+' - Set Bit?
2D9E  20.15   JR    NZ,2DB5H    Yes!
2DA0  1D      DEC   E           {Number Field Length - 1
2DA1  78      LD    A, B        String length = ?
2DA2  B7      OR    A
2DA3  28 10   JR    1, 2DB5H    ija, format string evaluated
2DA5  7E      LD    A, (H)      ;Load Characters
204   6 2D    SUB   ,_,         ;minus sign?
2DA8  28O     JR    Z,2DB0H     3yes!
2DAA  FE      CP    0FEH        ;plus sign?
2DAC  20 07   JR    NZ,2DB5H    ;no, output!
2DAE  3E 08   LD    As8         iBit 3 of the format flag for   = 1
2D1.0 C 4     ADD   y %         $bit 2 of the format flag for
                                ;sign behind number= 1
2DB2  82      ADD   A,D         ;to Link Total Format flag
2DB3  57      LD    D,A         ;and in D
2DB4  05      DEC   B           ;String Length - 1
2DB5  El      POP   HL          Load {Programme pointer
2DBb  Fl      POP   AF          ;Reload flags
2DB7  28.50   JR    1.2E09H     ;End of instruction? Yeah,
2DB9  CS      PUSH  BC          ;String length + Nachk0111111
                                to the stack
2DBA  D5      PUSH  EN          iFormat Flag + Null Field
                                to the stack
```

```
2DB:S  CD 37 23        CALL   2337H          Evaluate expression (to
2DBE   Dl              POP    EN             ;Load Flag+Number Field Length
2DBF   Cl              POP    :SC            ;String+Post Conast. load
20C0   CS              PUSH   l\C            and again on the stack
2DC1   IT              PUSH   HL             ;Programme pointer to stack
2DC2   43              LD     B,E            ;Number field length in B
2DC3   78              LD     +B             {+ decimal places
2DC4   B1              ADD    As C
2DC5   RD 19           CP     19H            ;total field length>= 25?
2DC7   D2 4A IE        JP     NC, 1E4AH      ia, FUNTION CODE - Error
2DCA   7A              LD     A, D           {Format flag in A
2DC1\  Fb B0           OR     80H            ;Set Bit 7 {Formatting')
2DCD   CD BE OF        CALL   0Fl\EH         i number in torr,dated string, etc.
20D0   CD A7 28        CALL   28A7H          and spend it
2D3    EI              POP    HL             Load {Programme pointer
2004   2:S             DEC    HL             {Programme pointer - 1
20D5   D7              RST    10H            Load next character
2DD    37              SCF                   Set Carry (for CRl
2D07   28 0D           JR     Z,2DEbH        {Anoel Sungsende? Yeah, jump
2DD9   32 EN 78        LD     478DEH), A     ;Save Characters
2DDC   FE 3.B          CP                    Semicolon 7
2DDE   28.05           JR     Z,2DE5H        5yes!
2EN0   FE 2C           CP                    ; Coma?
2EN2   C2 97 19        JP     NZ,1997H       no, SYNTAX ERROR
2EN5   D7              RST    10H            next character
2ENb   Cl              POP    BC             Load String Counter
2DE7   EB              EX     DE,HL          ; Programme pointer in DE
2EN8   EI              POP    Hl...          ;Load String
2DE9   E5              PUSH   Hl...          ;and back to the stack
2DEA   FS              PUSH   AF             Flags on the stack
2DEB   D5              PUSH   EN             {Programme pointer to stack
2DEC   7E              LD     A,!            Original string length in A
2DED   90              SUB    :S             ; string length =number discarded.
2DEE   23              INC    HL             String + I
2DEF   4E              LD     C, HL)         ;Load string address in HL
2DF0   23              INC    HL
2DF1   b               LD     H, (Hl.)
2DF2   69              LD     L,C
2DF3   16.00           LD     D,0            Number of characters processed
2DF5   SF              LD     E,A            EN
2DFb   19              ADD    HL, DE         ;+String = Address of
                                             {Reststrings
2DF7   78              LD     A,B            Residual stringlength > 0?
2DF8   l\7             OR     A
```

```
2DF9  C2 03 2D        JP    NZ,2003H      Jan Next
2DFC  18 06           JR    2E04H

                    format string end

2DFE  CD 49 2E        CALL  2E49H         ;'+' out of Number Field
2E01  CD 2A 83        CALL  032AH         ;Print Characters
2E04  E1              POP   HL            Load {Programme pointer
2E5   F1              POP   AF            {Reload flag, statement end?
2EO   C2 CB 2C        JP    NZ,2CCBH      {no, next number with same
                                          Format String
2E09  DC FE 20        CALL  C, 20FEH      Carry set? yes, output CR
2E0C  E3              EX    (SP>,HL       ;Programme pointer to stack
                                          ;Load String
2E0D  CD DD 29        CALL  29DDH         ;String from String
                                          ;and Clear Cache
2E10  E1              POP   HL            Load {Programme pointer
2E11  CJ b9 21        JP    2169H         ;Output flag on screen
                                          and done!

                    String Formatting

2E14  0E 01           LD    C,1           ;Indent '!',Character count= 1
2E16  3E              DEFB  3EH           iLD A,Fl  dummy instruction
2E17  F1              POP   AF            ;'7.', stack correction
2E18  05              DEC   B             {String Length - 1
2E19  CD 49 2E        CALL  2E49H         ;'+' Outside the box
2E1C  E1              POP   HL            ;Load programme pointer
2E1D  F1              POP   AF            ;Reload flag
2E1E  23 E9           JR    Z.2E09H       ;End of instruction? Yeah,
2E20  C5              PUSH  BC            ;String length on the stack
2E21  CD 37 23        CALL  2337H         ;Evaluate Expression
                                          {string to format)
2E24  CD F4 MA        CALL  0AF4H         ;Yield. No string? TYPE MISMATCH
2E27  C1              POP   BC            ;Load String Length
2E28  C5              PUSH  BC            ;and back to the stack
2E29  IT              PIJSH HL            ;Prog pointer to stack
2E2A  2A 21.79        LD    H, (7921H)    {String pointer of the to format
                                          ;Load Strings
2E2D  1               LD    B,C           {Number of characters greater than 2.
                                          door LEFT$ in B
2E2E  0E 08           LD    C,            ;Left Oftset =
2E38  C5              PUSH  BC            ;both parameters to the stack
2E31  CD 68 2A        CALL  2A68H         Format {string. The 1.
                                          ;o. The first '7.' disconnect
2E34  CD A 28         CALL  28AAH         ;Output Forated String
```

```
2E37 2A       LD      HL, 7921H)      {HL string pointer
2E3A F1       POP     AF              ;Number of characters in A
2E3B 9        SUB     (HU             Format string length
2E3C 47       LD      B,A             ;- Number of characters in
2E3D 3E 28    LD      A,' '           Load spaces
2E3F M4       INC     B               ; difference= 0?
2E40 05       DEC     B
2E41 CA D3    JP      Z,2DD3H         Yes; further
2E44 CD 2A    CALL    032AH           ;Print spaces
2E47 18 F7    JR      2E40H           {continue at 2E4MH
```

USING Subprogramme

```
2E49 F5       PIJSH   AF              ;AF secure
2E4A 7A       LD      A,D             ;Bit set in format flag?
2E4B B7       OR      A               i(can only be '+' bit)
2E4C 3E 2B    LD      A,'+'           Load in '+'
2E4E CA 2A    CALL    NZ.032AH        {Jas out
2E51 F1       POP     AF              ~F Reload
2E52 C9       RET
```

%#~##%%%#%~%% }With i !tH&

Existing row **output** when autotyped

```
2E53 60       LD      H,B             ;Line Address in HL
2E54 9        LD      L,C
2E55 23       INC     HL              ;Line Address After Line
2E5b 23       INC     HL
2E57 23       INC     HL
2E58 23       INC     HL
2E59 CD 7E    CALL    2B7EH           ;Edit line &.          in 1/0
2E5C 2A A7    LD      HL, (78A7Hl     ;Load butter address
2E5F CD 75    CALL    2B75H           {Print Line
2E62 C9       RET
```

Hffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

NODE - Execute Statement

```
2Eb3 CF       RST     8               '('                    2
2Eb4 28       DEFB    '('
2E5  CD 1C    CALL    2B1CH           Evaluate {operand in brackets
2Eb8 B7       OR      A               {FASHION (0)?
2E69 28 12    JR      Z,2E7DH         Yeah, jump
2EB  3D       DEC     4               ;- 1
2E6C 28 03    JR      Z,2E71H         4= 0? Yes, FASHION {L)!
```

```
2E6  CJ     JP          1E4AH              No, FUNCTIN CODE Error
            NODE ( 1 l)
2E7  16     LD          D,0                ; delete character= X'00'
2E7  YE     LD          A, 73BH)           Load output latch bytes
2E7  F      OR          8                  ;Set Bit J
2E7  32     LD          (783BH>,A          and save back
2E7  18     JR          2E87H              Continue at 2EBlH
            Set NODE (0)
2E7  1      LD          D, '               Delete = Blank
2E7  YE     LD          A, (783BH)         Load {0utputLatch byte
2E8  E      AND         0F7H               ;Delete Bit J
2E8  32     LD          (783BH),A          Save Jund
2E8  32     LD          (6800H),A4         Output Latch Byte
2EB  E5     PUSH        HL                 ;Programme pointer to Stark
2E8  21     LD          HL 70MH            Load start address
2EB  01     LD          BC, 2048           ;Counter = 2K byte
2E9  7A     LD          A, D               wildcard with delete character
2E92 77     LD          (H),A              to fill
2E9  23     INC         HL                 Image Address + 1
2E9  0B     DEC         BC                 ;counter - 1
2E9  78     LD          A+M                = 8 ?
2E9  BI     OR          C
2E9  20     JR          NZ.2E91H           No, continue
2E9  EI     POP         HL                 ;Load the programme pointer
2E9  CF     RST         8                  {Completion With ')◦ 3
2E9  29     DEFB        'O'
2E9  c9     RET
            ********************************************************************************
            ***************************
            LIST add-on routine
            (Strings output)
2E9  RD     CP          ,.                 {Start of string?
2E9  CA     JP          Z,2EBYH            yes, continue at 2EB3H
2EA  117    OR          A                  Is it a token?
2EA  F2     JP          P.2B89H            ;no, next character
2EA  C3     JP          2,B94H             yes, continue at 2.B94H
2EA  7E     LD          + (HL)             Load Characters
2EA  B7     OR          A                  ;End of Line Test
2EA  23     INC         li.                { indicator + l
```

```
2EAC  02          LD    BC),A       Transfer i characters to Putter
2EAD  approx      RET   or          {End of Line, Finished
2EAE  RD 22       CP    '-          {Strengthening?
2EB0  CA 89 2B    JP    Z.2B89H     yes further at 28%
2EB3  03          INC   BC          ; Buffer + I
2EB   15          DEC   D           In a buffer full ?
2EB5  CB          RET   or          Yeah, stop
2ER   18 Fl       JR    2EA9H       next character in string
```

}# # # # # # itlf %### t3 # # # k # tl # Milk

Interrupt Service Routine

(Video controller stops every 20115)

```
2EB8  FS          PUSH  AF          ;Save register contents
2EB9  C5          PUSH  BC
2EBA  D5          PUSH  EN
2EBB  E5          PUSH  HL
2EBC  CD 7D 78    CALL  787DH       RA} - Expansion output
2EBF  CD 7B 3F    CALL  JF7BH       ;Invert screen if required
                                    ;Print Butter
2EC2  CD DC 2E    CALL  2EDCH       ;Output Cursor/ Blink
2EC5  CD FD 2E    CALL  2EFDH       ;Query Keyboard
2EC8  F5          PUSH  AF          ;Save read character
2EC9  21.39.78    LD    H, 7839     ;Address flag 2
2ECC  CB 46       BIT   0, HL)      ;Carriage-Return Flag set?
2ECE  CC 1 39     CALL  1, 301BH    No, output characters   (Echo)
2ED1  Fl          POP   AF          ;Load the character
2ED2  CD 30 34    CALL  3436H       ;Make Summer sound
2ED5  El          POP   HL          Restore register contents
2ED   D1          POP   EN
2ED7  Cl          POP   BC
2EDB  F1          POP   AF
2ED9  FB          EI                ;Turn the Interupts back on
2EDA  ED 4D       RETI              1RETURN from Interrupt
```

fffffffflllfllllfltllNIIIIH

Print/Flash Cursor

```
2EDC  34.39.78    LD    7839Hl      Load Flag 2
2EDF  C 47        BIT   @,A         {Carriage-Return Flag set?
2EE1  Cl          RET   NZ          Yeah, done
2EE2  21 41 78    LD    H., 7841H   ;Address flashing
2EE5  35          DEC   (H)         ;- 1
2EE   Cl          RET   NZ          ;Not Mull, done!
```

```
2EE7  3E 10     LD      A, 16           ;Reset flashing counter
2EE9  32.41.    LD      (7841Hl,A
2EEC  2A        LD      HL, (7820%)     Load cursor address
2EDF  3E 40     LD      , 4@H           ;Set inverse bit to A
2EF1  Af.       XOR     (HL.            ;Character to cursor-post. invert
2EF2  77        LD      (HL),A
2EF3  c9        RET
```

%%3%%%%%%%%% %%%% %i -#% #% % t #t MM

Import a character from the keyboard

(Keyboard DCB Call)

```
2EF4  CD FD     CALL    2EFDH           ;Evaluate keyboard
2EF7  FS        PUSH    AF              ;Save Characters
2EF8  CD 0E     CALL    2F0EH           ;Reset Flags
2EFB  F1        POP     AF              ;Load the character again
2EFC  C9        RET
```

%1%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%1With

Evaluate keyboard once

Exp.: A = ASCII character read in
A= if no key is pressed

```
2EFD  34 O      LD      A,16800Hl       all lines of the key matrix les
2FO0  F CO      OR      11000100B       ;Columns 6 u. Hide 7
2F02  2F        CPL                     ;A Register Invert
2F03  RD 08     CP      0               if = @,no key pressed
2F05  28 87     JR      Z,2F0EH         then delete all flags
2F07  CD 28     CALL    2F28H           ;Analyse Hatrix Lines
2F0A  B7        OR      A               ;no ASCII character?
2F0B  C2 D7     JP      NZ,05D7H        ;but, check for Nehrfach.
      05
```

ttttttttttttttttttfffffffffffHtHtHtHtHt

Reset Flag Bits

```
2F0E  21.38.7   LD      HL,7838H        Address Flag 1
2F11  CB 56     BIT     2, HL           ;Function flag set?
2F13  28 08     JR      Z,2F1DH         {no!
2F15  3A 3A     LD      A, (783AHl      ;Load time counter
2F18  B7        OR      A               f= 0?
2F19  28 02     JR      Z,2F1DH         yes, do not delete function flag
2F1B  CB 96     RES     2, (p.)         ;Delete Function Flag
2F1D  7E        LD      , {H)           Load iflag l
2F1   E 06      AND     00009 1 1 8B    ;Delete to Radio+ Lnv Flag
F
```

```
2F  32 38     LD        (7838Hl,A          ;and save back
2F  AF        XOR       A                  Delete iCharacterPUtfer Bl
2F  32.36     LD        (7836Hl,A
2F  C9        RET
27
              ************************************************************************
              *****************************
              Evaluate keyboard line by line

              Exp.: A = Key Code or Null
2F  21 FE     LD        HL.68FEH           ;Address keyboard line 1
2F  0E 08     LD        1.7                ;Line counter= 8
2F  06        LD        B                  Column counter =
2F  7E        LD        A,(HLl             {Load Line Content
2F  F 04      OR        0000/1 OO          ;Hide column 2 (special)
2F  1F        RRA                          lowest bit in carry
2F  3020      JR        NC,2F62H           i= 0, key found
2F  10 FB     DJNZ      2F32H              ;next bit (8x)
2F  CB 05     RLC       L                  ;Address next line
2F  D         DEC       C                  ;Line counter - 1
2F  20 FI     JR        NZ,2F2DH           ;> 0? yes, next line
2F  06 4      LD        M4                 ;Column counter= 4 (column 2)
2F  21 DF     LD        HL.68DFH           Addressing Keyboard Line 6
2F  7E        LD        A, (HL             ;Load Line Contents
2F  CB 57     BIT       2.A                ;'-' Press?
2F  28 10     JR        Z,2F56H            {yes!
2F  C3 05     RLC       L                  ;Address keyboard line 7
2F  7E        LD        A, <HL>            ;Load Line Contents
2F  CB 57     BIT       2A                 ;RETURN - Key?
2F  28 0D     JR        Z,2F5AH            Yes!
2F  C3 05     RLC       L                  ;Address keyboard line 8
2F  7E        LD        A, (HL)            ;Load Line Contents
2F  CB 57     BIT       2.A                ':' Press the key?
2F  28 04     JR        Z,2F5EH            Yes!
2F  AF        XOR       A                  with A = back
2F  C9        RET
2F
56
2F  8.03      LD        C,3                ;Line counter= 3 (door '-')
56
2F  18O       JR        2Fblli
2F  0E 02     LD        1.2                ;Line counter= 2 (door RETURN)
2F  18 02     JR        2FM
2F  0E 01     LD        C,1                ;Line counter= 1 (for':')
2F  F M4      OR        008001080B         Hide column 2 again
e
2F
62  5F        LD        E,A                ;Record line contents in E !!!
```

```
2F63  3E il6          LD    A,6          From column and row counters
2F65  90              SUl\  1\           Offset for the keyboard tables
2F66  CB 27           Sl.A  A            identify,
2F68  CB 27           Sl.A  A            i
2F6A  Cl\ 27          SLA   A            {A = 8 ( 6 - B L + 8 - C
2F6C  C 08            ADD   A, B
2F6E  91              SUB   C
2F6F  ED 43 42 78     LD    (78421),B    Remember Row and Column counters
2F73  22 44 78        LD    !7844Hl, HL  ;Row Address 111
2F76  21 D9 01        LD    HL,01D9H     ;Keyboard Table 1 (o. SHIFT) or
2F79  4F              LD    CA           ;Table offset in BC
2F7A  06 O            LD    B,0
2F7C  3A Fl\ 68       LD    A,(68FBH!    ;Load Keyboard Line 3
2F7F  CB 57           BIT   2.A          Pressed the iSHIFT key?
2F81  20 04           JR    NZ,2F8DH     No!
2F83  21.38.78        LD    HL.7838H     ;Address Flag 1
2F86  CB C            SET   0, (HL)      ;Set SHIFT flag
2F88  21 09 02        LD    H, 0209      ;Keyboard Table 2 (m. SHIFT) or
2F8B   18 3D          JR    2FCAH        ;Read Code from Table
2F8D  3A FD 68        LD    A, (68FD1il  ;Load Keyboard Line 2
2F90  CB 57           BIT   2.A          ;CTRL - Press?
2F92  2il 39          JR    NZ,2FCDH     No!
2F94  3A 7F 68        LD    A, (687FH)   ;Load Keyboard Line 8
2F97  CB 57           l\IT  2s           ;':' - Hit the key? (INVERSE)
2F99  28 0E           JR    NZ, 2FA9H    No!
2F9B  21.38.78        LD    HL.7838H     Address Flag 1
2F9E  CB E            BIT   S. (Hl )     iWAIT flag set?
2FA0  28.04           JR    NZ, 2FA6H    Yes, ignore the key press
2FA2  7E              LD    A, (HL)      Load Flag 1
2FA3  EE 22           XOR   001 000 1 8B Invert {INVERSE flag,
                                         ;Set WAIT flag
2FA5  77              LD    (HL),A       ;Save flag l
2FA6  AF              XOR   A            iA = il
2FA7  Cl              POP   BC           ;1. Undo Rear Rebound
2FA8  C9              RET                {two levels back
2FA9  21.38.78        LD    HL.7838H     ;Address flag 1
2FAC  CB FE           SET   7, H.)       Set iCONTROL flag
2FAE  CR 56           BIT   2 HL)        ;FUNCTlON flag set?
2FBO  28 O5           JR    1.2FB7H      {no!
2FB2  21.69.02        LD    HL, 29H      ;Keyboard Table 4 <Features)
2FB5  18 13           JR    2FCAH        ;Read Code from Table
2FB7  3A BF 68        LD    A, (68:BFHJ) ;Load Keyboard Line 7
```

