co- ordinates have x values in the range 0 to 319, and y values ranging from 0 to 199. One byte is sufficient to hold all the possible values of y, but two bytes will be required to store values of x greater than 255. In BASIC, given the co-ordinates x and y, the corresponding bit is calculated using the following steps:

```
1000 HB=XAND248:VBYTE=INT(Y/8)
1010 RMY=YAND7:RMX=XAND7
1020 ROW=VBYTE*320+HB
1030 BYTE=BASE+ROW+RMY
1040 POKEBYTE,PEEK(BYTE)OR(2↑(7-RMX))
```

The corresponding machine code routine has to perform the same calculations. HB, ROW, BASE and BYTE all require two bytes, which makes the arithmetic processes more complex. Most of the coding is self-explanatory, but two sections of interest are those where y is divided by eight and VBYTE is multiplied by 320. Division by powers of two can be easily accomplished:

```
45       = 00101101
45/2 = 22 = 00010110
22/2 = 11 = 00001011
11/2 =  5 = 00000101
5/2  =  2 = 00000010
2/2  =  1 = 00000001
1/2  =  0 = 00000000
```

Each time a division by two is performed, the bits that go to make up the number being divided all move one place to the right. Hence division by eight can be achieved by three Logical Shifts Right (LSR). As we only need the integer part of the answer, we can conveniently ignore any bits that 'drop off' the right-hand end of the number. The LSR operation places the bit that is lost in the carry, so we could use it if we wished. Multiplication by two can, not surprisingly, be done by shifting one place to the left (ASL). We can use this fact to create a routine that will multiply VBYTE by 320. We can think of 320 as 5×64, and 64 itself as 2×2×2×2×2×2. Thus, if we take VBYTE, add it to itself five times and then perform six ASL's we will have effectively multiplied by 320. The only snag is that the answer may well be bigger than 255 and hence two bytes will be needed.

Finally, the two routines we have investigated are both called from within a BASIC program using SYS, and it is important that when the machine code routines have ended, the BASIC program can continue. During execution of a BASIC program, the interpreter makes use of the X, Y and A registers. As these registers are used by the machine code routines as well, it is important to store their contents on entering machine code and restore them on leaving the routine. The most convenient method of doing this is to use the stack. The values of the Y, X and A registers should be pushed onto the stack as soon as the machine code routine is entered and pulled off it before exit. Values for the co-ordinates, colours and flags can be passed to the machine code program by POKEing them into the locations specified, as demonstrated in the BASIC program.

```
1 GOTO 200
2 POKE 53265,PEEK(53265)AND223
3 POKE 53272,PEEK(53272)AND240OR4
4 STOP
200 REM  **** C64 HI-RES DEMO   ****
210 :
220 POKE 56,32:CLR:      REM LWR MEMTOP
250 GOSUB 3000:      REM INITIALISE S/R
350 :
360 REM  **** SET UP HIRES MODE ****
370 :
380 PRINT CC$:PRINT :PRINT
390 INPUT"FOREGROUND COLOUR";FG
400 INPUT"BACKGROUND COLOUR";BG
410 TT=FG*16+BG:     REM CALC. COLOUR
420 POKE COLOUR,TT: REM COL TO M/C S/R
430 POKE HRSFLG,1:  REM SET HIRES ON
440 POKE CLMFLG,1:  REM CLRHIRES SCRN ON
450 SYS BEGIN:      REM ENTER M/C S/R
460 :
500 REM  **** DRAW PATTERN      ****
510 :
515 Z=0:Y1=50:Y2=150:X1=160:SP=6
520 FORY=Y1 TO Y2 STEP SP
530 FORX= Y2-Z TO Y2+Z STEP SP
540 GOSUB1000:       REM PLOT POINT
550 NEXT X
555 Z=Z+SP
557 NEXT Y
560 :
565 GETJ$:IFJ$<>""THEN 565
570 GETA$:IF A$=""THEN 570:REM AWAIT KPRESS
580 :
600 REM  **** CLEAR HIRES SCREEN ****
605 POKE HRSFLG,1:POKE CLMFLG,1
610 SYS BEGIN
620 :
630 REM  ****   LINE DEMO        ****
640 X1=0:X2=300:Y1=0:Y2=190:SP=1
670 GOSUB1500:           REM LINE PLOT
690 :
695 GETJ$:IFJ$<>""THEN695
700 GETA$:IF A$=""THEN700:REM AWAIT KPRESS
710 :
720 REM  ****   RESTORE SCREEN   ****
730 :
740 POKE HRSFLG,0:   REM HRES OFF
750 SYS BEGIN
760 PRINT CC$:PRINT :PRINT
770 PRINTTAB(9)"****END OF PROGRAM****"
780 END
999 :
1000 REM *** HIRES PLOT SUBROUTINE ***
1010 :
1020 XHI=INT(X/HX):XLO=X-XHI*HX
1030 POKE XBYTE,XLO:POKE XPAGE,XHI:POKE YBYTE,Y
1035 SYS PLOT:        REM ENTER PLOT S/R
1040 RETURN
1500 REM *** LINE  PLOT SUBROUTINE ***
1550 C9=(Y2-Y1)/(X2-X1):C8=C9*X1-Y1
1600 FORX=X1 TO X2 STEP SP
1650 Y=X*C9-C8
1700 GOSUB 1000:          REM PLOT POINT
1750 NEXT X
1800 RETURN
3000 REM *** INITIALISE SUBROUTINE ***
3020 CC$=CHR$(147):       REM CLEAR SCRN
3025 HX=256
3030 HRSFLG=49408:        REM $C100
3040 CLMFLG=49409:        REM $C101
3050 COLOUR=49410:        REM $C102
3060 XBYTE =49411:        REM $C103
3070 XPAGE =49412:        REM $C104
3080 YBYTE =49413:        REM $C105
3085 BEGIN =49422:        REM $C10E
3090 PLOT  =49539:        REM $C183
3095 PRINT CC$:PRINT:PRINT
3100 PRINTTAB(9)"****M/CODE LOADER****"
3150 PRINTTAB(9)"1) M/CODE IS ON TAPE"
3200 PRINTTAB(9)"2) M/CODE IS IN DATA"
3250 PRINTTAB(9)"3) M/CODE IS IN MEM."
3300 PRINT"HIT OPTION NUMBER"
3350 FOR LP=0 TO 1 STEP 0
3400 GET OP$
3450 IF OP$>"0" AND OP$<"4" THEN LP=1
3500 NEXT LP
3600 ON VAL(OP$) GOSUB 4000,5000,6000
3900 RETURN
4000 REM** LOAD MCODE FROM TAPE S/R **
4100 PRINT "INSERT TAPE CONTAINING MACHINE CODE S/R"
4200 IF A=0 THEN A=1:LOAD "PLOTSUB.HEX",1,1
```