



Executable prg file [3ddotscroll.zip](#)

```
;USED ASSEMBLER 64tass v1.45

;THIS IS THE EXAMPLE CODE OF 3D DOT SCROLL ON THE C64
;NO IRQ, WITHOUT MSX, NO DOUBBLE BUFFERING THIS IS ONLY SAMPLE
;I VERY LIKE THIS ART ON THIS COMPUTER
;THE RULE IS SIMPLE - ON THE 3D TRAJECTORY DATA OF CHARSET ARE WRITTING
;THE TRAJECTORY WAS PREPARED IN MY PROGRAM CREATED IN DELPHI

;=====
;= C64 3D DOT SCROLL =
;= BY WEGI /BS/SMR/FTM =
;=====

        *= $0801

;BASIC LINE START
        .BYTE $0B,$08,$90,$06,$9E,$32
        .BYTE $30,$34,$39,$00,$A0,$00

;-----
COUNTLINES = 8 ;8 LINES IN CHAR
COUNTCHAR = 16 ;16 CHARS TO SHIFT
LENLOOP = $62 ;LENGTH LOOP FOR SPEEDCODE ITERATION

STOREPLOT = $04 ;4 VECTOR FOR STORE MISC DATA
VECTR1     = $06
VECTR2     = $08
VECTR3     = $0A

SCREEN     = $2000 ;SCREEN ADDRES
SPEEDCODE = $4000 ;UNROLED CODE ADDRES FOR DISPLAY DATA
SPEEDCLEAR = SPEEDCODE + ( COUNTCHAR * LENLOOP * COUNTLINES)+1
;SPEED CLEAR ALL 1024 PLOT

CHAR1      = $20 ;CHAR TO SHIFT (ON ZP)
CHAR2      = CHAR1 + COUNTCHAR
CHAR3      = CHAR2 + COUNTCHAR
CHAR4      = CHAR3 + COUNTCHAR
CHAR5      = CHAR4 + COUNTCHAR
```

```

CHAR6      = CHAR5  + COUNTCHAR
CHAR7      = CHAR6  + COUNTCHAR
CHAR8      = CHAR7  + COUNTCHAR
CSET2      = CHAR8  + COUNTCHAR

; ---
START

    SEI
    CLD
    LDX #$FB
    TXS
    LDA #$37
    STA $01
    JSR $FDA3
    JSR $FD15
    JSR $E3BF
    JSR $FF5B
    SEI
    LDA #<DRAW
    STA $0318
    STA $FFFA
    STA $FFFE
    LDA #>DRAW
    STA $0319
    STA $FFFB
    STA $FFFF
    JSR INITGRAPH ;ENABLE HIRES ETC.
    JSR FILLCHAR  ;FILL CHAR
    JSR SETTBADR  ;HELP PROC. FOR PREPARE DATA
    JSR MAKESPEEDCODE ;MAKE LONG AND BORNING CODE FOR DOTSCROLL
                        ;AND SETTING PLOTS FOR WAIT LOOK
    JSR SPEEDCODE ;NOW PLOTS WILL BE CLEAR
    JSR CLEARCHAR ;NOW CHAR BE CLEAR
    JSR MAKESPEEDCLEAR ;LIKE BEFORE FOR CLEAR PLOTS AND SET PLOTS
    JSR SPEEDCLEAR ;OK NOW CLEAR PLOTS

;=====
;HERE IS IRQ NMI AND BRK FOR NEVERENDING LOOP
;IN THIS SAMPLE WE DON'T WORK IN THE IRQ
;=====
DRAW

    SEI
    CLD
    LDX #$FB ;STACK INIT
    TXS
    LDA #$38 ;SHOW ALL 64 RAM (IN THIS SAMPLE NOT NECESSARY)
    STA $01
    JSR CLEARCHAR
    JSR SPEEDCLEAR ;CLEAR PLOTS
    STA POSSCROLL ;START SCROL FROM ZERO POS.

```

```
;AFTER INIT AND MAKE SPEEDCODE HERE IS MAINLOP
;AND ALL NECESSARY ROUTINES TO WORK DOT SCROLL
;REALLY NOT THAT BIG ;- )
;FROM $0801 TO $0A29
;OF COURSE SPEEDCODE FROM $4000 TO $7D03 STILL OVER HERE
```

MAINLOOP

```
JSR ROLCHAR ;SHIFT DATA FOR DOT SCROLL
LDA #$35 ;SHOW I/O
STA $01
LDX #$C8
CPX $D012
BNE *-3

INC $D020
LDA #$38 ;SHOW ALL RAM
STA $01
JSR SPEEDCLEAR ;CLEAR PLOTS ON THE BITMAP
JSR SPEEDCODE ;DISPLAY PLOTS OF CHARS ON 3D TRAJECTORY
LDA #$35 ;SHOW I/O VIC ETC.
STA $01
DEC $D020

LDA #$EF
CMP $DC01 ;SPACE?
BNE MAINLOOP
CMP $DC01
BEQ *-3
LDA #$38
STA $01
BRK ;GO TO DRAW OF COURSE
```

```
;=====
```

```
;CLEAR OR FILL CHAR DATA
```

```
;=====
```

CLEARCHAR

```
LDA #$00
.BYTE $2C
```

FILLCHAR

```
LDA #$FF
```

INITCHAR

```
LDX #$00
```

```
-
```

```
STA CHAR1,X
```

```
INX
```

```
CPX #COUNTCHAR*8+8
```

```
BCC -
```

```
RTS
```

```
;=====
```

ROLCHAR

;=====

;SHIFT ALL BITS FOR DOT SCROLL

;DATA ON ZERO PAGE FOR SPEED

;=====

ASL CSET2

ROL CHAR1 + COUNTCHAR-1

ROL CHAR1 + COUNTCHAR-2

ROL CHAR1 + COUNTCHAR-3

ROL CHAR1 + COUNTCHAR-4

ROL CHAR1 + COUNTCHAR-5

ROL CHAR1 + COUNTCHAR-6

ROL CHAR1 + COUNTCHAR-7

ROL CHAR1 + COUNTCHAR-8

ROL CHAR1 + COUNTCHAR-9

ROL CHAR1 + COUNTCHAR-10

ROL CHAR1 + COUNTCHAR-11

ROL CHAR1 + COUNTCHAR-12

ROL CHAR1 + COUNTCHAR-13

ROL CHAR1 + COUNTCHAR-14

ROL CHAR1 + COUNTCHAR-15

ROL CHAR1

ASL CSET2+1

ROL CHAR2 + COUNTCHAR-1

ROL CHAR2 + COUNTCHAR-2

ROL CHAR2 + COUNTCHAR-3

ROL CHAR2 + COUNTCHAR-4

ROL CHAR2 + COUNTCHAR-5

ROL CHAR2 + COUNTCHAR-6

ROL CHAR2 + COUNTCHAR-7

ROL CHAR2 + COUNTCHAR-8

ROL CHAR2 + COUNTCHAR-9

ROL CHAR2 + COUNTCHAR-10

ROL CHAR2 + COUNTCHAR-11

ROL CHAR2 + COUNTCHAR-12

ROL CHAR2 + COUNTCHAR-13

ROL CHAR2 + COUNTCHAR-14

ROL CHAR2 + COUNTCHAR-15

ROL CHAR2

ASL CSET2+2

ROL CHAR3 + COUNTCHAR-1

ROL CHAR3 + COUNTCHAR-2

ROL CHAR3 + COUNTCHAR-3

ROL CHAR3 + COUNTCHAR-4

ROL CHAR3 + COUNTCHAR-5

ROL CHAR3 + COUNTCHAR-6

ROL CHAR3 + COUNTCHAR-7

ROL CHAR3 + COUNTCHAR-8

ROL CHAR3 + COUNTCHAR-9

ROL CHAR3 + COUNTCHAR-10
ROL CHAR3 + COUNTCHAR-11
ROL CHAR3 + COUNTCHAR-12
ROL CHAR3 + COUNTCHAR-13
ROL CHAR3 + COUNTCHAR-14
ROL CHAR3 + COUNTCHAR-15
ROL CHAR3

