Program I

```
10 REM XCOL
20 REM By Ian Sharpe
30 REM (c) Computing with the Amstrad
40 REM
50 MEMORY &9FFF
60 FOR ADDR=&A000 TO &A19B:READ BYTE$
70 BYTEVAL=VAL("&"+BYTE$)
80 POKE ADDR,BYTEVAL
90 CHECK=CHECK+BYTEVAL
100 NEXT
110 IF CHECK<>37357 THEN PRINT"TYPING
 ERROR IN DATA":END
120 CALL &A000
130 PRINT"|XCOL to enable"
140 PRINT"|XDIS to disable"
150 PRINT"|SETHT,n to set level where
 n=0,1 or 2"
160 PRINT"BORDER a,b or INK x,a,b set
s top(a) and bottom(b) values"
170 NEW
180 DATA 21,47,A1,CB,46,C0,CB,C6
190 DATA CD,00,B9,3A,02,C0,A7,CA
200 DATA 1B,A0,21,E5,B7,11,D4,B7
210 DATA C3,21,A0,21,EA,B1,11,D9
220 DATA B1,22,43,A1,ED,53,45,A1
230 DATA CD,0C,B9,2A,45,A1,11,57
240 DATA A1,01,22,00,ED,B0,21,EB
250 DATA BD,11,40,A1,01,03,00,ED
260 DATA B0,01,4A,A0,21,63,A0,C3
270 DATA D1,BC,55,A0,C3,67,A0,C3
280 DATA 93,A0,C3,CB,A0,58,43,4F
290 DATA CC,5B,44,49,D3,53,45,54
300 DATA 48,D4,00,00,00,00,00,21
310 DATA 47,A1,CB,4E,C0,CB,CE,CD
320 DATA AF,A0,11,EB,BD,21,3D,A1
330 DATA 01,03,00,ED,B0,21,48,A1
340 DATA 0E,00,06,81,11,D7,A0,CD
350 DATA E0,BC,3E,01,CD,19,BD,32
360 DATA 55,A1,C9,21,47,A1,CB,4E
370 DATA C8,CB,BE,CD,AF,A0,21,40
380 DATA A1,11,EB,BD,01,03,00,ED
390 DATA B0,21,48,A1,C3,E6,BC,2A
400 DATA 45,A1,DD,21,57,A1,06,22
410 DATA 7E,F5,DD,7E,00,77,F1,DD
420 DATA 77,00,DD,23,23,10,F1,C9
430 DATA FE,01,C0,DD,7E,00,FE,03
440 DATA D0,C6,02,32,56,A1,C9,3A
450 DATA 55,A1,3D,32,55,A1,CA,ED
460 DATA A0,21,56,A1,BE,C0,ED,5B
470 DATA 45,A1,C3,25,BD,3E,06,32
480 DATA 55,A1,ED,5B,43,A1,C3,25
490 DATA BD,2A,43,A1,E5,11,79,A1
500 DATA CD,31,A1,2A,45,A1,E5,11
510 DATA 8A,A1,CD,31,A1,21,79,A1
520 DATA 22,43,A1,21,8A,A1,22,45
530 DATA A1,21,00,C0,01,00,40,36
540 DATA 00,23,0B,78,B1,C2,1F,A1
550 DATA E1,22,45,A1,E1,22,43,A1
560 DATA C9,7E,12,23,13,7E,06,10
570 DATA 12,13,10,FC,C9,C3,F9,A0
580 DATA 00,00,00,00,00,00,00,00
590 DATA 00,00,00,00,00,00,00,00
600 DATA 00,00,00,00,00,00,03,00
610 DATA 00,00,00,00,00,00,00,00
620 DATA 00,00,00,00,00,00,00,00
630 DATA 00,00,00,00,00,00,00,00
640 DATA 00,00,00,00,00,00,00,00
650 DATA 00,00,00,00,00,00,00,00
660 DATA 00,00,00,00,00,00,00,00
670 DATA 00,00,00,00,00,00,00,00
680 DATA 00,00,00,00,00,00,00,00
690 DATA 00,00,00,00,00,00,00,00
```

Program II

```
 ORG &A000

 ;*** initialise ***

 LD HL,flag   ;done before?
