



Basic Loader

```

10 REM *****
20 REM ** BASIC LOADER FOR **
30 REM ** ALTERNATE SCREENS **
40 REM *****
50 :
60 FOR I=49152 TO 49439
65 READ A:POKE I,A
70 CC=CC+A
75 NEXT
80 READ C$:IF C$<>CC THEN PRINT "CHECKSUM ERROR":STOP
100 DATA 255,255,0,0,191,255,0,0,255
110 DATA 191,0,0,191,191,0,0,255,191,0
120 DATA 0,255,191,0,0,255,255,0,173,1
130 DATA 192,201,1,208,30,169,0,141,6
140 DATA 192,169,64,141,7,192,32,224
150 DATA 192,162,0,32,246,192,173,2,192
160 DATA 141,32,208,173,3,192,141,33
170 DATA 208,173,0,192,208,6,32,32,193
180 DATA 76,139,192,169,0,141,4,192,141
190 DATA 5,192,174,0,192,173,5,192,24
200 DATA 105,4,202,208,250,24,105,28
210 DATA 141,5,192,24,105,36,141,7,192
220 DATA 173,0,192,162,4,10,202,208,252
230 DATA 24,105,112,141,0,192,173,24
240 DATA 208,41,15,13,8,192,141,24,208
250 DATA 173,5,192,141,136,2,173,1,192
260 DATA 208,6,174,0,192,32,211,192,201
270 DATA 2,240,4,32,189,192,96,174,0
280 DATA 192,32,246,192,32,211,192,32
290 DATA 32,193,169,0,141,6,192,169,64
300 DATA 141,7,192,162,0,32,211,192,32
310 DATA 189,192,96,173,6,192,133,251
320 DATA 173,7,192,133,252,169,0,133
330 DATA 253,169,216,133,254,32,3,193
340 DATA 96,189,9,192,141,32,208,189,18
350 DATA 192,141,33,208,96,173,6,192
360 DATA 133,253,173,7,192,133,254,169
370 DATA 0,133,251,169,216,133,252,32,3
380 DATA 193,96,173,32,208,157,9,192
390 DATA 173,33,208,157,18,192,96,162,3
400 DATA 160,0,177,251,145,253,136,208
410 DATA 249,230,252,230,254,202,48,10
420 DATA 208,240,177,251,145,253,160
430 DATA 231,208,232,96
440 DATA 36606:REM#CHECKSUM#

```

Alternate Screens

```

3 REM *****
5 REM ** **
10 REM ** ALTERNATE SCREENS **
11 REM ** **
12 REM *****
13 :
15 DN=0:REM FOR CASS DN=1
20 IFA=0 THEN A=LOAD "ALT SCREENS.HEX",DN,1
25 POKES5,0:POKES6,32:CLR:REM LOWER MEMTOP
40 SCNUMB=49152:REM SCREEN NUMBER
70 EDITFG=49153:REM 0-DISPLAY SCREEN
75 :REM 1-EDIT SCREEN
77 :REM 2-COPY C-RAM
80 ALT=49179:REM M/C START ADDRESS
85 BRDCOL=49154:REM BORDER COLOUR
87 PAPCOL=49155:REM PAPER COLOUR
90 :
95 REM **** MAIN MENU ****
96 :
100 PRINTCHR$(147):REM CLEAR SCREEN
105 PRINTCHR$(134):REM LT BLUE LETTERS
110 DN$="":REM 0 = CURSOR DOWN CHARS
130 PRINTTAB(0);DN$:"CBM 64 ALTERNATE SCREENS"
135 PRINTTAB(0);"-----"
140 PRINTTAB(5);DN$:"F1 - EDIT PICTURE"
160 PRINTTAB(5);DN$:"F3 - DISPLAY PICTURE SEQUENCE"
200 :
210 GETA$:IFA$="" THEN I0:REM WAIT KEYPRESS
220 IFA$="F1" THEN GOSUB1000
230 IFA$="F3" THEN GOSUB 1500
260 GOTO100
999 :
1000 REM **** EDIT SCREEN ****
1002 :
1005 EF=1:REM SET EDIT MODE
1010 PRINTCHR$(147)
1020 PRINTTAB(15);DN$:"EDIT MODE"
1030 PRINTDN$:INPUT"SCREEN NUMBER";SN#
1050 IFASC(SN#)<40 OR ASC(SN#)>55 THEN 1030
1060 PRINTDN$:INPUT"BORDER COLOUR";BC#
1070 IFVAL(BC#)>8 AND BC#<>0 THEN 1060
1080 PRINTDN$:INPUT"PAPER COLOUR";PC#
1090 IFVAL(PC#)>8 AND PC#<>0 THEN 1080
1100 :
1110 POKEDITFG,EF
1120 POKESCNUMB,VAL(SN#)
1130 POKEBRDCOL,VAL(BC#)
1140 POKEPAPCOL,VAL(PC#)
1150 SYS ALT
1155 REM ** WAIT FOR RETURN **
1162 INPUT"X":REM S = HOME CURSOR
1165 REM ** SAVE SCREEN **
1170 EF=2
1175 POKEDITFG,EF
1180 SYS ALT
1185 RETURN
1190 :
1500 REM **** DISPLAY SCREEN ****
1510 EF=0
1520 PRINT CHR$(147)
1545 SN#1
1550 POKEDITFG,EF
1555 POKESCNUMB,SN
1560 SYS ALT
1570 GET X$:IF X$="" THEN I570:REM WAIT KEYPRESS

