```
2C37  F5                    PUSH AF             ;Save flags
2C38  EF                    RST JT
2C39  B1                    DB 0B1H
                          ;CALL PGSLC1          ;Switch pages
2C3A  2A FB46               LD HL,(DATAAD)
2C3D  F1                    POP AF              ;Restore flags
2C3E  38 08                 JR C,FND1ST         ;Jump if first time
2C40  7E                    LD A,(HL)
2C41  FE 2C                 CP ','              ;A = ',' or 0FFH
2C43  28 13                 JR Z,MOREDATA
                            ;
2C45  23                  FNDNXT: INC HL        ;HL -> first byte of next line
2C46  CD 2D6B               CALL TOPCH
2C49  28 20                 JR Z,FNDEND         ;Jump if at end of program
2C4B  23                  FND1ST: INC HL
2C4C  23                    INC HL
2C4D  23                    INC HL
2C4E  23                  FNDLP:  INC HL        ;HL -> next char in line
2C4F  7E                    LD A,(HL)
2C50  FE FF                 CP 0FFH
2C52  28 F1                 JR Z,FNDNXT         ;Jump if end of line
2C54  FE 8F                 CP 143
2C56  20 F6                 JR NZ,FNDLP         ;Jump if not 'DATA'
                            ;
2C58                      MOREDATA:
2C58  23                    INC HL              ;HL -> next data item
2C59  7E                    LD A,(HL)
2C5A  FE FF                 CP 0FFH
2C5C  28 E7                 JR Z,FNDNXT
2C5E  CD 2945               CALL GGETST         ;On exit, HL -> ',' or 0FFH
2C61  22 FB46               LD (DATAAD),HL      ;Store new data address
2C64  3A FAD3               LD A,(CRNTPG)
2C67  32 FB48               LD (DATAPG),A       ;Store new data page (0-15)
2C6A  3C                    INC A               ;Resets zero flag
                          ;
2C6B  E1                  FNDEND: POP HL        ;H = old page
2C6C  F5                    PUSH AF             ;Save flags
2C6D  7C                    LD A,H
2C6E  EF                    RST JT
2C6F  B1                    DB 0B1H
                          ;CALL PGSLC1        ;Select old page
2C70  F1                    POP AF              ;Z = 1 if end of program
2C71  D1                    POP DE              ;Get text ptr
2C72  C9                    RET
                          ;
                          ;
                          ;
2C73  0F                    DB 15
                          ;
2C74  EF                  SREAD:  RST JT
2C75  BB                    DB 0BBH
                          ;CALL ENDLIN
2C76  C8                    RET Z
2C77  FE 2C                 CP ','
2C79  20 01                 JR NZ,READ2
2C7B  13                    INC DE
```

```
2C7C  CD 2C2C        READ2:  CALL FINDDATA
2C7F  28 07                  JR Z,READERR
2C81  CD 2927                CALL GETINP
2C84  28 EE                  JR Z,SREAD
2C86  EF                     RST ERRRST
2C87  3B                     DB 11+48
                     ;
2C88                 READERR:
2C88  EF                     RST ERRRST
2C89  3F                     DB 63
                     ;
                     ;
                     ;
2C8A  C9                     RET
2C8B  06                     DB 6
                     ;
2C8C                 SRESTORE:
2C8C  F7                     RST GETRST
2C8D                 RESTENT::
2C8D  3A FAD3                LD A,(CRNTPG)
2C90  F5                     PUSH AF
2C91  D5                     PUSH DE
2C92  C5                     PUSH BC
2C93  CD 06DE                CALL GETLN1
2C96  C1                     POP BC
2C97  28 06                  JR Z,REST1
2C99  78                     LD A,B
2C9A  B1                     OR C
2C9B  28 02                  JR Z,REST1
2C9D  EF                     RST ERRRST
2C9E  2A                     DB 42
2C9F  22 FB46        REST1:  LD (DATAAD),HL
2CA2  3A FAD3                LD A,(CRNTPG)
2CA5  CB FF                  SET 7,A
2CA7  32 FB48                LD (DATAPG),A
2CAA  D1                     POP DE
2CAB  F1                     POP AF
2CAC  EF                     RST JT
2CAD  B1                     DB 0B1H
                     ;CALL PGSLC1
2CAE  C9                     RET
                     ;
                     ;
2CAF  D7             SRUN::  RST SCRRST
2CB0  68 4D                  DB 68H,4DH
2CB2  CD 0744                CALL STACKS
2CB5  CD 27FB                CALL KILLVAR
2CB8  01 0000                LD BC,0
2CBB  CD 06DE                CALL GETLN1
2CBE  EB                     EX DE,HL
2CBF  E1             RUN2:   POP HL
2CC0  E1             RUN33:  POP HL
2CC1  AF             RUN3:   XOR A
2CC2  32 FD69                LD (CONTFLG),A
2CC5  EB                     EX DE,HL
```

Restore DATA pointer to line BC

```
2CC6  CD 2D6B           CALL TOPCH
2CC9  EB                EX DE,HL
2CCA  CA 0250           JP Z,BASIC2
2CCD  13                INC DE
2CCE  13                INC DE
2CCF  13                INC DE
2CD0  13                INC DE
2CD1  CD 28D2           CALL KBCHK
2CD4  ED 53 FD6A        LD (CONTAD),DE
2CD8  CD 2CDD           CALL RUN6
2CDB  18 E4             RUN55:: JR RUN3          ;******** Return address 2 *********
2CDD  CD 09F2           RUN6:   CALL BREAKMON
2CE0  3E 01                     LD A,1
2CE2  C2 2D56                   JP NZ,STOPPP
2CE5  1A                        LD A,(DE)
2CE6  21 26F7                   LD HL,RWTAB
2CE9  CD 0689                   CALL FINDJP
2CEC  13                RUN8::  INC DE
2CED  CD 2CFD                   CALL RUN4
2CF0  1A                RUN5::  LD A,(DE)        ;******** Return address 1 *********
2CF1  13                        INC DE
2CF2  FE 3A                     CP ':'
2CF4  28 E7                     JR Z,RUN6
2CF6  FE FF             RUN9:   CP 0FFH
2CF8  C8                        RET Z
2CF9  1A                        LD A,(DE)
2CFA  13                        INC DE
2CFB  18 F9                     JR RUN9
2CFD  E9                RUN4::  JP (HL)
2CFE  1A                ENDLIN::        LD A,(DE)
2CFF  FE FF             ENDLI1: CP 0FFH
2D01  C8                        RET Z
2D02  FE C3                     CP ELSE
2D04  C8                        RET Z
2D05  FE 3A                     CP ':'
2D07  37                        SCF
2D08  C8                        RET Z
2D09  3F                        CCF
2D0A  C9                        RET
2D0B  05                        DB 5
                        ;
2D0C                    SSOUND::
                        ;
                    C   INCLUDE SNDPORTS.INC
0002                C   VDPRG0  EQU     2;+192
0002                C   VDPRG1  EQU     2;+192
0006                C   SNDPTQ  EQU     6;+192
0003                C   SNDPTI  EQU     3;+192
0005                C   KEYPORT EQU     5;+192
                        ;
0028                    ERRORR  EQU     28H
0030                    GETRST  EQU     30H
                        ;
2D0C  3E FF             LOADSN::LD A,0FFH
2D0E  21 FE02                   LD      HL,TEMP-2
2D11  F5                        PUSH AF
```

```
2D12  F1              LOOP:  POP AF
2D13  71                     LD    (HL),C
2D14  23                     INC   HL
2D15  70                     LD    (HL),B
2D16  23                     INC   HL
2D17  3C                     INC   A
2D18  FE 08                  CP    8
2D1A  28 4D                  JR    Z,SNDEROR
2D1C  F5                     PUSH AF
2D1D  F7                     RST 6ETRST
2D1E  28 F2                  JR    Z,LOOP
                      ;
2D20  F1                     POP AF
2D21  D5                     PUSH DE
2D22  FE 03                  CP    3       ;If less than 4 direct output
2D24  20 05                  JR    NZ,MSSS
2D26  CD 08EB                CALL DIROUTP
2D29  D1                     POP DE
2D2A  C9                     RET   ;.
2D2B  FE 07           MSSS:  CP 7
2D2D  20 3A                  JR NZ,SNDEROR
2D2F  3A FE04                LD    A,(TEMP)
2D32  CD 0891                CALL  LOADER  ;Call set pointers rouine
                             ;On stack is ctrl buf addr, and in HL is snd buff addr.
2D35  E5                     PUSH  HL
2D36  D1                     POP   DE
2D37  CD 089E                CALL  LOAD2
2D3A  D5                     PUSH  DE      ;Save Address of Ctrl reg.
2D3B  EB                     EX    DE,HL   ;Point DE to sound buffer
2D3C  21 FE06                LD    HL,TEMP+2       ;Point HL to temp
                             ;Reg C allready contains number
2D3F  01 000C                LD BC,12 ;**************was 10
2D42  ED B0                  LDIR          ;Transfer mem
2D44  E1                     POP   HL      ;Get ctrl reg addr
2D45  E5                     PUSH  HL      ;Save it
2D46  34                     INC   (HL)    ;Increment OUT pointer
2D47  46                     LD    B,(HL)  ;Load into reg 'B'
2D48  3E 03                  LD    A,3     ;
2D4A  85                     ADD   A,L     ;Increment
2D4B  6F                     LD    L,A     ;
2D4C  78                     LD    A,B     ;
2D4D  BE                     CP    (HL)    ;Compare OUTpoint to end of buffer
2D4E  E1                     POP   HL
2D4F  20 02                  JR    NZ,LOADED       ;Return if not =
2D51  AF                     XOR   A       ;Put buffer out pointer to zero
2D52  77                     LD    (HL),A  ;Store start of buffer value to OUT pointer
2D53  D1              LOADED: POP DE
2D54  C9                     RET
                      ;
                      ;
                      ;
2D55  AF              SSTOP:  XOR A
2D56  F5              STOPPP::       PUSH AF
2D57  ED 53 FD6A             LD (CONTAD),DE
2D5B  3A FAD3                LD A,(CRNTPG)
2D5E  32 FD6C                LD (CONTAD+2),A
```

```
2D61  F1                    POP AF
2D62  3C              STOP::  INC A
2D63  32 FD69               LD (CONTFLG),A
2D66  C3 0259               JP BASIC
                      ;
2D69  EF              SNDEROR:      RST ERRORR
2D6A  31                    DB 49
                      ;
                      ;
2D6B  CD 068B         TOPCH:  CALL TPCH1
2D6E  C0                    RET NZ
2D6F  3A FAA9               LD A,(BASTPG)
2D72  EF                    RST JT
2D73  AF                    DB 0AFH
                     ;CALL PGCHK
2D74  C8                    RET Z
2D75  3C                    INC A
2D76  EF                    RST JT
2D77  F1 B0                 DB 0F1H,0B0H
                     ;CALL PGSLC1
                     ;CALL LDBSBT
2D79  AF                    XOR A
2D7A  3C                    INC A
2D7B  C9                    RET
                      ;
                      ;
                      PUBLIC  NODDY
0010                  SCRRST  EQU     10H
0028                  JT      EQU     28H
0008                  DEHL    EQU     8
0009                  TAB     EQU     9
0020                  CNT     EQU     20H
0008                  CLS     EQU     8
0040                  SCR     EQU     40H
0080                  EMB     EQU     80H
0007                  MESS    EQU     7
0005                  FULL    EQU     5
                      ;
2D7C  2A FA81         NODDY:  LD HL,(CALCST)
2D7F  22 FA83               LD (KBDBUF),HL
2D82  E5                    PUSH HL
2D83  01 0400         NODD2:  LD BC,1024
2D86  CD 3299               CALL CHKSTK
2D89  D7                    RST SCRRST
2D8A  68 65 6F 87           DB 68H,SCR+CNT+FULL,SCR+CNT+MESS+CLS,EMB+7,30,'Noddy>'
2D8E  1E 4E 6F 64
2D92  64 79 3E
2D95  E1                    POP HL
2D96  E5                    PUSH HL  ;RESTORE KBDBUF
2D97  22 FA83               LD (KBDBUF),HL
2D9A  EF                    RST JT
2D9B  E9 AA                 DB 0E9H,0AAH
                     ;CALL EDITOR
                     ;      CALL EDEND
2D9D  CA 0244               JP Z,BASIC0
2DA0  FE 06          NODD33: CP 6
```

```
2DA2  38 DF                   JR C,NODD2
2DA4  28 DD                   JR Z,NODD2      ;RETURN IF NO TITLE
2DA6  01 0007                 LD BC,7
2DA9  ED 42                   SBC HL,BC
2DAB  ED 53 FA83              LD (KBDBUF),DE ;MOVE KBD TO END OF TITLE
2DAF  EB                      EX DE,HL
2DB0  2B                      DEC HL
2DB1  CB FE                   SET 7,(HL)
2DB3  ED 52                   SBC HL,DE
2DB5  3E 20                   LD A,32
2DB7  BE              NODD4:  CP (HL)
2DB8  20 03                   JR NZ,NODD5
2DBA  23                      INC HL
2DBB  18 FA                   JR NODD4
2DBD  E5              NODD5:  PUSH HL
2DBE  11 2E9A                 LD DE,DIRRRR
2DC1  CD 1BC4                 CALL COMPSTR
2DC4  E1                      POP HL
2DC5  CA 2E9D                 JP Z,DIRTRY
2DC8  E5                      PUSH HL
2DC9  D7                      RST 10H
2DCA  4D                      DB SCR+CLS+FULL ;CLEAR MAIN SCREEN
2DCB  D1                      POP DE
2DCC  D5                      PUSH DE
2DCD  CD 2E39                 CALL FINDNOD
2DD0  20 0B                   JR NZ,NODD55
2DD2  C5                      PUSH BC
2DD3  EB                      EX DE,HL
2DD4  CD 02E8                 CALL STROUT
2DD7  E1                      POP HL
2DD8  CD 06CF                 CALL DELET1
2DDB  18 04                   JR NODD6
2DDD  EB              NODD55: EX DE,HL
2DDE  CD 1907                 CALL OUTSTR
2DE1  D7              NODD6:  RST SCRRST
2DE2  1E                      DB 30
2DE3  EF                      RST JT
2DE4  E9 AA                   DB 0E9H,0AAH
                      ;CALL EDITOR
                      ;       CALL EDEND
2DE6  D1                      POP DE
2DE7  28 9A                   JR Z,NODD2
2DE9  D5                      PUSH DE
2DEA  2A FA83                 LD HL,(KBDBUF)
2DED  CD 0A15                 CALL SQUASH
2DF0  C1                      POP BC
2DF1  EB                      EX DE,HL
2DF2  50                      LD D,B
2DF3  59                      LD E,C
2DF4  ED 42                   SBC HL,BC
2DF6  23                      INC HL
2DF7  E5                      PUSH HL
2DF8  44                      LD B,H
