

Twistin' and a-turnin'

Richard Sargent continues his look at screen manipulation and flips.

Last month we ended with a quick look at the two ROM routines SCR CHAR POSITION (&BC1A) and SCR DOT POSITION (&BC1D). These are general-purpose translation routines which are very useful to the machine-code programmer, since they accept either a character-screen coordinate or a graphics-screen coordinate and produce a 16-bit number pointing to a precise VDU RAM location. Routines like these simplify the code needed in programs which investigate portions of the screen, or which exchange sections of screen between VDU RAM and free RAM. Free RAM in this context is RAM which is above Basic and which is also not in danger of being over-written by the command SYMBOL, AFTER or by tape/disc-file operations. Before writing any routines this month, careful consideration must be given to their location within the free RAM - with quantities of screen-bytes moving around, some important code could easily get scrambled!

Room to experiment

You'll find that on a 434 or 664 the only sizeable chunk of memory available for storing screen-bytes is the 28K area from &4000 to &6FFF. This amount of space is sufficient for an entire alternate screen of 192 to be stored at &4000-&47FF and for a quarter-screen portion to be stored at &4000-&4FFF. The code required to manipulate the screen can therefore start at &4000 but it must finish at about &63FD since at this location the tape and disc files begin, followed closely by the user-defined graphics area.

Those readers familiar with the CPC628 will probably already know that this machine has a SCREENCOPY HEX routine on disc, and that the extra 64K of banked memory can hold four alternate screens. A screen can be swapped around in half-a-second, and, since the memory contents at &4000-&47FF is not disrupted by the move, HIMEM doesn't need to be moved right down to &3FFF. CPC628 users should be delighted to have this extra facility, but, just for this episode of the screen-manipulation series, they're being asked to pretend their machine hasn't got that additional 64K.

Character squares

The first three routines this month are concerned with manipulating the information in character squares. The



useful routine possessed by the 6128, COPYCHAR, will be introduced to CPC644 users, but the D-3-Y version COPYCHAR presented here is not quite the same as the one which resides in the 6128 Basic.

COPYCHAR

COPYCHAR is a machine-code routine called from Basic which looks at a screen location and reads the character found there in a pre-declared string variable. The new command, in its simplest form, takes the syntax AS-"@".CALL.&MEM.&A&. On the 6128 there is no need to declare the variable in advance, but the screen channel must be given and it is the current cursor position for that channel which is read. The following 6128 program segment reads the character at column 1, row 1 and prints it at column 8, row 3. The original character is not altered, and of course there is no obligation to reprint it (copy it) if you don't want to.

```
PRINT 1,1,CHR$(COPYCHAR(1,1,0,1,1,0,0,0))
```

For the 434, the same task is accomplished by:

```
PRINT 1,1,CHR$(PRINT COPYCHAR(1,1,0,1,1,0,0,0))
```

Figure 1 shows the assembly listing for a simple version of COPYCHAR which doesn't even bother with channel

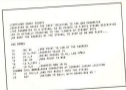


Figure 1/Listing 1

numbers. The code is relocatable, so it can go anywhere that's convenient.

However, COPYCHAR can be expanded beyond this simple version and some of the alternative arrangements which can be concocted offer the programmer some special effects. The first proposed change to COPYCHAR is to allow it to accept screen column and row coordinates, but to limit its operations to the current screen channel. This gives the format: `AB-W"CALL &ROSLC,R,MS` and the current cursor position is unaffected until such time as `AB` is copied by a `PRINT` command. So far so good, but there is a complication. The versions of COPYCHAR which are easiest to code use a ROM routine called `TXT RD CHAR` at `&B860` (Read a Character from the Text screen) and this routine is not infallible. The read is performed by comparing the matrix found on the screen with the matrices used to generate the characters. The problem is that the screen matrix can become altered in ways which will baffle the computer. One innocent way to muddle the screen matrix is to change paper and/or pen ink—the character may look the same but the bits in the screen bytes have undergone some transformations to encode the new colours. `TXT RD CHAR` will attempt to assemble the colours, but success is not guaranteed. The colour-decoded screen matrix is compared with the character matrix held in ROM and/or RAM, where Amstrad have defined 256 characters. The RAM UDGs are recognised. Failure to match with one of these causes `TXT RD CHAR` to generate a "carry false" flag and the routine COPYCHAR is left holding a null string. If the character is offset within its character square (as it might be if it was printed to the graphics screen using `TAG`) then it can't be recognised. If the character has a graphics line or point plotted on it then again it can't be recognised, even if it's sitting squarely in its character square.