```
2FBA   CB 57          BIT   2.A             ; RETURN depressed 7
2FBC   20 07          JR    NZ, 2FC5H       {ne1n'
2FBE   C D            SET   2s (H)          ;FUNCT!(Set JN flag
2FC0   AF             XOR   A               ;Reset time counter
2FC1   32 3A 78       LD    (7834/1,A
2FC4   C'i'           RET
2FC5   CR 9           RES   2, HL)          Delete iFUNCTION flag
2FC7   21.39.02       LD    HL,023'i'H      ;Keyword Table 3
2FCA   0'i'           ADD   HL,BC           ;table adr. + Offset
2FCB   7E             LD    #, (HL)         ;Read Code from Table
2FCC   C'i'           RET                   ;back
2FCD   3A 38.78       LD    A, (7838H)      ;Load flag 1
2FD0   E 81           AND   10000001B       ;SHIFT- o. CTRL flag set?
2FD2   28F            JR    Z,2FCAH         no, find Cbde
2FD4   AF             XOR   A               ;Ignore key
2FD5   El             POP   Hl              1. Reverse Rec. remove
2FD    C'i'           RET                   two levels back
```

%%%%#% % # } % # t i# kt lt }

Key Repeat

```
2FD7   21.38.78       LD    HL.7838H        ;Address flag 1
2FDA   CB E           BIT   5,(HLJ          ;ARTE flag set?
2FDC   28 25          JR    1, 3083H        {no!
2FDE   3A 3A 78       LD    A, (783H)       i Time Counter +1
2FE1   3C             INC   A
2FE2   32 3A 78       LD    (783H),A
2FE5   FE 2A          CP    42              i= 0.84 seconds?
2FE7   28 1.12        JR    Z,2FEBH         Yes!
2FE'i  AF             XOR   A               back with A =
2FEA   C'i'           RET
2FEB   7E             LD    A, HL)          Load Flag 1
2FEC   E DF           AND   11011111B       Delete {HARTE Flag
2FEE   F 4            0R    01000000R       ;REPEAT flag
2FFl!I 32 38 78       LD    (7838HJ,A       ;Save Flag 1
2FF3   AF             XOR   A               ;Reset time counter
2FF4   32 3A 78       LD    (783AH!iA
2FF7   CR 66          BIT   ,{HL)           ;2 keys pressed?
2FF'i' 20.04          JR    NZ,2FFFH        Yes!
2FFB   34 3 78        LD    A, 7836H)       ;Load key code from character buffer
2FFE   C'i'           RET                   ;back
```

```
2FFF  3A 37.78        LD    A(7837H)        ;Code for 2, key load
3002  C9              RET                   back
3003  CB 76           BIT   6, (HL)         ;REPEAT flag set?
3005  20 07           JR    NZ, 300EH       yes'
3007  CB EE           SET   5, (Hl)         ;Set WAIT flag
3009  AF              COLOUR A              ;Reset time counter
300A  32 YES 78       LD    (7834H),A
300D0 C9              RET                   with A = @back
300E  3A 3A 78        LD    A, (783AH)      ;Time counter+ 1
3011  3C              INC   A
3012  32.34.78        LD    !783AH1,A
3015  FE              CP    6               = 0.12 seconds?
3017  28 DA           JR    Z,2FF3H         ija'
3019  AF              XOR   A               with A= @
3814  C9              RET                   ;back
```

%%%#%#% lt # # # i # # ##i

Show entered character on screen

!ECHO - Function)

```
301B  B7              OR    A               A = 0? !no character)
301C  approx          RET   or              Yeah, done
301D  F5              PUSH  AF              ;Save Characters
301E  CD 39 30        CALL  3039H           ;Print Characters
3821  Fl              POP   AF              ;Load the character again
3022  FE D           CP    0DH             Was it a carriage return?
3024  CS              RET   or              Yes, done
3025  FE 01           CP    1               The BREAK?
3027  approx          RET   or              Yeah, ready too
3828  3A 39.78        LD    A, 7839H)       Load flag 2
302:B CB 47           BIT   0,              ;CR flag set?
382   CO              RET   NZ              Yeah, done
302E  3E 20           LD    A.32            ;Flash counter to dopp. Value !Pause)
38/38 32.41.78        LD    (7841Hl,A
3833  2A 20.78        LD    HL, (7820Hl     ;Load cursor address
303   C3 B2 3E        JP    3EB2H           ;Print character inverted
```

##Ei # ####t# %# MM # # # t lt

Direct output of a character or keyword

```
3839  21.38.78        LD    HL.7838H        ;Address flag 1
303C  CB 7E           BIT   7, (Hl...)      Set iCONTROL flag?
303E  CA 57 31        JP    Z,3157H         ;no, output characters
3041  B7              OR    A               Is it a token?
```

```
3 F     JP          P.3157H             {n1n, return characters
3 F     PUSH        AF                  Save {Characters
3 I     SUB         80H                 ;Delete Bit 7
3 3     INC         A                   ;+ 1 as a word counter
3 4     LD          B,A
3 2     LD          HL,1650H            ;Keyword tab address.
3 2     INC         HL                  ;Address pointer+ 1
3 C     BIT         7, (HU              {Start of a new word?
3 2     JR          Z.304DH             No, continue
3 1     DJNZ        304DH               ; word count - 1 = 0?
3 7     LD          A, HL)              ;yes, keyword search
3 C     CALL        3082H               Print {Characters from Table
3 7     LD          A, HL)              ;next character
3 C     BIT         7.A                 {new keyword?
3 2     JR          Z,3055H             No, continue to spend
3 F     POP         AF                  ;character from stack              1
                        Check to see if the token matches the character ' ('.
3 (     LD          M.22                ;Number of table elements
3 2     LD          HL.0299H            ;Table start address
3 E     CP          (HL)                ;Token =table entry?
3 2     JR          1, 307CH            Yeah, jump
3 2     INC         HL                  Address pointer + 1
3 1     DJNZ        3063H               ;next table entry
                         not in table,              DEF Special Treatment
3 F     CP          0RMH                DEF-Tol: en ?
3 c     RET         NZ                  No, done!
3 3     LD          A,'                 ;yes, to 'DEF FN'                co
3 C     CALL        3082H
3 3     LD          A, 'F'
3 C     CALL        3082H
3 3     LD          A,'N'
3 C     CALL        3082H
3 C     RET                             Ready!
        to prepare a character for output
3 I     AND         7FH                 Delete with 7
3 E     PUSH        HI.                 ;Stack string pointer
. C     CALL        3157H               ;Print Characters
3 E     POP         HL                  ;Load String
3 2     INC         HL                  String + !
3 C     RET
8 .
```

ff+fftfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffft
fftfffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffft
fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffft

```
                        Buffered output of characters
308B   F5                PUSH   AF              ;Save Characters
308c   34 3B 78           LD     A, (783.BHl     il/0 - Load Latch. Byte
308F   C.B SF             .BIT   3               {System in Graphics Mode
3091   28 17             JR     Z, 30H          No!
3093   E6 F7             AND    0F7H            .Delete Bit 3
3095   32 3 78           LD     (783BHl,A       il/0 - Save Latch. Byte
3098   32 00 68          LD     ( 6B80MH),A     ;toggle to text mode
389.B  01 00 02          LD     .BC,512         i.Text Mode Video Memory
309E   21.080.70          LD     H, 700H         delete (512 bytes)
30A1   CD BE JE          CALL   3EBEH           ;Delete Characters
30A4   23                INC    HL              ;Next Address
3845   I.B               DEC    BC              ;counter - l
30A6   79                LD     To              i= 0? (ready)
30A7   .B0               OR     .B
30A8   20 F7             JR     NZ, 3041H       ;no, next character
3844   Fl                POP    AF              ;Load character to output
38AM   21.39.78          LD     H., 7839H       ;Address flag 2
3IAE   CB BU.            :m     5, (Hl)         Set the initialisation flag?
38B0   CA O 31           JP     2, 310H         ;no, direct output of character
38B.3  RD 20             CP     20/i            Is it a control character?
30B5   D2 C0 30          JP     N, 30C0H        No!
                        111 types until print buffer is fully output
31B8   F5                PUSH   AF              ;Save Characters
308B9  3A AF 7A          LD     A, {7AFH)       ;Load buffer counter
30.BC  .B7               OR     A               = 0?
30.BD  20 FA             JR     NZ,30B9H        {no, wait!
30BF   Fl                POP    AF              Reload Characters
30C0   F3                DI                     Disable linterupts
30C1   2A BA 7A          LD     HL, (7A.B0H)    ;Load buffer pointer
30C4   77                LD     (HL),A          ;Transfer character to buffer
30C5   23                INC    HI.             ;Buffer pointer+ 1
30c    22.B0 7A          LD     (7 A.B0H), HL   and save back
30C9   21 AF 7A          LD     HL,7AAFH        ;Address buffer counter
30CC   34                INC    (HL)            + 1
30CD   F5                PUSH   AF              ;Save Characters
38CE   3A From 78        LD     A, (78Al)       Pointer on image output column
30D1   8                 ADD    A, (H)          ;+ Number of characters in buffer
30D2   32 AE 7A          LD     (7AAEH),A       ;= Line Position Pointer
30D05  Fl                POP    AF              ;Load the character again
30D    FB                EI                     ;Reset Interupts
```

```
3 R     CP          20H                 Was it in control characters?
3 D     JP          C,311AEH            jYes'
3 3     LD          A, 20               3 not more than 20 characters in
3 B     CP          (HLJ)               ;limit exceeded
3 D     JP          C,30DEH             Yes, wait until putter empties
3 C     RET
3 A     0R          A                   {Wait for b1s buffer to be empty
3 B     C           (HL/
3 2     JR          NZ,30E4H
3 C     RET                             finished
```

****

****

to print **an** interrupt buffer

```
J 3     LD          A,(7AAFH)           iBuffer counter       load
3 B     OR          A                   ;Buffer empty?
J C     RET         0r                  Yeah, done
3 4     LD          B,A                 in B as a loop counter
J 2     LD          HL, 7A1\2H          ;Load putter start address
J E     PUSH        HL                  ;and on the stack
J 7     LD          A, HL)              ;Load character from buffer
J 2     INC         H                   ;Putter Address + 1
3 I     PUSH        HL                  ;Buffer address on stack
J C     PUSH        BC                  ;Character counter on stack
3 C     CALL        310#                ;Print Characters
3 C     POP         BC                  ;Load character counter
3 E     POP         HL                  Load buffer address
3 1     DJNZ        30F2H               i Putter empty?
J E     POP         HL                  Yes, load buffer start address
3 2     LD          (7AB0HL,HL          ;Save as New Buffering
3 A     XOR         A                   ; Buffer Counter= 0
3 3     LD          (7MFH),A
3 C     RET
```

%# %%%4 %~~%1 1 1' t ##} Mt %

control of character output

```
3 C     CALL        030DH               ;Character to Cursorpos.          a
3 B     OR          A                   ;excerpt. Character= 0?
3 2     JR          Z,3110H             Yes, CR by RDLINE!
3 F     CP          0DH                 Is it a Carrigen return?
3 2     JR          NZ.31SAH            ;no, output characters
J F     PUSH        AF                  ;Save Characters
```

```
3111  2A 20 7B      LD    HL,(7B20HJ      ;Load cursor address
3114  3A A6 7B      LD    A,(7BA6HJ       Load {column pointer
3117  4F            LD    C,A             Transfer to B
311B  AF            XOR   A
3119  47            LD    B,A
311A  32 A6-78      LD    (784H)/A        {column pointer to column @
311D  ED 42         SBC   HL,BC           ;cursor address to start of line
311F  01 20 00      LD    BC.32           ;+ 1 line (32 characters)
3122  09            ADD   HL,BC
3123  7C            LD    , H             ; Address outside the image?
3124  FE 72         CP    72H
3126  F4 F3 33      CALL  P,33F3H         yes, roll up Picture 1
3129  22 20 78      LD    7820H), HL      ;Save new cursor address
312C  CD 53 00      CALL  0053H           ;Character from cursor position
312F  Fl            POP   AF              ;Load character to output
3130  B7            OR    A               ;CR by RDLINE {03E.3H) 2
3131  approx        RET   or              yes done
3132  CD AS 33      CALL  33A8H           ;Determine the status of the row
3135  RD 8          CP    B0H             ;Single line?
3137  CB            RET   or              {yes, done
3138  FE Bl         CP    BlH             First of a double line?
313A  20.05         JR    NZ,3141H        ;no, follow-up
313C  3D            DEC   A               ;both lines to single lines
 313  77            LD    (HL),A          Convert {
313E  23            INC   HL
313F  77            LD    (HL),A
3148  C9            RET                   ;finished
3141  3E 80         LD    A, 80lH         ;Single Line Sequence
3143  77            LD    (H),A           {uman de]n
3144  C9            RET                   {finished
```

#%ii Ei kt# # # i} } M

Print Characters or Run Control Functions

Introduction is 3157 or 315A

```
3145  CB 77         BIT   S               Inverted alpha. Signs?
3147  28.04         JR    Z.314DH         No, block graphics!
3149  C3 60 3F      JP    3F60H           Inverted Appearance Depends
314C  00            NOP                   From Background
314D  E6 8F         AND   BFH             ;Bits 4,5,6 delete
314F  47            LD    BA              ;Graphics in B
3150  3A 46 7B      LD    A, (7B46l)      ;Load colour code
3153  B             OR    B               Combine with graphic characters
3154  47            LD    B,A             ;Character in B
```

```
3155  18 5F            JR     31BH          {and spend
3157  CD 8D 3          CALL   030DH         ;Character to Cursorpos. again.
315A  R7               OR     A             ;bit 7 of the statement,
315   FA 45 31         JP     #, 3145H      Yeah, jump
315E  FE OD            CP     0DH           ;Carriage return?
3160  approx           RET    or            Yeah, done!
311   FE 08            CP     8             BACKSPACE?
3163  CA 27 32         JP     Z,3227H       Yes!
316   RD1              CP     1RH           ;Cursor up?
3168  CA 53 32         JP     Z,3253H       Yes!
316B  FE 0A            CP     0AH           ;Cursor down?
316O  CA 6D 32         JP     Z.326H        {yes!
3170  RD 08            CP     8             ; Cursor left?
3172  CA 27 32         JP     Z,3227H       {yes!
3175  FE 09            CP     9             ;Cursor right?
3177  CA B8 31         JP     Z, 31:BBH     Yes!
317A  FE 01            CP     1             BREAK?
317C  approx           RET    or            Yeah, done
317D  FE 7F            CP     7FH           RUIOUT?
317F  CA C 33          JP     Z.33CBH       Yes!
3182  RD 15            CP     15H           ;INSERT?
3184  CA C6 32         JP     1.32Ci        Yes!
3187  RD 18            CP     18H           ;Left arrow?
3189  CA 27 32         JP     Z,3227H       {yes!
318C  FFE 19           CP     19H           ;Right arrow?
318E  CA B8 31         JP     Z,31BSH       Yes!
3191  FE 1B            CP     1BH           ;Up arrow?
3193  CA 53 32         JP     Z,3253H       Yes!
319   FE 1C            CP     lCH           ;Cursor to top?
3198  CA 87 32         JP     Z,3287H       Yes!
319   FE 1D            CP     lH            ;Cursor to line start?
319D  CA bit 32        JP     1.32B4H       Yes!
31A0  FE 1F            CP     1FH           ;Delete Image?
31A2  CA 92 32         JP     1, 3292H      Yes!
31A5  RD 28            CP     2%#           ;Ignore remaining control
31A7  F8               RET    I'l
31A8  CJ CA JE         JP     3ECAH         continue with JECA (Backpack>
31AB  21.38.78         LD     HL.7838H      ;Address flag 1
31AU  CB 4E            BIT    1(H)          ilNVERSE flag set?
31B0  El               POP    Hl            ;Clean Stack
31B1  28 02            JR     Z,31B5H       No!
31R3  F 40             OR     4Wed          Yes, invert characters
31B5  47               LD     B,A           ;Character in B
```

```
J1B6  78          LD    A+B          ;Transfer character to A
31B7  77          LD    (HL),A       ;Output to Screen
31R8  CD BF 31    CALL  31BFH        ;Move cursor to a location
31BB  CD 50 80    CALL  0050H        oak to Cursorpos. secure
31RE  C9          RET
```

%ii±ME} Mi% tt % 3%3%±± #%}

Move the cursor to a character position

```
31BF  3A From 78  LD    A, (78ABH)   Load Column Pointers
31C2  3C          lNC   A            + 1
31CJ  RD 20       CP    32           iaa beginning of next line?
31C5  20 2B       JR    NZ,31F2H     no'
31C7  CD AB 33    CALL  33A8H        ; Get Row, Tu
31CA  FE 81       CP    81H          first of two lines?
J1CC  28 23       JR    Z,31F1H      Yes!
31CE  B7          CIR   #            4second of two lines?
31CF  20 35       JR    NZ.3216H     No!
31D1  47          LD    M,A          !Status in B
3102  34 39 78    LD    A, (7839H)   Flag 2 load
3105  CB 47       BIT   0,A          ;CR flag set?
31D7  78          LD    A,B          {Status back in A
31D8  C8          RET   or           ija axial two lines!
31D9  AF          XOR   A            ;next line= next line
31DA  23          INC   HL
J1DB  77          LD    (HL),A       {(= 08)
31DC  23          INC   HL           HL to status d. n.    Line
31DD  IT          PUSH  HL           Remember and
31EN  ED '+BA'+ 78 LD   BC (7844H)   Is this the last line?
31E2  8B          DEC   BC           ;(This routine collides)
31E3  0B          DEC   BC           {{ait programmes that do not)
31E4  B7          OR    A            (start at default)
31E5  ED 42       SBC   HL,BC
31E7  El          POP   HI...        ;Load the status of the line
31E8  38.07       JR    NC,31F1H     Yes; last line
31EA  7E          LD    To (lHL)     ; Status of row= 110?
31EB  B7          OR    A            if yes, follow-up line
31EC  20 03       JR    NZ,31F1H     {no!
31EE  EACH 80     LD    A,80H        ;Single line label.
31F0  77          LD    (HL),A
31Fl  AF          XOR   A            Column hands =
31F2  32 From 78  LD    (78ABh,A     Move {A to Column Pointers
31F5  2A 20.78    LD    H, 7820%)    ;Load cursor address
```

```
J1FB  01 01 00        LD     BC.I          + 1
J1FB  09              ADD    HL,BC
J1FC  7C              LD     A,H           outside the image?
31FD  FE 72           CP     72H
31FF  F4 FJ 33        CALL   P,33F3H       Yes, roll up one line
3202  22 20 78        LD     (7820HL,HL    ;Save Cursor Address
3205  C9              RET                  ;finished
3206  F5              PUSH   AF            ;111erken status
3287  ED SB 20 78     LD     EN, (7820H)   ;Load cursor address
320B  13              INC    EN            + 1
320C  7A              LD     A, D          ; outside the picture?
320D  FE 72           CP     72H
320F  28 10           JR     1.3221H       Yes!
3211  IT              PUSH   HL            {Status Address on Stack
3212  21.39.78        LD     HL.7839H      {Address Flag 2
3215  CB 46           BIT    0, H.)        {CR - Flag set ?
3217  28 07           JR     NZ.322tlH     Yes!
3219  CB 66           BIT    +,H)          ;1NUT - Flas set?
321B  28.03           JR     NZ.322tlH     Yes!
321D  CD 2C 33        CALL   332CH         to roll back a row
3220  El              POP    HL            Load Status Address
3221  Fl              POP    AF            ;Load Line Status
3222  3C              INC    A             ; Set Status= 81
3223  77              LD     (HL)+A        two-line!
3224  C3 D9 31        JP     31D9H
```

%%%%4%%%~%~#% # MM With MM

cursor one character to the left

```
3227  34 A 78         LD     A, (78H1      Load Column Pointers
322A  3D              DEC    4             ;- 1
3223  F2 35 32        JP     P,3235H       the same line!
322E  CD AS 33        CALL   33A8H         ;Get Row Status
3231  B7              OR     A             is this a follow-up line?
3232  C               RET    NZ            ;no, don't go back
3233  JE lF           LD     A.31          Column pointer to last column
3235  32 A 78         LD     (78A6H),A     and save
3238  01 01 00        LD     BC1           ; cursor address - 1
323B  2A 20.78        LD     HL, (7821H L)
323E  AF              XOR    A
323F  ED 42           SBC    HL,BC
3241  7C              LD     A, H          ; outside the picture?
3242  FE 78           CP     71H
```

```
3 D     JP          C, 324EH            Yes!
3 2     LD          (7820HL,HL          ;new cursor address returned
3 C     CALL        0053H               ;Character to Cursorpos. secure
3 C     RET
3 F     XOR         A                   Column hands = @ 1 . column)
3 3     LD          4784H),A
3 C     RET
2 9
```

#I% %##i}% ME % % # ± # ## 3 #%

Move cursor one line up

```
3 2     LD          HL.7839H            ;Address flag 2
3 C     BIT         (HL.)               ;INPUT flag set?
3 C     RET         NZ                  Ijas inadmissible!
3 0     LD          C.32                Line Length
3 2     LD          Hl., (7820H)        ;Load cursor address
3 A     XOR         A                   ;Delete Carry Flag
3 E     SBC         H,                  ;cursor address - 1 line
3 7     LD          A, H                off-screen?
3 F     CP          70H
3 F     RET         11                  $yes, don't go!
3 2     LD          (7828H), HL         ;Save Cursor Address
3 C     CALL        0053H               ;Character to cursorpas. secure
3 C     RET
2 9
```

%%#% Mi% % #t t # t

cursor one line down

```
3 2     LD          HL.7839H            ;Address flag 2
3 C     BIT         4, (HI)             Set iINPUT flag?
3 c     RET         NZ                  Yes, not allowed!
3 0     LD          BC, 32              Load Length of Line
3 2     LD          HI, (7820H)         ;Load cursor address
3 0     ADD         HL,BC               ;+ one line
3 7     LD          A, H                off the screen?
3 F     CP          72H
3 F     CALL        P, 3424H            ;yes, roll up one line
3 2     LD          (7820H) ,HL         ;Save Cursor Address
3 C     CALL        0053H               ;Character to Cursorpos. secure
3 C     RET
2 9
```

"**********

****

To the top of the screen.