ASL CSET2+3

ROL CHAR4 + COUNTCHAR-1
ROL CHAR4 + COUNTCHAR-2
ROL CHAR4 + COUNTCHAR-3
ROL CHAR4 + COUNTCHAR-4
ROL CHAR4 + COUNTCHAR-5
ROL CHAR4 + COUNTCHAR-6
ROL CHAR4 + COUNTCHAR-7
ROL CHAR4 + COUNTCHAR-8
ROL CHAR4 + COUNTCHAR-9
ROL CHAR4 + COUNTCHAR-10
ROL CHAR4 + COUNTCHAR-11
ROL CHAR4 + COUNTCHAR-12
ROL CHAR4 + COUNTCHAR-13
ROL CHAR4 + COUNTCHAR-14
ROL CHAR4 + COUNTCHAR-15
ROL CHAR4

ASL CSET2+4

ROL CHAR5 + COUNTCHAR-1
ROL CHAR5 + COUNTCHAR-2
ROL CHAR5 + COUNTCHAR-3
ROL CHAR5 + COUNTCHAR-4
ROL CHAR5 + COUNTCHAR-5
ROL CHAR5 + COUNTCHAR-6
ROL CHAR5 + COUNTCHAR-7
ROL CHAR5 + COUNTCHAR-8
ROL CHAR5 + COUNTCHAR-9
ROL CHAR5 + COUNTCHAR-10
ROL CHAR5 + COUNTCHAR-11
ROL CHAR5 + COUNTCHAR-12
ROL CHAR5 + COUNTCHAR-13
ROL CHAR5 + COUNTCHAR-14
ROL CHAR5 + COUNTCHAR-15
ROL CHAR5

ASL CSET2+5

ROL CHAR6 + COUNTCHAR-1
ROL CHAR6 + COUNTCHAR-2
ROL CHAR6 + COUNTCHAR-3
ROL CHAR6 + COUNTCHAR-4
ROL CHAR6 + COUNTCHAR-5
ROL CHAR6 + COUNTCHAR-6

```
ROL CHAR6 + COUNTCHAR-7
ROL CHAR6 + COUNTCHAR-8
ROL CHAR6 + COUNTCHAR-9
ROL CHAR6 + COUNTCHAR-10
ROL CHAR6 + COUNTCHAR-11
ROL CHAR6 + COUNTCHAR-12
ROL CHAR6 + COUNTCHAR-13
ROL CHAR6 + COUNTCHAR-14
ROL CHAR6 + COUNTCHAR-15
ROL CHAR6
```

```
ASL CSET2+6
```

```
ROL CHAR7 + COUNTCHAR-1
ROL CHAR7 + COUNTCHAR-2
ROL CHAR7 + COUNTCHAR-3
ROL CHAR7 + COUNTCHAR-4
ROL CHAR7 + COUNTCHAR-5
ROL CHAR7 + COUNTCHAR-6
ROL CHAR7 + COUNTCHAR-7
ROL CHAR7 + COUNTCHAR-8
ROL CHAR7 + COUNTCHAR-9
ROL CHAR7 + COUNTCHAR-10
ROL CHAR7 + COUNTCHAR-11
ROL CHAR7 + COUNTCHAR-12
ROL CHAR7 + COUNTCHAR-13
ROL CHAR7 + COUNTCHAR-14
ROL CHAR7 + COUNTCHAR-15
ROL CHAR7
```

```
ASL CSET2+7
```

```
ROL CHAR8 + COUNTCHAR-1
ROL CHAR8 + COUNTCHAR-2
ROL CHAR8 + COUNTCHAR-3
ROL CHAR8 + COUNTCHAR-4
ROL CHAR8 + COUNTCHAR-5
ROL CHAR8 + COUNTCHAR-6
ROL CHAR8 + COUNTCHAR-7
ROL CHAR8 + COUNTCHAR-8
ROL CHAR8 + COUNTCHAR-9
ROL CHAR8 + COUNTCHAR-10
ROL CHAR8 + COUNTCHAR-11
ROL CHAR8 + COUNTCHAR-12
ROL CHAR8 + COUNTCHAR-13
ROL CHAR8 + COUNTCHAR-14
ROL CHAR8 + COUNTCHAR-15
ROL CHAR8
```

```
INC CNTR0L
```

```
LDA CNTR0L
```

```
AND #$07 ;IF ALL CHAR WAS SHIFTED THEN SCROLL NEXT CHAR
```

```
BEQ MYSCROL
```

```
RTS
```

```
;=====
```

```

MYSCROL
;=====
;SIMPLE SCROLL ROUTINE (MAX 256 CHAR!!!)
;=====
    LDX POSSCROLL
    LDA TXTSCROL,X
    BNE +
    STA POSSCROLL
    LDA TXTSCROL
    +
    AND #$3F
    ASL
    ASL
    ASL      ;CHAR MULTIPLY 8 FOR ADDRES IN THE CHARGEN
    STA VECTR1
    LDA #$00
    ADC #$D0
    STA VECTR1+1
    PHP      ;STATUS REGISTER SAVE
    LDY #$07
    SEI
    LDA $01
    PHA      ;SAVE $01
    LDA #$33 ;HERE IS USED CHARGEN FROM C64 ROM
    STA $01  ;YOU CAN USED SOMETHING OWN
    -
    LDA (VECTR1),Y
    STA CSET2,Y
    DEY
    BPL -
    PLA      ;RECALL $01
    STA $01
    PLP      ;RECALL STATUS REGISTER FOR "I" (INTERRUPTS WAS BLOCKED?)
    INC POSSCROLL
    RTS

;=====
POSSCROLL .BYTE 0
CNTROL .BYTE 0
TXTSCROL .TEXT " _____ C64 3D DOT SCROLLER "
          .BYTE 0
;=====
;=====
;AFTER INIT ALL DATA AND PROC. BELOW CAN BE ERASE
;=====
*= $1000
INITGRAPH
;=====
;ENABLE HIRES, FILL COLLOR, CLEAR BITMAP
;=====