```

```

1572 IF X$=CHR$(13) THEN 1580:REM NORMAL SCREEN
1575 SN=SN+1:IF SN<9 THEN 1555
1577 SN=1:GOTO1555
1580 POKESCNUMB,0:SYS ALT
1600 RETURN

```

Commodore 64

```

*****
+---+ ALTERNATE +---+
+---+ SCREENS +---+
+---+ FOR +---+
+---+ CBM 64 +---+
*****
FROM #BFR TO #BFD
:
CRANLO=#00 :START OF COLOUR RAM
CRAMHI=#08
NCOLLO=#00 :NORMAL COLOUR
NCOBASE=#00 :BASE ADDRESS
NSCRHI=#04 :NORMAL SCREEN BASE ADDRESS
NVCPOK=#10 :NORMAL VIC REG VALUE
:
BLOCKS=#03 :NO OF 256 BYTE BLOCKS
EXTRA=#E7 :EXTRA AMOUNT TO 1000 BYTES
:
COLOFF=#24 :COLOUR OFFSET HIBYTE
SCROFF=#1C :SCREEN OFFSET HIBYTE
VICOFF=#70 :VIC CTRL REG OFFSET
VCMASK=#0F :VIC CTRL REG LOGIC MASK
VICREG=#D018 :SCREEN LOCATION CONTROL REG
EDOREG=#028E :SCREEN EDITOR KERNAL REG
BORDER=#0020 :BORDER COLOUR REG
PAPER=#0021 :BACKGROUND COLOUR REG
:
#*#000 :SET LOAD POINTER
:
SCNUMB +---+ :SCREEN NUMBER
EDITFG +---+ :EDIT MODE FLAG
BRDCOL +---+ :BORDER COLOUR
PAPCOL +---+ :PAPER COLOUR
:
SCBASE +---+ :SCREEN BASE STORAGE
CLBASE +---+ :COLOUR BASE STORAGE
VPOKE +---+ :TEMP STORE FOR VIC NUMBER
BRDTAB +---+ :BORDER COLOURS TABLE
PAPTAB +---+ :PAPER COLOURS TABLE
:
+---+ SAVE SCREEN 0 +---+
LDA EDITFG
CMP #00 :IF 0 OR 2
BNE CALC :THEN DON'T SAVE
:
LDA #NCOLLO
STA CLBASE
LDA #NCOLHI
STA CLBASE+1
JSR SAVE :SAVE CRAM TO CB
:
LDX #000
JSR SAVEBP :SAVE B/P REGS
:
LDA BRDCOL
STA BORDER :SET NEW B/P
LDA PAPCOL :COLOURS
STA PAPER
:
+---+ CALCULATE COLOUR BASE +---+
CALC
LDA SCNUMB
BNE NOZERO
JSR RESET :SET NORMAL REGS
JMP TESTFG
:
NOZERO
LDA #000
STA SCBASE
STA SCBASE+1 :INIT SCBASE
:
LDX SCNUMB :LOAD SCREEN NUMBER
LDA SCBASE+1 :HI BYTE ONLY
MULT
CLC
ADC #04
DEX
BNE MULT
:
CLC
ADC #SCROFF :ADD SCREEN OFFSET
STA SCBASE+1
:
CLC
ADC #COLOFF :ADD COLOUR OFFSET
STA CLBASE+1
:
+---+ SET VIC AND EDITOR REGISTERS +---+
LDA SCNUMB
LDX #004
:
ASL A
DEX
BNE MORE :MULT BY 16
:
CLC
ADC #VICOFF :ADD OFFSET
STA VPOKE
LDA VICREG
AND #VCMASK
ORA #VPOKE
STA VICREG :RESTORE VIC REG
STA EDREG :RESTORE EDREG
RTS