```
3287  21 00 70          LD     HL, 7000~       ;Load image start address
328A  22 20 78          LD     (7820H0, HL     ; to cursor address
328D  AF                XOR    A               Column hands =
328E  32 A6-78          LD     (78AH)
3291  C9                RET
```

%%% f % i %#kt # # # # t # # #Mt3%

```
                        Delete Screen 11
3292  21 00 70          LD     HL, 7000H       ;Load image start address
3295  22 20 78          LD     47820H) ,HL     ;in cursor address
3298  01 00 02          LD     BC.512          ;Length of Text Store
329B  CD BE 3E          CALL   3EBEH           Delete a character
329E  23                INC    HL              ;Image Address+ 1
329F  0B                DEC    BC              Length-1
32A   79                LD     A,              On the fine?
32A1  B0                OR     B
32A2  20 F7             JR     NZ.329BH        no, next character
32A4  AF                XOR    A               Column hands =
32A5  32 A 78           LD     (78A1),A
32A8  6 10              LD     B, 1            ;counter= 16 (count  Rows)
32AA  3E 80             LD     A,80H           Load one line status
32AC  21 D7 7A          LD     HL,7AD7H        ;Status byte 1. Address Row
32AF  77                LD     (HL)+A          ;Status= Single Line
32B8  23                INC    HL              ;Next row status byte
32B1  10 FC             DJNZ   32AFH           ;edited all lines?
32B3  C9                RET                    Yeah, done!
```

%%1%%%%%%%%%%%i%%%ttttttttttttte

```
                        cursor to start of line
324   2A 20.78          LD     HL, (7820H1     ;Load cursor address
32B7  3A A 78           LD     A, (78A6H1      Load Column Pointers
32BA  4F                LD     CA              to BC
32R   AF                XOR    A
32BC  47                LD     ,A
32BD  32 A 78           LD     784H),A         {column pointer =
32C0  ED 42             SBC    HL,BC           ;Cursor Address - Column Pointer
32C2  22 28 78          LD     (7828H) ,HL     ;Save new cursor address
32C5  C9                RET
```

%l # Wed %lt #% 1 % r} with %}%

```
                        INSERT - Function
```

```
32C6  CD A8 33        CALL   33A8H          ;Determine the status of the line
32C9  FE 81           CP     81H            First of two lines?
32CB  28.31           JR     Z.32FEH        Yes!
32CD  34 A 78         LD     A, (78A6H)     Load {column pointer
32D8  RD 1F           CP     31             {at the end of the line?
32D2  2B 25           JR     Z,32F9H        Yes!
32D4  4F              LD     CA             Columns in BC
32D5  AF              XOR    A
 32D  47              LD     B,A
32D7  2A 20.78        LD     HL, (7820H)    i Load cursor address
32DA  ED 42           SBC    HL,BC          ;- Column hands= Top of Line
32DC  0! 1F 00        LD     RC,31          ;last character of the line adress.
32DF  09              ADD    HL,BC
32E0  CD E9 3E        CALL   3EE9H          Test {last character in line
32E3  20.14           JR     NZ,32F9H       unequal spaces
32E5  E5              PUSH   HL             ;Last character address
32E6  Dl              POP    EN             ;in
32E7  2B              DEC    HL             iHL = penultimate character
32EB  3A A6 7B        LD     A, (784H)      ;Load column pointer
32EB  4F              LD     CA             ;Line Length - 1 - Column
32EC  3E 1F           LD     A.31           in BC for Block-Move LDDR
32EE  91              SUB    C
32EF  4F              LD     C,A
32F0  ED BB           LDDR                  iab Cursorpos.1 characters right
32F2  CD F6 3E        CALL   3EF6H          Insert spaces
32F5  32 3C 7B        LD     (783CH),A      Save Characters
32FB  C9              RET

32F9  CD A8 33        CALL   33A8H          Determine {Status of Row
32FC  7               OR     A              ;Sequential?
32FD  C8              RET    0r             Yes, done
32FE  FE O            CP     80H            ;Single line?
3300  28 1E           JR     Z,3320H        ija'
3302  3A A 7B         LD     A,(78A6Hl      Load Column Pointers
3305  4F              LD     C,A            ;in BC
 330  AF              XOR    A
3387  47              LD     B+A
330B  2A 2078         LD     HL, (7820H)    i Load cursor address
330B  ED 42           SBC    HL,BC          ;- Column hands = Top of row
338D  01 3F Oe        LD     BC.63          j+ 63
3310  09              ADD    HL,BC          i= End of double line
3311  CD E9 3E        CALL   3EE9H          last character = empty?
3314  c8              RET    NZ             no, no pasting possible
3315  E5              PUSH   Hl...          ;End Address
```

```
3316  Tue              POP   EN            1n EN
3317  2.B              DEC   HL            {Hi = Home - 1
3318  YES A6 78        LD    A, (7BA6H)    ;Number of characters to move
3311l 4F               LD    C,A           ;and in .BC
331C  JE 3F            LD    + 63          {(= 63 - column pointer)
331E  18 CE            JR    32EEH         ;Double Line 1 character scroll
                                           right
3320  IT               PUSH  HL            ;Status address on the stack
3321  CD 2C 33         CALL  332CH         Roll Screen 1
3324  El               POP   HL            Load iStatus Address
3325  3E 81            LD    A,BlH         ;Status of 1. Row= Set 81
3327  77               LD    (HL),A        ▬
3328  23               INC   HL            ;New Line Status
3329  AF               XOR   A             ;Declare as follow-up line
332A  77               LD    (HL),A        j(= 00)
3328  c9               RET
```

#%4#}f## ' ## #k##%## E ## K ## ## ###t#k k ##

One line screen from cursor position

Scroll Down

```
332C  2A 2078          LD    HL, (7820)    ;Load cursor address
332F  7C               LD    A,H           ;Is it the last line?
3330  FE 71            CP    71H
3132  20 23            JR    NZ, 335FH     {no!
3334  7D               LD    A,L           ;in the bottom half
3335  FE EO            CP    0E0H          and 2. Check address byte
3337  DA 5F 33         JP    C, 335FH      ;not last line!
333A  3A From 78       LD    A, (78Al)     Load {Column Zener
333   F5               PVSH  AF            ;and remember the stack
3J3E  3A D7 7A         LD    A, (7AD7H)    ;Status of 1. Load Row
3341  FE 81            CP    81H           is it a double line?
3343  20.88            JR    NZ, 334H      Roll {no, 1 line only
3345  E5               PUSH  HL            ;cursor address to the stack
3346  CD F3 33         CALL  33FJH         ;Roll up image 1
3349  El               POP   Hl…           Load cursor address
334A  CD 17 03         CALL  0317H         ;cursor address - 1 line
334D  IT               PUSH  HL            ;Cursor address to the stack
334E  CD F3 33         CALL  33F3H         ;Roll up image 1
3351  Et               POP   HL            ;Load cursor address
3352  CD 17 83         CALL  0317H         ;cursor address - 1 line
3355  Fl               POP   AF            Load {column pointer
3356  32 A 78          LD    (786H),A      and write back old value
J359  D1               POP   EN            Rear Rear Rear, from Stack
```

```
335w  El          POP   HL          {Status address of stack
3358  2           DEC   HL          ;Status Address - 1 Line
335C  E5          PUSH  HL          ;Status address on the stack
335   D5          PUSH  EN          {back flywheel. to the stack
335E  C9          RET               ;finished

335F  YES From 78 LD    (A, 178A6H) ;Load column pointer
3362  4F          LD    CA          to BC
3363  AF          XOR   A
3364  47          LD    B,A
3365  ED 42       SBC   HL,BC       Cursoradr. - Column hands
3367  01 40 00    LD    BC,         + 4
3364  89          ADD   HL,BC       {= initial address of the     Line
336B  E5          PUSH  Hl.         {remember
33C   EB          EX    DE,HL       and in
336D  21 00 72    LD    HL, 7200~   ;Picture address+ 1
3370  ED 52       SBC   LT,DE             ;- DE= Number to be bytes
3372  E5          PUSH  HL          {works
3373  Cl          POP   BC          ;Load byte counter in BC
3374  21 DF 71    LD    HL, 71DFH   ;End address penultimate
3377  11 FF 71.   LD    DE,71FFH    ;End address last row
337A  79          LD    A;          Byte count = 8 ?
337B  9           OR    B
337C  28 02       JR    1, 3380H    yes, no postponement
337E  ED B8       LDDR              No, image one line down
3388  Et          POP   HL          ;load the line address of the new
3381  CD 82 3F    CALL  JF02H       delete mark
3384  00          NOP
3385  12          LD    (EN),A      ;Delete new line
3386  1B          DEC   EN
3387  10 FC       DJNZ  3385H
3389  CD A8 33    CALL  33A8H       ;Determine the status of the
338C  E5          PUSH  Hl.         ;Status address in BC
338D  Cl          POP   BC
338E  21 E6 7A    LD    HL,7AE6H    ;Last Line Status Address
3391  E5          PUSH  Hl.         to the stack
3392  B7          OR    A
3393  ED 42       SBC   HL,BC       ;- Status address is  Line
3395  E5          PUSH  HL          ;difference= bytes to move
3396  Cl          POP   BC          in B as byte counter
3397  Et          POP   HL          H = Status address Line
3398  E5          PUSH  HL          ;DE= Last Line Status
3399  D1          POP   EN
339A  2B          DEC   HL
```

```
339 E    LDDR                          {Status byte down one line
339 3    LD          A, (7AH)          if last line is not
33A F    CP          81H               ;Double Line
33A C    RET         NZ                {then finished and back
33A 2    LD          HL, (7820Hl       ;load other cursor address
33A 1    JR          335FH             and roll another line
```

****

****

Set Row Status

Exp.: A = Line State

Hl = Status Address

```
33A 3    LD          A, (78A6Hl        Load {column pointer
33A 4    LD          C,                to BC
33A A    XOR         A
33A 4    LD          B,A
334 2    LD          Hl, (7820Hl       ;Load cursor address
33B E    SBC         H,BC              ;- Column hands = Top of row
33B I    PUSH        Hl…               Line address in BC
33B C    POP         BC
331 7    LD          A, B              Find Line Number
33 E     AND         0FH               {= Line Address - 788MH
331 C    SRI.        A                 j/ 2
33B 7    LD          B,A
33B C    R           C                 ;= ZMr.      i left half byte of C
33B C    SRI.        C                 move to right half-byte
33B C    SRI.        C
33C C    SRL         C
33C C    SRL         C
3JC 2    LD          HL,7AD7H          Attachment address of the status
33C 0    ADD         H,BC              ;+ Row
3JC 7    LD          HL)               ;Load Line Status
33C C    RET
```

Mt#Mi%Mil MM#tiM Hi#

RUBOUT Function

```
33C C    CALL        3JA8H             ;Get Line Status
33C F    CP          81H               first of a double line?
330 2    LD          HL, (7820Hl       ;Load cursor address
33D E    PUSH        HL                ;in
33D D    POP         EN
3JD 2    INC         HL                {H to next character position
33D 3    LD          A, (78A6Hl        Load {column pointer
```

```
3309  4F          LD    CA          ;in C
33DA  29 13       JR    Z.33EFH     double-line, jump
33DC  RD 1F       CP    31          End of line?
33EN  28 08       JR    Z.33EBH     ijas just delete this character
33E0  JE 1F       LD    A.31        Line Length
33E2  91          SUB   C           ;- Column hands
33E3  4F          LD    C,A         in BC as counter
33E4  AF          XOR   A
3YE5  47          LD    B,A
33Eb  ED B0       LDIR              ;Row one character
33EB  CD F 3E     CALL  JEFbH       Space to end of line
3JEB  CD 50 00    CALL  050H        ;Save characters to cursor position
33EE  C9          RET               finished
3JEF  3E 3F       LD    A.63        Load {length of two lines
33F1  18 EF       JR    33E2        Shorten over two lines

                  FFF

                  Scroll up one line.
                  The last line is filled with spaces.
33F3  11 00 70    LD    DE, 70808H  Address of 1. Row in DE
33F6  21 20 78    LD    HL, 7820H   Address of 2. Line in HL
33F9  81 EO 0I    LD    C, 488      ;.Byte counter = 15 lines
33FC  ED B        LDIR              illd Scroll one line up
33FE  CD 02 3F    CALL  3Fll2H      ;prepare dl!s delete
3401  00          NOP               A = Blank, B = Bytes/Leile
3482  12          LD    (EN),A      ;Delete last line
3403  13          INC   EN
3484  10 FC       DJNZ  3402H
340b  21 D7 7A    LD    HL,7AD7H    ;Status table also
3409  E5          PUSH  HL          ;Roll up one line
3404  D1          POP   EN          ;DE= Status Line 1
348B  23          INC   HL          iHL = Status Line 2
€340  01 F O      LD    , 15        iBC = Line counter
340F  ED Be       LDIR              from DE to HL (length 15)
3411  1A          LD    A, (DE)     ;Load the status of the last line
3412  FE B1       CP    B1H         was this a double line?
3414  28.03       JR    NZ.3419H    ;no, last line= single line
341   AF          XOR   A           ;yes, last line= next line
3417  18.82       JR    341BH
3419  3E 80       LD    A,80H       ;'88' = ID f, single line
341B  12          LD    (EN),       ;new ID for last row
341C  AF          XOR   A           ;column pointer= 0
```

```
341D  32 From 78      LD     (78ABh.A
3420  21 E0 71        LD     HL, 71EH       {HL = beginning of last line
3423  C9              RET                   ifert 19

                      Depending on the status of the first row

                      ent111Each one or two line rollers.
3424  3A D7 7A        LD     A,(7AD7HJ      ;Status 1. Load Row
3427  FE 81           CP     81H            ;= Double Line ?
3429  CC F3 33        CALL   Z.33F3H        Yes, roll a line
342C  CD F3 33        CALL   33F3H          ;roll a line
342F  C9              RET

                      ##k Mi}#%'i %i %i## #±# } # # #it

                      Print Acoustic Signal When Entered
3430  21.39.78        LD     HL.7839H       Address Fla 2
3433  B7              OR     A              ;character entered'
3434  288B            JR     NZ,3441H       Yes!
343   CB CE           SET    1.H)           no, set BUZZER-FIag
3438  01 FF 03        LD     BC.03FFH       ;Hold
343B  0B              DEC    .BC            ;up to' End of vert.Sync pulse
343C  79              LD     A; C
3430  .B0             OR     .B
343E  21 FB           JR     NZ.343BH
340   C9              RET                   {finished

3441  CB 46           BIT    0,(HI)         iCarriage-Return Flag set?
3443  c8              RET    NZ             Yeah, done
3444  FE 0D           CP     0DH            ;Character= Carriage-Return?
344   28 06           JR     Z.344EH        1yes'
3448  FE if           CP     01H            ;character= BREAK?
34411 20.04           JR     NZ,J450H       No!
344C  CB D6           SET    2, (HL)        ;Set BREAK Flag
344E  CB C            SET    8, HL.)        Set Carriage Return Flag

3450  E5              PUSH   HI.            ;Flag 2 - Address on stack
3451  21 A 08         LD     HL, BA0H       Load Frequency
3454  0! % 00         LD     BC.6           ;Load sound duration
3457  CD SC 34        CALL   345CH          ;Print beep sound
345A  El              POP    HI.            ;Flag 2 - Load Address
345B  c9              RET                   finished

                      HffffHIIHIHHFFFFhfft
```

```
                        sound output
                        Eing: Hl...= Frequency
                              BC = Length of
345C  YES 3B 78         LD    A, (783BH)      i Load Latch Byte
345F  57                LD    D,A             Transfer to D
3460  CD 69 34          CALL  3469H           Output on
3463  0B                DEC   BC              ;Length - 1
34    79                LD    To C            1= 8?
3465  1\0               OR    B
346   20 F8             JR    NZ.3460H        {no, continue
3468  C9                RET                   Yeah, done
3469  C5                PUSH  .BC             Backs up sound time
346A  7A                LD    A,D             ;I/0 Latch Byte in A
346B  EE 21             XOR   21H             Bits @u. 5 Complement
3460  32 00 68          LD    (680%),A        ; I/0 Output Latch Byte
3470  E5                PUSH  Hl.             ;Frequency counter in BC
3471  Cl                POP   BC
3472  purple            DEC   BC              ;form lower half wave of sound
3473  79                LD    A,C
3474  Be                OR    B
3475  20 FB             JR    NZ.3472H
3477  7A                LD    A, D            I/0 Latch Byte in A
3478  32 00 68          LD    (680H), A       and spend
347:B IT                PUSH  Hl...           ;Frequency counter in BC
347C  Cl                POP   BC
3470  8B3               DEC   BC              ;Top half-wave of clay
347E  79                LD    A,
347F  B                 OR    B
3481  28 FB             JR    NZ.347DH
3482  Cl                POP   BC              Load duration
3483  C9                RET
```

% Hit i~%} With t # It t t titt

Part of the initialisation routine

```
3484  CD AO 3F          CALL  JFAilH          ;Verify CTRL Key Pressed
                                              and background colour,
3487  3E 20             LD    A,' '           ;Basic I/0 Latch Setting
3489  32 32 78          LD    (783:BH1,A      {Mark!
348C  32 08 68          LD    (6808H),A       ;and spend
348F  JE JC             LD    ,@              ;Time counter= 60
3491  32.34.78          LD    1783AH),A
3494  3E 10             LO    , 1             Initialise the flashing counter
```

```
349    32.41.78          LD     (7841H),A
3499   AF                XOR    A              ;Buffer counter = M
349A   32 AF 7A          LD     (7AAFH1,A
349D   21 B2 7A          LD     HL,7AB2H       buffer pointer
34A0   22 BO 7A          LD     (7AB0HL,HL
34A3   3E C9             LD     A, WC9H        RET t. interrupt vector
34A5   C3 37 3E          JP     3E37H          ;Colour= Yellow
34A8   C9                RET                   ;not used
```

%# # kt k #k# ##k # ### # M Mt # # # t } # #k

CSAVE Statement

Output to Cassette

```
34A9   F3                DI                    ;Disable Interrupts
34AA   0E Flr'l          LD     C, OFOH        ;Set BASIC Programme Kenner
34AC   CD 58 35          CALL   3558H          {Vr spanner and programme nose
34AF   RD 3A             JP     C,JAFEH        if Carry=1, BREAK.
34B2   IT                PUSH   HL            ;Save the programme pointer
34B3   01 9A 91          LD     RC,410         ;3 ms gap on tape
34B6   0B                DEC    BC
2:87   79                LD     A, C
34B8   B                 OR     B
34R9   20 FB             JR     NZ, 3BH
34BB   CD F8 3A          CALL   3AF8H          ;Check for BREAK key
34BE   DD 21 23 78       LD     IX,7823H       ; Addresses checksum bytes
34C2   2A A 7            LD     HL,(78A4H1     Load {programme start address
34C5   7D                LD     sL             ;LSB Load startup address in A,
34C6   CD 11 35          CALL   3511H          to print on cassette
34C9   DD 77 00          LD     41),A          jund in probe byte (LSB)
34CC   AF                XOR    A              ; in LS} checksum bytes
34CD   DD 77 01          LD     !IX+ll,A
34D0   7C                LD     A, H           il'tSB Load start address in A
3401   CD 11 35          CALL   3511H          ;Run on cassette
341>'1 CD BE 38          CALL   388EH          ;Add to Checksum111111e
34D7   EB                EX     DE,HL          ;Start address in DE
34D8   2A F9 78          LD     HI,(78F9H)     Load End Address
34DB   7D                LD     A,L            3LSB Load end address in A
34DC   CD 11 35          CALL   3511H          ;Run on cassette
34DF   BE 38 CD          CALL   388EH          ;Add to Checkout
34E2   7C                LD     As             il'ISB Load End Address
34E3   CD 11 35          CALL   3511H          to print on cassette
34E    CD SE 38          CALL   38BEH          to add test tubes
34E9   CD FB 3A          CALL   JAFSH          Check to see if the BREAK key is
34EC   1A                LD     A, (EN         Load {Progree Byte
```