```

```
LDA #$18
STA $D018

LDA $D011
ORA #$20
STA $D011

LDX #$00
LDA #246
-
STA $0400,X
STA $0500,X
STA $0600,X
STA $06F8,X
INX
BNE -
STX POSScroll

LDX #>SCREEN
STX VECTR1+1
LDY #$00
STY VECTR1

LDA #$00
-
STA (VECTR1),Y
INY
BNE -
INC VECTR1+1
DEX
BNE -
RTS

;=====
;CALCULATE TB ROW ADDRESS IN THE BITMAP
;=====
SETTBADR
LDX #$00
LDA #>SCREEN
STX VECTR1
STA VECTR1+1
-
LDA VECTR1
STA TBADLO,X
LDA VECTR1+1
STA TBADHI,X

LDA VECTR1
CLC
ADC #$40
```



```

    STA VECTR1

    LDA VECTR1+1
    ADC #$01
    STA VECTR1+1
    INX
    CPX #25
    BCC -
    RTS

;-----
TBBIT
    .BYTE %10000000
    .BYTE %01000000
    .BYTE %00100000
    .BYTE %00010000
    .BYTE %00001000
    .BYTE %00000100
    .BYTE %00000010
    .BYTE %00000001

;---
TBADLO
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0

;---
TBADHI
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0
    .BYTE 0,0,0,0,0

;-----
XPOSS    .BYTE 0,0
YPOSS    .BYTE 0

;-----
CALCPLOTADD
;=====
;CONVERT DATA FROM XPOSS LO HI AND YPOS (BITMAP POSS.) AND RETURN
;TO ACC #<PLOT ADDRES, YREG #>PLOT ADDRES, XREG BIT POSS.
;=====
    LDA YPOSS
    LSR A
    LSR A
    LSR A
    TAX

    LDA YPOSS
    AND #$07
    TAY

```

```

    LDA XPOSS
    AND #$F8
    CLC
    ADC TBADLO,X
    STA STOREPLOT

    LDA TBADHI,X
    ADC XPOSS+1
    STA STOREPLOT+1

    LDA XPOSS
    AND #$07
    TAX

;SET PLOT THIS IS NOT NECESSARY ONLY FOR WAIT LOOK
    LDA (STOREPLOT),Y
    EOR TBBIT,X
    STA (STOREPLOT),Y
;
    RTS
;OR THIS IS ONLY EOR PLOT PROCEDURE
    TYA
    CLC
    ADC STOREPLOT
    STA STOREPLOT
    BCC +
    INC STOREPLOT+1
    +
    LDA TBBIT,X
    TAX
    LDA STOREPLOT
    LDY STOREPLOT+1
;IN ACC #<PLOT ADDRES, YREG #>PLOT ADDRES, XREG BIT POSS.
    RTS
;=====
MAKESPEEDCODE
;=====
;ITERATOR FOR GENERATE UNROLLED CODE
;FROM SPEEDCODE BASE
;=====
    LDA #<SPEEDCODE
    STA VECTR3
    LDA #>SPEEDCODE
    STA VECTR3+1

    LDY #$00
    STY CNTRCOLUMN
    LDA #<CHAR1
    STA CNTCHAR

MCO00
    LDA #<PLOTS

```

```

STA VECTR1
LDA #>PLOTS
STA VECTR1+1
LDA CNTRCOLUMN
ASL
ADC CNTRCOLUMN
ADC VECTR1
STA VECTR1
BCC +
INC VECTR1+1
+
LDY #$00
STY CNTRLINES
JSR STOREDATA

```

MCOD1

```

JSR CALCLOTADD ;IN ACC #<PLOT ADDRES, YREG #>PLOT ADDRES, XREG
BIT POSS.

```

```

STA DOT8
STA DOT8+5
STY DOT8+1
STY DOT8+6
STX DOT8+3
JSR NEXTROW
JSR CALCLOTADD
STA DOT7
STA DOT7+5
STY DOT7+1
STY DOT7+6
STX DOT7+3

```

```

JSR NEXTROW
JSR CALCLOTADD
STA DOT6
STA DOT6+5
STY DOT6+1
STY DOT6+6
STX DOT6+3
JSR NEXTROW
JSR CALCLOTADD
STA DOT5
STA DOT5+5
STY DOT5+1
STY DOT5+6
STX DOT5+3
JSR NEXTROW
JSR CALCLOTADD
STA DOT4
STA DOT4+5
STY DOT4+1
STY DOT4+6

```

```
STX DOT4+3
JSR NEXTROW
JSR CALCPLTADD
STA DOT3
STA DOT3+5
STY DOT3+1
STY DOT3+6
STX DOT3+3
JSR NEXTROW
JSR CALCPLTADD
STA DOT2
STA DOT2+5
STY DOT2+1
STY DOT2+6
STX DOT2+3
JSR NEXTROW
JSR CALCPLTADD
STA DOT1
STA DOT1+5
STY DOT1+1
STY DOT1+6
STX DOT1+3
JSR NEXTROW
LDA CNTCHAR
STA LITERA1
INC CNTCHAR

LDY #$00
-
LDA FCOD,Y
STA (VECTR3),Y
INY
CPY #LENLOOP+1
BNE -
LDA VECTR3
CLC
ADC #LENLOOP
STA VECTR3
BCC +
INC VECTR3+1
+
INC CNTRLINES
LDA CNTRLINES
CMP #COUNTCHAR
BEQ +
JMP MCOD1
+
INC CNTRCOLUMN
LDA CNTRCOLUMN
CMP #COUNTLINES
BEQ +
```

```

        JMP MCODE0
        +
        RTS
;=====
;MAKE STA $ADDRESS PLOT FOR FAST CLEAR ALL 1024 BITS
;NOT SUPER OPTIMIZE!!! COS DON'T ELLIMINATED THIS SAME ADDRES FOR
;ANY PLOTS
;=====
MAKESPEEDCLEAR
        LDA #<SPEEDCLEAR
        STA VECTR3
        LDA #>SPEEDCLEAR
        STA VECTR3+1
        LDA #<PLOTS
        STA VECTR1
        LDA #>PLOTS
        STA VECTR1+1
        LDA #$A9
        LDY #$00
        STA (VECTR3),Y
        TYA
        INY
        STA (VECTR3),Y
        LDA VECTR3
        CLC
        ADC #$02
        STA VECTR3
        BCC +
        INC VECTR3+1
        +
-
        LDY #$00
        LDA #$8D
        STA (VECTR3),Y
        LDY #$03
        LDA #$60
        STA (VECTR3),Y
        JSR STOREDATA
        JSR CALCPLOTADD
        TAX
        TYA
        LDY #$02
        STA (VECTR3),Y
        TXA
        DEY
        STA (VECTR3),Y
        LDA VECTR1
        CLC
        ADC #$03
        STA VECTR1
        BCC +