2DF9  4D                      LD C,L
2DFA  03                      INC BC
2DFB  03                      INC BC
```

```
2DFC  C5              PUSH BC
2DFD  D5              PUSH DE
2DFE  CD 2E0E         CALL NODDSP     ;FIND BC SPACES
2E01  D1              POP DE
2E02  C1              POP BC  ;RETURNS WITH HL POINTING AT START
2E03  71              LD (HL),C
2E04  23              INC HL
2E05  70              LD (HL),B
2E06  23              INC HL
2E07  EB              EX DE,HL
2E08  C1              POP BC
2E09  ED B0           LDIR
2E0B  C3 2DB3         JP NODD2
                      ;
2E0E  3A FAA6         NODDSP: LD A,(NBTPG)
2E11  32 FAD3                 LD (CRNTPG),A
2E14  CD 0344                 CALL TESTSP
2E17  EB                      EX DE,HL
2E18  30 11                   JR NC,NDSP2
2E1A  3A FA7A                 LD A,(LSTPG)
2E1D  EF                      RST JT
2E1E  AF                      DB 0AFH
                      ;CALL PGCHK
2E1F  20 02                   JR NZ,NDSP1
2E21  EF                      RST 28H
2E22  23                      DB 23H                ;"NO SPACE"
2E23  CD 06F3         NDSP1:CALL INCPG
2E26  32 FAA6                 LD (NBTPG),A
2E29  EF                      RST JT
2E2A  B0                      DB 0B0H
                      ;CALL LDBSBT
2E2B  E5              NDSP2:PUSH HL
2E2C  09                      ADD HL,BC
2E2D  22 FAA4                 LD (NBTOP),HL
2E30  EB                      EX DE,HL
2E31  CD 06E7                 CALL DETPN
2E34  CD 0644                 CALL PGSLC2
2E37  E1                      POP HL
2E38  C9                      RET
2E39  D5              FINDNOD::       PUSH DE
2E3A  CD 2E67                 CALL NODBAS
2E3D  D1                      POP DE
2E3E  CD 2E53         FN1:    CALL TOPNOD
2E41  C0                      RET NZ
                      ;
2E42  E5                      PUSH HL
2E43  23                      INC HL
2E44  23                      INC HL
2E45  CD 1BC4                 CALL COMPSTR
2E48  C1                      POP BC
2E49  C8                      RET Z ;FRAME FOUND,Z,HL=START,BC=START OF BLOCK
2E4A  01 0000                 LD BC,0
2E4D  3E FF                   LD A,0FFH
2E4F  ED B1                   CPIR
2E51  18 EB                   JR FN1
                      ;
```

```
2E53  CD 06BB          TOPNOD: CALL TPCH1
2E56  28 02                    JR Z,FNDTOP ;FOUND TOP THEREFORE TEST FOR TOP OF NODDY
2E58  AF                       XOR A
2E59  C9                       RET
2E5A  CD 066E          FNDTOP: CALL PGCHK1
2E5D  28 05                    JR Z,FNDTPX
2E5F  CD 2E88                  CALL NXTND
2E62  AF                       XOR A ;TEST FOR NODDY TOP
2E63  C9                       RET
2E64  AF               FNDTPX: XOR A
2E65  3C                       INC A
2E66  C9                       RET
2E67  3A FAA9          NODBAS: LD A,(BASTPG)
2E6A  32 FAD3                  LD (CRNTPG),A
2E6D  EF                       RST JT
2E6E  AE                       DB 0AEH
                       ;CALL BSTPHL
2E6F  CF                       RST DEHL
2E70  2A FAA7                  LD HL,(BASTOP)
2E73  A7                       AND A
2E74  ED 52                    SBC HL,DE
2E76  20 05                    JR NZ,NDBS1
2E78  CD 066E                  CALL PGCHK1
2E7B  20 0B                    JR NZ,NXTND
2E7D  CD 0644          NDBS1:CALL PGSLC2
2E80  ED 53 FAD6               LD (PGTOP),DE
2E84  2A FAA7                  LD HL,(BASTOP)
2E87  C9                       RET
2E88  D5               NXTND:  PUSH DE
2E89  CD 06F3                  CALL INCPG
2E8C  CD 064A                  CALL PGSLCT
2E8F  EF                       RST JT
2E90  AE                       DB 0AEH
                       ;CALL BSTPHL
2E91  CF                       RST DEHL
2E92  ED 53 FAD6               LD (PGTOP),DE
2E96  EF                       RST JT
2E97  B0                       DB 0B0H
                       ;CALL LDBSBT
2E98  D1                       POP DE
2E99  C9                       RET
                       ;
2E9A  44 49 D2         DIRRRR: DC 'DIR'
2E9D  D7               DIRTRY: RST SCRRST
2E9E  4D                       DB SCR+CLS+FULL
2E9F  CD 2E67                  CALL NODBAS
                       ;****
2EA2  CD 2E53          DIR111: CALL TOPNOD
2EA5  C2 2D83                  JP NZ,NODD2
2EA8  23                       INC HL
2EA9  23                       INC HL
2EAA  CD 1907                  CALL OUTSTR
2EAD  D7                       RST SCRRST
2EAE  09                       DB TAB
2EAF  01 0000                  LD BC,0
2EB2  3E FF                    LD A,0FFH
```

```
2EB4  ED B1                   CPIR
2EB6  18 EA                   JR DIR1111
                        ;
                        ;
0028                    ERRRST  EQU   28H
0010                    SCRRST  EQU   10H
0028                    JT      EQU   28H
                        ;
                        ;
2EB8  AF                NODENT: XOR A
2EB9  32 FDDA                   LD (MEM1+1),A
2EBC  2A FA81                   LD HL,(CALCST)
2EBF  2B                        DEC HL
2EC0  2B                        DEC HL
2EC1  2B                        DEC HL
2EC2  CB FE                     SET 7,(HL)
2EC4  23                        INC HL
2EC5  CD 2F91                   CALL NEWSC1
2EC8  ED 53 FA81                LD (CALCST),DE
2ECC  D5                        PUSH DE
                        ;
2ECD  0E 2A             MAINLP: LD C,'*'
2ECF  CD 2FC5                   CALL GETSTAR
2ED2  EF                        RST JT
2ED3  A6                        DB 0A6H
                        ;CALL GETCHAR
2ED4  CD 2F1A                   CALL FINDINST
2ED7  18 F4                     JR MAINLP
                        ;
                        ;
                        ;
2ED9                    NODLIST:
2ED9  3E 01                     LD A,1
2EDB  32 FD75                   LD (PRORPL),A
2EDE  CD 2F73                   CALL NDISPLAY
2EE1  D7                        RST SCRRST
2EE2  2D 0A                     DB 13+20H,10          ;Print CR,LF at end of Noddy page
2EE4  AF                        XOR A
2EE5  32 FD75                   LD (PRORPL),A
2EE8  C9                        RET
                        ;
                        ;
                        ;
2EE9  D5                GGGOTO: PUSH DE
2EEA  EF                        RST JT
2EEB  E5 E6 A7                  DB 0E5H,0E6H,0A7H
                        ;       CALL GETSPACE
                        ;       CALL GETCHAR
                        ;       CALL PUTSTACK         ;HL -> ',' or '.'
2EEE  7E                        LD A,(HL)
2EEF  FE 2C                     CP ','
2EF1  28 04                     JR Z,GOLABEL          ;Jump if a comma
2EF3  EF                        RST JT
2EF4  A8                        DB 0A8H
                        ;       CALL NEWSCREEN
2EF5  D1                        POP DE
```

```
2EF6  C9                    RET
                          ;
2EF7                      GOLABEL:
2EF7  EF                    RST JT
2EF8  A6                    DB 0A6H
                          ;       CALL GETCHAR
2EF9  4E                    LD C,(HL)           ;C = label
2EFA  C5                    PUSH BC             ;Save C
2EFB  EF                    RST JT
2EFC  AB                    DB 0ABH
                          ;       CALL NEWSCREEN
2EFD  C1                    POP BC
2EFE  CD 2FB8               CALL BRLOOP         ;Find label
2F01  D1                    POP DE
2F02  C9                    RET
                          ;
                          ;
                          ;
2F03  D1                  EXXXX:  POP DE
2F04  E1                    POP HL
2F05  2B                    DEC HL
2F06  22 FA81               LD (CALCST),HL
2F09  C9                    RET
                          ;
                          ;
                          ;
2F0A  EF                  STKNOD: RST JT
2F0B  E6 A7                 DB 0E6H,0A7H
                          ;CALL GETCHAR
                          ;CALL PUTSTACK
2F0D  7E                    LD A,(HL)
2F0E  FE 2C                 CP ','
2F10  28 F8                 JR Z,STKNOD
2F12  C9                    RET
                          ;
                          ;
                          ;
2F13  E5                  PAUSE:  PUSH HL
2F14  01 03E8               LD BC,1000
2F17  C3 2BE6               JP PAUO
                          ;
                          ;
2F1A                      FINDINST:
2F1A  CD FAA1               CALL USERNOD
2F1D  CD 09F2               CALL BREAKMON
2F20  20 E1                 JR NZ,EXXXX         ;Quit if break key pressed
2F22  E5                    PUSH HL             ;HL = screen address
2F23  21 2F42               LD HL,VJ1
2F26  01 000B               LD BC,11
2F29  ED B9                 CPDR
2F2B  28 04                 JR Z,FFFND
2F2D  CD 3028               CALL NODERR
2F30  31                    DB 49               ;Command not found
                          ;
2F31  21 2F43             FFFND:  LD HL,VJUMP
2F34  09                    ADD HL,BC
```

```
2F35  09                    ADD HL,BC
2F36  E3                    EX (SP),HL
2F37  C9                    RET
                      ;
                      ;
2F38  4C 53 45 44     CPVAL:  DB 'LSEDARGIPB'
2F3C  41 52 47 49
2F40  50 42
2F42  4F              VJ1:    DB 'O'
2F43  18 94           VJUMP:  JR NODLIST      ;*List
2F45  18 C3                   JR STKNOD       ;*Stack
2F47  18 18                   JR INKEY        ;*Enter
2F49  18 28                   JR NDISPLAY     ;*Display
2F4B  18 40                   JR RETURN       ;*Advance
2F4D  18 B4                   JR EXXXX        ;*Return
2F4F  18 98                   JR GGGOTO       ;*Goto
2F51  18 4E                   JR CCCOMP       ;*If
2F53  18 BE                   JR PAUSE        ;*Pause
2F55  18 5A                   JR BRANCH       ;*Branch
                    ;
                    ;Simply fall through to *Off
                    ;
2F57  CD 300F       OFF:    CALL STACKEM
2F5A  E5                    PUSH HL
2F5B  EB                    EX DE,HL
2F5C  CD 301C               CALL FINDIT
2F5F  E1                    POP HL
2F60  C9                    RET
                    ;
                    ;
                    ;
2F61  D5            INKEY:  PUSH DE
2F62  E5                    PUSH HL
2F63  D7                    RST SCRRST
2F64  6F 1E                 DB 6FH,30
2F66  EF                    RST JT
2F67  E9 AA                 DB 0E9H,0AAH
                    ;CALL EDITOR
                    ;CALL EDEND
2F69  D7                    RST SCRRST
2F6A  3F 45                 DB 3FH,45H
2F6C  1B                    DEC DE
2F6D  EB                    EX DE,HL
2F6E  CB FE                 SET 7,(HL)
2F70  E1            CLRSTK: POP HL
2F71  D1                    POP DE
2F72  C9                    RET
                    ;
                    ;
                    ;
2F73                NDISPLAY:
2F73  EF                    RST JT
2F74  A5                    DB 0A5H
                    ;CALL GETSPACE
2F75  CD 3009       DISP1:  CALL GETSUB
2F78  D5                    PUSH DE
```

```
2F79  E5                  PUSH HL
2F7A  3A FAD3             LD A,(CRNTPG)
2F7D  F5                  PUSH AF
2F7E  CD 2E39             CALL FINDNOD
2F81  20 16               JR NZ,ERROR1
2F83  D7                  RST SCRRST
2F84  4D                  DB 40H+5+8 ;CLS VS 5
2F85  CD 02E7             CALL STRHL
2F88  F1                  POP AF
2F89  EF                  RST JT
2F8A  B1                  DB 0B1H
                        ;CALL PGSLC1
2F8B  18 E3               JR CLRSTK
                        ;
                        ;
                        ;
2F8D  CD 300F           RETURN: CALL STACKEM
2F90                    NEWSCREEN::
2F90  EB                  EX DE,HL
2F91  CD 301C           NEWSC1: CALL FINDIT
2F94  D5                  PUSH DE
2F95  CD 2E39             CALL FINDNOD
2F98  D1                  POP DE
2F99  C4 3028           ERROR1: CALL NZ,NODERR
2F9C  3F                  DB 63
2F9D  22 FDD9             LD (MEM1),HL
2FA0  C9                  RET
                        ;
                        ;
                        ;
2FA1  EF                CCCOMP: RST JT
2FA2  A5                  DB 0A5H
                        ;CALL GETSPACE
2FA3  CD 3009             CALL GETSUB
2FA6  D5                  PUSH DE
2FA7  E5                  PUSH HL
2FA8  2A FA83             LD HL,(KBDBUF)
2FAB  CD 1BC4             CALL COMPSTR
2FAE  E1                  POP HL
2FAF  D1                  POP DE
2FB0  C0                  RET NZ
                        ;
                        ;Fall through to branch
                        ;
                        ;
2FB1  EF                BRANCH: RST JT
2FB2  E5 A6                     DB 0E5H,0A6H        ;Call GETSPACE, call GETCHAR
2FB4  4E                        LD C,(HL)           ;C = label
2FB5  2A FDD9                   LD HL,(MEM1)        ;HL -> top of Noddy page
2FB8  C5                BRLOOP: PUSH BC             ;Save label
2FB9  0E 5E                     LD C,'^'            ;C = label identifier
2FBB  CD 2FC5                   CALL GETSTAR        ;Find '^'
2FBE  EF                        RST JT
2FBF  A6                        DB 0A6H
                        ;       CALL GETCHAR        ;A = character
```

```
2FC0  C1              POP BC              ;Get label
2FC1  B9              CP C
2FC2  20 F4           JR NZ,BRLOOP
2FC4  C9              RET                 ;Return with HL -> required label
                      ;
                      ;
                      ;
2FC5  CD 2FFE         GETSTAR:CALL TESTBIT7
2FC8  B9                      CP C
2FC9  C8                      RET Z
2FCA  23                      INC HL
2FCB  18 F8                   JR GETSTAR
                      ;
                      ;
                      ;
2FCD                  PUTSTACK::
2FCD  7E                      LD A,(HL)
2FCE  FE 2E                   CP '.'