```
34ED  13           INC   EN          ; programme address + 1
34EE  CD 11 35     CALL  3511H       ;programme byte on cassette
34F1  CD BE 38     CALL  388EH       ;Add to Checksum
34F4  CD F8 3A     CALL  3AFBH       ;Check if BREAK key is pressed
34F7  DF           RST   18H         ;End of programme
34F8  20F2         JR    NZ.34ECH    ;no, output next byte
34FA  DD 7E 00     LD    A, )        Load LSB Checksum
34FD  CD 11 35     CALL  3511H       to print on cassette
3500  DD 7E 01     LD    A, (IX+1)   ;/SR Load Checksum
3503  CD 11 35     CALL  3511H       {on cassette output
350   0 14         LD    B, 20       ;as End ID
3508  AF           XOR   A           120 bytes Yoo'
3509  CD 11 35     CALL  3511H       ;Run on cassette
350C  10 FB        DJNZ  3509H
350E  E1           POP   HL          Reload {programme pointer
350F  FB           EI                ;Turn back on interrupts
3510  c9           RET               Ready!
```

#Mt # # # # } M # # i #i #k k #ME M} # # It

Writing a Byte to Cassette

Ring: **A** = byte to output

```
3511  F5           PUSH  AF          Save Register Content
3512  CS           PUSH  BC
3513  IT           PUSH  HL
3514  2E 08        LD    L8          bit count = 8
351b  7            LD    H, A        ;H = byte to output
3517  CD 42 35     CALL  3542H       ;Clock pulse output
351A  CB 04        RLC   H           highest bit in carry
351C  30 0D        JR    NC, 352BH   1= 0? yes, output @ bit
351E  CD 42 35     CALL  3542H       no, output 1-bit
3521  CD 42 35     CALL  3542H       ;by 2 consecutive.        Clock-
3524  2D           DEC   L           ;Bi counter - 1
3525  20F0         JR    NZ, 3517H   ;) 0? Yes, next bit
3527  E1           POP   HL          Restore register contents
3528  C1           POP   BC
3529  F1           POP   AF
352A  C9           RET               finished
```

0 - bit output

```
352B  3A 3B 78     LD    A 783MH)    ;I/0 Load Latch Byte
352E  F6 0         OR    b           ;Bits 1 u. Set 2
3530  32 00 68     LD    (800H),A    ;and spend
3533  0 99         LD    3.153       ;555 pause
```

```
3535  10FE            DJNZ   3535H
3537  E6 F9           AND    WF 9            bits fu. ? Delete newer
3539  32 00 68        LD     46880H),A       and spend
353C  O 99            LD     B, 153          ;555 pause
353E  10 VU           DJNZ   353EH
3540  18 E2           JR     3524H           ;finished

                      Print clock pulse

3542  3A 3B 78        LD     A, (783BH)      ;I/0 Load Latch Byte
3545  F6 06           OR     6               ;Bits 1 u. Set 2
3547  32 00 68        LD     (6800H),A       ;and spend
354A  €064            LD     B.76            1277 from
354C  10 VU           DJNZ   354CH
354E  E6 F9           AND    0F9H            Bits l u. 2 Delete again
3550  32 00 68        LD     (6800Hl,A       ;and spend
3553  06 4C           LD     B.776           ;277 pause
3555  10FE           DJNZ   3555H
3557  C9             RET                    ; finished

                      ##k # k # # k kl k k # # # # # # k # #####

                      Write header on cassette

                      (synchronisation, header,          programme name )
3558  CD 8C 35        CALL   358CH           ;Progra111111name in buffer
355B  06 FF           LD     1.255           ;Output Synchronisation Bytes
355D  3E 88           LD     A, Bi           _
355F  CD 11 35        CALL   3511H           Output Byte
3562  CD E8 3A        CALL   3AE8H           ;BREAK key pressed?
3545  D8              RET    C               Yes, cancel!
356   106 F5          DJNZ   355DH           1255 bytes counter
3568  0 05            LD     M,5             ;5 x X'FE' as header
3564  PER FE          LD     A,0FEH          ;issue
356C  CD 11 35        CALL   3511H           Output Byte
356f  CD E8 YES       CALL   JAEBH           ;BREAK key pressed?
3572  D8              RET    C               Yes, abort!
3573  10 F5           DJNZ   356AH           {Byte counter
3575  79              LD     A;              Load {Programme am-(File) Kenner
3576  CD 11 35        CALL   3511H           and write on cassette
3579  CD EB JA        CALL   3AE8H           ;BREAK key pressed?
357C  D8              RET    C               ;yes, cancel
357D  34 D 7A         LD     A, 7ADH)        Load length of name
3589  47              LD     B,A             ;in B as counter
3581  11 9D 7A        LD     DE, 7A9DH       ;Name start address in DE
3584  1A              LD     A, (DE)         ;Load the name character
```

```
3585  13              INC    EN             iAddress + 1
3586  CD 11 35        CALL   3511H          ;Print character on cassette
3589  10 F9           DJNZ   3584H          Names issued completely?
358B  C9              RET                   Yeah, done!
```

%%%%# t # i± } # i% ##M

Check programme (file) name and put it into buffer

Calling CSAVE, CLOAD, PRINT#, INPUT#

```
358C  8 10            LD     B, 1           imax. 16 characters
35BE  11 9D 7A        LD     DE, 7A9DH      ;Load initial buffer address
3591  7E              LD     A, (HL)        ;Load characters from programme
3592  FE 3A           CP     ':'            ;End of command?
3594  28 12           JR     1, 3548H       Yes!
3596  B7              OR     A              End of line?
3597  28 0F           JR     Z,35ABH        Yes!
3599  CF              RST    8              ; follows a 'O ?
359A  22              DEFB   ' '
359B  7E              LD     A (HL)         Character of the name in A
359C  B7              OR     A              End of line?
359D  28 09           JR     Z,35ABH        Yes!
359F  23              INC    Hl.            Image + 1
3540  RD 22           CP     '"'            End of the string?
35A2  28 04           JR     Z,35ABH        Yes!
35A4  12              LD     (EN),A         Retain characters in putter
35A5  13              INC    EN             iPufter Address + 1
3546  10 F3           DJNZ   359AB          ;next character
35AB  AF              XOR    A              iX'00' as Abschluj
35A9  12              LD     (EN),A         a buffer
35AA  EACH 11         LD     A, 17          Determine the length of the name
35AC  90              SUB    B
35AD  32 D6 7A        LD     (74DH),A       and note
35B0  C9              RET                   finished
```

%Mi%%%%%%%%%~~~~$ #MM%%

Check to see if loading messages should be reported.

If yes, prepare the output

```
35B1  3A 4C 78        LD     A, (784CH)     Load iOutput Flag
35B4  B7              OR     A              4= 0?
35B5  C               RET    NZ             Inein, l'suppress messages
35B6  3A JB 78        LD     A,(783BH1      !I/0 Load Latch Byte
359   Ci 5F           BIT    3.A            {Count ik-
35BB  28 IB           JR     1, 3508H       No!
```

```
351.lD  E6 F7          AND   0F7H           Yes, switch to text mode
351.lF  32 3B 78       LD    (783BH),A
35C2    32 00 68       Lfi   (6800H),4
35C5    CD 92 32       CALL  3292H          {Delete Screen
35C8    21 FF 71       LD    HL.71FFH       {Cursor to Last Position
35CB    22 20 78       LD    (7820H),HL
35CE    3E IF          LD    A.31           {column pointer to last column
35D0    32 A6 78       LD    (78A6HJ,A
35D3    34 E5 7A       LD    A, (7E5H)      ;Status of prel.   Load Row
3506    FE 81          CP    81H            {Double Line?
35D8    C0             RET   NZ             {no, ready
3509    3D             DEC   A              last and last.     Line as
35DA    32 ES 7A       LD    (7AE5H),A      Mark individual lines
35DD    32 E6 7A       LD    (7AE6H),A
35E0    C9             RET                  ;finished
```

%%%%%%%%%%~%%#%%}# i % EH

Find the beginning of the programme on the cassette

```
35E1    21 42 38       LD    HL.3842H       Address WAITING' - Load Text
35E4    CD F4 37       CALL  37F4H          ;Print Text
                  Injection of CLOAD
35E7    CD F8 3A       CALL  3AFBH          ;BREAK key pressed?
35EA    3A 00 68       LD    A, (6800H)     Read /0 Byte
35ED    C.B 77         BIT   6.A            Evaluate {pulse of cassette
35EF    20F            JR    NZ,35E7H       no pulse, back'
35F1    CD 8F 37       CALL  378FH          Read to t
35F4    38 F1          JR    C, 357H        was nothing, back
35F6    C.B.47         .Bit  @,A            Was that a one?
35F8    28 F7          JR    Z,35FlH        nemn+ next bit
35FA    06 07          LD    M7             ;7 Read more .Bits
JSFC    CD 8F 37       CALL  378FH          Read ;Bit
35FF    38 E6          JR    C, 35E7H       ;Time expired, once more'
3801    18 F9          DJNZ  35FCH          ;next bit
303     FE 80          CP    80H            is it a SYNC - .Byte?
3605    20 E0          JR    NZ,35E7H       No, continue searching
Move the cartridge to the end of SYNC
3607    CD 75 37       CALL  3775H          Read Byte
360A    DA E7 35       JP    C.35E7H        Time expired, again!
360     RD 8           CP    80H            ;SYNC-.Byte?
360F    28 F6          JR    ,3607H         ;yes, next .byte
the next 5 bytes must be X'FE'
3611    0 M4           LD    1.4            ;Counter= 4, da 1 • .Byte over.
3613    FE             CP    0FEH           = Header byte?
```

```
3b15  C2 E7 35        JP    NZ,35E7H        ;no, search 111tab
3b18  CD 75 37        CALL  3775H           ;Read next byte
361B  DA E7 35        JP    C.35E7H         ;Time expired, back'
361E  10 F3           DJNZ  3613H
                    Read programme id
 320  CD 75 37        CALL  3775H           Read {ID byte
3623  32 D2 7A        LD    {7AD2Hl,A       ;and Save
                    Read Names and Transfer to Buffer
 362  21 B2 74        LD    HL, 7AB2H       Load buffer address
3629  6 12            LD    B, 18           max, length
 32B  CD 75 37        CALL  3775H           ;Read Byte
362E  77              LD    (HL),A          and in buffer
362F  B7              OR    4               i= 0?
3638  28 06           JR    1, 3638H        Yeah, done
 332  23              INC   HL              ;Buffer Address+ 1
3633  10 F6           DJNZ  362BH           ;next byte of name
3535  C3 E7 35        JP    35E7H           more than 18 characters!
                      Print FOUND Message
3638  21 SA 38        LD    HL.J85AH        ;Address Message Text
363B  CD F4 37        CALL  37F4H           ;Text 'FOUND'
363E  21 B2 74        LD    HL,7AB2H        ;Address names in the router
3641  CD 14 38        CALL  3814H           ;and spend.
                    Check if the programme you are looking for
3644 21 B2 7A         LD    H, 7AR2H        ;Address buffer     {geles, name)
3647  11.91) 7A       LD    DE,7A9DH        ;Address entered name
3A  1A                LD    A, {DE)         Load byte of entered name
364R B7               OR    A               End ?
364C CS               RET   or              Yes found!
 34D  BE              CP    (HU             = Character in buffer?
J64E  C2 E7 35        JP    NZ,35E7H        {no, continue
3651  23              INC   HL              ;Buffer Address+ 1
3652  13              INC   EN              {dr, d, input. Name + 1
 353  18 FS           JR    3AI             ;next character
3655  C9              RET                   not used

%%%H%%%%%%%%%%%%%%%% ~%~ % 9%%%

                      CLOAD Statement
                    Read Progr on Cassette
3656  IT              PUSH  HL              ; Programme pointer to stack
3657  21.39.78        LD    HL.7839H        ;Address flag 2
354  CB               RES   6, (Hl.)        ;Clear CRUN Flag
365C  CB 9E           RES   3, {HL.)        i\lERIFY Flag Delete
J65E  El              POP   HL              Load {Programme pointer
```

Together with CLOAD, CRUN and VERIFY used

```
365F F3              DI                  Disable Interrupts
  36 CD ac 35        CALL   358CH        Remove {Name from programme text
3663 E5              PUSH   HL           ;Programme pointer to stack
  36 CD B1 35        CALL   35B1H        l'prepare the message output
3667 21 42 38        LD     H, 3842H     ;Text 'WAITING'
  36 CD F4 37        CALL   37F4H        ;and spend
366D CD E7 35        CALL   35E7H        ;Find programme on cassette
3670 3A D2 7A        LD     A, (7AD2H)   ;Load programme id
3673 FE F2           CP     0F2H         Is it simple data?
3675 28F             JR     Z.366DH      {yes; Keep surging!
3677 21.60.38        LD     HL.3860H     ;Text 'LOADIN6'
367A CD 04 38        CALL   3804H        ;and spend
367D DD 21 23 78     LD     IX.782JH     ;Address sum bytes per
3681 CD 68 38        CALL   3868H        ;Read start and end address
3684 DA 11 37        JP     ,3711H       ;Error? yes-LOADING ERROR
3687 IT              PUSH   HL           ;End address on stack
3688 ED 52           SBC    HL, DE       End - Start address = Byte counter
368A DA 11 37        JP     C,3711H      ;Start-> End Address? yes, mistake
368D ED 53 1E 78     LD     (781EH1,EN   ;Save Startup Address
3691 E5              PUSH   HL           Transfer {gtecount to B.
3692 C1              POP    BC
3693 E1              POP    HL           ;Load End Address
3694 3A 39.78        LD     A,7839H1     ;Load flag 2
 397 CB SF           BIT    3            ;VERIFY flag set?
3699 C2 42 37        JP     NZ.3742H     ;yes, to the VERIFY routine
369C CD 73 3F        CALL   3F73H        ;Read cassette byte
 39F 12              LD     (EN),        and save to RAN area
3640 CD 8E 38        CALL   388EH        ;Add to Checksum
3A4.3 13             INC    EN           ; programme address + 1
36A4 0B              DEC    RC           Eyte counter - 1
36A5 79              LD     A,€          = ?
3A B0                OR     B
36A7 20F3            JR     NZ, 369CH    no" continue reading
3649 CD 75 37        CALL   3775H        ;Read LSB Checkout
36AC DD BE 0         CP     (IX)         with calculated checksum
364F C2 11 37        JP     NZ,3711H     ;unequal, LOADING ERROR spend
36R82 CD 75 37       CALL   3775H        ;Read NSB test tube
6B5  DD BE 01        CP     !IX+11       with calculated checksum
3B8 C2 11 37         JP     NZ,3711H     ;unequal, LOADING ERROR spend
36B8 22 F9 78        LD     (78F9H1,HL   Save { Programme Address
36BE F.B             EI                  ;Reset interrupts
36BF 3E ilD          LD     A, DH        iCR - Issue LF
```

```
36c1  CD 8 30          CALL   308BH
36C4  3A D2 7A         LD     A, (7AD2H)      ;Load programme id
36C7  FE F1            CP     0F1H            a machine programme?
36C9  211114           JR     NZ.36CFH        {no'
                       Start programme
36CB  2A LE 7B         LD     H., (781EH)     ;Load Startup Address
36CE  E9               JP     (H)             and start
                       BASIC programme
 36F  212919           LD     HL, 1929H       ;Text 'READY'
36D2  CD A7 28         CALL   28A7H           and spend
3505  2A4 78           LD     HL, (7844H)     ;Load programme startup address
 3D8  E5               PUSH   HL              {and onto the stack
36D9  21.39.78         LD     HL.7839H        ;F!ag 2
36DC  CB 76            BIT    6, HL)          Set iCRUN flag?
36EN  20 03            JR     NZ.36E3H        yes, start
36E0  C3 ES lA         JP     AEBHH           No, to the main loop
                       Start BASIC programme
36E3  21.39.78         LD     HL.7839H        {Flag? address
36E6  CR B             RES    , {HL)          Delete iCRUN Flag
36EB  D1               POP    EN              Load programme start address
36E9  CD FC La         CALL   lAFCH           ;Refresh Row
 36C  CD 5 79          CALL   79B5H           {RA! Extension Output
36EF  CD 5D 1B         CALL   1B5DH           ;Delete Variable Table
36F2  CD B8 78         CALL   78BBH           {RAN-Erve1 terungsaus9ang
36F5  FF 21            LD     HL,0FFFFF       ;act. Line number= Direct
36F8  22 A2 78         LD     (78A2HJ,HL
36FB  21 EB 79         LD     HL.79EBH        ;I/O Putfer Addressing
36FE  1170115          LD     EN, 05701H      ;Address RUN Command
3701  1A               LD     A, (DE)         and transferred to I/O putter
3702  77               LD     (HL),A
3703  B7               Oll    A               End?
3704  28 04            JR     Z.370AH         Yeah, done
3706  23               INC    HL              ;Buffer Address+ 1
3707  13               INC    EN              ;text address + 1
3708  1 F7             JR     3701H           ;next character
370A  21 E7 79         LD     HL, 79E7H       ;I/O-Putter - 1 Addresses
370D  AF               OR     A
370E  C3 81 1A         JP     1A81H           Run {RUN Command
                       Report Load Error
3711  21 4A 38         LD     HL+ 3841        ;Text 'LOADING ERROR' address
3714  FB               EI                     ;Enable Interrupts
3715  CD A7 28         CALL   2BA7H           Print {Text
3718  F3               DI                     ;Enable Interrupts
```

```
3719   3A 478          LD     (A, 784CH)      Load Output Flag
371C   B7              OR     A               = 0 7
371D   C2 673          JP     NZ.3667H        No, other text output below.
372111 21 FF 71        LD     HL, 71FFH       ;cursor to last character
3723   22 20 78        LD     !782111H),HL
3726   3E lF           LD     A.31            {column pointer to last column
3728   32 From 78      LD     !78A6H>,A
372B   C3 7 3          JP     3667H           Try {again
```

~%%%%#%%$4% tt # kt H

CRUN Statement

Read and start a **programme**

```
372E   IT              PUSH   HL              {Programme pointer to stack
372F   21.39.78        LD     HL.7839H        Address Flag 2
3732   CB F            SET    6, HL.)         Set {CRUN Flag
3734   E1              POP    HL              Load { programme pointer
3735   C3 5F 3         JP     365FH
```

#%%%%%%%~~~%%1%%%%%%% #% }#dE

WERIFY Statement

Check a programme on cassette

```
3738   E5              PUSH   HL              Programme pointer to stack
3739   21.39.78        LD     HL, 7839H       ;Address flag 2
373C   CB EN           SET    3, (H)          ;Set VERIFV flag
373E   El              POP    HL              Load { programme pointer
373F   C3 5F 3         JP     35FH
```

HffH-1-fffffffffffffffffffffffffffffff

VERIFY - Set after common routine with CLOAD

```
3742   EB              EX     DE,HL           ;HL = Master Start Address
3743   CD 75 37        CAl…L  3775H           ;Read Byte from Cassette
3746   BE              CP     (Hl.)           {= Progranbyte ?
3747   28 09           JR     Z,37S2H         Yes!
3749   21 6c 37        LD     HL.376CH        ;Text 'VERIFV ' Addresses
374C   CD A7 28        CALL   28A7H           ;and spend
374F   C3 83 01        JP     0183H           ;Text 'ERROR'

3752   23              INC    Hl.             Programme Address + !
3753   0B              DEC    BC              Counter - 1
3754   79              LD     A, C            {done?
3755   B8              OR     B
```

```
3756  20EB            JR     NZ.3743H       ;no, next byte
3758  21.39.78        LD     HL.7839H       ;Address flag 2
375B  CB 9E           RES    3,(HI..)       ;Delete VERIFY flag
375D  21 6C 37        LD     H.37CH         ;Text 'VERIFY ' Address
3760  CD A7 28        CALL   28A7H          ; and spend
3763  218003          LD     HL.0380H       ;Text 'OK'
3766  CD A7 28        CALL   28A7H          ;and spend
379   C3 F 36         JP     JbCFH          ;back to BASIC