Delving deeper

`TXT RD CHAR` is a fairly high-level ROM routine with the intelligence to try colour combinations when attempting to read a character. In the section of the ROM which deals with screen-handling there are two low-level routines called `SCR UNPACK` (`&BC08`) and `SCR REPACK` (`&BC00`) and they actually deal with the difficult task of moving bits and bytes from the character matrix to the screen matrix and vice versa. `SCR REPACK` is particularly interesting since it is used by `TXT RD CHAR` to compress a screen matrix, which is 32 bytes in size for a Mode 0 character and 16 bytes in Mode 1, to the standard character matrix which is always 8 bytes in size. Obviously `SCR REPACK` can't cope with sorting out a multitude of paper and pen ink either (or also `TXT RD CHAR` would be more successful, but what it can do is build up a character matrix in 8 bytes of RAM to which the programmer has full access. From there it can (if you so wish) be expanded back into a screen matrix using `SCR UNPACK` (and some additional code) at a different VDU location. The advantages of having a screen character residing even temporarily in 8 contiguous bytes of RAM are substantial as we shall see later.

Returning to COPYCHAR, four versions are presented here, and each will run on any CPC computer. For reasons which will become apparent later, the `MS`

parameter is no longer needed.

`CALL &R00,I,COL,ROW` is the version which provides its own screen coordinates, in which case the current cursor position is left untouched (which is usually most helpful.) The significance of the zero (the third parameter) will be explained shortly.

```
CALL &R00,&B810,&B810,&B810,&B810,&B810,&B810,
&B810,&B810,&B810,0,0,0,0
```

is the long command of COPYCHAR version two. The integer array `C()` accepts the value of each byte in the compressed character matrix, and a subsequent printing of `C()` allows the screen character to be viewed easily in BIN format, and incorporated into a `SYMBOL` statement. The Basic program to do this is shown in figure 2/Listing 2. The machine-code routine for the fully-fledged COPYCHAR runs to some 236 bytes because the code includes other features yet to be described.

```
00 00 00000000 00 0000 0000 0000 0000 0000 0000
01 00 00 00 00000000 00 0000 0000 0000 0000
02 00 00 0000 0000 0000 0000 0000 0000 0000
03 00 00 0000 0000 0000 0000 0000 0000 0000
04 0000 0000 0000 0000
05 00 00 00 0000 0000 0000 0000
06 00 00 00 0000 0000 0000 0000
```

Figure 2/Listing 2

Rotate

Rotate is a machine-code extension of COPYCHAR which turns a character around on the screen, so that for example, a letter or number may be placed on its side and used to label the Y-axis of a graph. User-defined-graphics may also benefit from being rotated. The ideal rotate routine can be considered as a number of defined stages, in which ROM routines occasionally do some of the hard work:

1. Save the current cursor position.
2. Obtain location of character to be rotated. :TXT GET CURSOR
3. Pack character into RAM. :TXT RD CHAR
4. Rotate character in RAM 90, 180 or 270 degrees. :SCR REPACK
5. Unpack character back into VDU RAM.
6. Restore cursor position. :SCR UNPACK
:TXT SET CURSOR

The five ROM calls don't tell the whole story. `SCR REPACK`, for example, likes to know the pen ink colour for the current stream, coded to suit the prevailing screen mode. A call to `TXT GET PEN` (`&B800`) followed immediately by a call to `SCR INK ENCODE` (`&BC0C`) finds this information. The main problem, though, is that `SCR UNPACK` doesn't actually place the bits and bytes back onto the screen — it has to be aided and abetted by some extra machine code which turns out to be unreasonably long when asked to cope with modes 0 and 1. However, two tricks can circumvent the length problem. One is to take UDG number 285 out of general use and use it exclusively for rotation duties, and the other is to make the rotation command share the

COPYCHAR syntax. The rotation routine now becomes:

1. Save the current cursor position. `:TXT GET CURSOR`
2. Obtain location of character to be rotated. `:TXT GET CHAR`
3. Pack character into UDG 255 RAM. `:SCR REPACK`
4. Rotate character in RAM 90, 180 or 270 degrees.
5. Write UDG 255 to required location.
6. Restore cursor position. `:TXT SET CURSOR`

The extra ROM routine required in this version is TXT GET MATHX (@BBAN), which finds the address of UDG 255 (as required by SCR REPACK). The Basic syntax of the new command (the third version of COPYCHAR) is CALL @BBAN,N,COL,ROW where N is a number 1, 2 or 3 replacing the new third-parameter. The numbers simply indicate how many 90-degree rotations are required. COL and ROW are, as usual, the coordinates where the character is to be found, and it is actually rotated on the screen as well as being loaded into UDG 255. As with COPYCHAR version one, rather than show the entire coding, the principles at work are illustrated in the course of listing 3 (Figure 3) so that the points of interaction between the Basic and machine-code can be seen.