2FD0  28 10                   JR Z,PUTST1
2FD2  FE 2C                   CP ','
2FD4  28 0C                   JR Z,PUTST1
2FD6  3C                      INC A
2FD7  28 2C                   JR Z,ERROR2
2FD9  EB                      EX DE,HL
2FDA  CD 3EC9                 CALL STKTEST
2FDD  EB                      EX DE,HL
2FDE  ED A0                   LDI
2FE0  18 EB                   JR PUTSTACK
                      ;
2FE2  1B              PUTST1: DEC DE
2FE3  1A                      LD A,(DE)
2FE4  F6 80                   OR 80H
2FE6  12                      LD (DE),A
2FE7  13                      INC DE
2FEB  C9                      RET
                      ;
                      ;
                      ;
2FE9                  GETSPACE::
2FE9  CD 2FFE                 CALL TESTBIT7
2FEC  C0                      RET NZ
2FED  FE 2C                   CP ','
2FEF  C8                      RET Z
2FF0  23                      INC HL
2FF1  18 F6                   JR GETSPACE
                      ;
                      ;
                      ;
2FF3  23              GETCHAR::       INC HL
2FF4  CD 2FFE                 CALL TESTBIT7
2FF7  20 FA                   JR NZ,GETCHAR
2FF9  FE 2C                   CP ','
2FFB  28 F6                   JR Z,GETCHAR
2FFD  C9                      RET
                      ;
                      ;
```

```
                        ;
2FFE                    TESTBIT7:
2FFE  7E                        LD A,(HL)
2FFF  CB 7F                     BIT 7,A
3001  C8                        RET Z
3002  FE FF                     CP 0FFH
3004  C0                        RET NZ
                        ;
3005  CD 3028             ERROR2: CALL NODERR
3008  35                        DB 53;MISSING SYMBOL
                      ;
                      ;
                      ;
3009  D5              GETSUB: PUSH DE
300A  EF                        RST JT
300B  E6 A7                     DB 0E6H,0A7H
                      ;CALL GETCHAR
                      ;CALL PUTSTACK
300D  D1                        POP DE
300E  C9                        RET
                      ;
                      ;
                      ;
300F                  STACKEM:
300F  E5                        PUSH HL
3010  A7                        AND A
3011  2A FA81                   LD HL,(CALCST)
3014  ED 52                     SBC HL,DE
3016  E1                        POP HL
3017  C0                        RET NZ
3018  CD 3028                   CALL NODERR
301B  29                        DB 41                   ;Stack empty
                      ;
                      ;
                      ;
301C  E5              FINDIT: PUSH HL
301D  2B                        DEC HL
301E  2B              FLOOP:  DEC HL
301F  7E                        LD A,(HL)
3020  E6 80                     AND 80H
3022  28 FA                     JR Z,FLOOP
3024  23                        INC HL
3025  EB                        EX DE,HL
3026  E1                        POP HL
3027  C9                        RET
                      ;
                      ;
                      ;
3028  2A FDD9        NODERR: LD HL,(MEM1)
302B  7C                        LD A,H
302C  A7                        AND A
302D  28 05                     JR Z,NODER2
302F  D7                        RST SCRRST
3030  0C                        DB 12
3031  CD 02E7                   CALL STRHL
3034  C3 0028        NODER2: JP ERRRST
```

```
                        ;
                        ;
                        ;
                        ;
0080                    REM     EQU     128
0096                    GOTO    EQU     150
0097                    GOSUB   EQU     151
00C7                    THEN    EQU     199
00C6                    STEP    EQU     198
00C3                     ELSE   EQU     195
0028                    JT      EQU     28H
002B                    ERROR   EQU     28H
0020                    SYNRST  EQU     20H
00FF                    ENDMARK EQU     255
0018                    OSPRST  EQU     18H
00DC                    NNOT    EQU     220
00DA                    NAND    EQU     218
00DB                    NOR     EQU     219
00C2                    WDTOP   EQU     194
001B                    OSPRST  EQU     18H
0080                    REM     EQU     128
00FF                    ENDMARK EQU     255
008F                    DATA    EQU     143
                        ;
                        ;
002B                    ERROR   EQU     28H     ;RESTART ADDRESS FOR ERROR
0023                    SPCERR  EQU     35      ;'OUT OF SPACE'
0024                    RNGERR  EQU     36 ;'OUT OF RANGE'
0026                    UNKERR  EQU     38      ;'UNKNOWN VARIABLE'
0027                    ARRERR  EQU     39 ;'ARRAY ALREADY EXISTS'
                        ;
0000                    ROMBITS EQU     00H     ;NOT CP/M, ROM PAGE 0
0030                    GETRST  EQU     30H
                        ;
0400                    K       EQU     1024
                        ;
                        ;
                        ;ROUTINE TO PUT VARIABLE NAME ON STACK
                        ;
3037   1A               PSHNAM: LD A,(DE)       ;GET FIRST CHAR OF NAME
3038   F6 E0                    OR 0E0H         ;SET HIGH 3 BITS OF CHAR
303A   4F                       LD C,A          ;SAVE CHAR
303B   06 01                    LD B,1          ;B = LENGTH OF NAME SO FAR
303D   2A FA81                  LD HL,(CALCST)
3040   18 03                    JR FIRST
3042   77               LP:     LD (HL),A
3043   23                       INC HL
3044   04                       INC B           ;B = LENGTH OF NAME SO FAR
3045   13               FIRST:  INC DE
3046   1A                       LD A,(DE)
3047   CD 3B88                  CALL ALPHA1
304A   38 F6                    JR C,LP         ;LOOP IF A CHARACTER
304C   CD 3B97                  CALL DIGIT1
304F   38 F1                    JR C,LP         ;LOOP IF A DIGIT
                        ;
                        ;HERE IF AT END OF NAME
```

```
                        ;
3051  FE 24             EV:     CP '$'
3053  20 04                     JR NZ,ND
3055  CB B1                     RES 6,C         ;RESET BIT TO INDICATE STRING
3057  13                        INC DE
3058  1A                        LD A,(DE)
                        ;
3059  FE 28             ND:     CP '('
305B  CB A9                     RES 5,C         ;ASSUME NOT AN ARRAY
305D  20 03                     JR NZ,NAR       ;JUMP IF NOT AN ARRAY
305F  CB E9                     SET 5,C         ;RESET BIT TO INDICATE AN ARRAY
3061  13                        INC DE          ;DE -> FIRST SUBSCRIPT
                        ;
3062  1A               NAR:    LD A,(DE)       ;GET CURRENT CHAR 'JUST IN CASE'
3063  C3 315E                   JP STKLN1       ;PUT FIRST CHAR OF NAME AND NAME LENGTH ON STACK
                        ;
                        ;
                        ;
                        ;GETVAL - CALLED WITH VARIABLE NAME AND ONE BYTE COUNT ON STACK
                        ;RETURNS VARIABLE VALUE ON STACK
                        ;
                        ;ROUTINE TO GET VALUE OF A VARIABLE (ALL TYPES)
                        ;CALLED WITH DE = PTR TO TEXT
                        ;NAME , LENGTH OF NAME ON STACK
                        ;
3066  D5               GETVAL: PUSH DE         ;SAVE TEXT POINTER
3067  CD 3152                 CALL NAMELEN    ;BC = LENGTH OF NAME
306A  2B                      DEC HL          ;HL -> TYPE BYTE
                        ;
306B  CB 76                   BIT 6,(HL)
306D  20 0D                   JR NZ,GETVA1    ;JUMP IF NUMERIC
306F  CB AE                   RES 5,(HL)      ;RESET VARIABLE BIT
3071  C5                      PUSH BC         ;SAVE LENGTH OF NAME
3072  E5                      PUSH HL
3073  CD 30D5                 CALL SEARCH     ;LOOK FOR STRING
3076  E1                      POP HL
3077  C1                      POP BC          ;GET LENGTH OF NAME
3078  38 02                   JR C,GETVA1     ;JUMP IF FOUND -
307A  CB EE                   SET 5,(HL)      ;OTHERWISE SET ARRAY BIT
                        ;
307C  7E               GETVA1: LD A,(HL)
307D  F5                      PUSH AF         ;SAVE TYPE BYTE
307E  CD 3171                 CALL FINDVAR    ;GET UVI IN BC (AND TAKE NAME OFF STACK)
                        ;
3081  CD 34A0                 CALL UVItoSRA ;GET ADDRESS OF VARIABLE INTO HL
3084  CD 34DF                 CALL GST5SRA ;GET VALUE ONTO STACK
                        ;
3087  F1                      POP AF          ;RESTORE TYPE BYTE
3088  D1                      POP DE          ;GET TEXT PTR
3089  CB 77                   BIT 6,A
308B  28 0C                   JR Z,ISCHR      ;JUMP IF CHAR ARRAY OR STRING
                        ;
308D  CB 6F            NONAU:  BIT 5,A
308F  C8                      RET Z           ;RETURN IF SIMPLE NUMERIC
                        ;
                        ;HERE IF NUMERIC ARRAY
```

```
                        ;
3090  CD 33A5                   CALL SUBSCR
3093  D5                        PUSH DE         ;SAVE NEW TEXT PTR
3094  CD 3505                   CALL GSTSVA
3097  D1                  NUMEND: POP DE          ;GET TEXT PTR
3098  C9                        RET
                        ;
                        ;HERE IF ARRAY OR STRING
                        ;
3099  CB 6F               ISCHR:  BIT 5,A
309B  20 1B                     JR NZ,ISARR     ;JUMP IF ARRAY
                        ;
                        ;HERE IF STRING
                        ;
309D  1B                        DEC DE          ;DE -> '(' OR '$'
309E  1A                        LD A,(DE)
309F  13                        INC DE
30A0  FE 28                     CP '('
30A2  28 11                     JR Z,SUBS       ;JUMP IF STRING HAS SUBSCRIPTS
                        ;
30A4  D5                        PUSH DE         ;SAVE TEXT PTR
30A5  CD 34AD                   CALL GETLEN     ;BC = LENGTH OF STRING
30A8  EB                        EX DE,HL
30A9  CD 3486                   CALL POPVA1     ;AHL = VA (STRING) [CALC STACK IS RESET]
30AC  11 0002                   LD DE,2
30AF  19                        ADD HL,DE
30B0  CE 00                     ADC A,0         ;AHL = VA (FIRST CHAR IN STRING)
30B2  57                        LD D,A          ;SAVE A
30B3  18 11                     JR PUTS1        ;PUT STRING ON STACK
                        ;
30B5  CD 34B7       SUBS:   CALL STKDIM1    ;PUT DIMENSION OF 1 ON THE STACK, AND FALL THROUGH TO:
                        ;
30B8  CD 33A5       ISARR:  CALL SUBSCR     ;AHL = VA (ELEMENT REQUIRED), BC = NO. OF ELEMENTS LEFT
30BB  D5                    PUSH DE         ;SAVE NEW TEXT PTR
30BC  57                    LD D,A          ;SAVE A
                        ;
                        ;HERE IF STRING OR CHAR ARRAY
                        ;
30BD  3A FDDD       PUTSTR: LD A,(MEM1+4)   ;A = NO. OF SUBSCRIPTS MISSING
30C0  A7                    AND A
30C1  20 03                 JR NZ,PUTS1
30C3  01 0001               LD BC,1         ;BC = 1 IF NO SUBSCRIPTS MISSING
30C6  C5            PUTS1:  PUSH BC
30C7  78                    LD A,B
30C8  B1                    OR C
30C9  7A                    LD A,D          ;RESTORE A
30CA  28 03                 JR Z,MOVED
30CC  CD 3508               CALL GSTnVA     ;TRANSFER STRING TO STACK
                        ;
30CF  C1            MOVED:  POP BC          ;BC = LENGTH OF STRING
30D0  CD 315B               CALL STKLEN     ;PUT LENGTH ON STACK
30D3  18 C2                 JR NUMEND       ;QUIT
                        ;
                        ;
                        ;
                        ;SEARCH - RETURNS WITH CARRY SET IF NAME IS IN TABLE , BC = UVI
```

```
                        ;
30D5  2A FA81           SEARCH: LD HL,(CALCST)
30D8  B7                        OR A
30D9  ED 42                     SBC HL,BC       ;HL = START OF NAME
30DB  EB                        EX DE,HL        ;DE = START OF NAME
30DC  2A FA7B                   LD HL,(VARNAM)  ;HL = NAME TABLE
30DF  01 0000                   LD BC,0 ;BC = UVI SO FAR
                        ;
30E2                    LOOK1STR:
30E2  03                         INC BC                 ;INCREMENT UVI
30E3  7E                        LD A,(HL)
30E4  D6 FF                     SUB 0FFH               ;END OF TABLE ?
