(12288/64). The next block of data is for the explosion, sprite 1. If we set the pointer in location 2041 to 193, then the data must start at 12352. These are the values we are using:

Sprite Number	0	1	2	3
Sprite Pointer	192	193	194	195
63 Bytes Of Sprite Data	12288 to 12350	12352 to 12414	12416 to 12478	12480 to 12542

Notice that one byte remains unused at the end of each block of sprite data. The parts of the program listing that read the sprite data of memory and specify the sprite pointers are contained in lines 2000 to 2210.

MANIPULATING SPRITES

The VIdeo Control (VIC) chip has several special registers that are used to control sprites. The first location of the VIC chip is 53248, and it is simpler for our program to describe the locations of all the other registers as relative to this. If we let V=53248, the next location of the VIC chip, 53249, can be termed V+1 and so on. V should be defined, with other variables, at an early stage (see line 100).

The colour of each sprite is set by POKEing a colour code number (in the range 0 to 15) into a special register. Each of the eight sprites has its own colour register; these run from V+39 to V+46. For example, to colour the ship black we simply POKE the colour code 0 into location V+39. The other sprites can be coloured in the same way (see lines 2220 to 2250).

Positioning sprites on the screen will be discussed in greater detail in the next instalment. For now it is sufficient to know that the x co-ordinate of sprite 0 is held in location V, the y co-ordinate for sprite 0 is held in location V+1 ; the x and y co-ordinates for sprite 1 are held in V+2 and V+3 respectively, and so on up to location V+15 (see lines 2260 to 2280).

Sprites can be expanded horizontally, vertically, or in both directions, by a factor of two. The ship and sub sprites may seem rather squashed horizontally, but we will now expand them to twice their original length. In fact all four sprites will be expanded horizontally. The VIC chip register controlling horizontal expansion is V+29, which is easier to use than the other registers we have discussed. Instead of using eight different registers to control the attributes of each of the eight sprites, all that is required is to switch the function on or off. Therefore only one bit within the register is required to control the horizontal expansion for each sprite. If a sprite is to be expanded horizontally, the corresponding bit in the V+29 register must be set to 1. The following table shows the POKE required to expand all the four sprites we have defined:

Sprite Number	7	6	5	4	3	2	1	0
Contents Of V+29	0	0	0	0	1	1	1	1

= 15 (decimal)

Expansion in the vertical direction is controlled by V+23. The explosion, sprite 1, is expanded vertically and horizontally, thus doubling its size (see lines 2290 to 2310).

Our final task is to turn the required sprites on. A single bit in the VIC chip register, V+21, is used to switch each sprite on or off. In the Subhunter game only the ship and the sub are initially turned on (lines 2310 to 2360).

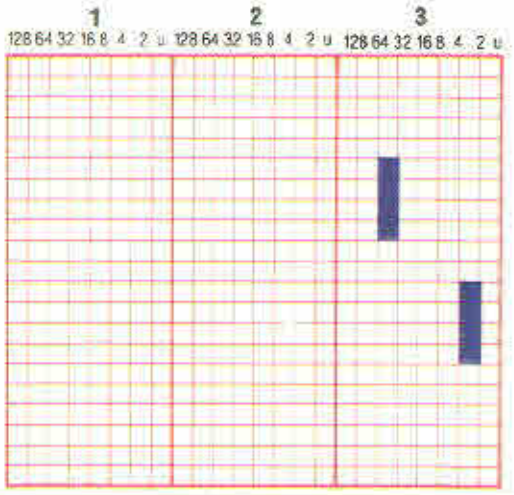
Once you have typed all of the listing in, you should test that the sprite data has been read correctly. To do this, run the program and break into it using RUN or STOP when the timer appears at the top of the screen. Entering the following statements, without line numbers, will position and display all four sprites created by the routine.

```
POKEV,160          (Ship's co-ordinate)
POKEV+2,240:       (Explosion's x and y
  POKEV+3,100       co-ordinates)
POKEV+4,160:       (Depth charge's
  POKEV+5,100       x and y co-ordinates)
POKEV+6,100:       (Submarine's x and y
  POKEV+7,100       co-ordinates)
POKEV+21,15        (Turns on sprites 0 – 3
```

If the program stops with an 'OUT OF DATA ERROR' message, check how many numbers there are in the DATA statements. There should be 63 for each sprite. If the program crashes and the keyboard fails to respond, make sure that V has been declared in line 100. It is always a good idea to SAVE your program before running it.