```

```

        INC VECTR1+1
        +
        LDA VECTR3
        CLC
        ADC #$03
        STA VECTR3
        BCC +
        INC VECTR3+1
        +

        LDA VECTR1+1
        CMP #>EPL0T
        BNE -
        LDA VECTR1
        CMP #<EPL0T
        BNE -
        RTS

;===
;NEXT 8 DOT'S
;=====
NEXTROW
        LDA VECTR1
        CLC
        ADC #24
        STA VECTR1
        BCC +
        INC VECTR1+1
        +
STOREDATA
        LDY #$00
        LDA (VECTR1),Y
        STA XPOSS
        INY
        LDA (VECTR1),Y
        STA YPOSS
        INY
        LDA (VECTR1),Y
        STA XPOSS+1
        RTS

;=====
CNTRLINES .BYTE 0
CNTRCOLUMN .BYTE 0
CNTCHAR .BYTE 0
;=====
;BELOW IS THE SPEEDCODE BASE FOR ITERATION
;=====

FCOD
LITERA1 = *+1

        LDA $77

```

```

ASL      ;WHY ASL AND TAX? COS LOWEST BIT IN NYBBLES NOT WORKING
TAX      ;IN ILLEGAL OPCODE $8B (ANE... ANX) SO AFTER ASL
AND #$10 ;HI BIT IN CARRY LOWEST BIT IS SECOND SAVE ACC TO XREG
BEQ +    ;AND CHECK THE LOWEST BIT IN HI NYBBLE (AND #$10)

```

```
DOT4 = *+1
```

```

LDA $1000 ;HERE WILL BE OVERWRITE ADDRES OF PLOT
EOR #$10  ;AND HERE BIT POSS. OF PLOT
STA $1000

```

```
+
```

```
BCC +
```

```
DOT8 = *+1
```

```

LDA $1000
EOR #$10
STA $1000

```

```
+
```

```

.BYTE $8B , $80 ;ANX #$80
BEQ +

```

```
DOT7 = *+1
```

```

LDA $1000 ;AND LIKE BEFORE...
EOR #$10
STA $1000

```

```
+
```

```

.BYTE $8B , $40 ;ANX #$40 YOU UNDERSTAND I HOPPE
BEQ +

```

```
DOT6 = *+1
```

```

LDA $1000
EOR #$10
STA $1000

```

```
+
```

```

.BYTE $8B , $20
BEQ + ;IF BIT IS SET SO SET BIT ON THE BITMAP

```

```
DOT5 = *+1
```

```

LDA $1000
EOR #$10
STA $1000

```

```
+
```

```

.BYTE $8B , $08 ;IF BIT IS NOT SET GO TO THE NEXT
BEQ +

```

```
DOT3 = *+1
```

```

LDA $1000
EOR #$10
STA $1000

```

```
+
```

```

.BYTE $8B , $04
BEQ +

```

```
DOT2 = *+1
```

```

LDA $1000
EOR #$10
STA $1000

```

```
+
```

```

    .BYTE $8B , $02
    BEQ +
DOT1 = *+1
    LDA $1000
    EOR #$10
    STA $1000
+
    RTS
;=====
;MAIN TRAJECTORY 1024 PLOT POSS. IN THE HIRES SCREEN
;=====
PLOTS

; LOW X POS, Y POS, HI X POS ...
.BYTE 13 , 56 , 0 , 13 , 57 , 0 , 13 , 58 , 0 , 13 , 59 , 0 , 13 , 61 , 0 ,
13 , 62 , 0 , 13 , 63 , 0 , 13 , 64 , 0
.BYTE 14 , 56 , 0 , 14 , 57 , 0 , 14 , 58 , 0 , 14 , 59 , 0 , 14 , 61 , 0 ,
14 , 62 , 0 , 14 , 63 , 0 , 14 , 64 , 0
.BYTE 15 , 56 , 0 , 15 , 57 , 0 , 15 , 58 , 0 , 15 , 59 , 0 , 15 , 60 , 0 ,
15 , 62 , 0 , 15 , 63 , 0 , 15 , 64 , 0
.BYTE 16 , 55 , 0 , 16 , 57 , 0 , 16 , 58 , 0 , 16 , 59 , 0 , 16 , 60 , 0 ,
16 , 62 , 0 , 16 , 63 , 0 , 16 , 64 , 0
.BYTE 18 , 55 , 0 , 18 , 57 , 0 , 18 , 58 , 0 , 18 , 59 , 0 , 18 , 60 , 0 ,
18 , 61 , 0 , 18 , 63 , 0 , 18 , 64 , 0
.BYTE 19 , 55 , 0 , 19 , 56 , 0 , 19 , 58 , 0 , 19 , 59 , 0 , 19 , 60 , 0 ,
19 , 61 , 0 , 19 , 63 , 0 , 19 , 64 , 0
.BYTE 20 , 55 , 0 , 20 , 56 , 0 , 20 , 57 , 0 , 20 , 59 , 0 , 20 , 60 , 0 ,
20 , 61 , 0 , 20 , 62 , 0 , 20 , 64 , 0
.BYTE 21 , 55 , 0 , 21 , 56 , 0 , 21 , 57 , 0 , 21 , 58 , 0 , 21 , 60 , 0 ,
21 , 61 , 0 , 21 , 62 , 0 , 21 , 63 , 0
.BYTE 22 , 54 , 0 , 22 , 56 , 0 , 22 , 57 , 0 , 22 , 58 , 0 , 22 , 59 , 0 ,
22 , 61 , 0 , 22 , 62 , 0 , 22 , 63 , 0
.BYTE 23 , 54 , 0 , 23 , 55 , 0 , 23 , 57 , 0 , 23 , 58 , 0 , 23 , 59 , 0 ,
23 , 60 , 0 , 23 , 62 , 0 , 23 , 63 , 0
.BYTE 24 , 54 , 0 , 24 , 55 , 0 , 24 , 56 , 0 , 24 , 58 , 0 , 24 , 59 , 0 ,
24 , 60 , 0 , 24 , 61 , 0 , 24 , 63 , 0
.BYTE 24 , 54 , 0 , 24 , 55 , 0 , 24 , 56 , 0 , 24 , 57 , 0 , 24 , 59 , 0 ,
24 , 60 , 0 , 24 , 61 , 0 , 24 , 62 , 0
.BYTE 25 , 53 , 0 , 25 , 54 , 0 , 25 , 56 , 0 , 25 , 57 , 0 , 25 , 58 , 0 ,
25 , 60 , 0 , 25 , 61 , 0 , 25 , 62 , 0
.BYTE 26 , 53 , 0 , 26 , 54 , 0 , 26 , 55 , 0 , 26 , 57 , 0 , 26 , 58 , 0 ,
26 , 59 , 0 , 26 , 60 , 0 , 26 , 62 , 0
.BYTE 27 , 52 , 0 , 27 , 54 , 0 , 27 , 55 , 0 , 27 , 56 , 0 , 27 , 58 , 0 ,
27 , 59 , 0 , 27 , 60 , 0 , 27 , 61 , 0
.BYTE 28 , 52 , 0 , 28 , 53 , 0 , 28 , 55 , 0 , 28 , 56 , 0 , 28 , 57 , 0 ,
28 , 58 , 0 , 28 , 60 , 0 , 28 , 61 , 0
.BYTE 29 , 52 , 0 , 29 , 53 , 0 , 29 , 54 , 0 , 29 , 55 , 0 , 29 , 57 , 0 ,
29 , 58 , 0 , 29 , 59 , 0 , 29 , 61 , 0
.BYTE 30 , 51 , 0 , 30 , 52 , 0 , 30 , 54 , 0 , 30 , 55 , 0 , 30 , 56 , 0 ,
30 , 58 , 0 , 30 , 59 , 0 , 30 , 60 , 0
.BYTE 31 , 51 , 0 , 31 , 52 , 0 , 31 , 53 , 0 , 31 , 54 , 0 , 31 , 56 , 0 ,