30E6  D0                        RET NC
30E7  CD 1BC4                   CALL COMPSTR
30EA  20 F6                     JR NZ,LOOK1STR
30EC  37                        SCF                    ;INDICATE FOUND
30ED  C9                        RET
                        ;
                        ;
                        ;
                        ;CODE TO SET UP NEW VARIABLE
                        ;
30EE  C5                NEWVAR: PUSH BC         ;SAVE LENGTH OF NAME
30EF  21 0005                   LD HL,5
30F2  09                        ADD HL,BC ;HL = SPACE WE NEED
30F3  44                        LD B,H
30F4  4D                        LD C,L
                        ;
                        ;CREATE SPACE BY MOVING STACK AND SIMPLE VALUES
                        ;
30F5  C5                        PUSH BC         ;SAVE AMOUNT OF NEW SPACE NEEDED
30F6  CD 3299                   CALL CHKSTK     ;CHECK ENOUGH ROOM TO MOVE STACK
30F9  2A FAB1                   LD HL,(CALCST)
30FC  ED 5B FA7D                LD DE,(VALBOT)
3100  B7                        OR A
3101  ED 52                     SBC HL,DE
3103  44                        LD B,H
3104  4D                        LD C,L          ;BC = NO OF BYTES TO MOVE
3105  19                        ADD HL,DE ;HL = CURRENT CALCST
3106  D1                        POP DE  ;DE = SPACE NEEDED
3107  D5                        PUSH DE
3108  19                        ADD HL,DE
3109  EB                        EX DE,HL        ;DE = NEW CALCST
310A  2A FA81                   LD HL,(CALCST)
310D  ED 53 FA81                LD (CALCST),DE
3111  2B                        DEC HL  ;HL = FIRST BYTE TO MOVE
3112  1B                        DEC DE          ;DE = NEW TOP BYTE OF STACK
3113  ED B8                     LDDR            ;MOVE STACK AND VALUES
                        ;
3115  D1                        POP DE  ;GET NUMBER OF BYTES MOVED
3116  2A FA7F                   LD HL,(CALCBOT)
3119  19                        ADD HL,DE
311A  22 FA7F                   LD (CALCBOT),HL
311D  2A FA7D                   LD HL,(VALBOT)
3120  19                        ADD HL,DE
                        ;
```

```
                        ;ZERO NEW VALUE
                        ;
3121  06 05                      LD B,5
3123  2B                ZLP:    DEC HL
3124  36 00                     LD (HL),0
3126  10 FB                     DJNZ ZLP
3128  22 FA7D                   LD (VALBOT),HL
                        ;
                        ;INSERT NEW VARIABLE NAME
                        ;
312B  2B                        DEC HL  ; HL = LAST BYTE OF NAME TABLE
312C  36 FF                     LD (HL),0FFH ;INSERT STOPPER BYTE
312E  EB                        EX DE,HL
312F  C1                        POP BC          ;GET LENGTH OF VARIABLE NAME
3130  1B                        DEC DE
3131  2A FA81                   LD HL,(CALCST)
3134  2B                        DEC HL          ;HL = HIGHEST BYTE ACTUALLY ON STACK
3135  ED B8                     LDDR            ;TRANSFER NAME FROM STACK TO TABLE
3137  23                        INC HL
3138  18 30                     JR RCALCST      ;RESET CALC STACK AND RETURN
                        ;
                        ;
                        ;
                        ;FNDINIT - FIND VARIABLE WHOSE NAME IS ON STACK (LENGTH BC) OR INVENT NEW ONE
                        ;IN EITHER CASE , RETURNING UVI IN BC
                        ;
313A                    FNDINIT:
313A  C5                        PUSH BC         ;SAVE VARIABLE LENGTH
313B  CD 30D5                   CALL SEARCH
313E  30 09                     JR NC,NOTFND
                        ;
                        ;FOUND - REMOVE VARIABLE NAME FROM STACK
                        ;
3140  2A FA81                   LD HL,(CALCST)
3143  D1                        POP DE  ;GET VARIABLE LENGTH
3144  B7                        OR A
3145  ED 52                     SBC HL,DE
3147  18 21                     JR RCALCST      ;RESET CALC STACK AND RETURN
                        ;
3149  60                NOTFND: LD H,B          ;HL = UVI
314A  69                        LD L,C
314B  C1                        POP BC          ;GET LENGTH OF NAME
314C  E5                        PUSH HL         ;SAVE UVI
314D  CD 30EE                   CALL NEWVAR     ;SET UP NEW VARIABLE
3150  C1                        POP BC          ;RESTORE UVI
3151  C9                        RET
                        ;
                        ;
                        ;
                        ;NAMELEN TAKES LENGTH OF VARIABLE NAME OFF STACK AND RESETS IT
                        ;(PRESERVES A,DE)
                        ;
3152                    NAMELEN:
3152  2A FA81                   LD HL,(CALCST)
3155  2B                        DEC HL
3156  4E                        LD C,(HL)       ;BC = LENGTH OF NAME
```

```
3157  06 00                 LD B,0
3159  18 0F                 JR RCALCST      ;RESET CALC STACK AND RETURN
                      ;
                      ;
                      ;
                      ;STKLEN PUTS LENGTH OF STRING ON STACK
                      ;(PRESERVES A,BC,DE)
                      ;
315B                  STKLEN:
315B  2A FAB1               LD HL,(CALCST)
315E  C5              STKLN1: PUSH BC
315F  01 0002               LD BC,2
3162  CD 3299               CALL CHKSTK     ;ROOM FOR 2 BYTES ON STACK ? (PRESERVES HL)
3165  C1                   POP BC
3166  71                   LD (HL),C
3167  23                   INC HL          ;PUT LENGTH OF STRING ON STACK
3168  70                   LD (HL),B
3169  23                   INC HL
                      ;
316A                  RCALCST:
316A  22 FAB1               LD (CALCST),HL
316D  C9                   RET
                      ;
                      ;
                      ;
316E                  FINDLEN:
316E  CD 3152               CALL NAMELEN    ;BC = LENGTH OF NAME
                      ;
                      ;
                      ;FIND - CALLED WITH VARIABLE NAME ON STACK , BC = LENGTH
                      ;RETURNS UVI OF VARIABLE IN BC (OR ABORTS)
                      ;
3171  C5              FINDVAR:       PUSH BC ;SAVE VARIABLE LENGTH
3172  CD 30D5                 CALL SEARCH
3175  38 02                   JR C,FOUND
                      ;
3177  EF              FNDERR: RST ERROR
3178  26                      DB      UNKERR  ;'VARIABLE NOT FOUND'
                      ;
3179  2A FA81         FOUND:  LD HL,(CALCST)
317C  D1                      POP DE
317D  B7                      OR A
317E  ED 52                   SBC HL,DE
3180  18 E8                   JR RCALCST      ;RESET CALC STACK AND RETURN
                      ;
                      ;
                      ;
                      ;UPDATE - CALLED WITH VALUE , NAME AND NAME LENGTH ON STACK
                      ;UPDATES VARIABLE WITH VALUE
                      ;
3182  D5              UPDATE: PUSH DE                 ;SAVE TEXT PTR
3183  CD 3152                 CALL NAMELEN            ;BC = LENGTH OF NAME (FROM STACK)
3186  2B                      DEC HL
3187  7E                      LD A,(HL)               ;A = FIRST CHAR OF NAME
3188  CB 6F                   BIT 5,A
318A  28 07                   JR Z,NOTARR             ;JUMP IF NOT AN ARRAY
```

```
                        ;
                        ;HERE IF ARRAY
                        ;
318C  CB 77                     BIT 6,A
318E  C2 325C                   JP NZ,ARRVAR            ;JUMP IF NUMERIC ARRAY
3191  CB AE                     RES 5,(HL)              ;CHAR ARRAY MIGHT BE STRING
                        ;
                        ;HERE IF SIMPLE NUMERIC OR STRING
                        ;
3193  CB 77              NOTARR: BIT 6,A
3195  28 0D                     JR Z,STRVAR             ;JUMP IF STRING
                        ;
                        ;SIMPLE NUMERIC
                        ;
3197  CD 313A                   CALL FNDINIT
319A  C5                        PUSH BC                 ;SAVE UVI
319B  CD 34A0                   CALL UVItoSRA           ;HL -> DESCRIPTOR
319E  CD 34EE                   CALL PST5SRA            ;UPDATE VARIABLE
31A1  C1                        POP BC                  ;GET UVI
31A2  D1                        POP DE                  ;GET TEXT PTR AND QUIT
31A3  C9                        RET
                        ;
                        ;STRING VARIABLE
                        ;
31A4  E5                STRVAR: PUSH HL
31A5  C5                        PUSH BC                 ;SAVE LENGTH OF NAME
31A6  CB EE                     SET 5,(HL)
31A8  CD 30D5                   CALL SEARCH             ;LOOK FOR CHAR ARRAY WITH SAME NAME
31AB  C1                        POP BC                  ;GET LENGTH OF NAME
31AC  E1                        POP HL                  ;HL -> FIRST CHAR
31AD  7E                        LD A,(HL)
31AE  DA 325C                   JP C,ARRVAR             ;JUMP IF CHAR ARRAY NOT STRING
31B1  CB AE                     RES 5,(HL)              ;RESET VARIABLE BIT
                        ;
31B3  E5                        PUSH HL
31B4  C5                        PUSH BC
31B5  CD 30D5                   CALL SEARCH             ;CARRY FLAG SET IF STRING FOUND
31B8  C1                        POP BC
31B9  E1                        POP HL
31BA  F5                        PUSH AF                 ;SAVE FLAGS
31BB  CD 313A                   CALL FNDINIT
31BE  CD 34A0                   CALL UVItoSRA           ;HL -> STRING DESCRIPTOR
31C1  F1                        POP AF                  ;GET FLAGS
31C2  38 67                     JR C,OLDSTR             ;JUMP IF STRING EXISTS
                        ;
31C4  D1                NEWSTR: POP DE                  ;GET TEXT PTR
31C5  D5                        PUSH DE                 ;SAVE TEXT PTR
31C6  E5                        PUSH HL                 ;SAVE HL -> STRING DESC
31C7  01 0001                   LD BC,1                 ;ASSUME NO SUBSCRIPT
31CA  1B                        DEC DE                  ;DE -> '(' or '$'
31CB  1A                        LD A,(DE)               ;GET CHAR
31CC  FE 28                     CP '('
31CE  20 22                     JR NZ,NEWS0             ;JUMP IF NO SUBSCRIPTS
31D0  13                        INC DE                  ;DE -> FIRST SUBSCRIPT
                        ;
                        ;***************
```

```
31D1  F7                  RST GETRST          ;BC = SUBSCRIPT
31D2  20 07               JR NZ,SUBERR        ;JUMP IF SUBSCRIPT > 64K
                        ;***************
                        ;
31D4  1A                  LD A,(DE)           ;GET NEXT CHAR IN TEXT
31D5  FE 29               CP ')'
31D7  28 19               JR Z,NEWSO          ;JUMP IF NO MORE SUBSCRIPTS
                        ;
31D9  C5                  PUSH BC             ;SAVE FIRST SUBSCRIPT
                        ;
                        ;***************
31DA  F7                  RST GETRST
31DB  C2 331D           SUBERR: JP NZ,MULTERR ;JUMP IF SUBSCRIPT > 64K
                        ;***************
                        ;
31DE  1A                  LD A,(DE)
31DF  FE 29               CP ')'
31E1  20 94               JR NZ,FNDERR        ;ERROR IF MORE THAN ONE SUBSCRIPT
                        ;                     ("UNDEFINED VARIABLE")
31E3  60                  LD H,B
31E4  69                  LD L,C              ;HL = 2ND SUBSCRIPT
31E5  CD 34AD             CALL GETLEN         ;BC = LENGTH OF STRING ON STACK
31E8  A7                  AND A
31E9  ED 42               SBC HL,BC
31EB  38 1D               JR C,NEWERR         ;"NO SPACE" ERROR IF HL < BC
31ED  09                  ADD HL,BC           ;HL = 2ND SUBSCRIPT
31EE  2B                  DEC HL
31EF  C1                  POP BC              ;BC = FIRST SUBSCRIPT
31F0  18 06               JR NEWSX
                        ;
31F2  0B                NEWSO:  DEC BC        ;DECREMENT SUBSCRIPT
31F3  60                  LD H,B
31F4  69                  LD L,C              ;HL = SUBSCRIPT - 1
31F5  CD 34AD             CALL GETLEN         ;BC = LENGTH OF NEW STRING FROM STACK
                        ;                     (PRESERVES HL)
31F8  09                NEWSX:  ADD HL,BC     ;HL = MINIMUM SPACE NEEDED FOR STRING
31F9  38 0F               JR C,NEWERR         ;JUMP IF > 64K - 1
31FB  44                  LD B,H
31FC  4D                  LD C,L              ;BC = LENGTH OF STRING
                        ;
31FD  11 0040             LD DE,64
3200  62                  LD H,D
3201  6B                  LD L,E
3202  E5                NEWS1:  PUSH HL
3203  AF                  XOR A               ;CLEAR A AND CARRY
3204  ED 42               SBC HL,BC
3206  E1                  POP HL
3207  30 05               JR NC,NEWS2
3209  19                  ADD HL,DE
320A  38 71             NEWERR: JR C,CHARO    ;JUMP IF SPACE NEEDED > 64K - 1
320C  18 F4               JR NEWS1
320E  1E 02             NEWS2:  LD E,2
3210  19                  ADD HL,DE
3211  CE 00               ADC A,0
3213  44                  LD B,H
3214  4D                  LD C,L              ;BC = NEXT INTEGER MULTIPLE OF 64 + 2 BYTES
```

```
                        ;                               FOR MAX STRING LENGTH
                        ;DIMENSION CHAR ARRAY OF 1 DIMENSION, LENGTH BC AT TOP OF ARRAYS
                        ;
3215  E1                        POP HL                ;HL -> STRING DESCRIPTOR
3216  ED 5B FA7B                LD DE,(VARNAM)        ;BOTTOM OF VARIABLES
321A  A7                        AND A
321B  ED 52                     SBC HL,DE             ;HL = RELATIVE STRING DESCRIPTOR
321D  5F                        LD E,A                ;EBC = SPACE NEEDED FOR STRING
                        ;
                        ;********** TSSTSPC MOVES DESCRIPTOR IF NECESSARY **********
                        ;
321E  CD 32AD                   CALL TSSTSPC          ;TEST ROOM FOR STRING (PRESERVES BC,HL)
                        ;
3221  ED 5B FA7B                LD DE,(VARNAM)
3225  19                        ADD HL,DE             ;HL -> STRING DESCRIPTOR
                                                        (TESTSPC MAY HAVE MOVED VARIABLES, ETC.)