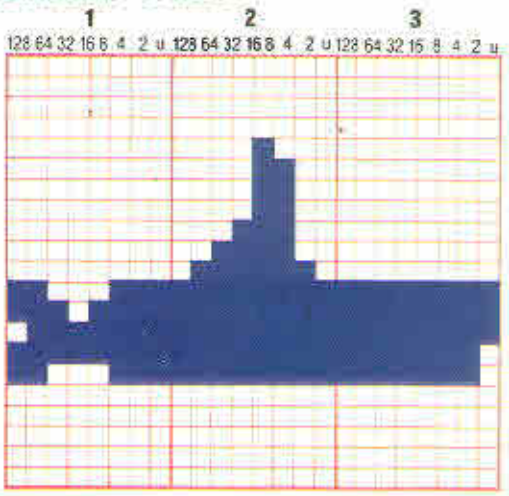
```
1 REM***C64 GRAPHICS*********
90 POKE 55,0:POKE 56,40:CLR
:REM LOWER MEMTOP
100 V=53248:FL=40:SC=0
110 GOSUB 1000
:REM SCREEN SETUP (see p217
120 GOSUB 2000
:REM SPRITE CREATION
2000 REM***SPRITE CREATION***
2020 REM** READ SHIP DATA  **
2030 FOR I=12288 TO 12350
2040 READ A:POKE I,A:NEXT I
2060 REM** READ EXP DATA   **
2070 FOR I=12352 TO 12414
2080 READ A:POKE I,A:NEXT I
2100 REM** READ CHRG DATA  **
2110 FOR I=12416 TO 12478
2120 READ A:POKE I,A:NEXT I
2140 REM** READ SUB DATA   **
2150 FOR I=12480 TO 12542
2160 READ A:POKE I,A:NEXT I
2180 REM** SET POINTERS    **
2190 POKE 2040,192:POKE 2041,
193:POKE 2042,194:POKE
2043,195
2220 REM** SET COLOURS     **
2230 POKE V+39,0:POKE V+40,1:
POKE V+41,0:POKE V+42,0
2260 REM**INIT SHIP COORDS **
2270 POKE V+1,80:XX=160
2290 REM** EXPAND SPRITES  **
2300 POKE V+29,15:POKE V+23,2
2320 REM** TURN ON SPRITES **
2330 POKE V+21,9
2340 RETURN
2350:
6000 REM** SHIP DATA       **
6010 DATA 0,0,0,0,0,0,0,0,0
6020 DATA 0,128,0,0,192,0,0,
192,0
6030 DATA 0,192,0,1,224,0,1,
224,0
6040 DATA 3,224,0,3,248,128,
3,253,0
6050 DATA 15,254,16,31,255,48,
255,255,255
6060 DATA 127,225,254,63,255,
254,31,255,252
6070 DATA 0,0,0,0,0,0,0,0,0
6100 REM** EXPLODE DATA    **
6110 DATA 0,0,0,0,0,0,0,16,0,
0,8,0,4,16
6120 DATA 0,9,2,64,1,56,128,
12,255,144
6130 DATA 1,230,40,5,151,0,11,
121,0,1
6140 DATA 169,0,25,214,96,0,
236,48,6,24
6150 DATA 152,3,90,0,0,51,0,
0,96,128,0
6160 DATA 64,0,0,0,0,0,0,0,0
6200 REM** DEPTH CHRG DATA **
6210 DATA 0,0,0,0,0,0,0,0,0,
0,0,0,0,0
6220 DATA 0,0,0,32,0,0,32,0,
0,32,0,0,32,0
6230 DATA 0,0,0,0,0,0,0,0,0
6240 DATA 2,0,0,2,0,0,2,0,0,
2,0,0
6250 DATA 0,0,0,0,0,0,0,0,0
6260 DATA 0,0,0,0,0,0,0,0,0
6300 REM** SUBMARINE DATA  **
6310 DATA 0,0,0,0,0,0,0,0,0,
0,0,0
6320 DATA 0,8,0,0,12,0,0,12,0
6330 DATA 0,12,0,0,28,0,0,60,0
6340 DATA 0,126,0,199,255,255,
239,255,255,127,
255,255
6360 DATA 255,255,254,199,
255,254
6370 DATA 0,0,0,0,0,0,0,0,0,
0,0,0,0,0
```

DEPTH CHARGES – 2



1	2	3
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	32
0	0	32
0	0	32
0	0	32
0	0	0
0	0	0
0	0	2
0	0	2
0	0	2
0	0	2
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

SUBMARINE – SPRITE 3



1	2	3
0	0	0
0	0	0
0	0	0
0	0	0
0	8	0
0	12	0
0	12	0
0	12	0
0	28	0
0	60	0
0	126	0
199	255	255
239	255	255
127	255	255
255	255	254
199	255	254
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0