LISTING 3 - EXAMPLE CODE

```
CALL @BBAN 1,100,100,100
LN 2,1,100,100
LN 3,1,100,100
LN 4,1,100,100
LN 5,1,100,100
CALL @BBAN 1,100,100,100
LN 6,1,100,100
LN 7,1,100,100
LN 8,1,100,100
LN 9,1,100,100
LN 10,1,100,100
LN 11,1,100,100
LN 12,1,100,100
LN 13,1,100,100
LN 14,1,100,100
LN 15,1,100,100
LN 16,1,100,100
LN 17,1,100,100
LN 18,1,100,100
LN 19,1,100,100
LN 20,1,100,100
LN 21,1,100,100
LN 22,1,100,100
LN 23,1,100,100
LN 24,1,100,100
LN 25,1,100,100
LN 26,1,100,100
LN 27,1,100,100
LN 28,1,100,100
LN 29,1,100,100
LN 30,1,100,100
LN 31,1,100,100
LN 32,1,100,100
LN 33,1,100,100
LN 34,1,100,100
LN 35,1,100,100
LN 36,1,100,100
LN 37,1,100,100
LN 38,1,100,100
LN 39,1,100,100
LN 40,1,100,100
LN 41,1,100,100
LN 42,1,100,100
LN 43,1,100,100
LN 44,1,100,100
LN 45,1,100,100
LN 46,1,100,100
LN 47,1,100,100
LN 48,1,100,100
LN 49,1,100,100
LN 50,1,100,100
LN 51,1,100,100
LN 52,1,100,100
LN 53,1,100,100
LN 54,1,100,100
LN 55,1,100,100
LN 56,1,100,100
LN 57,1,100,100
LN 58,1,100,100
LN 59,1,100,100
LN 60,1,100,100
LN 61,1,100,100
LN 62,1,100,100
LN 63,1,100,100
LN 64,1,100,100
LN 65,1,100,100
LN 66,1,100,100
LN 67,1,100,100
LN 68,1,100,100
LN 69,1,100,100
LN 70,1,100,100
LN 71,1,100,100
LN 72,1,100,100
LN 73,1,100,100
LN 74,1,100,100
LN 75,1,100,100
LN 76,1,100,100
LN 77,1,100,100
LN 78,1,100,100
LN 79,1,100,100
LN 80,1,100,100
LN 81,1,100,100
LN 82,1,100,100
LN 83,1,100,100
LN 84,1,100,100
LN 85,1,100,100
LN 86,1,100,100
LN 87,1,100,100
LN 88,1,100,100
LN 89,1,100,100
LN 90,1,100,100
LN 91,1,100,100
LN 92,1,100,100
LN 93,1,100,100
LN 94,1,100,100
LN 95,1,100,100
LN 96,1,100,100
LN 97,1,100,100
LN 98,1,100,100
LN 99,1,100,100
LN 100,1,100,100
```

```
LN 100,1,100,100
LN 101,1,100,100
LN 102,1,100,100
LN 103,1,100,100
LN 104,1,100,100
LN 105,1,100,100
LN 106,1,100,100
LN 107,1,100,100
LN 108,1,100,100
LN 109,1,100,100
LN 110,1,100,100
LN 111,1,100,100
LN 112,1,100,100
LN 113,1,100,100
LN 114,1,100,100
LN 115,1,100,100
LN 116,1,100,100
LN 117,1,100,100
LN 118,1,100,100
LN 119,1,100,100
LN 120,1,100,100
LN 121,1,100,100
LN 122,1,100,100
LN 123,1,100,100
LN 124,1,100,100
LN 125,1,100,100
LN 126,1,100,100
LN 127,1,100,100
LN 128,1,100,100
LN 129,1,100,100
LN 130,1,100,100
LN 131,1,100,100
LN 132,1,100,100
LN 133,1,100,100
LN 134,1,100,100
LN 135,1,100,100
LN 136,1,100,100
LN 137,1,100,100
LN 138,1,100,100
LN 139,1,100,100
LN 140,1,100,100
LN 141,1,100,100
LN 142,1,100,100
LN 143,1,100,100
LN 144,1,100,100
LN 145,1,100,100
LN 146,1,100,100
LN 147,1,100,100
LN 148,1,100,100
LN 149,1,100,100
LN 150,1,100,100
LN 151,1,100,100
LN 152,1,100,100
LN 153,1,100,100
LN 154,1,100,100
LN 155,1,100,100
LN 156,1,100,100
LN 157,1,100,100
LN 158,1,100,100
LN 159,1,100,100
LN 160,1,100,100
LN 161,1,100,100
LN 162,1,100,100
LN 163,1,100,100
LN 164,1,100,100
LN 165,1,100,100
LN 166,1,100,100
LN 167,1,100,100
LN 168,1,100,100
LN 169,1,100,100
LN 170,1,100,100
LN 171,1,100,100
LN 172,1,100,100
LN 173,1,100,100
LN 174,1,100,100
LN 175,1,100,100
LN 176,1,100,100
LN 177,1,100,100
LN 178,1,100,100
LN 179,1,100,100
LN 180,1,100,100
LN 181,1,100,100
LN 182,1,100,100
LN 183,1,100,100
LN 184,1,100,100
LN 185,1,100,100
LN 186,1,100,100
LN 187,1,100,100
LN 188,1,100,100
LN 189,1,100,100
LN 190,1,100,100
LN 191,1,100,100
LN 192,1,100,100
LN 193,1,100,100
LN 194,1,100,100
LN 195,1,100,100
LN 196,1,100,100
LN 197,1,100,100
LN 198,1,100,100
LN 199,1,100,100
LN 200,1,100,100
```