376C  0D              DEFB   0DH            ;Text Definition 'VERIFY'
376D  56 45 52 49     DEFN   'VERIFY'
      46.59% 20
3774  80              DEFB   0
```

#Mi kt t litt t}i %# #Hit MM&

Read Cassette Byte

Exp.: = byte read
Carry set if read error

```
3775  C5              PIISH  JC             ;Save Register
3776  D5              PUSH   EN
3777  6 88            LD     38             bit counter = 8
3779  CD 8F 37        CALL   378FH          Read a bit
377C  38 8E           JR     C.378CH        ;Read error!
377E  11 F9           DJNZ   3779H          ;next bit
3781  D1              POP    EN             ;Reload Tab
3781  C1              POP    BC
3782  32 D3 7A        LD     (7AD3WH),A     ;Save read byte
3785  CD FB 3A        CALL   3AFBH          ;Check BREAK Key
3788  3A D3 7A        LD     A, (7AD3H)     ;Reload read byte
378B  C9              RET                   ;finished
error return
378C  D1              POP    EN             Reload register
378D  C1              POP    BC
37BE  C9              RET
```

4111%14~%Mt ± Et MM

Read bit from cassette

```
37BF  C5              PUSH   BC             Secure RC
3798  @1 FF 07        LD     BC,1117FFH     ;Load lert for timer
3793  34.80.8         LD     A, (6880H)     ;Load J/O Port
379   CI 77           BIT    ,A             1 – Level ?
3798  28 08           JR     Z,37A2H        Yes continue
```

```
379A  0B              DEC   BC            ;Time value - 1
379B  79              LD    A,C           ;Time expired?
379C  1\0             6R    B
379D  20F4            JR    NZ.3793H      ;no, read again
379F  C1              POP   BC            Reconnect iBC vom Stack
37A0  37              SCF                 ;Set Carry Flag
37A1  C9              RET                 ;Fault Reversal
                      Check if clock pulse
37A2  3A 08 68        LD    A,t6800Hl     ; Load I/0 Port
37A5  Ci 77           BIT   in            il - Level ?
37A7  20 EA           JR    NZ.3793H      yes, W level too short'
37A9  34 00 68        LD    A, i6808H)    Load i/0 Port
37AC  CB 77           BIT   b6+           i1 - Level ?
37A   28E3            JR    NZ.3793H      yes, 0-level here too short
37B0  O 52            LD    B.82          i30e delay
37B2  10 VU           DJNZ  37B2H         pass un-pulse
37B   3A 00 68        LD    , (B880H)     i Load I/0 Port
37B7  Cl 77           BIT   6.            Now 1 level?
37B9  28 09           JR    NZ,37C4H      yes, correct clock pulse
37BB  34 00 68        LD    A, t6800Hl    ; Load I/0 Port
J7JE  CB 77           BIT   6sA           ; Pul change?
37CI  28 F9           JR    Z,37BBH       ;Wait for pulse end
37C2  18 CC           JR    3790H         no correct clock pulse
                      Eat number of pulses
37C't O 54            LD    R, 90         Set Counter to ms
37C6  0E III          LD    C,@           Delete Result Tab
37C8  34 00 68        LD    A, (688H)     Read I/0 Port
37CB  CB 77           BIT   S             ;Wait for negative flank
37CD  28B             JR    Z.37DAH       There it is!
37CF  10 F7           DJNZ  37C8H         Read i111
                      Number of pulses - 1 = result bit
3701  79              LD    A,C           {Number of pulses in A
37D2  3D              DEC   A             j- 1
3703  1F              RRA                 Down. bit into the Carrg
3714  CJ 12           RL    D             iCarry bit in D (bytes seep)
371.6 Cl              POP   BC            Restore iBC
37D7  7A              LD    A,D           ;Transfer result from D to A
3708  17              OR    4             Delete iCarry Flag
37D9  C9              RET                 finished
                      Check pulse stability and wait for 2, pulse
37DA  YES OO 8        LD    A,<68111H)    Load i/0 Port
37DD  Cl 77           BIT   fine          Back to 1?
37DF  21 EE           JR    NZ.37CFH      Yeah, too short!
37E1  34.80.68        LD    A, 8Mi)       i Load 1/0 Port
```

```
37E4  B 77          BIT                 back to ! level?
 37E  20 E7         JR      NZ.37CFH    yes" still too short!
37E8  €0            INC     C           ;pulse counter + 1
37E9  34 00 68      LD      A, (6880%)  ; Load I/O Port
37EC  CB 77         BIT     6.A         ;Pulse out?
37EE  208DF         JR      NZ.37CFH    Yes, maybe 2. Record pulse
37Case 10 F7        DJNZ    37E9H       ;Time expired?
37F2  18DD          JR      37D1H       yes; Count Pulses
```

#%%% %# % k # # ###

Delete last line and **output** 11message

```
37F4  3A 4C 78      LD      A, (74CH)   Output Load Flag
37F7  B7            OR      A           = 0 2
37F8  C0            RET     NZ          No output!
37F9  11 part 71    LD      DE, 71EOH   to address last line
37FC  6 20          LD      B.32        ;Line length as counter
37FE  CD F 3E       CALL    3EFbH       ;space in last line
3801  13            INC     EN
3802  10 FA         DJNZ    37FEH
3804  3A 4C 78      LD      A, 784CHl   Load Output Flag
3807  7             OR      A           = 0 ?
3808  C0            RET     NZ          3no, no output!
3809  CD 0E 3F      CALL    3F0EH       AB. Reverse From Background If
380C  7E            LD      A, {H)      ;Load text character
380D  B7            OR      4           End of text?
J80E  C8            RET     0r          Yeah, done!
3B0F  12            LD      (EN),A      Transfer to Image
3810  13            INC     EN          ;Image Address+ 1
3811  23            INC     HI.         ;text address + 1
3812  18 F8         JR      380CH       !Send next character
```

%%%%+% k It#I}}t }} $ MM t H} #

Print programme/file ID and names

```
3814  3A 4C 78      LD      A, (784CHl  ;Load Output Flag
3817  B7            OR      A           4= 8?
3818  C0            RET     NZ          ;no, suppress output
3819  11 E9 71      LD      DE, 71E9H   ;Last Line Image Location
381C  E5            PUSH    HI.         Save address of name
381D  34 D2 7A      LD      A,(7AD2Hl   Load {Programme ID
3828  E6 OF         AND     IJFH        Delete top half byte
3822  21 3F 38      LD      HL.382FH    ;Address identification table
3825  85            ADD     A,L         ;+ Id
```

```
3826  6F              LD     L,A
3827  3E 00           LD     A, 0
3829  SC              ADC    + ␣
382A  67              LD     H,
382B  CD 21 3F        CALL   3F21H          {ID abh from background off,
382E  00 08           2x NOP
3830  12              LD     (EN) ,A        ;':' as separator
3831  13              INC    EN             Image Address + 2
3832  13              INC    EN
3833  El              POP    HL             ;Load name address
3834  7E              LD     A, HL)         ;Load character from name
3835  B7              OR     A              End?
3838  addrox          RET    1              ia. done!
3837  CD 33 3F        CALL   3F33H          ;character abh, from background,
383A  13              INC    EN             ;.Image Address + 1
383B  23              INC    HL             ;Name Address+ 1
383C  18 F6           JR     3834H          next character
383E  C9              RET                   ;unused

                      identification table

383F  14 82 04        DEFB   141, 02H,04H   ;Codes for T, B, D

                      #%4i%%%% %%% Fit ±}i %Mi With

                      Cassette Routine Text Definitions

3842  57 41 49 54     DEFII  'WAITING'
      49 4E 47
3849  08              DEFB   0
384A  00              DEFB   0DH
384B  4C 4F 1 4 49    DEFM   'LOADING ERROR'
      4E 47 28 45 52
      52 4F 52
3858  0D0 00          DEFB   8DH, 00H
385A  F 55 4E 44      DEFII  'FOOND'
385F  00              DEFB   0
3860  4C 4F 41 44    DEFI'!  'LOADING'
      49 4E 47
3867  08              DEFB   0

                      fffffffffffffffHlllllllfffffffffff
```

```
                                Read the start and end address of the cassette
3868  CD 75 37      CALL      3775H          Read Byte
38611 Profit        RET    C                 Read errors!
€38   5F            LD     E,A               ;LSB Start Address in E
386   DD 77 00      LD       (1X),A4         and in LSB checksum
3870  AF            XOR    A                 3MR checksum =
3871  DD 77 01      LD      (IX+1),A
3874  CD 75 37      CALL      3775H          Read {Byte
3877  Profit        RET    C                 Read errors!
3878  57            LD     D,A               ;NSB Start Address in D
3879  CD 8E 38      CALL      388EH          and add to checksum
387C  CD 75 37      CALL      3775H          Read Byte
387F  DS            RET    C                 ;Read error
3880  6F            LD     n                 ;LSB End Address in L
3881  CD BE 38      CALL      388EH          ;and aut PrüfSlUl!llte add
3884  CD 75 37      CALL      3775H          Read {Byte
3887  Profit        RET    C                 ;Read error!
3888  67            LD     H,A               iHSB End Address in H
3889  BE 38 CD      CALL      388EH          ;and add to checkout!'le
388C  B7            OR     A                 Delete Carry Flag
388D  C9            RET                      {finished

%%i %# %%% %~%~%~H~%Mt t

                                Determine Checksum

                                Ring: IX = address of checksum bytes
                                      A     read-in byte
388E  DD 8b 00      ADD       A, (IX)        ilSB Add checksum to characters
3891  DD 77 00      LD         (IX),A        ;and save again
3894  3E 08         LD       A+0             A =
3896  DD SE 01      ADC       A,<IX+ll       Add iNSB checksum to carry
3899  DD 77 01      LD        (II+±),A       ;and save again
389C  C9            RET                      ;finished

#%%i%%% %% %% ~~%1%#%%3 % M # Wed

                                COLOUR Statement
389D  7E            LD          A,<HU        Load next programme character
389E  FE 2C         CP        ',             = comma?
380   28 20         JR         Z,38C2H       ;yes, just change background
38A2  CD IC 211     CALL    2BICH            1. Evaluate Expression
38A5  B7            OR      A                = ?
3846  CA 4A 1E      JP       Z, 1E4AH        ;yes, FUNCTION CODE - Error
```

```
38A9  FE 09          CP    9              :> 8?
38AB  D2 4A 1E       JP    NC,1E4AH       Yes, FUNCTION CODE Error
38AU  3D             DEC   A              ;Colour Code - 1
38AF  E6 07          AND   7              ;limit to 0 - 7
3BB1  CB 27          SLA   A              ;Move to top half byte
38B3  CB 27          SLA   A
38.BS CB 27          SLA   A
38B7  CB 27          SLA   A
38B9  32.46.78       LD    (784bH),A      and save
38BC  7E             LD    + (HT)         ;Load next character
38BD  B7             0R    A              ; End of line?
38BE  CS             RET   or             Yes, done!
38BF  FE 3A          CP    :'             {Command end?
38C!  approx         RET   or             Yeah, done!
38C2  CF             RST   8              {next character comma?
38C3  2C             DEFB  ',
38C4  CD 1C 2B       CALL  2B1CH          Yeah,     Evaluate Expression
38C7  B7             0R    A              1= 0?
3BC8  20 0C          JR    NZ, 38Di       {no!
38CA  3A JB 7B       LD    A, !7B3BH1     Load i/o-Latch Byte
3BCD  CB A7          RES                  ;Set background to green
38CF  32 3B 78       LD    (783BH), A     save again
3BD2  32 00 68       LD    (6800H), A     ;and output via 1/0 port
3BD5  C9             RET                  ;finished

38D6  FE 01          CP                   = 1 ?
38DB  C2 4A 1E       JP    NZ,1E4AH       ;no, FUNCTION CODE - Error
38DB  3A 3B 78       LD    A, (783BH)     Load i/0 Latch Byte
38EN  C.B E7         SET   +              ;Set background to red
38E0  32 3 78        LD    (783BH),A      save again
38E3  32 00 68       LD    (60OH) A       ;and output via I/0 port
38Eb  C9             RET                  ;finished
```

%%#Mt} With# tri tt #Et Mil t }#&

POINT function add-on routine

```
38E7  0E C0          LD    C, @CMH        ;2.Bit Load Naske
3BE9  C3 09          RRC   C              according to the pixel position
38EB  10 FC          DJNZ  38E9H          ;Move right in byte
3BED  1A             LD    A, (DE)        Load Byte from Video Memory
38EE  A1             AND   C              Link to Taske
38EF  47             LD    .B,A           Save result in .B
3BF0  79             LD    AC             ;Load mask in A
J8F1  CB 88          RRC   B              Byte and mask since right
```

```
3BF3  CB 0F        RRC   A              ;slide up mask to far right
38F5  FE 03        CP    3              check whether mask is right
3487  20 FB        JR    NZ,38F1H       Keep pushing!
3BF9  78           LD    A, B           ;Load colour code from B
38FA  3C           INC   A              ;+ 1 for result ( 1 • 4)
30FB  IT           PUSH  HL             ; Programme pointer to stack
38FC  CD 8D 09     CALL  098DH          Convert to 1-Bit Integer        (X)
38FF  E1           POP   HL             ;Reload the programme pointer
3900  C3 0F 39     JP    390FH          check that ') follows
```

%# With Wed # # M # # it kt t% t # % # f kt

SET and RESET Statement Completion Routine

```
3903  47           LD    B,A            ;SET mask transferred to B
3904  1A           LD    A, (DE)        ;Load bytes from image store
3905  Al           AND   C              ;Bits for adr. Delete Pixels
390   12           LD    (DEHA          ;Write back byte
3907  F1           POP   AF             ;Load Function Flag
3908  B7           OR    A              ; RESET Statement ?
3909  F2 0F 39     JP    P.390FH        Yeah, done!
€390  1A           LD    A, {DE)        ;SET, Reload Byte
390D  B            OR    B              Bits for adr. Set Pixels
390   12           LD    <DE>,A         Write Byte
390F  CF           RST   8              {parameter with ')' completed?
3910  29           DEFB  ')'
3911  C9           RET                  ijas finished
```

%Mi%Hi With t%#%Mt # HE

COPY Statement

```
3912  F3           DI                   ;Disable Interrupts
3913  IT           PUSH  Hl             ;Programme pointer to stack
3914  3A 3B 78     LD    A, (783BHJ)    ;Load I/0 Latch Byte
3917  CB 5F        BIT   3.A            ;Calculator in Gratik Nodus?
3919  C2 8E 39     JP    NZ.39BEH       ;yes, graphics output
                   Text Mode Screen Output
39C   21 00 70     LD    Hl, 7000H      ;Home Image Address
391F  0E 10        LD    , 1            ;Line counter= 16
3921  0 28         LD    B.32           ;Column counter= 32
3923  7E           LD    A, (HL)        ;Load characters from image
3924  B7           OR    A              ;Blockgraphic - Characters?
3925  F2 2D 39     JP    , 392D%        No!
3928  CD 73 2C     CALL  2C73H          ;Print block graphics
392B  18 16        JR    3943H
```

```
392D  C3 44 3F        JP     3F44H          ;Verify that characters inverted 1st
3930  00              NOP                   ;if no, continued at 3938H
3931  E6 3F           AND    3FH            ;Bits 6 and 7 delete
3933  CD 56 39        CALL   3956H          {Inverted Character Output
3936  18 0B           JR     3943H
3938  E6 3F           AND    3FH            Delete Bits and 7
393A  CB 6F           BIT    5.A            Zechen { IFH ?
393C  20 02           JR     NZ, 39401      {no!
393E  F6 4            0R     4MH            ;yes, + 40H for true ASCII character
3940  CD BA 3A        CALL   JABAH          {Print Characters on Printer
3943  23              INC    HL             ;Image Address+ 1
3944  10DD            DJNZ   3923H          ;End of Line?
3946  3E 0D           LD     A,0DH          Yes, issue CR/LF
3948  CD BA 3A        CALL   JABAH
394B  CD FB 3A        CALL   3AFBH          ;BREAK key pressed?
394E  0D              DEC    C              ;Line counter - 1
394F  79              LD     As             = ?
3958  B7              OR     A
3951  20 CE           JR     NZ.3921H       ;no, next line
3953  El              POP    HL             Load {Programme pointer
3954  FB              EI                    ;Enable Interrupts
3955  C9              RET                   {finished
                      Print Inverted Characters
3956  FS              PUSH   AF             Save {Register to Stack
3957  CS              PUSH   BC
3958  DS              PUSH   EN
3959  IT              PUSH   HL
395A  F               LD     L,A            ;character to be issued in HL
395B  26 00           LD     H,8
395D  3E 08           LD     A, 8           ;Switch Printer to Graphics Nodus
395F  CD BA 3A        CALL   3ABAH
 392  06 04           LD     B,4            Character code 1 5
 394  IT              PUSH   HL             as table offset
3965  D1              POP    EN
  39  B7              OR     A
3967  ED SA           ADC    HL, DE
\ 399  10 FC           DJNZ   3967H
396M  IT              PUSH   HL             ;Table offset in BC
396C  Cl              POP    BC
396D  21 94 38        LD     H, 3B94#       Table t. inverted characters
3970  09              ADD    HL,BC          ;+ Offset
3971  3E FF           LD     A,0FFH         il Row points left of character
3973  CD BA 3A        CALL   3ABAH          ;issue
3976  &6 5            LD     M,5            15 rows of points from table
```

```
3978  7E              LD    A, (HL)        spend
3979  23              INC   HL
397A  CD BA 3A        CALL  3ABAH
397D  10 F9           DJNZ  3978H
397F  JE FF           LD    A, OFFH        1 row of points to the right of
3981  CD BA 3A        CALL  3ABAH          ;Print Character
3984  3E 0F           LD    A, FH          ;Printer back in text—Nockls
3986  CD BA 3A        CALL  3ABAH          ;Switch
3989  El              POP   HL             Restore register contents
398A  Dl              POP   EN
398B  Cl              POP   BC
398C  Fl              POP   AF
398D  9               RET                  Ready!
                      Print image in graphic mode
398E  AF              XOR   A              ;Reset interval counter
398F  32 D 7A         LD    (74D),A
3992  32 D 7A         LD    (74D6),A
3995  3E 08           LD    A, 8           {printer in graphic l'lodus
3997  CD BA 3A        CALL  3ABAH          ;Switch
399A  DD 21 D2 7A     LD    IX,<7AD2H)     Load URL Graphics Buffer
399E  21 00 70        LD    HL, 7000~      ;Load image beginning address
39Al  11 00 00        LD    EN,@           ;Reset print pattern
3944  0E C0           LD    C, Wed         ;Bits 6.7 in Shitt Mask
3946  CD FB 3A        CALL  3AF8H          ;BREAK key pressed?
39A9  E5              PUSH  HL             ;Image Address to the Stack
39AA  CD C9 05        CALL  05C9           ;Delete Graphics Buffer
                      combine 3 lines into one graphic character each
39AD  06 03           LD    , 3            ;Line counter= 3
39AF  7E              LD    A, HL)         Load Graphics Byte from Image
398   Al              AND   C              Select 1 Pixel with Shift Bag
39B1  C5              PUSH  BC             Counter + Shift mask on stack
39/82 47              LD    B,A            the selected pixel in B
39/83 CB 08           RRC   B              ;so often with the SHIFT-11aske
39B5  C.B 08          RRC   .B             Rotate right until the two
39B7  CB O            RRC   C              ;bit pixel bits
39B9  CB 09           RRC   C              i0 and 1 are
39RB  79              LD    A; C
39BC  RD 83           CP    3              il'laske right?
39BE  C2 B3 39        JP    NZ.39B3H       No, move on
39C1  78              LD    A+             ;Pixel back in A
39C2  Cl              POP   BC             ;Counter+ Shift—Reload Nasal
39C3  FE lil3         CP    3              ;Pixel colour= red?
39C5  28 0D           JR    Z,39D4H        Yes!
39C7  FE 82           CP    2              ;Pixel colour= blue?
```