                        ;
                        ;
3226  E5                        PUSH HL               ;SAVE DESCRIPTOR
3227  CD 34C1                   CALL DIM1             ;DIMENSION STRING OF LENGTH BC
322A  E1                        POP HL                ;HL -> STRING DESCRIPTOR
                        ;
322B  D1                OLDSTR: POP DE                ;GET TEXT PTR
322C  E5                        PUSH HL               ;SAVE DESCRIPTOR
322D  D5                        PUSH DE               ;SAVE TEXT PTR
322E  CD 34DF                   CALL GST$GRA          ;PUT STRING DESCRIPTOR ON STACK
3231  CD 34B7                   CALL STKDIM1          ;CHANGE TOP OF STACK TO DIMENSION OF 1
3234  D1                        POP DE                ;GET TEXT PTR
3235  AF                        XOR A                 ;RESETS BIT 6, INDICATING STRING
                        ;                               (NEEDED BY SUBSCR ROUTINE)
3236  CD 326C                   CALL CHAR             ;UPDATE STRING
3239  E1                        POP HL                ;HL -> STRING DESCRIPTOR
323A  23                        INC HL
323B  23                        INC HL
323C  23                        INC HL                ;HL -> STRING LENGTH IN DESCRIPTOR
323D  D5                        PUSH DE               ;SAVE TEXT PTR
323E  50                        LD D,B
323F  59                        LD E,C                ;DE = NEW LENGTH
3240  3A FDDD                   LD A,(MEM1+4)         ;GET NO. OF SUBSCRIPTS MISSING
3243  FE 01                     CP 1
3245  28 10                     JR Z,OLDST1           ;JUMP IF 1 SUBSCRIPT MISSING
3247  E5                        PUSH HL
3248  CF                        RST DEHL
                        ;LD E,(HL)
                        ;       INC HL
3249  2A FDDB           ;       LD D,(HL)             ;DE = OLD STRING LENGTH
                                LD HL,(MEM1+2)        ;HL = LAST SUBSCRIPT - 1
324C  09                        ADD HL,BC             ;HL = 'NEW' LENGTH
324D  EB                        EX DE,HL              ;DE = 'NEW' LENGTH, HL = OLD LENGTH
324E  A7                        AND A
324F  ED 52                     SBC HL,DE             ;IS OLD LENGTH < NEW LENGTH ?
3251  E1                        POP HL
3252  38 03                     JR C,OLDST1           ;JUMP IF YES
3254  A7                        AND A
3255  28 03                     JR Z,OLDQT            ;JUMP IF NO. OF SUBSCRIPTS MISSING = 0
3257  73                OLDST1: LD (HL),E
3258  23                        INC HL
```

```
3259  72                    LD (HL),D           ;STORE NEW STRING LENGTH
325A  D1              OLDQT:  POP DE              ;GET TEXT PTR
325B  C9                      RET
                      ;
                      ;
325C  F5              ARRVAR: PUSH AF             ;SAVE TYPE BYTE
                      ;
325D  CD 3171                 CALL FINDVAR
3260  CD 34A0                 CALL UVItoSRA
3263  CD 34DF                 CALL GST5SRA        ;PUT ARRAY DESCRIPTOR ON STACK
                      ;
3266  F1                      POP AF              ;GET TYPE BYTE
3267  CB 77                   BIT 6,A
3269  D1                      POP DE
326A  20 21                   JR NZ,NUMARR        ;JUMP IF NUMERIC ARRAY
                      ;
                      ;CHAR ARRAY
                      ;
326C  CD 33A5         CHAR:   CALL SUBSCR         ;AHL -> ELEMENT REQUIRED
326F  D5                      PUSH DE             ;SAVE NEW TEXT PTR
3270  E5                      PUSH HL
                      ;
3271  60                      LD H,B
3272  69                      LD L,C              ;HL = NO. OF CHARS LEFT
3273  CD 34AD                 CALL GETLEN         ;BC = LENGTH OF STRING FROM STACK
3276  ED 53 FA81              LD (CALCST),DE      ;RESET STACK PTR
327A  A7                      AND A
327B  ED 42                   SBC HL,BC
327D  38 2C           CHAR0:  JR C,NOSPACE        ;ERROR IF NO. OF CHARS ON STACK TOO LARGE
                      ;
327F  E1                      POP HL              ;AHL = VA (ELEMENT REQUIRED)
3280  57                      LD D,A              ;SAVE A
3281  78                      LD A,B
3282  B1                      OR C
3283  28 06                   JR Z,CHAR1          ;JUMP IF NULL STRING
3285  7A                      LD A,D              ;RESTORE A
3286  C5                      PUSH BC
3287  CD 3537                 CALL PSTnVA         ;COPY BC BYTES FROM STACK TO VA (AHL)
328A  C1                      POP BC
328B  D1              CHAR1:  POP DE              ;GET TEXT PTR AND RETURN
328C  C9                      RET
                      ;
                      ;
328D  CD 33A5         NUMARR: CALL SUBSCR         ;AHL -> ELEMENT REQUIRED
3290  D5                      PUSH DE             ;SAVE TEXT PTR
3291  CD 3534                 CALL PST5VA         ;COPY 5 BYTES FROM STACK TO VA (AHL)
3294  D1                      POP DE
3295  C9                      RET
                      ;
                      ;
                      ;
3296  01 0030         CHKS30:LD BC,30H
                      ;
                      ;CHKSTK = ABORTS IF NOT ENOUGH SPACE TOP ADD BC BYTES TO STACK
                      ;PRESERVES A,BC,DE,HL
                      ;
```

```
3299                    CHKSTK:
3299  D5                        PUSH DE
329A  E5                        PUSH HL
329B  2A FA81                   LD HL,(CALCST)
329E  09                        ADD HL,BC       ;HL = NEW VALUE OF CALCST
329F  38 0A                     JR C,NOSPACE    ;ERROR IF HL > 64K
                        ;
32A1                    CHKST1:
32A1  ED 5B FA92                LD DE,(STKLIM)
32A5  37                        SCF
32A6  ED 52                     SBC HL,DE       ;NEW CALCST < STKLIM + 1 ?
32A8  E1                        POP HL
32A9  D1                        POP DE
32AA  D8                        RET C           ;IE NEW CALCST <= STKLIM ?
                        ;
32AB                    NOSPACE:
32AB  EF                        RST ERROR
32AC  23                        DB      SPCERR  ;'OUT OF SPACE'
                        ;
                        ;
                        ;
                        ;TEST FOR ENOUGH ROOM LEFT IN VIRTUAL MEMORY FOR ARRAY OF SIZE BC BYTES
                        ;
32AD                    TSTSPC:
32AD  F5                        PUSH AF
32AE  E5                        PUSH HL
32AF  C5                        PUSH BC         ;SAVE MAIN REGISTERS
32B0  D5                        PUSH DE
                        ;
32B1  2A FACC                   LD HL,(ARRTOP)
32B4  3A FACE                   LD A,(ARRTOP+2)
32B7  09                        ADD HL,BC
32B8  CE 00                     ADC A,0         ;AHL = 'NEW' TOP OF ARRAYS
32BA  F5                        PUSH AF
32BB  E5                        PUSH HL         ;SAVE NEW ARRTOP
                        ;
32BC  2A FA7B                   LD HL,(VARNAM)
32BF  3A FA7A                   LD A,(LSTPG)    ;AHL = LRA OF TOP OF VM + 1
32C2  CD 3593                   CALL LRAtoVA    ;AHL = VA OF TOP OF VM + 1
32C5  4F                        LD C,A
32C6  EB                        EX DE,HL        ;CDE = VA OF TOP OF VM + 1
32C7  E1                        POP HL
32C8  F1                        POP AF          ;AHL = VA OF NEW ARRTOP
32C9  F5                        PUSH AF
32CA  E5                        PUSH HL         ;SAVE AHL
                        ;
32CB  A7                        AND A
32CC  ED 52                     SBC HL,DE
32CE  99                        SBC A,C         ;AHL = NEW ARRTOP - (TOP OF VM + 1)
32CF  38 1D                     JR C,ROOM       ;JUMP IF ENOUGH ROOM
                        ;
32D1  A7                        AND A
32D2  20 D7             TEST0:  JR NZ,NOSPACE   ;IF A > 0 THEN ERROR
                        ;
32D4  7C                        LD A,H
32D5  B5                        OR L
```

Check at least
BC bytes between
(CALCST) and (STKLIM)

```
32D6  28 16                   JR Z,ROOM       ;JUMP IF AHL = 0
                              ;
                              ;NOT ENOUGH ROOM IN VM. NEED TO MOVE UP NAMES,VALUES,ETC IN TOP 16K.
                              ;CHECK FOR ENOUGH ROOM ON CALC. STACK
                              ;
32D8  CD 0C9D                 NOROOM: CALL DJVL4              ;DE should be old (CALCST)
32DB  2A FA81                         LD HL,(CALCST)
32DE  ED 4B FA7B                      LD BC,(VARNAM)
                              ;
                              ;NOW MOVE EVERYTHING FROM OLD (VARNAM) TO OLD (CALCST) UP BC BYTES
                              ;
32E2  A7                              AND A
32E3  E5                              PUSH HL
32E4  ED 42                           SBC HL,BC               ;HL = NO. OF BYTES TO MOVE
32E6  44                              LD B,H
32E7  4D                              LD C,L
32E8  E1                              POP HL                  ;OLD CALC STACK
32E9  EB                              EX DE,HL
32EA  1B                              DEC DE                  ;DE -> NEW TOP OF VALUES
32EB  2B                              DEC HL                  ;HL -> OLD TOP OF VALUES
32EC  ED B8                           LDDR                    ;MOVE NAMES,VALS,ETC, STARTING FROM TOP
                              ;
                              ;
32EE  E1                      ROOM:   POP HL
32EF  F1                              POP AF                  ;AHL = VA (NEW ARRTOP)
32F0  22 FACC                         LD (ARRTOP),HL
32F3  32 FACE                         LD (ARRTOP+2),A         ;STORE NEW ARRTOP
                              ;
32F6  D1                              POP DE
32F7  C1                              POP BC                  ;SIZE OF ARRAY
32F8  A7                              AND A
32F9  ED 42                           SBC HL,BC
32FB  DE 00                           SBC A,0                 ;AHL = OLD ARRTOP = VA (NEW STRING)
32FD  C5                              PUSH BC
32FE  D5                              PUSH DE
                              ;
32FF  CD 355D                         CALL ZEROnVA            ;FILL ARRAY WITH ZEROS
                              ;
3302  D1                              POP DE
3303  C1                              POP BC
3304  E1                              POP HL
3305  F1                              POP AF                  ;RESTORE MAIN REGISTERS AND RETURN
3306  C9                              RET
                              ;
                              ;
                              ;
                              ;
                              ;MULT16 - MULTIPLIES BC BY DE , LEAVING RESULT IN DEHL
                              ;
3307  21 0000                 MULT16: LD HL,0
330A  3E 10                           LD A,16
330C  29                      MULLP:  ADD HL,HL
330D  CB 13                           RL E
330F  CB 12                           RL D                    ;SHIFT RESULT , CARRY = ADD OR NOT
3311  30 04                           JR NC,NOADD
3313  09                             ADD HL,BC
```

```
3314  30 01                     JR NC,NOADD     ;CARRY OUT OF HL ?