Basic loader for machine code

```
LN 100,1,100,100
LN 101,1,100,100
LN 102,1,100,100
LN 103,1,100,100
LN 104,1,100,100
LN 105,1,100,100
LN 106,1,100,100
LN 107,1,100,100
LN 108,1,100,100
LN 109,1,100,100
LN 110,1,100,100
LN 111,1,100,100
LN 112,1,100,100
LN 113,1,100,100
LN 114,1,100,100
LN 115,1,100,100
LN 116,1,100,100
LN 117,1,100,100
LN 118,1,100,100
LN 119,1,100,100
LN 120,1,100,100
LN 121,1,100,100
LN 122,1,100,100
LN 123,1,100,100
LN 124,1,100,100
LN 125,1,100,100
LN 126,1,100,100
LN 127,1,100,100
LN 128,1,100,100
LN 129,1,100,100
LN 130,1,100,100
LN 131,1,100,100
LN 132,1,100,100
LN 133,1,100,100
LN 134,1,100,100
LN 135,1,100,100
LN 136,1,100,100
LN 137,1,100,100
LN 138,1,100,100
LN 139,1,100,100
LN 140,1,100,100
LN 141,1,100,100
LN 142,1,100,100
LN 143,1,100,100
LN 144,1,100,100
LN 145,1,100,100
LN 146,1,100,100
LN 147,1,100,100
LN 148,1,100,100
LN 149,1,100,100
LN 150,1,100,100
LN 151,1,100,100
LN 152,1,100,100
LN 153,1,100,100
LN 154,1,100,100
LN 155,1,100,100
LN 156,1,100,100
LN 157,1,100,100
LN 158,1,100,100
LN 159,1,100,100
LN 160,1,100,100
LN 161,1,100,100
LN 162,1,100,100
LN 163,1,100,100
LN 164,1,100,100
LN 165,1,100,100
LN 166,1,100,100
LN 167,1,100,100
LN 168,1,100,100
LN 169,1,100,100
LN 170,1,100,100
LN 171,1,100,100
LN 172,1,100,100
LN 173,1,100,100
LN 174,1,100,100
LN 175,1,100,100
LN 176,1,100,100
LN 177,1,100,100
LN 178,1,100,100
LN 179,1,100,100
LN 180,1,100,100
LN 181,1,100,100
LN 182,1,100,100
LN 183,1,100,100
LN 184,1,100,100
LN 185,1,100,100
LN 186,1,100,100
LN 187,1,100,100
LN 188,1,100,100
LN 189,1,100,100
LN 190,1,100,100
LN 191,1,100,100
LN 192,1,100,100
LN 193,1,100,100
LN 194,1,100,100
LN 195,1,100,100
LN 196,1,100,100
LN 197,1,100,100
LN 198,1,100,100
LN 199,1,100,100
LN 200,1,100,100
```

Figure 3/Listing 3

Figure 4



Double-height

A double-height character option is always useful to have: it's the fourth version of COPYCHAR, and only involves three dozen bytes of extra code. The routine already in use can produce a copy-character (rotated or normal) in UDG 253, and enlarging this character in the vertical axis is simply a matter of spreading the 8-bytes of UDG 253 across the 16-bytes of UDGs 253/254 and then printing UDG 253 on the screen-row beneath UDG 254. A double-height character is generated in response to the syntax:

```
CALL @0000.4.10.000
```

The fifth parameter, the L, tells the machine code to enlarge the character at COL,ROW coordinates by splitting it between the two UDGs. If N is greater than zero, the character will be rotated before it's enlarged. The new character then lies dormant until reprinted by LOCATE X,Y:PRINT CHR\$(253):LOCATE X,Y-1:PRINT CHR\$(254), or until specified in two string variables for later use by A\$=CHR\$(253):B\$=CHR\$(254). Double-height characters look particularly pleasant in mode 0.

Mixing modes

Having produced a double-height character in any mode, the question which now arises is this: can characters of one particular mode be printed onto a screen of a different mode? The answer is yes, but with some difficulty. One novel way that I've come across is to use interrupts to fool the computer into thinking that the top third of the screen is set to one mode, the middle third to another mode, and the bottom third to yet another mode. This method uses a long machine-code program, running constantly in the background, and the screen is always divided in the system of horizontal zones. Rather more in keeping with our screen-manipulation theme is a Basic program which uses the UDG-matrices to calculate the shape of a character which is then plotted onto the screen. The routine is shown in figure 4. Note the way in which all the standard Anstrad characters are moved out of ROM into RAM (this meaning UDG 32 to UDG 252 inclusive) by the single command SYMBOL AFTER 32. Once in RAM they can be PEKED and reprinted in various sizes anywhere on the screen. It's an interesting program with which to experiment. You can use a machine-code routine at &9124 to find the address of any particular character. In Basic, type

```
CALL @0000.4.10.000,253,253,9124,000000,11
```

and the address of the matrix for character 253 will be

printed. If you're using SYMBOL AFTER, as you will be in the mix-mode program, remember that it will only work before HIMEM is altered by the MEMORY command. Thus the sequence of events should be:

1. Reset the Anstrad.
2. SYMBOL AFTER 32.
3. Set HIMEM at least as low as &9123.
4. Load &9124-&9147 and use CALL &9124 to find out where UDG 32 is.
5. Load mix-mode program, check that address for UDG 32 is correct.
6. Use the mix-mode program.