 BIT 0,(HL)
 RET NZ
 SET 0,(HL)
 CALL &B900   ;find OS version
 LD A,(&C002) ;and set vector
 AND A        ;address to suit
 JP Z,osv1    ;is v 1.0
 LD HL,&B7E5  ;is v 1.1
 LD DE,&B7D4
 JP xcol1

.osv1

 LD HL,&B1EA
 LD DE,&B1D9

.xcol1

 LD(vec1),HL
 LD(vec2),DE
 CALL &B90C     ;restore ROM state
 LD HL,(vec2)   ;copy initial ink
 LD DE,vecstore ;values to store
 LD BC,34
 LDIR
 LD HL,&BDEB   ;store jumpblock
```

```
LD DE,jpstore ;entry for later
LD BC,3       ;use
LDIR
LD BC,rsxtable  ;enable RSX
LD HL,workspace
JP &BCD1

.rsxtable

WORD nametable
JP enable
JP disable
JP setht

.nametable

DEFB   "XCO","L"+&80
DEFB   "XDI","S"+&80
DEFB  "SETH","T"+&80
NOP

.workspace

DEFS 4

;*** enable interrupt ***

.enable

LD HL,flag ;already enabled?
BIT 1,(HL)
RET NZ
SET 1,(HL)
CALL swap       ;swap vector/store
LD DE,&BDEB     ;patch
LD HL,clrpatch  ;SCR MODE CLR
LD BC,3
LDIR
LD HL,tickblock ;enable interrupt
LD C,0
LD B,&81
LD DE,rtn
CALL &BCE0
LD A,1
CALL &BD19
LD(counter),A
RET

;*** disable interrupt ***

.disable

LD HL,flag ;not enabled?
BIT 1,(HL)
RET Z
RES 1,(HL)
CALL swap
LD HL,jpstore   ;restore
LD DE,&BDEB     ;SCR MODE CLR
LD BC,3
LDIR
LD HL,tickblock
JP &BCE6        ;disable

;** swap store/vectors ***

.swap

LD HL,(vec2)
LD IX,vecstore
LD B,34

.swaploop

LD A,(HL)
PUSH AF
LD A,(IX)
LD(HL),A
POP AF
LD(IX),A
INC IX
INC HL
DJNZ swaploop
RET

;***** set height *****

.setht

CP 1
RET NZ
LD A,(IX)
CP 3
RET NC
ADD 2
LD(count),A
RET

;*** switch colours routine ***

.rtn

LD A,(counter) ;decrement counter
DEC A
LD(counter),A
JP Z,rtn1       ;0=frame flyback

LD HL,count
CP(HL)          ;switch point?
RET NZ
LD DE,(vec2)
JP &BD25        ;set new colours

.rtn1           ;recharge counter

LD A,6
LD(counter),A
LD DE,(vec1)
JP &BD25        ;reset colours

;**** clear screen ****

.cls

LD HL,(vec1) ;set inks to bgrnd
PUSH HL
LD DE,zvec1
CALL setvec
LD HL,(vec2)
PUSH HL
LD DE,zvec2
CALL setvec
LD HL,zvec1
LD(vec1),HL
LD HL,zvec2
LD(vec2),HL
LD HL,&C000
LD BC,&4000

.clsloop     ;scr mem=0's

LD(HL),0
INC HL
DEC BC
LD A,B
OR C
JP NZ,clsloop
POP HL          ;restore cols
LD(vec2),HL
POP HL
LD(vec1),HL
RET
.setvec
 LD A,(HL)
 LD(DE),A
 INC HL
 INC DE
 LD A,(HL)
 LD B,16
```

```
.setvecloop

 LD(DE),A
 INC DE
 DJNZ setvecloop
 RET
;*** program w. s. ****

.clrpatch

 JP cls
.jpstore   DEFS 3
.vec1      DEFS 2
.vec2      DEFS 2
.flag      DEFS 1
.tickblock DEFS 13
.counter   DEFS 1
.count     DEFS 3
.vecstore  DEFS 34
.zvec1     DEFS 17
.zvec2     DEFS 17

end
```

Program III

```
100 MODE 0
110 CALL &BC02
120 CALL &BB4E
130 DEFINT a-z
140 !XCOL
150 FOR x=0 TO 15
160   INK x,x,(x+16) MOD 27
170 NEXT
180 BORDER 9,18
190 FOR i=1 TO 14 STEP 13
200   FOR j=0 TO 3
210     FOR k=0 TO 2
220       FOR l=1 TO 16 STEP 5
230         LOCATE 1,i+j*3+k
240         PAPER j*4+(l-1)/5
250         PRINT STRING$(5,32);
260       NEXT l
270     NEXT k
280   NEXT j
290 NEXT i
300 WHILE INKEY(47)<0
310 WEND
```