```

```

ORA VPOKE
STA VICREG :SET VIC REG
:
LDA SCBASE+1
STA EDREG :SET EDITOR REG
:
+---+ TEST STATUS OF EDIT FLAG +---+
TESTFG
LDA EDITFG
BNE NOLOAD :IF 0 DISPLAY MODE
LDX SCNUMB
JSR LOADBP :LOAD B/P TO REGS
:
NOLOAD
CMP #02
BEQ CONT :IF 2 THEN SAVE RAM
:
JSR LOAD :LOAD COL TO CRAM
RTS
:
CONT
LDX SCNUMB
JSR SAVEBP :SAVE BP REGS
JSR SAVE :SAVE CRAM TO COL
JSR RESET :SET NORMAL REGS
LDA #NCOLLO
STA CLBASE :LOAD CBASE WITH C0 BASE
LDA #NCOLHI
STA CLBASE+1
LDX #000
JSR LOADBP :RELOAD ORIG BORD/PAP COLS
JSR LOAD :LOAD NRMAL SCR COLS
RTS
:
+---+ TRANSFER TO RAM S/R +---+
LOAD
LDA CLBASE
STA FROM :LOAD 0 PAGE PTRS
LDA CLBASE+1
STA FROM+1
:
LDA #CRANLO
STA TO
LDA #CRAMHI
STA TO+1
:
JSR COPY :COPY RAM AREA
RTS
:
+---+ LOAD BORDER & PAPER COLS S/R +---+
LOADBP
LDA BRDTAB,X
STA BORDER
LDA PAPTAB,X
STA PAPER
RTS
:
+---+ TRANSFER FROM RAM S/R +---+
SAVE
LDA CLBASE
STA TO
LDA CLBASE+1
STA TO+1
:
LDA #CRANLO
STA FROM
LDA #CRAMHI
STA FROM+1
:
JSR COPY :COPY RAM AREA
RTS
:
+---+ SAVE BORDER & PAPER COLS S/R +---+
SAVEBP
LDA BORDER
STA BRDTAB,X
LDA PAPER
STA PAPTAB,X
RTS
:
+---+ COPY 1000 BYTES S/R +---+
COPY
LDX #BLOCKS
LDY #000
NEXT
LDA (FROM),Y
STA (TO),Y
DEY
BNE NEXT
INCBLOC
INCBLOC
DEX
BNE FINISH
BNE NEXT
LDA (FROM),Y
STA (TO),Y
LDY #EXTRA
BNE NEXT :EXTRA BYTES
FINISH
RTS
:
+---+ RESET VIC AND EDIT REGS S/R +---+
RESET
LDA #NCOLLO
STA CLBASE
LDA #NCOLHI
STA CLBASE+1
LDA VICREG
AND #VCMASK
ORA #VPOKE
STA VICREG :RESTORE VIC REG
LDA #NSCRHI
STA EDREG :RESTORE EDREG
RTS

```