Screen saving

Saving and recalling the whole screen was a topic dealt with in Part 2 of this series. Its extravagant use of memory makes it of limited use, but there is no reason why smaller portions of the screen shouldn't be saved, recalled, and moved around. The routine to do this occupies approximately 100 bytes of machine code and, to make it useful, requires from between 1.5k to 3.5k of RAM as screen-storage space. This is considerably better than the 16k normally required to save the entire screen.

The Basic syntax of the screen-save command is:

```
CALL @0000.4.10.1.10.000
```

ADDR is the memory location reserved for holding the screen-portion. Y1 and Y2 are the top and bottom coordinates on the Y axis. X1 and X2 are the right and left coordinates on the X axis. D is the direction of transfer. Zero to save a screen-portion. Non-zero to load a screen-portion.

Remember that although the process is called screen-saving options of screen copy and screen move are also available. To copy a portion of screen it is only necessary to save it and then recall it to a different location. To move a portion, simply copy it and then, using a Basic window, erase the original. These techniques will be explained in more detail shortly.

The X coordinates used are the normal graphics-screen values from 0 to 639 but the Y values run from 0 to 192 (ie the true vertical resolution of the graphics screen, (The video-controller is responsible for 'stretching' the vertical resolution to a perceived 400 units). To keep the machine-code routine short, X and Y are not checked for

Number of zones required	1	2	3	4
Bytes available for each zone	1200	800	675	624
Addr 1 (chars=characters, 00' 0000)	784	580	540	184
Addr 2 (chars=characters, 00' 0000)	580	580	580	88
Addr 3 (chars=characters, 00' 0000)	580	580	580	112
Beginning of zone 1 (0)	---	---	---	---
Beginning of zone 2 (0)	---	---	---	---
Beginning of zone 3 (0)	---	---	---	---
Beginning of zone 4 (0)	---	---	---	---