3316  13                        INC DE
3317  3D                NOADD:  DEC A
3318  20 F2                     JR NZ,MULLP
331A  7A                        LD A,D
331B  B3                        OR E
331C  C8                        RET Z           ;RETURN IF DE = 0
                          ;
331D                      MULTERR:
331D  EF                        RST ERROR
331E  22                        DB 34           ;"OUT OF RANGE"
                          ;
                          ;
                          ;
                          ;CODE FOR DIM STATEMENT
                          ;
331F  CD 3152             DODIM:  CALL NAMELEN            ;BC = LENGTH OF VARIABLE NAME (TAKEN OFF STACK)
3322  2B                          DEC HL                  ;HL -> FIRST CHAR IN NAME
3323  CB 6E                       BIT 5,(HL)
3325  28 7C                       JR Z,BADSUB
3327  C5                          PUSH BC                 ;SAVE NAME LENGTH
3328  E5                          PUSH HL
3329  D5                          PUSH DE                 ;SAVE TEXT PTR
                          ;
332A  CD 30D5                     CALL SEARCH             ;LOOK FOR VARIABLE NAME IN NAMES TABLE
332D  30 02                       JR NC,NOTPRES           ;JUMP IF NOT FOUND
                          ;
332F  EF                  DIMERR: RST ERROR
3330  27                        DB 39                   ;'REDIMENSIONED ARRAY'
                          ;
3331                      NOTPRES:
3331  D1                        POP DE                  ;TEXT PTR
3332  E1                        POP HL                  ;HL -> FIRST CHAR NAME
                          ;
3333  CB 76                     BIT 6,(HL)              ;WHAT TYPE OF ARRAY ?
3335  20 15                     JR NZ,DONS5             ;JUMP IF NUMERIC
                          ;
3337  C1                        POP BC                  ;GET NAME LENGTH
3338  C5                        PUSH BC
3339  D5                        PUSH DE
333A  E5                        PUSH HL
                          ;
333B  CB AE                     RES 5,(HL)              ;LOOK FOR STRING WITH SAME NAME
333D  CD 30D5                   CALL SEARCH             ;ERROR IF FOUND -
3340  38 ED                     JR C,DIMERR             ;"RE-DIMENSIONED ARRAY"
                          ;
3342  E1                       POP HL                  ;HL -> FIRST CHAR
3343  D1                       POP DE                  ;GET TEXT PTR
3344  CB EE                    SET 5,(HL)              ;SET ARRAY BIT
3346  E5                       PUSH HL                 ;HL -> FIRST CHAR
3347  21 0001                  LD HL,1                 ;CHARACTER ARRAY - ELEMENT SIZE IS 1
334A  18 04                    JR DONSIZ
                          ;
334C  E5                  DONS5:  PUSH HL               ;HL -> FIRST CHAR
334D  21 0005                     LD HL,5               ;NUMERIC ELEMENT SIZE
3350  CD 343D             DONSIZ: CALL NUMELS           ;HL = ARRAY SIZE
```

```
3353  7C                    LD A,H
3354  B5                    OR L
3355  28 C6                 JR Z,MULTERR        ;JUMP IF ARRAY SIZE = 0 ("OUT OF RANGE")
                      ;
3357  EB                    EX DE,HL            ;DE = LOW (ARRAY SIZE), HL = TEXT PTR
3358  E3                    EX (SP),HL          ;(SP) = TEXT PTR, HL -> FIRST CHAR IN NAME
3359  ED 43 FDDC            LD (MEM1+3),BC      ;SAVE NO.OF DIMENSIONS
335D  AF                    XOR A
335E  B0                    OR B                ;ERROR IF NO. OF DIMS > 255
335F  C2 32AB               JP NZ,NOSPACE
3362  CB 76                 BIT 6,(HL)          ;TEST FIRST CHAR
3364  20 09                 JR NZ,DOZE          ;JUMP IF NUMERIC ARRAY
3366  79                    LD A,C
3367  3D                    DEC A
3368  20 05                 JR NZ,DOZE          ;JUMP IF MORE THAN ONE DIMENSION
336A  CB AE                 RES 5,(HL)          ;CHANGE FIRST CHAR TO INDICATE STRING
336C  32 FDDC               LD (MEM1+3),A       ;ZERO CURRENT LENGTH
                      ;
336F  EB              DOZE: EX DE,HL            ;HL = SIZE OF ARRAY
3370  CB 11                 RL C
3372  CB 10                 RL B                ;BC = 2 * DIM
3374  C5                    PUSH BC       ;SAVE 2 * DIM
3375  AF                    XOR A               ;CLEAR A
3376  09                    ADD HL,BC
3377  CE 00                 ADC A,0
3379  5F                    LD E,A
337A  44                    LD B,H
337B  4D                    LD C,L              ;EBC = NO. OF BYTES NEEDED FOR ARRAY
                      ;
337C  2A FACC               LD HL,(ARRTOP)
337F  3A FACE               LD A,(ARRTOP+2)
3382  22 FDD9               LD (MEM1),HL
3385  32 FDDB               LD (MEM1+2),A       ;SAVE ADDRESS OF NEW SPACE
                      ;
3388  CD 32AD               CALL TSSTSPC        ;MAKE SPACE FOR ARRAY
                      ;                         (PRESERVES A,BC,DE,HL)
                      ;
338B  C1                    POP BC              ;BC = 2 * DIM
338C  CD 3537               CALL PSTnVA         ;WRITE DIMENSIONS FROM STACK TO FREESPACE
                      ;
338F  D1                    POP DE        ;DE = TEXT PTR
3390  C1                    POP BC              ;GET LENGTH OF ARRAY NAME
3391  D5                    PUSH DE       ;SAVE TEXT PTR
                      ;
3392  CD 313A               CALL FNDINIT        ;SET UP VARIABLE NAME FOR THIS ARRAY
3395  CD 34A0                     CALL UVItoSRA ;GET ADDRESS OF ARRAY DESCRIPTOR
                      ;
3398  11 FDD9               LD DE,MEM1
339B  EB                    EX DE,HL
339C  01 0005               LD BC,5
339F  ED B0                 LDIR          ;UPDATE ARRAY DESCRIPTOR
                      ;
33A1  D1                    POP DE              ;GET TEXT POINTER
33A2  C9                    RET
                      ;
                      ;
```

```
                        ;
33A3  EF                BADSUB: RST ERROR
33A4  24                        DB 36           ;"SUBSCRIPT ERROR"
                        ;
                        ;
                        ;SUBSCR STARTS WITH
                        ;               VA , NO OF DIMS ON STACK , DE = TEXT PTR
                        ;
                        ;RETURNS WITH   AHL = VA OF ARRAY ELEMENT
                        ;               DE = NEW TEXT PTR
                        ;               BC = NO. OF ELEMENTS LEFT
                        ;
33A5  F5                SUBSCR: PUSH AF         ;SAVE ARRAY TYPE
33A6  2A FA81                   LD HL,(CALCST)
33A9  2B                        DEC HL          ;HIGH BYTE OF COUNT SHOULD BE ZERO
33AA  2B                        DEC HL
33AB  46                        LD B,(HL)       ;B = (LOW BYTE OF) NUMBER OF DIMENSIONS
33AC  22 FA81                   LD (CALCST),HL
33AF  21 0000                   LD HL,0         ;HL = OFFSET SO FAR
                        ;
33B2  C5                SUBLP:  PUSH BC         ;SAVE COUNT
33B3  E5                        PUSH HL         ;SAVE OFFSET
                        ;
33B4  CD 3464                   CALL GETDIM     ;HL = NEXT DIMENSION
33B7  EB                        EX DE,HL        ;DE = DIM, HL = TEXT PTR
33B8  C1                        POP BC          ;BC = OFFSET
33B9  E5                        PUSH HL         ;SAVE TEXT PTR
33BA  CD 3307                   CALL MULT16     ;HL = BC * DE = OFFSET * DIM
33BD  D1                        POP DE          ;GET TEXT PTR
33BE  C1                        POP BC          ;GET DIM COUNT
                        ;
                        ;(MEM1, MEM1+1 CONTAIN DIMENSION)
                        ;
33BF  1A                        LD A,(DE)       ;GET NEXT CHAR
33C0  FE 29                     CP ')'
33C2  28 4E                     JR Z,SUBCHK     ;JUMP IF NO MORE SUBSCRIPTS
33C4  FE D4                     CP 212
33C6  28 49                     JR Z,SUBCH0
33C8  FE 2C                     CP ','
33CA  28 45                     JR Z,SUBCH0
33CC  CD 2CFE                   CALL ENDLIN
33CF  28 40                     JR Z,SUBCH0
                        ;
33D1  C5                        PUSH BC         ;SAVE DIM COUNT
33D2  E5                        PUSH HL         ;SAVE OFFSET
33D3  2A FDD9                   LD HL,(MEM1)    ;HL = DIM
                        ;
33D6  E5                        PUSH HL         ;SAVE DIM
                        ;
                        ;************ GETRST USES MEM1 ***************
                        ;
                        ;************
33D7  F7                        RST GETRST      ;GET NEXT SUBSCRIPT, DE = NEW TEXT PTR , BC = SUBSCRIPT
33D8  C2 331D           BIGSUB: JP NZ,MULTERR   ;SUBSCRIPT > 64K ?
                        ;************
                        ;
```

```
33DB  E1                  POP HL          ;GET DIM
                          ;
33DC  0B                  DEC BC          ;DECREMENT SUBSCRIPT
33DD  A7                  AND A
33DE  ED 42               SBC HL,BC       ;IS DIMENSION < SUBSCRIPT ?
33E0  22 FDD9             LD (MEM1),HL    ;SAVE NO. OF ELEMENTS LEFT
33E3  38 BE               JR C,BADSUB
33E5  28 BC               JR Z,BADSUB     ;JUMP IF YES
33E7  ED 43 FDDB          LD (MEM1+2),BC  ;SAVE SUBSCRIPT - 1
33EB  E1                  POP HL          ;GET OFFSET
33EC  09                  ADD HL,BC       ;HL = (OFFSET * DIM) + (SUBSCRIPT - 1) = NEW OFFSET
                          ;
33ED  C1                  POP BC          ;B = DIM COUNT
33EE  10 C2               DJNZ SUBLP      ;JUMP IF MORE DIMENSIONS
                          ;
33F0  1A                  LD A,(DE)
33F1  FE 29               CP ')'          ;ANY MORE SUBSCRIPTS ?