```
39C9  28 0E          JR    Z.39D9H        yes'
39CB  FE 01          CP    1              {pixel colour = yellow .
39CD  28 10          JR    Z,39DFH        Yes!
39CF  11 00 00       LD    EN, 0          green - print pattern= set 0000
39D02 18 0F          3R    39E31i
39D4  11 E0 E0       LD    DE, 0E0EOH     ;red - Print Pattern= E0E0 Set
39D7  180A           JR    39E3H
39D9  16 40          LD    D.40H          {blue - Print Pattern = Set 4040
39DB  1E A0          LD    E, 040H
39DD  18 04          JR    J9E3H
39DF  16A            LD    D, 0AH         yellow - set pressure pattern = A4
39E1  1E 40          LD    E, 40H
                   pressure pattern of three Combine Rows
39E3  DD 7E 00       LD    A,(IX)         1. Load bytes from buffer
39Eb  C 3F           SRL   A              Bits previous, line 3 bitpos.
39EB  CB 3F          SRL   A              slide right
39EA  C3 3F          SRL   A
39EC  E5             PUSH  HL             Backing up the image address
39ED  21 D3 7A       LD    HL, 7AD3H      ;Graphic Putter+ 1
39F8  CD A 3A        CALL  3AbAH          when carry, in buffer.+1.
39F3  E1             POP   HL             Load ;Image Address'
39F4  B2             OR    D              ;Byte 1 of the print pattern in
39F5  DD 77 00       LD    (IX,           ;1. Write back buffer bytes
39FB  DD 7E 82       LD    A, ( IX+2i     ;3. Load bytes from buffer
39FB  3F             SRL   A              Bits past. Line 3 Bitpos,
39FD  C 3F           SRL   A              Move ;Right
39FF  C 3F           SRL   A
3A01  IT             PUSH  HI.            Backing up the image address
3A02  21 D5 7A       LD    LT,7AD5H       ;Graphics Buffer+ 3
3A05  CD bA 3A       CALL  3AbAH          if carry, in buffer + 3.
3A08  E1             POP   HL             Load ;Image Address'
3489  B3             OR    E              ;Byte 2 of print pattern in putter
3404  DD 77 82       LD    IX+2),A        {3. Write back buffer bytes
3AID  3E 20          LD    A 32           ;Image Address' + 1 Line
3A0F  85             ADD   A,L
3A10  bf             LD    L+
3A11  3E 08          LD    A,0
3A13  SC             ADC   A,H
3414  b7             LD    H,A
3A15  10.58          DJNZ  3Ab7H          ;3 lines edited?
3A17  CD 73 3A       CALL  3A73H          Print Yas!
                   the next 3 pixels in the same row
3A1A  E1             POP   ed             Load image address
3A11  C 39           SRI…  C              Shift–11aske 2 bits
```

```
3A1D  CB 39          SRL   C              slide right
3A1F  79             LD    To C           ;3-byte series finished?
3420  B7             OR    A
3A21  20 83          JR    NZ,39A6H       no, next row in the same byte
3A23  23             INC   HL             yes, picture address+ 1
3A24  7D             LD    A,L            ;End of line?
3A25  E 1F           AND   lFH
3A27  C2 A4 39       JP    NZ,39A4H       ;no, next byte
3A2A  CD E2 3A       CALL  3AE2H          ;CR/LF for new line
                     Address next line
3A2D  3A D6 7A       LD    A, (7ADH)      ;Load interval counter
 343  3C             INC   A              3+ I
3A31  FE 03          CP    3              . Row?
3A33  20 01          JR    NZ,3A36H       No!
3A35  A              XOR   A              interval counter =
3A36  32 D 7A        LD    (7AD/),4       new value in interval counter
3A39  20.04          JR    NZ,3A3FH
3A3B  3E 40          LD    0.64           {A = Length of two lines
343D  18 02          JR    3A41H
3A3F  3E 20          LD    A, 32          iA = Line Length
3A41  85             ADD   A,L            H + one o, two lines
3A42  6F             LD    L,A
3A43  3E 00          LD    A,0
3A45  BC             ADC   A,H
3A46  7              LD    H,A
3A47  FE 78          CP    78H            outside the image?
3449  D2 SF 3A       JP    NC,3ASFH       Yeah, done!
3A4C  RD 77          CP    77H            Last line?
3AAE  C2 A 39        JP    NZ,39A4H       No!
3A51  70             LD    A,L
3A52  FE E0          CP    OE0H
3A54  D A4 39        JP    C, 3944#       No!
3A57  3E FF          LD    A,0FFH         ;Set 'last row' identifier
3A59  32 06 7A       LD    (7AD6Hl,A
3ASC  C3 A4 39       JP    39AAH          ;Print next line
                     Graphic expression created!
345F  3E OF          LD    A, l, IFH      Switch Printer to Text-l'lodus
3A61  CD BA 3A       CALL  3ABAH
3A64  El             POP   HL             Load {Programme pointer
3A65  FB             EI                   Enable Interrupt
34    C9             RET                  Ready!
3A67  C3 AF 39       JP    39AFH          What's that?
                     Carry into the next buffer byte at SHIFT
3A6A  02 70 3A       JP    NC 347MH       No Carrg!
```

```
3AD    CR C          SET   @.(HL)          {Carry, bit in n. Buffer Byte = 1
3AF    C9            RET
3A70   CB 8          RES   0, HL)         ;Bit 0 wrong, next putter byte = 0
YES72  C9            RET
                        Print graphic putter
3473   CD 85 YES     CALL  3A85H          Emit {buffer 1+2
3A47   DD 23         INC   IX
3A78   DD 23         INC   IX
3A7A   CD 85 YES     CALL  YES85H         ;Print buffer 3+4
JA7D   DD 2B         DEC   IX
3A7F   DD 2B         DEC   IX
3A81   CD 85 YES     CALL  3A85H          ;Buffer 1+2 from9eben
3A84   C9            RET
                     to print a row of dots for graphic printing
3A85   DD 7E 01      LD    A; (IX+1)      via carry from Putfer+1   Load J
3A88   CB 0F         RRC   A              to carry
3A8A   DD 7E 00      LD    A, IX)         ;Load putter + 0 or 2
3A8D   FS            PUSH  AF             to Stack
3A8E   YES D6 7A     LD    A, {7ADH)      ;Load interval counter
3A91   FE 02         CP    2              ;J. Line?
3A9J   28.1D         JR    Z,JAB2H        Yes!
3A95   FE OI         CP    1              2. Row?
'YES97 28 16         JR    Z,JAAFH        Yes!
3499   Fl            POP   AF             ;Buffer+ 0 or 2 Load
3A9A   17            RLA                  Carry bit in low. bit position
3A9B   FS            PUSH  AF             $Work Character to Emit
3A9C   YES D6 7A     LD    A, (7AD6H)     ;Load interval counter
3A9F   FE FF         CP    OFFH           Last line?
3AA1   20.05         JR    NZ, 3448H      No!
3AA3   Fl            POP   AF             Exp. Reload Characters
3AA4   E6 07         AND   7              just output the lower J bits
3AAi,  18 01         JR    3AA9H
3AA8   Fl            POP   AF             ;excl. Reload Characters
34A9   F6 80         OR    80H            Set Bit 7
JAAB   CD BA JA      CALL  JABAH          ;Print graphic character
3AAE   C9            RET                  Ready!
3AAF   Fl            POP   AF             ;Load character to output
3AB    18 E9         JR    JA9BH          ;output unchanged
JAB2   Fl            POP   AF             Load character to output
3AB3   1F            RRA                  on carry in maximum. bit position
JAB4   18 ES         JR    JA9BH            {and spend
```

#k# # i % it # %MM}i

Print Characters on the Printer

```
3AR     B7              OR    A              ;Block or Reverse            ?
3AB7    Profit JA       JP    M, 3AD8H       Yes!
JABA    FS              PUSH  AF             ;excl. Save Characters
34RB8   ES 3A CD        CALL  3AE8H          ;BREAK key pressed?
3ABE    D2 C4 3A        JP    NC,33AC4H      No!
3AC1    F1              POP   AF            Exp. Load Characters
3AC2    37              SCF                  Set CarryFlag
3AC3    C9              RET                  ;and back
3AC4    DB 00           IN    A, (00)        ;Read Port 0
34c     CB 47           BIT   0,             BUSY?
3AC8    20 FI           JR    NZ,3ABBH       ija" wait
3ACA    F1              POP   AF             ;excl. Load Characters
3ACB    D3 0E           OUT   (@EH),A        Output Byte
3ACD    D3 0D           OUT   (0DH) , A      ;Run Strobe
JACF    FE OD           CP    0DH            ;Carriage Return?
3AD1    37              SCF                  {Set Carry
3AD2    3F              CCF                  ; Reset Carry
3AD3    C0              RET   NZ             No!
3AD4    3E 04           LD    A, MAH         Print Line Feed
3AD6    18 E2           JR    JABAH
3AD8    C 77            BIT   6.A            inverted character?
3ADA    CA 73 2C        JP    Z,2C73H        no, output block graphics
  3D    E6 3F           AND   3FH            ;Bits 6.7 Delete
3ADF    C3 56 39        JP    3956H             ;Print inverted character
```

%#t±}Hi %%%% %~%M~Hi%%%

Output carriage return (from driver 5A6)

```
3AE2    3E 0            LD    A, DH          ;Load code for CR
JAE4    CD BA YA        CALL  JABAH          and spend
3AE7    C9              RET
```

With tt ± i tat} Mt3

Verify that the BREAK button is depressed

etc. Carry=1 when pressed

```
  3AE8  B7              OR    A              ;Delete Carry
JAE9    3A FD 68        LD    A, (68FLUH)    ;Load Keyboard Line 2
JAEC    C 57            BIT   2.A            ;CTRL key?
JAEE    C0              RET   NZ             $no!
3AEF    JA DF 68        LD    A, (68DFH)     ;Load Keyboard Line 6
```

```
3 3     SCF                                 Set Carry Flag
3 C     BIT         24A                     ;BREAK key?
3 C     RET         or                      Yes!
3 3     CCF                                 no" carry flag              d
3 C     RET
n c
        ************************************************************************
        ************************************************************************
                              Check BREAK Key
                        when pressed,  Interrupt execution
3 E     CALL        3AEBH                   ;Check BREAK Key
3 D     RET         NC                      ;not pressed'
3 E     POP         HL                      ;Remove bounce address
3 E     POP         Hl                      Load {Programme pointer
3 3     LD          A, (7839H)              ;Load flag 2
3 E     AND         0B7H                    iCRUN and Delete VERIFY Flag
3 3     LD          (7839Hl,A               Save flag 2
3 3     LD          A, 1
3 E     EI                                  Enable Interrupt
3 C     JP          1DA0H                   ;back to BASIC
1 3
        ************************************************************************
        ******************
        Wait until output buffer on screen output
        fully issued
3 3     LD          (A, 1789CH)             Load i6Device Type
3 E     OR          A                       i= 0? (Screen)
3 C     JP          NZ,21b4H                ;no, back immediately
3 3     LD          A, 7AAFH)               ;Load buffer counter
3 E     OR          A                       = 0? (buffer empty)
3 2     JR          N7, 3M13H               No, wait
3 C     JP          21b4H                   Jas back
R 3
        %With# iit}t # Mt
        when image output buffer is empty, load column pointer
3 3     LD          A, {7A4FH)              ;Load putter counter
3 E     OR          A                       ; Buffer empty?
3 c     RET         NZ                      ;no, back
3 3     LD          A, (7846H)              yes, load column pointer
3 C     RET
8 c
        ************************************************************************
        ************************************************************************
```

```
3B25  21 EF 68      LD     HL.68EFH        ;Address keyboard 4
3B28  CB 66         BIT    4.CHU           Space key pressed?
3B2A  20 18         JR     NZ, 3R44H       No!
3B2C  CD 48 3B      CALL   3B48H           ;Unlock Key
3B2F  CB 66         BIT    (HL)            wait until the key is released
3B31  28 FC         JR     Z,3B2FH
3B33  CD 48 3B      CALL   3B48H           Unlock Key
3R36  CD F8 3A      CALL   JAFBH           ;BREAK key pressed?
3B39  CB            BIT    4, (H)          ;no, space key pressed again?
3B3B  20 F9         JR     NZ, 3R36H       No, wait
3BJD  CD 48 3       CALL   3B48H           Unlock {key
3M48  CB 66         BIT    4+ (HL)         wait for key to release
3B42  28 FC         JR     Z,3B40H
3B44  FF 21         LD     HL,0FFFFF       act.Line number = Direct button
3:847 C9            RET
                  Unlock Key
31148 21 FF 07      LD     HL.7FFH         ;Queue Counter
3B4B  211           DEC    HL              {counter - 1
3B4C  7D            LD     A,L             i= 0?
3B4D  B4            OR     H
3B4E  20 FB         JR     NZ,3B4BH        {no, continue counting
3B50  21 EF 68      LD     HL.68EFH        ;Address keyboard 4
3B53  C9            RET                    ;finished'
```

fit%%i i 4%%%%~~~%

print data on cassette at PRINTi

```
3154  CD 11 35      CALL   3511H           Write {Byte to Cassette
31157 C9            RET
```

#% %4%±~1#%~$%%3% i3 %4~

Write header on cassette at PRINTi

```
3B58  F3            DI                     ;Disable Interrupts
31159 23            INC    HI.             programme pointer + {
3B5A  0E F2         LD     , 0F2H          ;Load tag
3B5C  CD 58 35      CALL   3558H           ;sync bytes, header, connoisseur,
                                           ;Emit filename on cassette
3B5F  RD 3A         JP     C,3AFEH         BREAK! End
3B62  2B            DEC    HL              { Programme pointer - 1
3B63  CF            RST    8               is filename with ₇₇     and       ?
3B64  22            DEFB   ..
35    CF            RST    8               Then comes a coma?
```

```
3R6     2C                  DEFB    ¹
3B67    C9                                      Yes, done.'

                            %## # # i # # t %. # ## #i # # i #

                            in INPUT# read the header and filename of the cassette

3B68    F3                  DI                  ;Disable Interrupts
3B69    23                  INC     HL          {Programme pointer + 1
JB6A    CD SC 35            CALL    358CH       ;Filename
3B6D    23                  DEC     HL          {Programme pointer- 1
3B6E    CF                  RST      8          ;Name with'"'?
3B6F    22                  DEF8    ,. .
3B70    CF                  RST      8          Then a comma follows?
3871    2C                  DEFB    .'  ¹
3B72    E5                  PUSH    HL          Yes, programme pointer to the stack
3873    CD BI 35            CALL    35B1H       Prepare {message output
317     21 42 38            LD      HL.3842H    ;Text 'WAITING'
3B79    CD F4 37            CALL    37F4H       ;and spend
387C    CD E7 35            CALL    35E7H       ;Find File on Cassette
3B7F    3A D2 7A            LD      A, !7AD2H)  Load File Kenner
3B82    FE F2               CP      0F2H        {correct for data?
3B84    20 F6               JR      NZ, 3B7CH   No, keep surfing
3B8b    Et                  POP     HL          ija, load programme pointer
3B87    C9                  RET                 done!

                            %%Mt%HE ttt} Mt Et# }# With Wed

                            Read data from cassette on INPUTt

3M88    CD 75,337           CALL    3775H       ;Read cassette byte
3B8B    FE 0D               CP      IDH         End of sentence?
3B8D    Co                  RET     HZ          No Back
3B8E    FS                  PUSH    AF          ;Save Characters to Stack
3B8F    CD F9 20            CALL    20F9H       Output iCR/LF
3B92    F1                  POP     AF          ;Load the character again
3M93    C9                  RET                 Ready!

                            fiHH!fftH*****************************

                            Pixel table for inverse character output on
                            the printer. Per character 5 bytes
3B94    Cl BE A2 NE BI      DEF8    %CIH, OBEH, 042, 04EH, B1H      ;8
3B99    83 ED EE ED 83      DEF8    83H, 8ED0H, 0EEH, 0EDH, 83H     {A
3M9E    8 B B M CI          DEF8    80H, 6BH, 0RH, 8BH, CIH         1B
3MA3    C1 BE BE DD         DEFB    0C1H, 8REH, OBEH, 8BEH, 0DDH    5c
```

```
3BA8      80 BE BE Cl    DEF.B        80H,BEH,0BEH,0.BEH,0C1H        D
3BAD    80 .B6 B B RE    DEF.B     80H4, 0BH, 0BR6H, 0B6H, 0BEH      1E
3.BB2     80 F F6 F6 FE  DEF.B     80H,0F6H,0F6H,0F6H,0F6H,0FEH      F
3BB7        Cl BE AE SC    DEFB        0C1H,BEH,0BEH,lilAEH,8CH     1G
JBBC    80 F7 F7 F7 80     DEFB   80\, 0F 7H, OF 7H, OF 7H, 80H     {H
3BC1     FF RE 80 BE FF    DEFB       0FFH, 0REH, 8MH, 0EH, OFFH    ;I
3BC6    DF BF BF C0 FE   DEF.B      0DFH,0.BFH,0.BFH,0C0H,0FEH      J
3BCB 8 F7 Eli DD BE       DEFB    B80\ 0F 7H, OE Bra, ODDH, 0REH    K
3.BD0   80 BF BF BF BF     DEFB    80H, 0RFH, QBFH, ORF H, OBFH     iL
3RD5    80 FD FJ FD 80     DEFB 80H, 0F D0H, 0F 3H, OF DH, 80H      ;
3.BDA    80 FD FB F7 80    DEFB         80H,0FDH,0FBH,0F7H,80H      N
JBDF        Cl BE BE Cl    DEFB       0C1H,0.BEH,0BEH,0.BEH,0C1H    ;0
3.BE4    80 F6 F6 F6 F9    DEFB    80H,0F6H,0F6H,0F6H,0F6H,0F9H     ;P
3BE9    Cl BE AE DE Al     DEFB         0C1H,0BEH,0ÄH,0DEH,0A1H     ;4
3BEE    80 F E D B         DEFB   80H, 8F 6H, OE 6H, 0D6H, OB9H     iR
3BF3   D9 B6 B6 B CD      DEF.B       0D9H,0.B6H,0B6H,0.B6H,0CDH    ;S
3BF8      FE FE 8 FE      DEFB    0FEH,0FEH,80H,0FEH,0FEH,0FEH     iT
3.BFD C0 BF BF .BF C0     DEF.B 0C0H,0BFH,0BFH,0BFH,0BFH,0C0H      ;U
3C02    FS E7 9F E7 F8     DEFB         0F8H,0E7H,9FH,0E7H,0F8H     ;V
3C07    80 DF E7 DF 80     DEFB     80H, 0DFH, OE7H, DFH, BRA       ;
3C0C    9C ED F7 EB 9C   DEF.B    9CH, 8EDH, OF 7H, 0EBH, 9CH      ;X
3C11 FC F.B 87 F.B FC     DEFB        0FCH,0F.BH,87H,0F.BH,0FCH     ;Y
3C16    9E AE .B6 BA BC    DEFB        9EH, MAEH, 0BH, 0RAH, @RH    12
3C111       FF 80 BE FF    DEFB         0FFH,80H,0BEH,0BEH,0FFH     j[
3C20   FD FB F7 EF DF      DEFB  OF DH, OF BH, OF 7H, OEFH, @DFH    ;6
3C25      BE BE 80 FF    DEF.B       OFFH, REH, 0BEH, 80H, 0FFH     ; ]
3C2A    FB FD 80 FD FB     DEFB      0FBH, 0FDH, 80H, 0FDH, 0FBH    ;F
3C2F   F7 EJ D6 F7 F7      DEFB         0F7H,0EJH,0D6H,0F7H,0F7H     1E
3C34         FF FF FF      DEFB      OFFH, OFFH, OFFH, OFFH, OFFH
3C39      FF FF A0 FF    DEF.B         0FFH,0FFH,0A0H,0FFH,0FFH     4
3C3E      FF FB FF FF      DEFB    OFFH, OF BH, OFFH, OF 8H, OFFH     1
3C43 EB 80 ER 8 ED        DEFB    0EBH,80H,0EBH,80H,0EDH             ;#
3C4B   DB D6 80 D6 ED      DEFB         0DBH,0D6H,B0H,0D6H,0EDH      j$
3C4D   D9 E9 F7 CB CD    DEF.B         0D9H,0E9H,0F7H,0CBH,0CDH     3%
3C52   C9 D6 A9 DF AF      DEFB       0C9H,0D6H,lilA9H,0DFH,0AFH     and
3C57      F7 FB FC FF      DEFB    8F 7H, 0F 8H, 0F CH, FFH, FFH     ▪
3CSC    FF EJ DD BE FF    DEF.B        0FFH,0E3H,0DDH,0.BEH,0FFH     j(
3C61    FF BE DD EJ FF     DEFB     FFH, OBEH, ODDH, OE 3H, WFFH     (j)
3C66     D6 EJ 80 EJ D5    DEFB       006H, 0EJH, 80H, 0EJH, 0DSH    i*
3C6.B   F7 F7 Cl F7 F7     DEFB      0F 7H, 0F 7H, 0C1H, F 7H, 0F    ;+
3C70       DF C7 F7 FF     DEFB     0DFH, OC7H, OF7H, OFFH, NFFH     ,
3C75    F7 F7 F7 F7 F7     DEFB    OF 7H, 0F 7H, OF 7H, OF 7H, 0F    ;-
3C7A       FF 9F 9F FF     DEFB      OFFH, 09FH, Q9FH, QFFH, 0FFH    ;.
3C7F   EN EF F7 F B FD     DEFB         0DEH,0EFH,0F7H,0FBH,0FDH     j/
```