Table of saving options. The screen-storage area is &1600 bytes.

```

      0010 0
      0011 0
      0020 0
      0030 0
      0040 0
      0050 0
      0060 0
      0070 0
      0080 0
      0090 0
      0100 0
      0110 0
      0120 0
      0130 0
      0140 0
      0150 0
      0160 0
      0170 0
      0180 0
      0190 0
      0200 0
      0210 0
      0220 0
      0230 0
      0240 0
      0250 0
      0260 0
      0270 0
      0280 0
      0290 0
      0300 0
      0310 0
      0320 0
      0330 0
      0340 0
      0350 0
      0360 0
      0370 0
      0380 0
      0390 0
      0400 0
      0410 0
      0420 0
      0430 0
      0440 0
      0450 0
      0460 0
      0470 0
      0480 0
      0490 0
      0500 0
      0510 0
      0520 0
      0530 0
      0540 0
      0550 0
      0560 0
      0570 0
      0580 0
      0590 0
      0600 0
      0610 0
      0620 0
      0630 0
      0640 0
      0650 0
      0660 0
      0670 0
      0680 0
      0690 0
      0700 0
      0710 0
      0720 0
      0730 0
      0740 0
      0750 0
      0760 0
      0770 0
      0780 0
      0790 0
      0800 0
      0810 0
      0820 0
      0830 0
      0840 0
      0850 0
      0860 0
      0870 0
      0880 0
      0890 0
      0900 0
      0910 0
      0920 0
      0930 0
      0940 0
      0950 0
      0960 0
      0970 0
      0980 0
      0990 0
      1000 0
      1010 0
      1020 0
      1030 0
      1040 0
      1050 0
      1060 0
      1070 0
      1080 0
      1090 0
      1100 0
      1110 0
      1120 0
      1130 0
      1140 0
      1150 0
      1160 0
      1170 0
      1180 0
      1190 0
      1200 0
      1210 0
      1220 0
      1230 0
      1240 0
      1250 0
      1260 0
      1270 0
      1280 0
      1290 0
      1300 0
      1310 0
      1320 0
      1330 0
      1340 0
      1350 0
      1360 0
      1370 0
      1380 0
      1390 0
      1400 0
      1410 0
      1420 0
      1430 0
      1440 0
      1450 0
      1460 0
      1470 0
      1480 0
      1490 0
      1500 0
      1510 0
      1520 0
      1530 0
      1540 0
      1550 0
      1560 0
      1570 0
      1580 0
      1590 0
      1600 0
      1610 0
      1620 0
      1630 0
      1640 0
      1650 0
      1660 0
      1670 0
      1680 0
      1690 0
      1700 0
      1710 0
      1720 0
      1730 0
      1740 0
      1750 0
      1760 0
      1770 0
      1780 0
      1790 0
      1800 0
      1810 0
      1820 0
      1830 0
      1840 0
      1850 0
      1860 0
      1870 0
      1880 0
      1890 0
      1900 0
      1910 0
      1920 0
      1930 0
      1940 0
      1950 0
      1960 0
      1970 0
      1980 0
      1990 0
      2000 0
      2010 0
      2020 0
      2030 0
      2040 0
      2050 0
      2060 0
      2070 0
      2080 0
      2090 0
      2100 0
      2110 0
      2120 0
      2130 0
      2140 0
      2150 0
      2160 0
      2170 0
      2180 0
      2190 0
      2200 0
      2210 0
      2220 0
      2230 0
      2240 0
      2250 0
      2260 0
      2270 0
      2280 0
      2290 0
      2300 0
      2310 0
      2320 0
      2330 0
      2340 0
      2350 0
      2360 0
      2370 0
      2380 0
      2390 0
      2400 0
      2410 0
      2420 0
      2430 0
      2440 0
      2450 0
      2460 0
      2470 0
      2480 0
      2490 0
      2500 0
      2510 0
      2520 0
      2530 0
      2540 0
      2550 0
      2560 0
      2570 0
      2580 0
      2590 0
      2600 0
      2610 0
      2620 0
      2630 0
      2640 0
      2650 0
      2660 0
      2670 0
      2680 0
      2690 0
      2700 0
      2710 0
      2720 0
      2730 0
      2740 0
      2750 0
      2760 0
      2770 0
      2780 0
      2790 0
      2800 0
      2810 0
      2820 0
      2830 0
      2840 0
      2850 0
      2860 0
      2870 0
      2880 0
      2890 0
      2900 0
      2910 0
      2920 0
      2930 0
      2940 0
      2950 0
      2960 0
      2970 0
      2980 0
      2990 0
      3000 0
      3010 0
      3020 0
      3030 0
      3040 0
      3050 0
      3060 0
      3070 0
      3080 0
      3090 0
      3100 0
      3110 0
      3120 0
      3130 0
      3140 0
      3150 0
      3160 0
      3170 0
      3180 0
      3190 0
      3200 0
      3210 0
      3220 0
      3230 0
      3240 0
      3250 0
      3260 0
      3270 0
      3280 0
      3290 0
      3300 0
      3310 0
      3320 0
      3330 0
      3340 0
      3350 0
      3360 0
      3370 0
      3380 0
      3390 0
      3400 0
      3410 0
      3420 0
      3430 0
      3440 0
      3450 0
      3460 0
      3470 0
      3480 0
      3490 0
      3500 0
      3510 0
      3520 0
      3530 0
      3540 0
      3550 0
      3560 0
      3570 0
      3580 0
      3590 0
      3600 0
      3610 0
      3620 0
      3630 0
      3640 0
      3650 0
      3660 0
      3670 0
      3680 0
      3690 0
      3700 0
      3710 0
      3720 0
      3730 0
      3740 0
      3750 0
      3760 0
      3770 0
      3780 0
      3790 0
      3800 0
      3810 0
      3820 0
      3830 0
      3840 0
      3850 0
      3860 0
      3870 0
      3880 0
      3890 0
      3900 0
      3910 0
      3920 0
      3930 0
      3940 0
      3950 0
      3960 0
      3970 0
      3980 0
      3990 0
      4000 0
      4010 0
      4020 0
      4030 0
      4040 0
      4050 0
      4060 0
      4070 0
      4080 0
      4090 0
      4100 0
      4110 0
      4120 0
      4130 0
      4140 0
      4150 0
      4160 0
      4170 0
      4180 0
      4190 0
      4200 0
      4210 0
      4220 0
      4230 0
      4240 0
      4250 0
      4260 0
      4270 0
      4280 0
      4290 0
      4300 0
      4310 0
      4320 0
      4330 0
      4340 0
      4350 0
      4360 0
      4370 0
      4380 0
      4390 0
      4400 0
      4410 0
      4420 0
      4430 0
      4440 0
      4450 0
      4460 0
      4470 0
      4480 0
      4490 0
      4500 0
      4510 0
      4520 0
      4530 0
      4540 0
      4550 0
      4560 0
      4570 0
      4580 0
      4590 0
      4600 0
      4610 0
      4620 0
      4630 0
      4640 0
      4650 0
      4660 0
      4670 0
      4680 0
      4690 0
      4700 0
      4710 0
      4720 0
      4730 0
      4740 0
      4750 0
      4760 0
      4770 0
      4780 0
      4790 0
      4800 0
      4810 0
      4820 0
      4830 0
      4840 0
      4850 0
      4860 0
      4870 0
      4880 0
      4890 0
      4900 0
      4910 0
      4920 0
      4930 0
      4940 0
      4950 0
      4960 0