```



```

31 , 57 , 0 , 31 , 58 , 0 , 31 , 60 , 0
.BYTE 31 , 50 , 0 , 31 , 51 , 0 , 31 , 53 , 0 , 31 , 54 , 0 , 31 , 55 , 0 ,
31 , 57 , 0 , 31 , 58 , 0 , 31 , 59 , 0
.BYTE 32 , 49 , 0 , 32 , 51 , 0 , 32 , 52 , 0 , 32 , 53 , 0 , 32 , 55 , 0 ,
32 , 56 , 0 , 32 , 57 , 0 , 32 , 59 , 0
.BYTE 33 , 49 , 0 , 33 , 50 , 0 , 33 , 52 , 0 , 33 , 53 , 0 , 33 , 54 , 0 ,
33 , 56 , 0 , 33 , 57 , 0 , 33 , 58 , 0
.BYTE 34 , 48 , 0 , 34 , 50 , 0 , 34 , 51 , 0 , 34 , 52 , 0 , 34 , 54 , 0 ,
34 , 55 , 0 , 34 , 57 , 0 , 34 , 58 , 0
.BYTE 35 , 48 , 0 , 35 , 49 , 0 , 35 , 51 , 0 , 35 , 52 , 0 , 35 , 53 , 0 ,
35 , 55 , 0 , 35 , 56 , 0 , 35 , 57 , 0
.BYTE 35 , 47 , 0 , 35 , 49 , 0 , 35 , 50 , 0 , 35 , 51 , 0 , 35 , 53 , 0 ,
35 , 54 , 0 , 35 , 56 , 0 , 35 , 57 , 0
.BYTE 36 , 47 , 0 , 36 , 48 , 0 , 36 , 49 , 0 , 36 , 51 , 0 , 36 , 52 , 0 ,
36 , 54 , 0 , 36 , 55 , 0 , 36 , 56 , 0
.BYTE 37 , 46 , 0 , 37 , 47 , 0 , 37 , 49 , 0 , 37 , 50 , 0 , 37 , 52 , 0 ,
37 , 53 , 0 , 37 , 55 , 0 , 37 , 56 , 0
.BYTE 38 , 45 , 0 , 38 , 47 , 0 , 38 , 48 , 0 , 38 , 50 , 0 , 38 , 51 , 0 ,
38 , 53 , 0 , 38 , 54 , 0 , 38 , 56 , 0
.BYTE 39 , 45 , 0 , 39 , 46 , 0 , 39 , 48 , 0 , 39 , 49 , 0 , 39 , 51 , 0 ,
39 , 52 , 0 , 39 , 54 , 0 , 39 , 55 , 0
.BYTE 39 , 44 , 0 , 39 , 46 , 0 , 39 , 47 , 0 , 39 , 49 , 0 , 39 , 50 , 0 ,
39 , 52 , 0 , 39 , 53 , 0 , 39 , 55 , 0
.BYTE 40 , 44 , 0 , 40 , 45 , 0 , 40 , 47 , 0 , 40 , 48 , 0 , 40 , 50 , 0 ,
40 , 51 , 0 , 40 , 53 , 0 , 40 , 54 , 0
.BYTE 41 , 43 , 0 , 41 , 45 , 0 , 41 , 46 , 0 , 41 , 48 , 0 , 41 , 49 , 0 ,
41 , 51 , 0 , 41 , 52 , 0 , 41 , 54 , 0
.BYTE 42 , 43 , 0 , 42 , 44 , 0 , 42 , 46 , 0 , 42 , 47 , 0 , 42 , 49 , 0 ,
42 , 50 , 0 , 42 , 52 , 0 , 42 , 53 , 0
.BYTE 43 , 42 , 0 , 43 , 44 , 0 , 43 , 45 , 0 , 43 , 47 , 0 , 43 , 48 , 0 ,
43 , 50 , 0 , 43 , 51 , 0 , 43 , 53 , 0
.BYTE 44 , 41 , 0 , 44 , 43 , 0 , 44 , 45 , 0 , 44 , 46 , 0 , 44 , 48 , 0 ,
44 , 49 , 0 , 44 , 51 , 0 , 44 , 53 , 0
.BYTE 44 , 41 , 0 , 44 , 43 , 0 , 44 , 44 , 0 , 44 , 46 , 0 , 44 , 47 , 0 ,
44 , 49 , 0 , 44 , 51 , 0 , 44 , 52 , 0
.BYTE 45 , 40 , 0 , 45 , 42 , 0 , 45 , 44 , 0 , 45 , 45 , 0 , 45 , 47 , 0 ,
45 , 49 , 0 , 45 , 50 , 0 , 45 , 52 , 0
.BYTE 46 , 40 , 0 , 46 , 42 , 0 , 46 , 43 , 0 , 46 , 45 , 0 , 46 , 47 , 0 ,
46 , 48 , 0 , 46 , 50 , 0 , 46 , 52 , 0
.BYTE 47 , 39 , 0 , 47 , 41 , 0 , 47 , 43 , 0 , 47 , 45 , 0 , 47 , 46 , 0 ,
47 , 48 , 0 , 47 , 50 , 0 , 47 , 51 , 0
.BYTE 48 , 39 , 0 , 48 , 41 , 0 , 48 , 42 , 0 , 48 , 44 , 0 , 48 , 46 , 0 ,
48 , 48 , 0 , 48 , 49 , 0 , 48 , 51 , 0
.BYTE 49 , 39 , 0 , 49 , 40 , 0 , 49 , 42 , 0 , 49 , 44 , 0 , 49 , 46 , 0 ,
49 , 47 , 0 , 49 , 49 , 0 , 49 , 51 , 0
.BYTE 50 , 38 , 0 , 50 , 40 , 0 , 50 , 42 , 0 , 50 , 44 , 0 , 50 , 45 , 0 ,
50 , 47 , 0 , 50 , 49 , 0 , 50 , 51 , 0
.BYTE 51 , 38 , 0 , 51 , 40 , 0 , 51 , 42 , 0 , 51 , 43 , 0 , 51 , 45 , 0 ,
51 , 47 , 0 , 51 , 49 , 0 , 51 , 51 , 0
.BYTE 52 , 38 , 0 , 52 , 40 , 0 , 52 , 41 , 0 , 52 , 43 , 0 , 52 , 45 , 0 ,
52 , 47 , 0 , 52 , 49 , 0 , 52 , 51 , 0