33F3  28 26               JR Z,SUBEND     ;JUMP IF NO
                          ;
33F5  E5                  PLUS1: PUSH HL  ;SAVE OFFSET
33F6  2A FDD9             LD HL,(MEM1)
33F9  E5                  PUSH HL         ;SAVE NO. OF ELEMENTS LEFT
                          ;
                          ;***********
33FA  F7                  RST GETRST      ;BC = SUBSCRIPT = NO. OF CHARS, DE = NEW TEXT PTR
33FB  20 D8               JR NZ,BIGSUB    ;JUMP IF SUBSCRIPT TOO BIG
                          ;***********
                          ;
33FD  E1                  POP HL          ;HL = NO. OF ELEMENTS LEFT
33FE  1A                  LD A,(DE)       ;GET NEXT CHAR
33FF  FE 29               CP ')'
3401  20 A0               JR NZ,BADSUB    ;ERROR IF MORE SUBSCRIPTS
3403  A7                  AND A
3404  ED 42               SBC HL,BC
3406  38 9B               JR C,BADSUB     ;JUMP IF NO. OF ELEMENTS LEFT < NO. OF CHARS
3408  ED 43 FDD9          LD (MEM1),BC    ;STORE NO. OF CHARS
340C  E1                  POP HL          ;GET OFFSET
340D  06 FF               LD B,0FFH       ;INDICATES 1 TOO MANY SUBSCRIPT
340F  18 04               JR SUBCH1
                          ;
3411  1B                  SUBCH0: DEC DE
3412  10 8F               SUBCHK: DJNZ BADSUB  ;ERROR IF MORE THAN 1 SUBSCRIPT MISSING
3414  04                  INC B           ;B = 1
3415  F1                  SUBCH1: POP AF  ;GET TYPE BYTE
3416  F5                  PUSH AF         ;SAVE TYPE BYTE
3417  CB 77               BIT 6,A
3419  20 88               JR NZ,BADSUB    ;JUMP IF NUMERIC ARRAY (MUST HAVE RIGHT NO. OF SUBS)
                          ;
341B  F1                  SUBEND: POP AF  ;GET TYPE BYTE
341C  13                  INC DE          ;DE = NEW TEXT PTR
341D  D5                  PUSH DE         ;SAVE TEXT PTR
341E  57                  LD D,A          ;D = TYPE BYTE
341F  78                  SUBEN1: LD A,B
3420  32 FDDD             LD (MEM1+4),A   ;STORE NO. OF SUBSCRIPTS MISSING
3423  44                  LD B,H
3424  4D                  LD C,L          ;BC = OFFSET
```

```
                        ;
3425  CB 72             BIT 6,D
3427  28 0C             JR Z,SUBSTR     ;JUMP IF STRING ARRAY
                        ;
3429  CD 3483           CALL POPVA      ;AHL = VA OF BASE OF ARRAY
                        ;               (PRESERVES BC,DE)
342C  CD 345B           CALL NUMIND
                        ;
                        ;HERE AHL = BASE + 5 * OFFSET
                        ;
342F  D1                SUBRET: POP DE          ;DE = TEXT PTR
3430  ED 4B FDD9             LD BC,(MEM1)    ;BC = NO. OF ELEMENTS LEFT
3434  C9                      RET
                        ;
3435  CD 3483           SUBSTR: CALL POPVA      ;AHL = VA (BASE)
3438  09                      ADD HL,BC
3439  CE 00                   ADC A,0         ;AHL = BASE + OFFSET
343B  18 F2                     JR SUBRET
                        ;
                        ;
                        ;
                        ;NUMELS CALCULATES THE NUMBER OF ELEMENTS OF THE ARRAY
                        ;ENTER WITH DE = TEXT PTR
                        ;       HL = SIZE OF A SINGLE ELEMENT
                        ;EXIT WITH DE = NEW TEXT PTR
                        ;       AHL = ELEMENT SIZE * NUMBER OF ELEMENTS
                        ;               (can now be greater than 64K)
                        ;       BC = NUMBER OF DIMENSIONS
                        ;       AND EACH DIMENSION ON CALCULATOR STACK
                        ;
343D  01 0000           NUMELS: LD BC,0         ;BC=NUMBER OF DIMENSIONS
3440  03                DIMLP:  INC BC          ;INCREMENT DIMENSION COUNT
3441  C5                        PUSH BC         ;AND SAVE IT
3442  E5                        PUSH HL         ;SAVE SIZE
                        ;
                        ;**********
3443  F7                        RST GETRST      ;DE = NEW TEXT PTR , BC = DIMENSION
3444  20 92                     JR NZ,BIGSUB    ;JUMP IF SUBSCRIPT TOO BIG
                        ;**********
3446  CD 315B                   CALL STKLEN     ;PUT DIMENSION ON STACK
                        ;
3449  E1                        POP HL          ;HL = SIZE
344A  D5                        PUSH DE         ;SAVE TEXT PTR
344B  EB                        EX DE,HL        ;DE = SIZE
344C  CD 3307                   CALL MULT16     ;DEHL := DE * BC = SIZE * DIM
344F  7B                        LD A,E          ;AHL = ELEMENT SIZE * NUMBER OF ELEMENTS
3450  08                       EX AF,AF'       ;SAVE A
3451  D1                      POP DE          ;DE = TEXT PTR
3452  C1                      POP BC          ;BC = DIM COUNT
3453  1A                      LD A,(DE)
3454  FE 29                   CP ')'
3456  20 EB                   JR NZ,DIMLP
3458  13                      INC DE
3459  08                      EX AF,AF'       ;RESTORE A
345A  C9                      RET
                        ;
```

```
                        ;
                        ;
                        ;NUMIND - CALCULATES ADDRESS FOR NUMERIC ARRAY
                        ;ENTERED WITH VA (BASE) IN AHL , OFFSET IN BC , RESULT (VA) IN AHL
                        ;
345B    16 05           NUMIND: LD D,S
345D    09              NUMI2:  ADD HL,BC
345E    CE 00                   ADC A,0
3460    15                      DEC D
3461    20 FA                   JR NZ,NUMI2
3463    C9                      RET             ;AHL = BASE + 5 * OFFSET
                      ;
                      ;
                      ;
                      ;GETDIM - EXPECTS TO FIND A VIRTUAL ADDRESS ON STACK
                      ;INCREMENTS THIS VA BY 2 , LEAVING DIMENSION (FROM VA) IN HL
                      ;(PRESERVES DE)
                      ;
3464    D5            GETDIM: PUSH DE
3465    CD 3483               CALL POPVA
3468    F5                    PUSH AF         ;SAVE VA
3469    E5                    PUSH HL
346A    11 FDD9               LD DE,MEM1      ;2 BYTE WORK AREA
346D    01 0002               LD BC,2
3470    CD 351D               CALL GETnVA     ;GET DIMENSION INTO SAVE AREA
3473    E1                    POP HL
3474    F1                    POP AF
3475    01 0002               LD BC,2         ;CALCULATE NEW VA
3478    09                    ADD HL,BC
3479    CE 00                 ADC A,0
347B    CD 3493               CALL PSHVA
347E    2A FDD9               LD HL,(MEM1)    ;GET DIMENSION INTO HL
3481    D1                    POP DE
3482    C9                    RET
                      ;
                      ;
                      ;
                      ;POPVA PRESERVES DE
                      ;
3483    2A FA81       POPVA:  LD HL,(CALCST)
3486    D5            POPVA1: PUSH DE
3487    2B                    DEC HL
3488    7E                    LD A,(HL)
3489    2B                    DEC HL
348A    56                    LD D,(HL)
348B    2B                    DEC HL
348C    5E                    LD E,(HL)
348D    22 FA81               LD (CALCST),HL
3490    EB                    EX DE,HL
3491    D1                    POP DE
3492    C9                    RET
                      ;
                      ;
                      ;
3493    EB            PSHVA:  EX DE,HL
3494    2A FA81               LD HL,(CALCST)
```

```
3497  73                    LD (HL),E
3498  23                    INC HL
3499  72                    LD (HL),D
349A  23                    INC HL
349B  77                    LD (HL),A
349C  23                    INC HL
349D  C3 316A         RCALC1: JP RCALCST
                      ;
                      ;
                      ;
                      ;UVItoSRA - CALLED WITH UVI IN BC , RETURNS WITH SRA OF VALUE IN HL
                      ;
                      ;
34A0                  UVItoSRA:
34A0  60                      LD H,B
34A1  69                      LD L,C
34A2  29                      ADD HL,HL
34A3  29                      ADD HL,HL
34A4  09                      ADD HL,BC
34A5  EB                      EX DE,HL        ;DE = 5 * BC
34A6  2A FA7F                 LD HL,(CALCBOT)
34A9  B7                      OR A
34AA  ED 52                   SBC HL,DE       ;VALUE IS AT CALBOT - 5 * UVI
34AC  C9                      RET
                      ;
                      ;
                      ;
                      ;GETLEN RETURNS IN BC THE TOP TWO BYTES BENEATH THE CALC STACK PTR,
                      ;WHICH IS NOT CHANGED. (HL IS PRESERVED)
                      ;
34AD  EB              GETLEN: EX DE,HL
34AE  2A FA81                 LD HL,(CALCST)
34B1  2B                      DEC HL
34B2  46                      LD B,(HL)
34B3  2B                      DEC HL
34B4  4E                      LD C,(HL)               ;BC = LENGTH OF STRING ON STACK
34B5  EB                      EX DE,HL
34B6  C9                      RET
                      ;
                      ;
                      ;
                      ;STKDIM1 CHANGES THE TOP TWO BYTES BENEATH THE CALC STACK PTR TO 0001
                      ;
34B7                  STKDIM1:
34B7  2A FA81                 LD HL,(CALCST)
34BA  2B                      DEC HL
34BB  36 00                   LD (HL),0
34BD  2B                      DEC HL
34BE  36 01                   LD (HL),1
34C0  C9                      RET
                      ;
                      ;
                      ;
                      ;HL -> STRING DESCRIPTOR
                      ;
34C1  ED 5B FACC      DIM1:   LD DE,(ARRTOP)
```

```
34C5  3A FACE          LD A,(ARRTOP+2)
34C8  EB               EX DE,HL              ;AHL = NEW TOP OF ARRAYS
34C9  A7               AND A
34CA  ED 42            SBC HL,BC
34CC  DE 00            SBC A,0
34CE  EB               EX DE,HL              ;ADE = VA (NEW STRING)
34CF  73               LD (HL),E
34D0  23               INC HL
34D1  72               LD (HL),D
34D2  23               INC HL
34D3  77               LD (HL),A             ;STORE VA (NEW STRING) IN DESCRIPTOR
                       ;
34D4  0B               DEC BC
34D5  0B               DEC BC                ;BC = MAX LENGTH OF STRING
34D6  CD 315B          CALL STKLEN           ;PUT LENGTH ON STACK
                       ;
34D9  EB               EX DE,HL              ;AHL = VA (NEW STRING)
34DA  01 0002          LD BC,2
34DD  18 58            JR PSTnVA             ;MOVE LENGTH FROM STACK TO VA (NEW STRING)
                       ;
                       ;
                       ;GST5SRA - GETS 5 BYTES ONTO STACK FROM ADDRESS HELD IN HL
                       ;
34DF  01 0005          GST5SRA:      LD BC,5
34E2                   GSTnSRA:
34E2  CD 3299          CALL CHKSTK           ;CAN I PUT THESE BYTES ON STACK ?
34E5  ED 5B FA81       LD DE,(CALCST)
34E9  ED B0            LDIR                  ;SHIFT BYTES
34EB  EB               EX DE,HL
34EC  18 AF            JR RCALC1
                       ;
                       ;
                       ;
                       ;PST5SRA - PUTS 5 BYTES FROM STACK TO ADDRESS HELD IN HL
                       ;
34EE  01 0005          PST5SRA:LD BC,5
                       ;
34F1                   PSTnSRA:
34F1  EB               EX DE,HL       ;DE HOLDS 'TO' ADDRESS
34F2  2A FA81          LD HL,(CALCST)
34F5  B7               OR A
34F6  ED 42            SBC HL,BC
34F8  22 FAB1          LD (CALCST),HL ;CALCULATE NEW CALST
34FB  ED B0            LDIR           ;MOVE BYTES
34FD  C9               RET
                       ;
                       ;
                       ;
                       ;VIRTUAL MEMORY MANAGEMENT ROUTINES
                       ;
                       ;
                       ;
34FE  79               RAMPGC: LD A,C                ;A = NEW RAM PAGE
34FF  32 FAD2          SRAMPG: LD (PAGE),A
3502  D3 00                    OUT (PGPORT),A         ;SET RAM PAGE
```

```
3504  C9                    RET
                            ;
                            ;
                            ;
                            ;GSTnVA = GET BC BYTES FROM VA (IN AHL) TO STACK
                            ;GST5VA = GET 5 BYTES FROM VA TO STACK
3505  01 0005               GST5VA: LD BC,5
3508  C5                    GSTnVA: PUSH BC               ;SAVE COUNT
3509  CD 3299                       CALL CHKSTK           ;ABORT IF CAN'T ADD BC BYTES TO STACK
350C  ED 5B FA81                    LD DE,(CALCST)
3510  CD 351D                       CALL GETnVA           ;MOVE BYTES
3513  C1                            POP BC
3514  2A FA81                       LD HL,(CALCST)
3517  09                            ADD HL,BC             ;ADJUST STACK POINTER
3518  18 83                         JR RCALC1
                            ;
                            ;
                            ;
                            ;GETnVA = MOVE BC BYTES FROM VA (AHL) TO SRA (DE)
                            ;GET5VA = MOVE 5 BYTES FROM VA (AHL) TO SRA (DE)
                            ;
351A  01 0005               GET5VA: LD BC,5
351D  CD 3577               GETnVA: CALL SAVEPAGE
3520  04                            INC B                 ;NECESSARY
3521  F5                    GETLP:  PUSH AF
3522  CD 34FF                       CALL SRAMPG           ;SET CORRECT RAM PAGE
3525  7E                            LD A,(HL)
3526  12                            LD (DE),A             ;TRANSFER BYTE
3527  F1                            POP AF
3528  CD 35AC                       CALL INCLRA           ;INCREMENT PTRS
352B  13                            INC DE
352C  0D                            DEC C
352D  20 F2                         JR NZ,GETLP           ;DECREMENT AND TEST COUNT
352F  10 F0                         DJNZ GETLP
3531  08                    GETRET: EX AF,AF'             ;A = OLD (PAGE)
3532  18 CB                         JR SRAMPG             ;RESET (PAGE) AND RETURN
                            ;
                            ;
                            ;
                            ;PSTnVA = PUT BC BYTES FROM STACK TO VA (IN AHL)
                            ;PST5VA = PUT 5 BYTES FROM STACK TO VA
                            ;
3534  01 0005               PST5VA: LD BC,5
3537  EB                    PSTnVA: EX DE,HL
3538  2A FA81                       LD HL,(CALCST)
353B  B7                            OR A
353C  ED 42                         SBC HL,BC
353E  22 FA81                       LD (CALCST),HL
3541  EB                            EX DE,HL              ;DE = NEW TOP-OF-STACK
3542  18 03                         JR PUTnVA             ;MOVE BYTES
                            ;
                            ;
                            ;
                            ;PUTnVA = MOVE BC BYTES TO VA (AHL) FROM SRA (DE)
                            ;PUT5VA = MOVE 5 BYTES TO VA (AHL) FROM SRA (DE)
                            ;
```

```
3544  01 0005          PUT5VA: LD BC,5
3547  CD 3577          PUTnVA: CALL SAVEPAGE
354A  04                       INC B
354B  F5               PUTLP:  PUSH AF
354C  CD 34FF                  CALL SRAMPG          ;SET CORRECT RAM PAGE
354F  1A                       LD A,(DE)
3550  77                       LD (HL),A            ;TRANSFER BYTE
3551  F1                       POP AF
3552  CD 35AC                  CALL INCLRA          ;INCREMENT PTR5
3555  13                       INC DE
3556  0D                       DEC C
3557  20 F2                    JR NZ,PUTLP          ;DECREMENT AND TEST COUNT
3559  10 F0                    DJNZ PUTLP
355B  18 D4                    JR GETRET
                       ;
                       ;
                       ;
                       ;ZEROnVA - ZEROES EBC BYTES , STARTING AT THE VIRTUAL ADDRESS HELD IN AHL
                       ;
355D                   ZEROnVA:
355D  CD 3577                  CALL SAVEPAGE
3560  1C                       INC E                ;NECESSARY
3561  F5               ZEROLP: PUSH AF
3562  CD 34FF                  CALL SRAMPG          ;SET CORRECT RAM PAGE
3565  F1                       POP AF
3566  36 00                    LD (HL),0
3568  CD 35AC                  CALL INCLRA
356B  0B                       DEC BC
356C  57                       LD D,A               ;SAVE A
356D  78                       LD A,B
356E  B1                       OR C
356F  7A                       LD A,D               ;RESTORE A
3570  20 EF                    JR NZ,ZEROLP
3572  1D                       DEC E
3573  20 EC                    JR NZ,ZEROLP
3575  18 BA                    JR GETRET
                       ;
                       ;
                       ;
3577                   SAVEPAGE:
3577  08                       EX AF,AF'
3578  3A FAD2                  LD A,(PAGE)
357B  08                       EX AF,AF'            ;A' = CURRENT PAGE, THEN FALL THROUGH TO VAtoLRA
                       ;
                       ;
                       ;VAtoLRA = CONVERT VA IN AHL TO LRA IN AHL (PRESERVES BC,DE)
                       ;
357C                   VAtoLRA::
357C  DD 21 FA7A               LD IX,LSTPG
3580                   VALRA1::
3580  D5                       PUSH DE
3581  CB 14                    RL H                 ;SHIFT MSB OF H INTO A
3583  8F                       ADC A,A
3584  B7                       OR A                 ;CLEAR MSB OF H
3585  CB 1C                    RR H
3587  11 4000                  LD DE,16*K
```

```
358A  19                    ADD HL,DE       ;HL = OFFSET (FOR NON-FINAL PAGE)
358B  DD BE 00              CP (IX)         ;LAST PAGE ?