      4970 0
      4980 0
      4990 0
      5000 0
      5010 0
      5020 0
      5030 0
      5040 0
      5050 0
      5060 0
      5070 0
      5080 0
      5090 0
      5100 0
      5110 0
      5120 0
      5130 0
      5140 0
      5150 0
      5160 0
      5170 0
      5180 0
      5190 0
      5200 0
      5210 0
      5220 0
      5230 0
      5240 0
      5250 0
      5260 0
      5270 0
      5280 0
      5290 0
      5300 0
      5310 0
      5320 0
      5330 0
      5340 0
      5350 0
      5360 0
      5370 0
      5380 0
      5390 0
      5400 0
      5410 0
      5420 0
      5430 0
      5440 0
      5450 0
      5460 0
      5470 0
      5480 0
      5490 0
      5500 0
      5510 0
      5520 0
      5530 0
      5540 0
      5550 0
      5560 0
      5570 0
      5580 0
      5590 0
      5600 0
      5610 0
      5620 0
      5630 0
      5640 0
      5650 0
      5660 0
      5670 0
      5680 0
      5690 0
      5700 0
      5710 0
      5720 0
      5730 0
      5740 0
      5750 0
      5760 0
      5770 0
      5780 0
      5790 0
      5800 0
      5810 0
      5820 0
      5830 0
      5840 0
      5850 0
      5860 0
      5870 0
      5880 0
      5890 0
      5900 0
      5910 0
      5920 0
      5930 0
      5940 0
      5950 0
      5960 0
      5970 0
      5980 0
      5990 0
      6000 0
      6010 0
      6020 0
      6030 0
      6040 0
      6050 0
      6060 0
      6070 0
      6080 0
      6090 0
      6100 0
      6110 0
      6120 0
      6130 0
      6140 0
      6150 0
      6160 0
      6170 0
      6180 0
      6190 0
      6200 0
      6210 0
      6220 0
      6230 0
      6240 0
      6250 0
      6260 0
      6270 0
      6280 0
      6290 0
      6300 0
      6310 0
      6320 0
      6330 0
      6340 0
      6350 0
      6360 0
      6370 0
      6380 0
      6390 0
      6400 0
      6410 0
      6420 0
      6430 0
      6440 0
      6450 0
      6460 0
      6470 0
      6480 0
      6490 0
      6500 0
      6510 0
      6520 0
      6530 0
      6540 0
      6550 0
      6560 0
      6570 0
      6580 0
      6590 0
      6600 0
      6610 0
      6620 0
      6630 0
      6640 0
      6650 0
      6660 0
      6670 0
      6680 0
      6690 0
      6700 0
      6710 0
      6720 0
      6730 0
      6740 0
      6750 0
      6760 0
      6770 0
      6780 0
      6790 0
      6800 0
      6810 0
      6820 0
      6830 0
      6840 0
      6850 0
      6860 0
      6870 0
      6880 0
      6890 0
      6900 0
      6910 0
      6920 0
      6930 0
      6940 0
      6950 0
      6960 0
      6970 0
      6980 0
      6990 0
      7000 0
      7010 0
      7020 0
      7030 0
      7040 0
      7050 0
      7060 0
      7070 0
      7080 0
      7090 0
      7100 0
      7110 0
      7120 0
      7130 0
      7140 0
      7150 0
      7160 0
      7170 0
      7180 0
      7190 0
      7200 0
      7210 0
      7220 0
      7230 0
      7240 0
      7250 0
      7260 0
      7270 0
      7280 0
      7290 0
      7300 0
      7310 0
      7320 0
      7330 0
      7340 0
      7350 0
      7360 0
      7370 0
      7380 0
      7390 0
      7400 0
      7410 0
      7420 0
      7430 0
      7440 0
      7450 0
      7460 0
      7470 0
      7480 0
      7490 0
      7500 0
      7510 0
      7520 0
      7530 0
      7540 0
      7550 0
      7560 0
      7570 0
      7580 0
      7590 0
      7600 0
      7610 0
      7620 0
      7630 0
      7640 0
      7650 0
      7660 0
      7670 0
      7680 0
      7690 0
      7700 0
      7710 0
      7720 0
      7730 0
      7740 0
      7750 0
      7760 0
      7770 0
      7780 0
      7790 0
      7800 0
      7810 0
      7820 0
      7830 0
      7840 0
      7850 0
      7860 0
      7870 0
      7880 0
      7890 0
      7900 0
      7910 0
      7920 0
      7930 0
      7940 0
      7950 0
      7960 0
      7970 0
      7980 0
      7990 0
      8000 0
      8010 0
      8020 0
      8030 0
      8040 0
      8050 0
      8060 0
      8070 0
      8080 0
      8090 0
      8100 0
      8110 0
      8120 0
      8130 0
      8140 0
      8150 0
      8160 0
      8170 0
      8180 0
      8190 0
      8200 0
      8210 0
      8220 0
      8230 0
      8240 0
      8250 0
      8260 0
      8270 0
      8280 0
      8290 0
      8300 0
      8310 0
      8320 0
      8330 0
      8340 0
      8350 0
      8360 0
      8370 0
      8380 0
      8390 0
      8400 0
      8410 0
      8420 0
      8430 0
      8440 0
      8450 0
      8460 0
      8470 0
      8480 0
      8490 0
      8500 0
      8510 0
      8520 0
      8530 0
      8540 0
      8550 0
      8560 0
      8570 0
      8580 0
      8590 0
      8600 0
      8610 0
      8620 0
      8630 0
      8640 0
      8650 0
      8660 0
      8670 0
      8680 0
      8690 0
      8700 0
      8710 0
      8720 0
      8730 0
      8740 0
      8750 0
      8760 0
      8770 0
      8780 0
      8790 0
      8800 0
      8810 0
      8820 0
      8830 0
      8840 0
      8850 0
      8860 0
      8870 0
      8880 0
      8890 0
      8900 0
      8910 0
      8920 0
      8930 0
      8940 0
      8950 0
      8960 0
      8970 0
      8980 0
      8990 0
      9000 0
      9010 0
      9020 0
      9030 0
      9040 0
      9050 0
      9060 0
      9070 0
      9080 0
      9090 0
      9100 0
      9110 0
      9120 0
      9130 0
      9140 0
      9150 0
      9160 0
      9170 0
      9180 0
      9190 0
      9200 0
      9210 0
      9220 0
      9230 0
      9240 0
      9250 0
      9260 0
      9270 0
      9280 0
      9290 0
      9300 0
      9310 0
      9320 0
      9330 0
      9340 0
      9350 0
      9360 0
      9370 0
      9380 0
      9390 0
      9400 0
      9410 0
      9420 0
      9430 0
      9440 0
      9450 0
      9460 0
      9470 0
      9480 0
      9490 0
      9500 0
      9510 0
      9520 0
      9530 0
      9540 0
      9550 0
      9560 0
      9570 0
      9580 0
      9590 0
      9600 0
      9610 0
      9620 0
      9630 0
      9640 0
      9650 0
      9660 0
      9670 0
      9680 0
      9690 0
      9700 0
      9710 0
      9720 0
      9730 0
      9740 0
      9750 0
      9760 0
      9770 0
      9780 0
      9790 0
      9800 0
      9810 0
      9820 0
      9830 0
      9840 0
      9850 0
      9860 0
      9870 0
      9880 0
      9890 0
      9900 0
      9910 0
      9920 0
      9930 0
      9940 0
      9950 0
      9960 0
      9970 0
      9980 0
      9990 0