```
3(84)  Cl AE B6 BA Cl     DEFB   0C1H, OAEH, 0BH, 0RAH, OC1H     10
3C89   FF BD 80 BF FF     DEFB   OFFH, 0RDH, 80H, 0BFH, 0FFH     j1
3C8E   9D AE B6 RA BD     DEFB   9DH, 0EH, 0BH, OBAH, @RDH       Ti-
3C93   DD BB BB BB C9     DEFB   0DDH, 0RBH, 0RBH, 0RBH, 0C9H   13
JC98   E7 EB ED 80 EF     DEFB   0E 7H, 0OE BH, @EDH, 8MH, 0EFH {4
JC9D   D8.1\ADAC6         DEFB   0D8H,0BAH,0DAH,0DAH,0DAH,0C6H  ;s
JCA2   CI B6 B B6 CF      DEFB   0C1H, 0BH, 0BH, @B6H, 0CFH     16
JCA7   FC FE 8 FA FC      DEFB   OF CH, OFEH, BRA, OF AH, OF CH ;7
JCAC   C9 B B R C9        DEFB   0C9H, 0BH, 0RH, @B6H, &C9H     ;8
3CB1   F9 B6 B6 B6 Cl     DEFB   0F9H,0B6H,0B6H,0B6H,0C!H       9
JCBb   FF C9 C9 FF        DEFB   0FFH,0C9H,0C9H,0FFH,0FFH       ;
3CBB   BF C4 E4 FF FF     DEFB   0BFH,0C4H,0E4H,0FFH,0FFH       5
JCC0   F7 EB DD DE        DEFB   0F7H,0EBH,0DDH,0DEH,0DEH       ;
JCC5   EB EB EB EB EB     DEFB   OE Bra, OE Bra, 0E Bra, OE Bra, ▬
JCCA   EN DE DD EB F7     DEFB   0DEH,0DEH,0DDH,0EBH,0F7H       ;>
JCCF   FD FE A6 FA FD     DEFB   0FDH,0FEH,0A6H,0FAH,0FDH       ±7
```

%## # # # # # # # #l # ik %### # # # k # ### %## d#±

Feh lernel Run

Eing. E = Error ID
HI. = Error table address

```
JCD4   CB JB              SRL    E              ;error number/ 2
3CD6   1C                 INC    E              + 1
3CD7   7E                 LD     A, (HL.)       Load iB!lte from error table
3CDB   23                 INC    HL             ;table address+ 1
JCD9   B7                 OR     A              {new Hel dun9 ?
3CDA   F2 07 JC           JP     P,3CD7H        No!
3CDD   10                 DEC    E              {Feh]ernumr - 1 = 0?
JCDE   20 F7              JR     NZ,3CD7H       {nin, not the right dose
3CE0   E 7F               AND    7FH            ;Delete Bit 7
3CE2   CD 2A 03           CALL   032AH          Output Byte
JCES   7E                 LD     As (HL)        ;next B!lte from Error tab.   load
JCE6   23                 INC    HL             ;table address+ 1
JCE7   B7                 OR     A              New 11 message?
3CE8   F2 E2 3C           JP     P,JCE2H        no, output byte
JCEB   C9                 RET                   Yeah, done
```

%Hit i ;i Mi} Mt~ fit ##i Mi t}&

table *of* error messages

```
JCEC   CE                 DEFB   'N'+80H        ;NEXT WITHOUT FOR
JCED   45 58 54 20 57     DEFN   'EXT WITHOUT FOR'
```

```
      49 54 48 4F 55
      54 20 46 4F 52
3CFC  D3                DEFB  'S'+80H         iSYNTAX
3CFD  59 4E 54 41 58    DEFM  'YNTAX'
3D02  D2                DEFB  'R'+80H         iRETURN WITHOUT GOSUB
3D03  45.54             DEFM  'ET'
3D05  27                DEFB  27H
3D06  4E 20 57 49 54    DEFN  'N WITHOUT 60SUB'
      48 4F 55 5 28
      47 4F 53 55 42
3D15  CF                DEFB  '0'+80H         ;OUT OF DATA
3D6   55 54 28 4F 46    DEFN  'UT OF DATA'
      2@ 44 41 54 41
3D020 C                 DEFB  F' +8MH         FUNCTION CODE
3D21  55 4E 43 54 49    DEf11 'UNCTION CODE'
      4F 4E 20 43 4F
      4 45
3D2D  CF                DEFB  '0'+80H         ;OVERFLOW
3D2E  5 45 52 46 4C     DEFN  'VERFLOW'
      4F 57
3D35  CF                DEFB  '0'+80H         ;OUT OF NENORY
3D36  55 54 20 4F 4     DEf11 'UT OF
      20 4D 45 4D 4F
      52.59
3D42  D5                DEFB  'U'+80H         ;UNDEFINED STATEMENT
343   4E 44 45 46       DEFII 'NDEF'
3147  27                DEFB  27H
3D48  44 20 53 54 41    DEFM  'D STATEMENT'
      54 45 4D 45 4E
      54
3D53  C2                DEFB  'B'+80H         ;BAD SUBSCRIPT
3D54  41 44 28 53 55    DEFN  'AD SUBSCRIPT'
      42 53 43 52 49
      50.54
3D    D2                DEFB  'R+80/          ;REDil1ENSIONED ARRAY
3D61  45 44 49 4D       DEf11 'EDIN'
3D65  27                DEFB  27H
3D66  44 2 41 52 52     DEF11 'D ARRAY'
      41 59
3D6D  C4                DEFB  'D'+80H         ;DIVISION BY ZERO
3D6E  49 5 49 53 49     DEF11 '!VISION BY ZERO'
      4F 4E 20 42 59
      28 5A 45 52 4F
3D7D  C9                DEFB  'I'+80H         ;ILLEGAL DIRECT
```

```
3D7E   4C 4C 45 47 41      DEFM   'LLEGAL DIRECT'
       4C 20 44 49 52
       45 43 54
3D8B   D4                  DEFB   'T'+80H          ;TYPE MISMATCH
3D8C   59 50 45 20 4       DEFM   'VPE
       49 53 4D 41 54
       43.48
3D098  CF                  DEF.B  '0'+80H          ; OUT OF SPACE
3D99   55 54 20 4F 4       DEFM   'UT OF SPACE'
       20.53.50.41.43
       45
3DA4   D3                  DEF.B  'S'+80H          ;STRING T00 LON6
3DA5   54 52 49 4E 47      DEFM   'TRING TOO LON6'
       20 54 4F 4F 20
       4C 4F 4E 41
3D.B3  C6                  DEF.B  'F'+80H          ;FORMULA TOO COMPLEX
JD.84  4F 52 4D 55 4C      DEFII  'ORIIULA TOO COIIPLEX'
       1  20 54 4F 4F
       20 43 4F 4D 5111
       €45.58
3DC6   CJ                  DEF.B  ' €'+80H         ;CAN'T CONTINUE
3DC7   41 4E               DEFII  'AN'
3DC9   27                  DEFB   27H
3DCA   54 20 43 4F 4E      DEFN   'T CONT'
       54
3DII   CE                  DEF.B  'N'+B8H          ;NO RESU1'IE
3DD1   4F 20 52 45 53      DEFN   '0 RES\JNE'
       55 4D 45
3DD9   D2                  DEFB   'R'+801          ;RESVNE WITHOUT ERROR
3DDA   45 53 55 4 45       DEFN   'ESU1'IE
       20 57 49 54 48
       4F 55 54
3DE7   D5                  DEFB   'U'+80H          ;UNPRINTABLE ERROR
JDE8   4E 50 52 49 4       DEFN   'NPRINTABLE'
       54 41 42 4C 45
3DF2   CD                  DEFB   'I'I'+80H        ;1'IISSING OPERAND
3DF3   49 53 53 49 4       DEFN   'ISSING OPERAND'
       47 20 4F 50 45
       52 41 4E 44
3Eo!   C2                  DEFB   'B'+00%~         ;BAD FILE DATA
3E02   41 44 28 4 49       DEFN   'AD FILE DATA'
       4C 45.2044.41
       54.41
JEIiiE C4                  DEFB   'D'+SIIIH        DISK C01'11'1AND
```

```
3E0F  49      DEFM                 'ISK COl1NAND'
      4F
      44
3E1A  3F      DEFM                 '?SYNTAX ERROR'
      41
      52
3E27  0D 8111 DEFB                 0DH, 8MH

              %%%% ±# # %3#± #tf i iti±

3E29  7E      LD                   A, (Hl)
3E2A  117     OR                   A
3E21  28.8    JR                   NZ.3E34H
3E2D  3E      LD                   A,'
3E2F  77      LD                   (HL),A
3E30  23      INC                  HL
3E31  A       XOR                  A
3E32  77      LD                   (HL),A
3E33  2B      DEC                  HL
3E34  2B      DEC                  HL
3E35  Ft      POP                  AF
3E36  C9      RET

              hffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
              ftfftfftfftfftfftfftfftfftffffffffftffffHtfftfffffffffffffffffffffffffffff

              Initialisation Part

              Default colour = set yellow
3E37  32 7D   LD
3EYA  3E 10   LD                   A, 10H
3EJC  32 4    LD
3E3F  C9      RET

              Hfffffffffffllllttftftffffffffffffffffffffffffffffffffffffffffffffffffffffffll

              Extra routine to read a line <RDLll'E)

              green background and black plot

              transfer the data from the image to 1/0-llutter
3E48  7E      LD                   A, (H)
3E41  Cl!     BIT                  in
3E43  28.05   JR                   Z,3E4AH
3E45  FE 88   CP                   80%
3E47  DA SD   JP                   C, 3E5DH
      3F
```

```
3E4A  Cl              POP   .BC          If not INPUT, then
                                         ;Graphics and Inverse only in
                                         ;Str1ngs approved
3E4B  11 53 JE        LD    DE,JESJH     ;Backjump address in stack
3E4E  D5              PUSH  EN
3E4F  CS              PUSH  BC
3E50  CJ 02 05        JP    0502H        {Check text ID (BREAK)
3E53  Profit          RET   C            ;BREAK' back to BASIC
3E54  21 lA JE        LD    HL,YElAH     ;Text 'SYNTAX ERROR"
3E57  CD A7 28        CALL  28A7H        and spend
3E5A  CJ E3 03        JP    83E3H        back to line input
3E5D  RD 2            CP    '"+40H       {String Identifier?"> 7
3ESF  20.39           JR    NZ, 3E9A}    No, go on
3E61  E BF            AND   0BFH         Delete Mt 6
3E63  12              LD    (EN),        ;Character in 1/0 Bufter
3E64  23              INC   HL           ;Image Address
3E65  13              INC   EN           butter address +1
3E66  05              DEC   B            ;Character counter - 1
3E67  CA EE           JP    Z,04EEH      if , stop the transfer

                      Inlet of 3EAF on green background

3E6A  7E              LD    A,(HL)       ;Load character from
3E6B  CB 7F           BIT   7.A          ;graphic character?
3E6D  20O             JR    NZ.3E75H     Yes!
3E6F  CB 77           BIT   6n           inverted character
3E71  20 ilC          JR    NZ,3E7FH     No!
3E73  18 06           JR    3E7BH        Yes!
3E75  E6 8F           AND   8FH          {Graphics, Bits 4546 Delete
3E77  F6 8            OR    8MH          ;Set Bit 7
3E79  18 17           JR    3E92H
3E7B  F6 CO           OR    0CM8H        Set Bits and 7
3E7D  18 13           JR    3E92H
3E7F  FE 62           CP    •+'+4%}      ;it's a '"  7
3E81  20.89           JR    NZ,3E8CH     No!
JE83  IT              PUSH  HL           Save H
3EB4  21.39.78        LD    HL.7839H     ;Address flag 2
3E87  CB 66           BIT   +(H)         INPUT-Kcaando?
3E89  El              POP   HL           ;HL 111 Reload
JEBA  28O             JR    Z,JE9AH      no - from now on, graphic and
                                         ;Inverse not allowed
3EBC  CB 6F           BIT   5.A          ;characters in real ASCII code
```

```
3E8E    28 02          JR      Z.3E92H         Iunandein if false
3E908   E6 BF          AND     0BFH            ; (delete bit if necessary)
3E92    12             LD      (EN),           ;character in I/0 buffer
3E93    23             INC     ut..            Image Address + 1
3E94    13             INC     EN              Bufferaddress + 1
3E95    10 D3          DJNZ    3E6AH           i counter - 1
3E97    C3 EE          JP      04EEH           ;= @,then finished
3E9A    CR 6F          BIT     5.A             ;characters in real ASCII code
3E9C    28 02          JR      Z,3EA0H         Convert $if wrong
3E9E    E BF           AND     0BFH            (Bit 6 delete if necessary)
3EA0    12             LD      (EN),A          ;Character in I/O Buffer
3EAI    23             INC     HL              {Image Address + 1
3EA2    13             INC     EN              ; Butfer Address + 1
3EA3    10 9B          DJNZ    3E40H           ;counter - 1
3EA5    C3 EE 04       JP      04EEH           ;= W transmission stopped
3EA8    3A 18.78       LD      A, (7818H)      depending on background in
3EAB    B7             OR      A               ;appropriate routine.
3EAC    C2 B8 04       JP      NZ, 0BH         {black background
3EAF    C3 6 3E        JP      3E6AH           ;Green Background
```

%%% #~4~~ 3 %% %%13%%%

Invert Characters

```
3EB2    34 18 78       LD      A, (7818H)      Load iBackground Flag
3ER5    B7             OR      A               Black background?
3EBb    20 03          JR      NZ,JEBBH        Yes!
3EB8    CB B           RES     6, (HL)         ;green, delete 6 bit
3EBA    C9             RET
3ERB    CB F           SET     6, (H)          ;black, set bit 6
3EBD    C9             RET
```

FFF

Provide delete character for image delete routine

```
3EYE    3 18 78        LD      A, (7818H)      ;Load Background Flag
3EC1    B7             OR      A               Black background?
3EC2    3E 20          LD      A,' '           Load spaces
3EC4    20 02          JR      NZ,3ECBH        black, it's okay!
3EC6    F6 40          OR      401             Yar, hmm. Set Bit
3EC8    77             LD      t..i,A            ;Space in Image
3EC9    C9             RET
```

```
                        **********************************

                        Invert depending on the

                                Run Background
JECA   F5           PUSH   AF          ;Save character to output
JECB   3A 18.78     LD     A, (7818H}  ;Load Background Flag
JECE   B7           OR     A           Black background?
JECF   28 87        JR     Z,3ED8H     {no - green!
3ED1   Fl           POP    AF          {Reload Character
3ED2   E 3F         AND    3FH         ;Bits 6 and 7 delete
3ED4   E5           PUSH   HL          ;main routine inverted,
3ED5   C3 AB 31     JP     31ABH       ;if necessary,
3ED8   Fl           POP    AF          ;Load the character again
3ED9   Fb 4111      OR     4111H       Set Bit b
3EDB   IT           PUSH   Hl...       H Secure
JEDC   21.38.78     LD     HL.7838H    ;Address flag 1
3EDF   CB 4E        BIT    1, HL)      ; invert?
YEE1   El           POP    Hl...       ;Hl. restore
3EE2   28,082       JR     Z,EEbH      {no!
3EE4   E BF         AND    01.FH       Delete iBit b
YESb   C3 B5 31     JP     31B5H       {finished

#%lt# M MM Mk # Hi i #t i kM#kk Hi

                        Test characters to Blank
3EE9   3A 18.78     LD     A, (781H)   ;Load Background Flag
JEEC   B7           OR     A           Black background?
JEED   7E           LD     , HL.)      ;Load Characters
3EEE   20.83        JR     NZ,3EFJH    Ijas black!
3EF8   RD 6         CP     ''+40H      Check with bit 6 =I
3EF2   C9           RET
3EFJ   RD 20        CP     ' '         Check with Bit =
3EF5   C9           RET

ffffffffHffH 11111111111 I lfffffffffffff

                        Insert spaces in INSERT and RIJBOUT
JEFb   3A 18.78     LD     A, (7818H)  ;Load Background Flag
3EF9   B7           OR     A           ;black?
3EFA   3.28         LD     A,' '       Load spaces
JEFC   20 02        JR     NZ,3F011H   Yes!
3EFE   F6           OR     40H         ; to set green bit
3F88   12           LD     (EN),       in Image Memory
```

```
  3F01  C9              RET

                        %%% 3# % # # #}f##'f # £ # tk E

                        door roll routines prepare line deletion

  3F02  6 20            LD    3.32          Load Length of Line
  3F04  YES 18 78       LD    A, 7818H1     ;Load Background Flag
3F1117  B7              OR    A             Black?
  3F08  3E 20           LD    A,' '         Load spaces
  3F0A  C111            RET   NZ            Yeah, done
  3F0R  F 4             OR    40H           to set green bit
3F111D  C9              RET

                        %%%%~ %. % %#~$%} Mt#t#IttiE

                        Help routine for loading a cassette
                        for the correct presentation of the l'teldungen,
                        depending on the background,

                        Call of 3809H (general information in last line)

  3F0E  11 E0 71        LD    DE, 71E111H   ;Address last line
  3F11  YES 18 78       LD    A, (7818H)    ;Load Background Flag
  3F14  B7              OR    A             Black?
  3F15  C0              RET   NZ            $Yes; no action
  3F16  Fl              POP   AF            Clean up the stack
  3F17  7E              LD    A, HL)        ;Load text character
  3F18  B7              OR    A             Extender?
  3F19  approx          RET   or            Yeah, done
  JFlA  CB 117          RES   O             Delete {Bt
  3F1C  12              LD    (EN) ,A       ;Print character inverted
  3F1D  13              INC   EN            ;Image Address+ 1
  3F1E  23              INC   HL            Text address + 1
  3F1F  18F             JR    3F17H         ;next byte

                        Author of 382BH (File Identifier Output)

3F21    34 18 78        LD    A, 7818H1     Load iBackground Flag
  3F24  B7              OR    A             Black?
3F25    7E              LD    A, (HI)       ;Load Kenner
3F26    20 07           JR    NZ,3F2FH      Yes!
3F28    CB F7           SET   6.A           ;green, Set 6 Bit
3F2A    12              LD    (EN),A        Transfer to Image
3F2B    13              INC   EN            ;Image Address+ l
3F2C    3E 7A           LD    A,':'+4M4     ; separator t. to load green
3F2E    C9              RET
```

```
3F2F  12          LD    (EN),A      Kenner 1ns Bl& {f,   black)
3F30  EACH 3A     LD                Load {separator
3F32  C9          RET

                  Invoke 38J7H (output of data/Pr ogr to name!
3F33  F5          PUSH  AF          Save Characters
3F34  3A 18 7B    LD    (A, 17818H) {Background flag  Jaden
3F37  1i7         OR    A           Black?
3F38  20.05       JR    NZ,3F3FH    Yes!
3F3A  Fl          POP   AF          load grn' character
3F3B  Fb 40       0R    40H         {Set Bit 6 (f, black darst.)
3F3D  12          LD    (EN),A      ;Print Character on Image
3F3E  C9          RET
3F3F  Fl          POP   AF          Black! Reload Characters
3F40  E6 3F       AND   3FH         ;Delete bits b and 7
3F42  12          LD    (EN),A      ;Print Character on Image
3F43  C9          RET

                  # k kt F #k ###k # ##ik # kk# # #
                  # k # #k ###

                  (CPY Statement helper routine

                  called from 3921H
3F44  F5          PUSH  AF          ;Save Characters
3F45  3A 18.78    LD    A,17B18Hl   Load iBackground Flag
3F4B  B7          OR    A           Black?
3F49  20 09       JR    NZ.3F54H    Yes!
3F4B  Fl          POP   AF          ;green! Reload Characters
3F4C  CB 77       BIT   b           ;inverted character?
3F4E  C2 38 39    JP    NZ.3938H    ;no, normal output
3F51  C3 31 39    JP    3931H       ;yes, inverted output
3F54  Fl          POP   AF          ;black! Reload Characters
3F55  CB 77       BIT   6sA         ;inverted character?
3F57  CA 38 39    JP    Z.3938H     ;no, normal output
3F5A  C3 31 39    JP    3931H       ija inverted output
3F5D  C3 31 39    JP    3931H       ;Not used!