```

```
.BYTE 53 , 38 , 0 , 53 , 39 , 0 , 53 , 41 , 0 , 53 , 43 , 0 , 53 , 45 , 0 ,  
53 , 47 , 0 , 53 , 49 , 0 , 53 , 51 , 0  
.BYTE 55 , 37 , 0 , 55 , 39 , 0 , 55 , 41 , 0 , 55 , 43 , 0 , 55 , 45 , 0 ,  
55 , 47 , 0 , 55 , 49 , 0 , 55 , 51 , 0  
.BYTE 56 , 37 , 0 , 56 , 39 , 0 , 56 , 41 , 0 , 56 , 43 , 0 , 56 , 45 , 0 ,  
56 , 47 , 0 , 56 , 49 , 0 , 56 , 51 , 0  
.BYTE 57 , 37 , 0 , 57 , 39 , 0 , 57 , 41 , 0 , 57 , 43 , 0 , 57 , 45 , 0 ,  
57 , 47 , 0 , 57 , 49 , 0 , 57 , 51 , 0  
.BYTE 58 , 37 , 0 , 58 , 39 , 0 , 58 , 41 , 0 , 58 , 43 , 0 , 58 , 45 , 0 ,  
58 , 48 , 0 , 58 , 50 , 0 , 58 , 52 , 0  
.BYTE 60 , 37 , 0 , 60 , 40 , 0 , 60 , 42 , 0 , 60 , 44 , 0 , 60 , 46 , 0 ,  
60 , 48 , 0 , 60 , 50 , 0 , 60 , 52 , 0  
.BYTE 61 , 38 , 0 , 61 , 40 , 0 , 61 , 42 , 0 , 61 , 44 , 0 , 61 , 46 , 0 ,  
61 , 48 , 0 , 61 , 50 , 0 , 61 , 53 , 0  
.BYTE 63 , 38 , 0 , 63 , 40 , 0 , 63 , 42 , 0 , 63 , 44 , 0 , 63 , 47 , 0 ,  
63 , 49 , 0 , 63 , 51 , 0 , 63 , 53 , 0  
.BYTE 64 , 38 , 0 , 64 , 41 , 0 , 64 , 43 , 0 , 64 , 45 , 0 , 64 , 47 , 0 ,  
64 , 49 , 0 , 64 , 52 , 0 , 64 , 54 , 0  
.BYTE 66 , 39 , 0 , 66 , 41 , 0 , 66 , 43 , 0 , 66 , 46 , 0 , 66 , 48 , 0 ,  
66 , 50 , 0 , 66 , 52 , 0 , 66 , 55 , 0  
.BYTE 68 , 39 , 0 , 68 , 42 , 0 , 68 , 44 , 0 , 68 , 46 , 0 , 68 , 49 , 0 ,  
68 , 51 , 0 , 68 , 53 , 0 , 68 , 56 , 0  
.BYTE 69 , 40 , 0 , 69 , 42 , 0 , 69 , 45 , 0 , 69 , 47 , 0 , 69 , 50 , 0 ,  
69 , 52 , 0 , 69 , 54 , 0 , 69 , 57 , 0  
.BYTE 71 , 41 , 0 , 71 , 43 , 0 , 71 , 46 , 0 , 71 , 48 , 0 , 71 , 51 , 0 ,  
71 , 53 , 0 , 71 , 55 , 0 , 71 , 58 , 0  
.BYTE 73 , 42 , 0 , 73 , 44 , 0 , 73 , 47 , 0 , 73 , 49 , 0 , 73 , 52 , 0 ,  
73 , 54 , 0 , 73 , 57 , 0 , 73 , 59 , 0  
.BYTE 75 , 43 , 0 , 75 , 45 , 0 , 75 , 48 , 0 , 75 , 50 , 0 , 75 , 53 , 0 ,  
75 , 56 , 0 , 75 , 58 , 0 , 75 , 61 , 0  
.BYTE 78 , 44 , 0 , 78 , 46 , 0 , 78 , 49 , 0 , 78 , 52 , 0 , 78 , 54 , 0 ,  
78 , 57 , 0 , 78 , 60 , 0 , 78 , 62 , 0  
.BYTE 80 , 45 , 0 , 80 , 48 , 0 , 80 , 50 , 0 , 80 , 53 , 0 , 80 , 56 , 0 ,  
80 , 59 , 0 , 80 , 61 , 0 , 80 , 64 , 0  
.BYTE 83 , 46 , 0 , 83 , 49 , 0 , 83 , 52 , 0 , 83 , 55 , 0 , 83 , 58 , 0 ,  
83 , 60 , 0 , 83 , 63 , 0 , 83 , 66 , 0  
.BYTE 85 , 48 , 0 , 85 , 51 , 0 , 85 , 54 , 0 , 85 , 57 , 0 , 85 , 59 , 0 ,  
85 , 62 , 0 , 85 , 65 , 0 , 85 , 68 , 0  
.BYTE 88 , 50 , 0 , 88 , 53 , 0 , 88 , 56 , 0 , 88 , 59 , 0 , 88 , 61 , 0 ,  
88 , 64 , 0 , 88 , 67 , 0 , 88 , 70 , 0  
.BYTE 91 , 52 , 0 , 91 , 55 , 0 , 91 , 58 , 0 , 91 , 61 , 0 , 91 , 64 , 0 ,  
91 , 67 , 0 , 91 , 70 , 0 , 91 , 73 , 0  
.BYTE 94 , 54 , 0 , 94 , 57 , 0 , 94 , 60 , 0 , 94 , 63 , 0 , 94 , 66 , 0 ,  
94 , 69 , 0 , 94 , 72 , 0 , 94 , 75 , 0  
.BYTE 97 , 56 , 0 , 97 , 59 , 0 , 97 , 62 , 0 , 97 , 65 , 0 , 97 , 68 , 0 ,  
97 , 72 , 0 , 97 , 75 , 0 , 97 , 78 , 0  
.BYTE 101 , 58 , 0 , 101 , 61 , 0 , 101 , 65 , 0 , 101 , 68 , 0 , 101 , 71  
, 0 , 101 , 74 , 0 , 101 , 78 , 0 , 101 , 81 , 0  
.BYTE 105 , 60 , 0 , 105 , 64 , 0 , 105 , 67 , 0 , 105 , 71 , 0 , 105 , 74  
, 0 , 105 , 77 , 0 , 105 , 81 , 0 , 105 , 84 , 0  
.BYTE 108 , 63 , 0 , 108 , 67 , 0 , 108 , 70 , 0 , 108 , 74 , 0 , 108 , 77
```

```
, 0 , 108 , 80 , 0 , 108 , 84 , 0 , 108 , 87 , 0
.BYTE 112 , 66 , 0 , 112 , 69 , 0 , 112 , 73 , 0 , 112 , 77 , 0 , 112 , 80
, 0 , 112 , 84 , 0 , 112 , 87 , 0 , 112 , 91 , 0
.BYTE 117 , 69 , 0 , 117 , 73 , 0 , 117 , 76 , 0 , 117 , 80 , 0 , 117 , 84
, 0 , 117 , 87 , 0 , 117 , 91 , 0 , 117 , 95 , 0
.BYTE 121 , 72 , 0 , 121 , 76 , 0 , 121 , 80 , 0 , 121 , 83 , 0 , 121 , 87
, 0 , 121 , 91 , 0 , 121 , 95 , 0 , 121 , 98 , 0