```

Figure 5: Assembly source code

legality. Furthermore, the surface area of the screen-rectangle delimited by XR, XL, YT, YB is not calculated to ascertain whether it exceeds the amount of memory set aside to receive it. The user must either enter accurate figures to begin with, or else check them in Basic before calling @MOV. In the program presented here, the maxi-screen size of the saved-area has been set to fairly reasonable value, 4160 (5760 bytes), and the user has a number of saving options, which are shown in the table, figure 4.

To use this table first decide what screen mode you wish to work in and how many different saved screen-ports will be needed. If all the saves involve Moves and Copies then the storage RAM won't become blocked because the same RAM can be used over and over again. If, however, the storage of a screen-portion is to be a long-term affair, then the table will establish exactly what you can and can't do. Storing two screen-ports in mode 0 will give a maximum of 90 character-squares per save (arranged in any rectangular shape). If this is insufficient for your needs, you will have to make do with saving just one portion, which gives you a maximum of 180 characters to play with. Alternatively, you may change to a higher-resolution mode. This won't produce more bytes of storage space, but it will allow more character-squares to be saved at one time.

Special effects

Three points are immediately illustrated by the demonstration program. Firstly, the speed of operation of the save/recall cycle is directly proportional to the size

of the area being moved. This can be shown simply by adjusting the starting values of XR, XL, YT, and YB on line 116, but the same effect can be put into action by reducing the size of the recall instruction. For example, line 116 can be left alone and line 219 amended so that the XL value reads XL-32. The copied screen-ports are now reduced in size and come onto the screen faster. This brings us to point number two: material not screened is simply left in the RAM storage area. However, XL-32 increases the size of the copy-area and places a certain amount of garbage onto the copy-portion. XR-32 has the same effect as XL-32 and XR-32 has the same effect as XL-32. Point three concerns YT and YB. Changing either YT to YT-1 or YB to YB-1 causes the copied portions of text to slant. The text is readable in modes 2 and 1, but not really so in mode 0. YT-1 or YB-1 causes the slant to go the other way, and a + or - 2 causes a much coarser slant. A pointless exercise on text, but very interesting on graphics and colour displays! As always, make sure that the computed values of XR, XL, YT and YB stay within the legal range, otherwise the machine code program could crash.

The source code for the screen save routine is reproduced in figure 5. It's essentially two nested loops, one for the screen's X axis and one for its Y axis. The direction of transfer is held in the C register and the current screen mode (found by the call to SCR GET MODE) is held in the B register. At the heart of the inner loop is the important call to SCR DOT POSITION, which correctly calculates VDU RAM addresses from the X-Y screen coordinates.