                  HffffffHHHHIHfHHHHHFFFFF

                  On-screen character output helper

                  Adjusting the inversion to the background colour
                  3149H usage
3F611 F5          PUSH  AF          ;Save Characters
3F61  34 18 78    LD    A,(7818)    ;Load Background Flag
```

```
3F  B7          OR    A              Shuarz?
3F65 20%        JR    NZ,JF6DH       Yes!
JF67 Fl         POP   AF             green! Reload Characters
3F68 E6 JF      AND   3FH            ;Delete bits b and 7
JFbA C3 54 31   JP    315.ltH
3F6D Fl         POP   AF             ;black! Reload Characters
3F6E E6 7F      AND   7FH            ;Delete Bit 7
3F78 C3 5-4 31  JP    315.ltH
```

%3%%%t%%~%1%%%it t}

Cassette Scan Utility

369CH usage

```
JF73 CD 75 37   CALL  3775H          Read Cassette Byte
3F76 Dll        RET   NC             Okay!
3F77 El         POP   HL             {Return address of stack
3F78 C3 11 37   JP    3711H          Read errors!
```

fffHfHffhffffffff*HfHffffffff

When choosing a new background colour

Convert brain content
Called from **the** interrupt service routine

```
3F7B  3A 19.78  LD    A, (7819H)     All right. Background flag
3F7E  47        LD    B,A
3F7F  3A 18.78  LD    A <7818Hl      -
3F82  B8        CP    B
3F83  CA E8 30  JP    1,38EB8#       yes" to character output
3F8b  32 19 78  LD     (7819Hl,A     All right. Background flag
3F89  21.80.78  LD    HL,708Wed      ;Load image start address
JFBC  01 080 02 LD    BC.512         Screen Greetings
JFBF  7E        LD    A, (H)         ;Load Characters
3F90  B7        OR    A              ;Block Graphics 2
3F91  FA 97 3F  JP    N,3F97H        ;Yes, leave unchanged
3F94  EE 40     XOR   40H            ;Bit 6 !Invert) tilt
3F96  77        LD    (H)            ;and write back
3F97  23        INC   HL             Image Address + 1
3F98  O         DEC   BC             Character counter - 1
3F99  78        LD    A, B           -
3f9A  Bl        OR    C
3F9B  20F2      JR    NZ,3f8FH       no, next character
3F9D  C3 E8 3o  JP    30EB8H         Yeah, done!
```

With tttiitWith}it

- Z

```
                        Check whether during initialisation
                        the CTRL button is pressed
3FA0   3A FD 68         LD      A, (68FDH)      ;Load Keyboard Line 2
3FA3   CB 57            BIT     254             ;CTRL key pressed?
3FA5   3E 20            LD      A,' '           spaces in A
3FA7   20.08            JR      NZ,3FB1H        Not actuated!
3FA9   F6 40            OR      40H             {Set Bit {Black Background.)
3FAB   32 18 78         LD      (7818H),A       Background flags for black
3FAE   32 19 78         LD      (7819H),A       ;Set
3FP.1  32 3C 78         LD      (783CHJ,A       Spaces as Cursor Backup
3FB4   C3 C9 01         JP      01C9H           {Continued initialisation
```

Directory of the Rt-variable } vectors and -lenger, which are BASIC - Interpreter. {with cross references to ROM)

The range 7800H to 7835H is used during system initialisation from the ROM range.

7800	C3 96 1C	JP	1C96H	;RST B - Vector 10000, 0679
7803	C3 78 1D	JP	1D78H	iRST 10 Vector **;0010**
7806	C3 **90** 1C	JP	1C90H	;RST 18 Vector 70018
7809	C3 D9 25	JP	25D9H	;RST 20 Vector 10020
780C	C9 **00 00**	RET		;RST 28 Vector 1,0028
780F	C9 **00 00**	RET		;RST **30** - Vector **1,0030**
7812	FB	EI		iRST 38 Vector
7813	C9 **00**	RET		**;(**not addressed)

7815	01	Keyboard Device Control Block (DCB)
7816	Fit 2E	DCB-Kenner **i002B** Driver Address
7818	00	Background flag (@=green, 1=black) i048E 3EA8 3EB2 3EBE 3ECB 3EE9 3EF6 ;3F04 3F11 3F21 3F34 3F45 3F61 3F7B
7819	**00**	All right. ;3F7B **3F86** 3FAE
781A	**00**	Background
7B1B	4B 49	'AI'

Device Control Block (DCB) screen
(unused in LASER 110-318> DCB identifier (deleted) ;0033

781D **00** 781E

Pointer to programme start address

00 00 at CLOAD 538D 36CB.

7820 **00** 78 cursor address 10050 0311 034D 03E8 041 01E 042A **50468 0505 055 2083** 2EEC 30833 3113 129 31F5 3202 3207 323:8 3247 325C ;3266 3276 3280 328A 3295 3284 32C2 ;32D7 33 08 332C 33A3 33DI 35CB 3657

```
7822 00
7823 00 00

                        Checksum for Cassettes Input/Output
                                                ;34BE 367D

7825    06
7826    8D 05           Printer Device Control Block (DCB)
7828    43              DCB identifier          i003B
7829    00              Driver Address
782A    O               Lines/Be te+l
782B    50.52           line counter

782D    C3 00           'PR'
        50
7831.1  C7 00          JP      5000H           ;unused ;u
        00
7833    3E 00           RST 0                   nused

7835    C9

7836

                        Buffer BI for 1. Key Code Multiple
7837                    Key Operation           {603 086038 0635 065D 01 2F2 2FFR

                        Putter B2 for 2. Key Code Multiple
7838                    Key Operation           {@5FF 62E 3B 064 068 2FFF
                                                for unknown DCB identifier A=0
                        LD      A,0             103D1
                        RET
                                                i0517 051C 05D7 05F4 0625 05 2F0E
                        FLAG 1                  i2F20  2F83  2F9B  2FA9  2FCD  2FD7
                        Bit 7 - CONTROL-Flag    2FF0 ;31!139 31AB 3EDC
                        Bit - REPEAT-Flag
                        Bit 5 - WAIT-Flag
                        Bit 4 - B2 status
                        flag Bit 3 - Bl status
                        flag Bit 2 - FUNCTION
                        flag Bit 1 - INVERSE
                        flag Bit 0 - SHIFT flag
7839                    FLAG 2                  10183 03E3 1!1405 0425 1!14CE 051F 052C
                        Bit 7 - unused           ;0567 2EC9 2EDC 31!128 304 31D02 3212
                        Bit 6 - CRUN-F1a9        13253 32D 34308 3694 36D9 3E3 372F
                        Bit 5 - Ini-Flag t, buffered output       ;3739 3758 3AFE
                        Bit 4 - Flag f. INPUT Statement           {3B3 3EB84
                        Bit 3 - VERIFV flag
                        Bit 2 - BREAK flag
                        Bit 1 - BUZZER Flag
```

Bt @-Carriage-Return Flag

783A	time counter	;05DF 05E6 2F15 2FC1 2FDE 2FE2 2FFA ;J00A 300E 3012 3491
783B	INPUT/OUTPUT Latch	;2C41 2E73 2E78 2E7F 2E84 308C 3095 1345€ 3489 352B 3542 35B6 35BF 38CA i38CF 38DB 38E0 3 914
783C	Character lock for cursor display	**;0054 030E** 32FS 3FB1
7831)-	unused	
7840 7841	flash	;2EE2 2EE9 3030 3496 10348
7842-7843	Keyboard Query Cache (row/column) counter	;0611 0643 2F6F
7844-7845	Keyboard Query Cache (matrix address)	{015 47 2F73
7846	colour	;0173 3150 38B9 3E3C
7847-7843	**Output Flag f. Notification output at** cassettes I/0 **QO** - News 111 suppressed)	
784C	code	;35B1 3719 37F4 3804 3814
784D-787C	unused unused	
787D C9	**00 0o**	**RAI'l extension output of the** interrupt service routine i2EBC 3E37

```
                        The range 7880 - 78A5 is filled when initialising from clelfl
                        ROM range

                        Subprogramme for Division
                        SUB     0               iSubtraction 22 - Z1
7880 D6 00                                      is modified before each call. 10075
                                                08RB 08CA
7882  F                 LD      L,A
7883  7C                LD      A,H
7884  EN 0              SBC     As              108B6
7886  67                LD      H,A
7887  78                LD      A, B
7888  EN 00             SBC     A,              ;88B1
788A  47                LD      B,A
788B  PER               LD      A+@             ;08C4 08D2 08F0 08F4
788D  C9                RET

                        USR - Start address
                        pre-populated 111it FUNCTION-CODE Error
788E  4A 1E                                     12815

7890  40 E6 4D         Multiplier f, RND       ;14F0

                        INP Subprogramme
7893  DR 00             IN      A, (0)          i2AF5 2AF2 2B1
7895  C9                RET

                        OUT Subprogramme

789   D3 00             OUT     (),A            i2AFE 2B14

7899  00                INKEYf Cache
                                                ;019F 01AD 1DA5
789A4 00                ERR last error code
                                                ;19B7 1A2B !FBE 1FB8 2400
789B  00                Printer position in the line
                                                ;038F 03B1 03B7 2005 211B 214E
789C  00                Device flag (0=Picture, !=Printer, 80—Cassettes
                                                €1032F 038 209 2098 20CC 2144 2169
                                                ;2171 2B2B 3B0C
789D  40                Line length on a full screen (pre-loaded with 64)
                                                i20DD
789E  30                last tab position (pre-filled with 48)
                                                ;2123
789F  00                unused
```

7848 47 *7B* String range start address
i00F6 1917 1B90 1E9C 1F4C 27E5 28C3
i28F0

78A2 FE *FF* current line number i197E 1994 19A2 1A36 lCCl 1D41 1DC!
i1DF2 1EB9 1EC9 1EF0 1FD6 231C 2829
i2B36 36F8

78A4 E9 7A Start address of the programme text
i191B 1AFB 1BZC AB4D 15D 1D92
1F4 ;31DE 34C2 36D5

78A6-78A7 Output Image Column Pointer
10410 0415 0551 2089 28EI 2153
27F5 ;30CE 3114 311A 31BF 31F2 3227
3235;3231 5 324F 328E 32A5 32B7
32BD 32CD ;3302 3318 333A 3356 335F
3JA8 33AE ;33D6 341D 3500 3728 JB21

78A7-78A8 Pointer to input/output buffer !from 79E8)
1008B 1ADB 1BC6 1C84 21AF 21C3 2BbA
i2B7F 2E5C

78A9 Input Flag (= Cassette)
72186 **2143 220E**

78AA last random number 5 1503 152
78AB 101D5
78AC ;1510,152F
78AD

78AE

D111 Statement Flag

7BAF i2b0E 26EA 2707 272F 2757

Tgp of value in !-Register

02 = Integer i01C4 09D3 0FE1 22FC 2374 2399 2410
03 = Strins 52465 2509 253 2716 274 27CR 2804
= simple accuracy ;2891
B8 = Double Precision

78B0

Flag for intermediate code generation at
DATA Operation Code during expression
analysis

78B1-78B2
78B3-78B4 i1B1C 1BDC 1C7 2408 2451

End address of BASIC memory range
100F2 049F 1B7A 1E84 28E6

pointer to string cache

; 1B9E 1DMA 2889 289R 28F8 295 29FF

78B5-78D2 String Cache (1@ x 3 bytes)
!1 Byte - length, 2 bytes - address in string range) ;1B9B
1DB7 28F4

7803-7805 1. string cache
(or top) 12854 2884 01B5 2914 2427 2457

78Db-78D7 Pointer to last free byte in string range
i1B7D 27E9 2897 28C7 28D3 28E9
2955 ;297F 29E8

78D8-78D9 General address cache Format flag **t.**
String output of a number
;0FDC 0FFB 1034 1289 2JSE 2368
2JCE ;2752 2784 2935 2940

DATA Line Number i1991 22AA
78DA-78DB and i217F 21F5 2207 226F 2CCB 2CD5 2DD9

78DC

index blocking flag
; 1MAA 1CA3 2657 265

78DD RESUME/RETURN - Flag ;IAAA 1EF8 1FEA

7BDE
PRINT USING DATA flag
intermediate buffer for INPUT
othe
rs
78DF-7BE8 general address memory
e.g., **NE** for FOR/NEXT
Adr. d, LET Variable Table
;1B61 1BAF 1D16 1F27 22BC 2328

AUTO Input - Flag **(** - no AUTO)
78E1 11453 1B53

CAR - Line number ;1A3F 1A6E 2032
78E2-78EJ CAR - Increased ;140 2019
7BE4-78E5
Current line address !FFFF = Direct play) i19BA 1A9E 1D25
7BE6-78E7 1DB4 1DE 219

7BEB-7BE9 Pointer to BASIC stack
;19AE 1B95 1C2 1D28 1EE5 22C 2325

78EA-7BEB Number of the line where the last error occurred
11945 19C1 1A0? IFD3 24DF

7BEC-
78ED number of the line where the last error occurred !,-Option on LIST!
{1948 1ABI 1E53 2B5B

7BEE-
78EF Address of the line where the error occurred 319BD IFF

Error handling drone address (ON ERROR) i19D0 1B74 1FB4

78F0-
78F1 Error - F!ag (error=255, RESUl'IE=fl)
;1986 19D6 1B6F lFAF

7BF2 The address of the decimal point **ia** Print buffer
10CE 1197 129 2343 2346 2537 2703 i27C5

78F3-78F4 Line 111111 **with the** last interruption occurring <END, STOP, BREAK>
i19C9 lDCB lDEF

7BF5-7BF6 Address **of the** line where the *last* break took place i19CD 1B77 1EN4

Pr to End Address

7BF7-
7BFB Begin of variable table
i1AC2 1ACD 1BSA 1B83 1E90 1F53 2664 ;2903 2BE5 2BF1 3D8 36BB

7BF9-7BFA End address *of* variable table
Start of the l''!atrix table
i1B86 266E 26BB 2907

78FB-7BFC Initial address **of** free memory (behind *the* ftatrix table)
4194 1B89 240 **26** 2711 2779 2704 12922

7BFD-78FE

7BFF-7900 Pointer to DAT line ;1D96 21F0

Type Code Table

7901 A **1B** 1E2A
7902 26.ltA
7903 **B**
7904 C
7905 D
790 E
7907 F
7908 G
7909 H
7904 I
790 J
790C K
790D L
790E M
790f N
7910 O
7911 p
7912 Q
7913 **R**
7914 S
7915 T
7916 U
7917 V
7918 W
7919 X
791A V
791B Or

TRACE FLA (0 = TRON, AF = TROFF) i lD-4-4 1Df9

X-Register

791C zus. Move Right Byte **10B98** CB6 CEO 8CF7 **0OE 8D54**

INT STRING SINGLE DOUBLE

791D LSB LSB LSB LSB 40A408 04E6 0BA0 08D20 003 0048 EIE i12E2 23B7 2-4-43
791E - 273 -
791F 104E9 243F
7920 10AC5

LSB 7921	ADR	LSB	LSI!	i01BF 073C 0866 0991! 0945 09B5 O9RF i09CB 0A03 0A80 0A9A 0ACC 01!46 0C5B ;0F37 0F4E 1343 1352 1426 1F3E 20C9 ;2395 23FD 2433 246D 248 2545 2562 i25D2 26DB 26E4 288C 2991 299E 29DA i2CC6 2E2A 2E37
7922 11SB	ADR	LSI!	LSB	
7923		11SB	11SB	i095A 0982 09AA 09BA 09DF 0A1A 0A62 ;0BBA 0CDA **0090** 0DCC 0E14 1422 2487 308719 077 0 78E 0797 0810 08FD 0919 1094 0955 0969 0648F 0BA0 8B59 0C80 ;0005 0E30 12 08 1445 15C6 2605
7924		EXP	EXP	

7925 Cache for arithmetic operations.
e.g. ;07C3 0D15 1535

Y - Register
(splitting as X-Register)

7926-792E 09F4 9FC 0A49 0D33 0045 0DF 0DFC **508E04** 1213

792F unused 10CA

7930-7949 pressure buffer ;0FF5 1037 10%

7944-7951

Additional multiplication and division register with double accuracy
i0DF9 0E07 0E26

RAl'I vectors for floppy commands
preloaded with 'JP 012DH' (DISK-CONNAND - Error)

7952	CVI Statement	30093 162
7955	FN - Statement	12524
7958	CVS Statement	112B8
795B	DEF Statement	;1882
795E	CVD Statement	;t62A
7961	EOF Statement	5162c

7964	LOC Statement	i162E
7967	LOF Statement	;1631'1
796	MKI$ Statement	i1632
796D	MS$ Statement	i1634
7970	MKD$ Statement	3163
7973	CIID Statement	;182C
7976	TIME$ Statement	12510
7979	OPEN Statement	i1866
797C	FIELD Statement	i1868
797F	GET statement	i186A
7982	PUT Statement	i1BbC
7985	CL OSE-Anei sun9	i186E
7988	LOAD} Statement	; 1870
798B	MERGE Statement	;1872
798E	NAME Statement	i1874
7991	KILL Statement	;1876
7994	& - Statement	;24C8
7997	LSET Instruction	;1878
799A	RSET Statement	i187A
799D	INSTR Instruction	**i2506**
79A0	SAVE Statement	;187C
79A3	UNE instruction	;no reference

	pre-populated with 'BRT'	
79Ab	from ERROR routine	il9EC
79A9	from USR routine	i27FE
79AC	Start BASIC Loop	ilAlC
79AF-79B1	unused	
791\2	**from programme input**	ilAl
79B5	End of programme input	ilAEC 36EC
791\8	End of programme input	i1AF2 36F2
79B.B	from NEW and END	i 1.BBC 1DB0
7911E	PRINT End Query	i2174
79C1	data output	€1032
79C4	Import v Keyboard	10358
79C7	RUN Execution	; 1EA
79CA	Home PRINT Statement	;2F
79CD	PRINT Statement	€520
79D0	PRINT Statement	i2103
79D3	PRINT Statement	;2108,2141
79D6	INPUT Statement	12194
79D9	MID$ as Statement	i2AEC
79DC	INPUT Statement	52220
79DF	READ +INPUT+ LIST	;2278 2B44

79E2-79E4 unused

79E5 3A 00 I/0 buffer header ;0080
79E7 2C ;1A73 370A

Input/Output Buffer 10421 0461 0531 0542 0560 3FB
79E8-7A9C

BASIC stack during initialisation ;00AC
79F8

Programme/File Name - Cassette On/Out
7A9D-7AAD Cache
;3581 358E 3647

Column display on the screen ;2127 30D2
7AAE

Additional Buffer for Buffered Video Output

Number of characters i' Buffer
;053A 30B9 30C9 30E8 3102 349A 3B13
7AAF ;3BIC

338C1 3C 30FE 30
7AB0-7AB1 Buffer-Pointer
130EE 349D 326 363E 344

7AB2-7AD1 Buffer Area

7AD2 Byte buffer for graphic printing, SOUND and so on. Cassette I/0
0SCD 2BFD 2C12 3623 3670 3bC4 381D
3399A 3B7F
7AD3
7AD 3782 3788 39ED
7AD5
13482

74D counter f. o.a.Buffer + Length for cassettes 1/0
4357D 354D0 398F 3992 3A2D 3A36
3A59 ;3A8E 'JAC/6

Line statuses for screen lines (80=Single Line, 81=Double Line, 00=Subsequent Line)

7AD7	Row 1	**503F8** 3240 333 33C5 3406 3424
7ADB	Cell 2	
7AD9	Line J	
7ADA	Row 4	
7ADB	Row 5	
7ADC	Row 6	
7ADD	Row 7	
?ADE	Row 8	
7ADF	Row 9	
7AE	Row 10	
7AE1	Row 11	
7AE2	Row 12	
7AE3	Row 13	
7AE4	Row 14	
7AE5	Row 15	435D3 35D4
7AE6	Row 16	;33BE 339D JSDD
7AE7		
7AEB	Pr ogr to Begin	**{0048**

The entry-level models for students

(L.ALeR

HOME-COMPUTER

LASER 210, 8 KByte RAM,
erweiterbar um 16 oder 64 KByte,
8 Farben, Programmsprache BASIC.

LASER 310 mit gleicher Ausstattung wie Laser 210,
aber 18 KByte RAM und mit Schreibmaschinen-Tastatur.

Floppy Disk Controller für 2 Laufwerke
mit LASER-DOS, Speicherkapazität 80 KByte.

Generalimporteur: SANYO VIDEO Vertrieb GmbH & Co.
Lange Reihe 29, D-2000 Hamburg 1, Tel. 040/28010 45-49

In a short time the horn computers Sanyo Laser and VZ 200 have acquired a considerable following, especially young fans. Their play instinct is unbroken, as is their curiosity and their fun experimenting. One of them is that you can fully see what the device has to offer, that you know what it looks "in there". To penetrate and explore everything, these ROM listings help: a great "roadmap" in the last mysteries of the ROM memory, clearly structured and detailed comments.

DJ VOGEL-BUCHVERLAG
g"/ WÜRZBURG

ISBN 3-8023-0852-2