358E  20 01                 JR NZ,NOADJ
3590  19              ADX:  ADD HL,DE       ;LAST PAGE - ADD ANOTHER 16K OFFSET
3591  D1              NOADJ: POP DE
3592  C9                    RET
                    ;
                    ;
                    ;
                    ;LRAtoVA - CONVERTS A LRA IN AHL TO A VA IN AHL
                    ;
3593  E6 0F          LRAtoVA:      AND 0FH ;REMOVE ANY SPURIOUS PORT BITS
3595  DD 21 FA7A            LD IX,LSTPG
3599  DD BE 00              CP (IX)         ;LAST PAGE ?
359C  11 C000               LD DE,48*K      ;ADD 48K = SUB 16K
359F  20 03                 JR NZ,NTLAST
35A1  11 8000               LD DE,32*K      ;ADD 32K = SUB 32K
35A4  19              NTLAST: ADD HL,DE     ;HL = OFFSET WITHIN PAGE
35A5  CB 14                 RL H
35A7  CB 2F                 SRA A           ;SHIFT LSB OF A INTO MSB OF H
35A9  CB 1C                 RR H
35AB  C9                    RET
                    ;
                    ;
                    ;
                    ;INCLRA = INCREMENT LRA (IN AHL)
                    ;
35AC  23            INCLRA: INC HL          ;INCREMENT OFFSET
35AD  DD 21 FA7A    INCLR1: LD IX,LSTPG
35B1  DD BE 00              CP (IX)         ;LAST PAGE ?
35B4  C8                    RET Z           ;YES - DONE
35B5  D5                    PUSH DE
35B6  11 4000               LD DE,16*K
35B9  19                    ADD HL,DE       ;END OF PAGE REACHED ?
35BA  38 04                 JR C,INCL1
35BC  ED 52                 SBC HL,DE       ;RESTORE OFFSET
35BE  D1                    POP DE
35BF  C9                    RET
35C0  3C            INCL1:  INC A           ;INCREMENT PAGE COUNT
35C1  19                    ADD HL,DE       ;ADJUST OFFSET
35C2  DD BE 00              CP (IX)         ;JUST MOVED TO LAST PAGE ?
35C5  20 CA                 JR NZ,NOADJ
35C7  18 C7                 JR ADX
                    ;
                    ;
                    ;
                    ;BLOCK is called to save/load BASIC program block
                    ;Returns NZ if no more blocks, else Z.
                    ;
35C9  D9            BLOCK:: EXX
35CA  78                    LD A,B
35CB  3C                    INC A
35CC  28 05                 JR Z,QUITNZ             ;Return with NZ if B = FFH
35CE  78                    LD A,B
35CF  B2                    OR D
35D0  B3                    OR E
```

```
35D1  20 03                 JR NZ,BLOCK1        ;Return with NZ if B'D'E' = 0
35D3  3C              QUITNZ: INC A             ;Resets zero flag
35D4  D9                    EXX
35D5  C9                    RET
                      ;
35D6  E5              BLOCK1: PUSH HL           ;Save low (start address)
35D7  D5                    PUSH DE             ;Save low (bytes left in program)
                      ;
35D8  CD 34FE               CALL RAMPGC         ;Switch ram page
                      ;
35DB  7A                    LD A,D
35DC  D6 40                 SUB 40H
35DE  57                    LD D,A
35DF  78                    LD A,B
35E0  DE 00                 SBC A,0
35E2  47                    LD B,A              ;BDE = BDE - 16K = new bytes left
35E3  38 0F                 JR C,END1           ;Jump if bytes left < 16k (i.e. last block)
                      ;
35E5  F1                    POP AF              ;Clear stack
                      ;
35E6  7C                    LD A,H
35E7  C6 40                 ADD A,40H
35E9  67                    LD H,A
35EA  79                    LD A,C
35EB  CD 35AD               CALL INCLR1         ;Call INCLRA + 1
35EE  4F                    LD C,A              ;CHL := CHL + 16K = new start address
                      ;
35EF  E5                    PUSH HL
35F0  21 4000               LD HL,16*1024
35F3  E3                    EX (SP),HL          ;Put 16K on stack
                      ;
35F4  AF              END1:   XOR A             ;Set zero flag
35F5  D9                    EXX                 ;Swap registers
35F6  D1                    POP DE
35F7  E1                    POP HL              ;DE = size of block, HL = starting address
35F8  C9                    RET                 ;Z = 0 if end of program
                      ;
                      ;
                      ;
35F9                  ADJDES::
35F9  F5                    PUSH AF
35FA  C5                    PUSH BC
35FB  2A FB49               LD HL,(DESAVE)
35FE  08                    EX AF,AF'
35FF  7C                    LD A,H
3600  FE FB                 CP 0FBH
3602  30 11                 JR NC,DJVLB
3604  08                    EX AF,AF'
3605  A7                    AND A
3606  78                    LD A,B
3607  01 4000               LD BC,4000H         ;Assume 512 -> 500
360A  28 05                 JR Z,DJVLA          ;Jump if correct assumption
360C  A7                    AND A
360D  20 06                 JR NZ,DJVLB         ;Jump if A,B > 0
360F  06 C0                 LD B,0C0H
3611  09             DJVLA: ADD HL,BC
```

```
3612  22 FB49                 LD (DESAVE),HL
3615  C1              DJVLB:  POP BC
3616  F1                      POP AF
3617  C9                      RET
                      ;
                      ;
                      ;
                      ;The kbd routine returns after finding the first valid character
                      ;If control is pressed then upper and lower case is ignored.
                      ;If alpha lock is found then alpha flag is set.
                      ;If shift is found then shifted characters are generated
                      ;Numeric keypad is operated by shift lock
                      ;Subsequent auto repeats flag provides shorter period.
                      ;7 alpha lock on/off
                      ;2 shift lock
                      ;
3618  C5              KBDP0:  PUSH BC
3619  D5                      PUSH DE
361A  E5                      PUSH HL
361B  CD FD51           SSS:  CALL USERIO
361E  E1                      POP HL
361F  D1                      POP DE
3620  C1                      POP BC
3621  C9                      RET
3622                  KEY1::
3622  3E FB           START:  LD A,251
3624  32 FD7E                 LD (LASTDR),A
3627  D3 05                   OUT (DR),A
3629  DB 05                   IN A,(DR)
362B  CB 47                   BIT 0,A
362D  28 07                   JR Z,CNTRL
362F  CD 3643                 CALL KBNORMAL
3632  C4 37C3                 CALL NZ,ASCMAP
3635  C9                      RET
3636  CD 3643         CNTRL:  CALL    KBNORMAL
3639  C4 37C3                 CALL    NZ,ASCMAP
363C  C8                      RET Z
363D  E6 1F                   AND 1FH
363F  32 FD7D                 LD (LASTASC),A
3642  C9                      RET
3643  CD 00AD         KBNORMAL:       CALL SHIF
3646  28 32                   JR Z,NSHIFT ;NORMAL SHIFT
3648  CD 36AA                 CALL SCAN
364B  C8                      RET Z
364C  57                      LD D,A
364D  01 3721                 LD BC,BASE
3650  CD 3718                 CALL LOOK
3653  47                      LD B,A
3654  FE 7F                   CP 127
3656  28 03                   JR Z,KB8
3658  FE 20                   CP 32
365A  3F                      CCF
365B  3A FA91           KB8:  LD A,(KBFLAG)
365E  38 04                   JR C,KBN999
3660  CB 57                   BIT 2,A
3662  20 13                   JR NZ,NSH1
```

```
3664  CB 7F            KBN999: BIT 7,A
3666  78                       LD A,B
3667  28 1B                    JR Z,NCONT
3669  FE A0                    CP 160
366B  28 1C                    JR Z,NORM11
366D  FE 1D                    CP 29
366F  28 22                    JR Z,NORM2
3671  FE 61                    CP      61H
3673  D8                       RET     C
3674  FE 7B                    CP 7BH
3676  D0                       RET NC
3677  7A               NSH1:   LD      A,D
3678  18 04                    JR      NSH2
367A  CD 36AA          NSHIFT: CALL SCAN
367D  C8                       RET Z
367E  01 3772          NSH2:   LD BC,UPPER
3681  CD 3718                  CALL LOOK
3684  47               NCONT:  LD B,A
3685  FE A0            NORM1:  CP 160
3687  20 0A                    JR NZ,NORM2
3689  3A FA91          NORM11: LD A,(KBFLAG) ;GET FLAGS
368C  EE 80                    XOR 80H
368E  32 FA91                  LD (KBFLAG),A
3691  AF                       XOR A
3692  C9                       RET
3693  FE 1D            NORM2:  CP 29   ;28 =SCROLL/PAGE
3695  C0                       RET NZ ;NORMAL CHAR
3696  5F                       LD E,A ;save 28
3697  3A FA91                  LD A,(KBFLAG)
369A  CB 6F                    BIT 5,A
369C  28 01                    JR Z,NORM3
369E  1D                       DEC E
369F  EE 20            NORM3:  XOR 32
36A1  32 FA91                  LD (KBFLAG),A
36A4  7B               NORM4:  LD A,E
36A5  32 FD7D                  LD (LASTASC),A
36A8  A7                       AND A
36A9  C9                       RET
                       ;MATRIX SCAN
36AA  06 08            SCAN:   LD B,8 ;B=8=DRIVE COUNTER
36AC  AF                       xor a   ;a=0 reset carry
36AD  4F                       ld c,a ;SENSE COUNTER=0
36AE  3D                       dec a   ;a=ff
36AF  17               SCAN2:  RLA ;A=FE,FD,FB,F7,EF,DF,BF,7F
36B0  F5                       PUSH AF ;SAVE DRIVE
36B1  32 FD7E                  LD (LASTDR),A
36B4  D3 05                    OUT (DR),A
36B6  DB 05                    IN A,(SENSE1)
36B8  FE FF                    CP 0FFH ;IF FF THEN NO SENSE.TRY OTHER DRIVE
36BA  28 23                    JR Z,SCAN3
36BC  F5               CHECK1: PUSH AF ;SAVE SENSE
36BD  3E 02            CH10:   LD A,2 ;CHECK FOR SHIFT
36BF  B8                       CP B    ;IF B=2 THEN RESET SHIFTS
36C0  20 05                    JR NZ,CH11
36C2  F1                       POP AF
36C3  CB F7                    SET 6,A
```