.BYTE 126 , 75 , 0 , 126 , 79 , 0 , 126 , 83 , 0 , 126 , 87 , 0 , 126 , 91
, 0 , 126 , 95 , 0 , 126 , 98 , 0 , 126 , 102 , 0
.BYTE 131 , 79 , 0 , 131 , 83 , 0 , 131 , 87 , 0 , 131 , 91 , 0 , 131 , 95
, 0 , 131 , 98 , 0 , 131 , 102 , 0 , 131 , 106 , 0
.BYTE 136 , 82 , 0 , 136 , 86 , 0 , 136 , 90 , 0 , 136 , 94 , 0 , 136 , 98
, 0 , 136 , 103 , 0 , 136 , 107 , 0 , 136 , 111 , 0
.BYTE 141 , 86 , 0 , 141 , 90 , 0 , 141 , 94 , 0 , 141 , 98 , 0 , 141 , 103
, 0 , 141 , 107 , 0 , 141 , 111 , 0 , 141 , 115 , 0
.BYTE 147 , 90 , 0 , 147 , 94 , 0 , 147 , 98 , 0 , 147 , 102 , 0 , 147 ,
107 , 0 , 147 , 111 , 0 , 147 , 115 , 0 , 147 , 120 , 0
.BYTE 153 , 93 , 0 , 153 , 98 , 0 , 153 , 102 , 0 , 153 , 107 , 0 , 153 ,
111 , 0 , 153 , 115 , 0 , 153 , 120 , 0 , 153 , 124 , 0
.BYTE 159 , 97 , 0 , 159 , 102 , 0 , 159 , 106 , 0 , 159 , 111 , 0 , 159 ,
115 , 0 , 159 , 120 , 0 , 159 , 124 , 0 , 159 , 129 , 0
.BYTE 165 , 101 , 0 , 165 , 106 , 0 , 165 , 110 , 0 , 165 , 115 , 0 , 165 ,
120 , 0 , 165 , 124 , 0 , 165 , 129 , 0 , 165 , 134 , 0
.BYTE 171 , 105 , 0 , 171 , 110 , 0 , 171 , 115 , 0 , 171 , 119 , 0 , 171 ,
124 , 0 , 171 , 129 , 0 , 171 , 134 , 0 , 171 , 138 , 0
.BYTE 178 , 109 , 0 , 178 , 114 , 0 , 178 , 119 , 0 , 178 , 124 , 0 , 178 ,
128 , 0 , 178 , 133 , 0 , 178 , 138 , 0 , 178 , 143 , 0
.BYTE 185 , 113 , 0 , 185 , 118 , 0 , 185 , 123 , 0 , 185 , 128 , 0 , 185 ,
133 , 0 , 185 , 138 , 0 , 185 , 143 , 0 , 185 , 147 , 0
.BYTE 191 , 117 , 0 , 191 , 122 , 0 , 191 , 127 , 0 , 191 , 132 , 0 , 191 ,
137 , 0 , 191 , 142 , 0 , 191 , 147 , 0 , 191 , 152 , 0
.BYTE 198 , 121 , 0 , 198 , 126 , 0 , 198 , 131 , 0 , 198 , 136 , 0 , 198 ,
141 , 0 , 198 , 146 , 0 , 198 , 151 , 0 , 198 , 156 , 0
.BYTE 205 , 124 , 0 , 205 , 129 , 0 , 205 , 135 , 0 , 205 , 140 , 0 , 205 ,
145 , 0 , 205 , 150 , 0 , 205 , 155 , 0 , 205 , 161 , 0
.BYTE 212 , 128 , 0 , 212 , 133 , 0 , 212 , 138 , 0 , 212 , 143 , 0 , 212 ,
149 , 0 , 212 , 154 , 0 , 212 , 159 , 0 , 212 , 164 , 0
.BYTE 219 , 131 , 0 , 219 , 136 , 0 , 219 , 141 , 0 , 219 , 147 , 0 , 219 ,
152 , 0 , 219 , 157 , 0 , 219 , 163 , 0 , 219 , 168 , 0
.BYTE 226 , 134 , 0 , 226 , 139 , 0 , 226 , 144 , 0 , 226 , 150 , 0 , 226 ,
155 , 0 , 226 , 161 , 0 , 226 , 166 , 0 , 226 , 171 , 0
.BYTE 233 , 136 , 0 , 233 , 142 , 0 , 233 , 147 , 0 , 233 , 153 , 0 , 233 ,
158 , 0 , 233 , 164 , 0 , 233 , 169 , 0 , 233 , 174 , 0
.BYTE 240 , 138 , 0 , 240 , 144 , 0 , 240 , 150 , 0 , 240 , 155 , 0 , 240 ,
161 , 0 , 240 , 166 , 0 , 240 , 172 , 0 , 240 , 177 , 0
.BYTE 246 , 140 , 0 , 246 , 146 , 0 , 246 , 152 , 0 , 246 , 157 , 0 , 246 ,
163 , 0 , 246 , 168 , 0 , 246 , 174 , 0 , 246 , 179 , 0
.BYTE 253 , 142 , 0 , 253 , 148 , 0 , 253 , 153 , 0 , 253 , 159 , 0 , 253 ,
164 , 0 , 253 , 170 , 0 , 253 , 175 , 0 , 253 , 181 , 0
.BYTE 3 , 143 , 1 , 3 , 149 , 1 , 3 , 154 , 1 , 3 , 160 , 1 , 3 , 166 , 1 ,
3 , 171 , 1 , 3 , 177 , 1 , 3 , 182 , 1
```

```
.BYTE 9 , 144 , 1 , 9 , 149 , 1 , 9 , 155 , 1 , 9 , 161 , 1 , 9 , 166 , 1 ,  
9 , 172 , 1 , 9 , 177 , 1 , 9 , 183 , 1  
.BYTE 14 , 144 , 1 , 14 , 150 , 1 , 14 , 155 , 1 , 14 , 161 , 1 , 14 , 167  
, 1 , 14 , 172 , 1 , 14 , 178 , 1 , 14 , 183 , 1  
.BYTE 19 , 144 , 1 , 19 , 150 , 1 , 19 , 155 , 1 , 19 , 161 , 1 , 19 , 166  
, 1 , 19 , 172 , 1 , 19 , 178 , 1 , 19 , 183 , 1  
.BYTE 24 , 144 , 1 , 24 , 149 , 1 , 24 , 155 , 1 , 24 , 160 , 1 , 24 , 166  
, 1 , 24 , 171 , 1 , 24 , 177 , 1 , 24 , 182 , 1  
.BYTE 28 , 143 , 1 , 28 , 148 , 1 , 28 , 154 , 1 , 28 , 159 , 1 , 28 , 165  
, 1 , 28 , 170 , 1 , 28 , 176 , 1 , 28 , 181 , 1  
.BYTE 32 , 141 , 1 , 32 , 147 , 1 , 32 , 152 , 1 , 32 , 158 , 1 , 32 , 163  
, 1 , 32 , 169 , 1 , 32 , 174 , 1 , 32 , 179 , 1  
.BYTE 35 , 139 , 1 , 35 , 145 , 1 , 35 , 150 , 1 , 35 , 156 , 1 , 35 , 161  
, 1 , 35 , 166 , 1 , 35 , 172 , 1 , 35 , 177 , 1  
.BYTE 38 , 137 , 1 , 38 , 143 , 1 , 38 , 148 , 1 , 38 , 153 , 1 , 38 , 159  
, 1 , 38 , 164 , 1 , 38 , 169 , 1 , 38 , 175 , 1  
.BYTE 41 , 135 , 1 , 41 , 140 , 1 , 41 , 145 , 1 , 41 , 151 , 1 , 41 , 156  
, 1 , 41 , 161 , 1 , 41 , 166 , 1 , 41 , 172 , 1  
.BYTE 43 , 132 , 1 , 43 , 137 , 1 , 43 , 142 , 1 , 43 , 148 , 1 , 43 , 153  
, 1 , 43 , 158 , 1 , 43 , 163 , 1 , 43 , 168 , 1  
.BYTE 45 , 129 , 1 , 45 , 134 , 1 , 45 , 139 , 1 , 45 , 144 , 1 , 45 , 149  
, 1 , 45 , 155 , 1 , 45 , 160 , 1 , 45 , 165 , 1  
.BYTE 46 , 126 , 1 , 46 , 131 , 1 , 46 , 136 , 1 , 46 , 141 , 1 , 46 , 146  
, 1 , 46 , 151 , 1 , 46 , 156 , 1 , 46 , 161 , 1  
.BYTE 47 , 122 , 1 , 47 , 127 , 1 , 47 , 132 , 1 , 47 , 137 , 1 , 47 , 142  
, 1 , 47 , 147 , 1 , 47 , 152 , 1 , 47 , 157 , 1  
.BYTE 47 , 119 , 1 , 47 , 123 , 1 , 47 , 128 , 1 , 47 , 133 , 1 , 47 , 138  
, 1 , 47 , 143 , 1 , 47 , 147 , 1 , 47 , 152 , 1  
.BYTE 48 , 115 , 1 , 48 , 119 , 1 , 48 , 124 , 1 , 48 , 129 , 1 , 48 , 134  
, 1 , 48 , 138 , 1 , 48 , 143 , 1 , 48 , 148 , 1  
.BYTE 47 , 111 , 1 , 47 , 115 , 1 , 47 , 120 , 1 , 47 , 125 , 1 , 47 , 129  
, 1 , 47 , 134 , 1 , 47 , 139 , 1 , 47 , 143 , 1  
.BYTE 47 , 107 , 1 , 47 , 111 , 1 , 47 , 116 , 1 , 47 , 120 , 1 , 47 , 125  
, 1 , 47 , 129 , 1 , 47 , 134 , 1 , 47 , 138 , 1  
.BYTE 46 , 103 , 1 , 46 , 107 , 1 , 46 , 112 , 1 , 46 , 116 , 1 , 46 , 120  
, 1 , 46 , 125 , 1 , 46 , 129 , 1 , 46 , 134 , 1  
.BYTE 46 , 99 , 1 , 46 , 103 , 1 , 46 , 107 , 1 , 46 , 112 , 1 , 46 , 116 ,  
1 , 46 , 120 , 1 , 46 , 124 , 1 , 46 , 129 , 1  
.BYTE 44 , 95 , 1 , 44 , 99 , 1 , 44 , 103 , 1 , 44 , 107 , 1 , 44 , 111 ,  
1 , 44 , 116 , 1 , 44 , 120 , 1 , 44 , 124 , 1  
.BYTE 43 , 91 , 1 , 43 , 95 , 1 , 43 , 99 , 1 , 43 , 103 , 1 , 43 , 107 , 1  
, 43 , 111 , 1 , 43 , 115 , 1 , 43 , 119 , 1  
.BYTE 42 , 87 , 1 , 42 , 91 , 1 , 42 , 95 , 1 , 42 , 99 , 1 , 42 , 102 , 1  
, 42 , 106 , 1 , 42 , 110 , 1 , 42 , 114 , 1  
.BYTE 40 , 83 , 1 , 40 , 87 , 1 , 40 , 90 , 1 , 40 , 94 , 1 , 40 , 98 , 1 ,  
40 , 102 , 1 , 40 , 106 , 1 , 40 , 110 , 1  
.BYTE 38 , 79 , 1 , 38 , 83 , 1 , 38 , 86 , 1 , 38 , 90 , 1 , 38 , 94 , 1 ,  
38 , 98 , 1 , 38 , 101 , 1 , 38 , 105 , 1  
.BYTE 37 , 75 , 1 , 37 , 79 , 1 , 37 , 82 , 1 , 37 , 86 , 1 , 37 , 90 , 1 ,  
37 , 93 , 1 , 37 , 97 , 1 , 37 , 101 , 1  
.BYTE 35 , 71 , 1 , 35 , 75 , 1 , 35 , 79 , 1 , 35 , 82 , 1 , 35 , 86 , 1 ,
```

```
35 , 89 , 1 , 35 , 93 , 1 , 35 , 96 , 1
.BYTE 33 , 68 , 1 , 33 , 71 , 1 , 33 , 75 , 1 , 33 , 78 , 1 , 33 , 82 , 1 ,
33 , 85 , 1 , 33 , 89 , 1 , 33 , 92 , 1
.BYTE 31 , 65 , 1 , 31 , 68 , 1 , 31 , 71 , 1 , 31 , 75 , 1 , 31 , 78 , 1 ,
31 , 81 , 1 , 31 , 85 , 1 , 31 , 88 , 1
.BYTE 29 , 61 , 1 , 29 , 65 , 1 , 29 , 68 , 1 , 29 , 71 , 1 , 29 , 74 , 1 ,
29 , 78 , 1 , 29 , 81 , 1 , 29 , 84 , 1
.BYTE 27 , 58 , 1 , 27 , 61 , 1 , 27 , 65 , 1 , 27 , 68 , 1 , 27 , 71 , 1 ,
27 , 74 , 1 , 27 , 77 , 1 , 27 , 81 , 1
.BYTE 25 , 55 , 1 , 25 , 58 , 1 , 25 , 61 , 1 , 25 , 65 , 1 , 25 , 68 , 1 ,
25 , 71 , 1 , 25 , 74 , 1 , 25 , 77 , 1
.BYTE 23 , 52 , 1 , 23 , 55 , 1 , 23 , 58 , 1 , 23 , 61 , 1 , 23 , 64 , 1 ,
23 , 68 , 1 , 23 , 71 , 1 , 23 , 74 , 1
.BYTE 21 , 50 , 1 , 21 , 53 , 1 , 21 , 56 , 1 , 21 , 59 , 1 , 21 , 61 , 1 ,
21 , 64 , 1 , 21 , 67 , 1 , 21 , 70 , 1
```

EPL0T

```
.TEXT "END OF DATA"
```

From:

<http://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

http://codebase64.org/doku.php?id=base:3d_dot_scroll

Last update: **2015-04-17 04:30**

