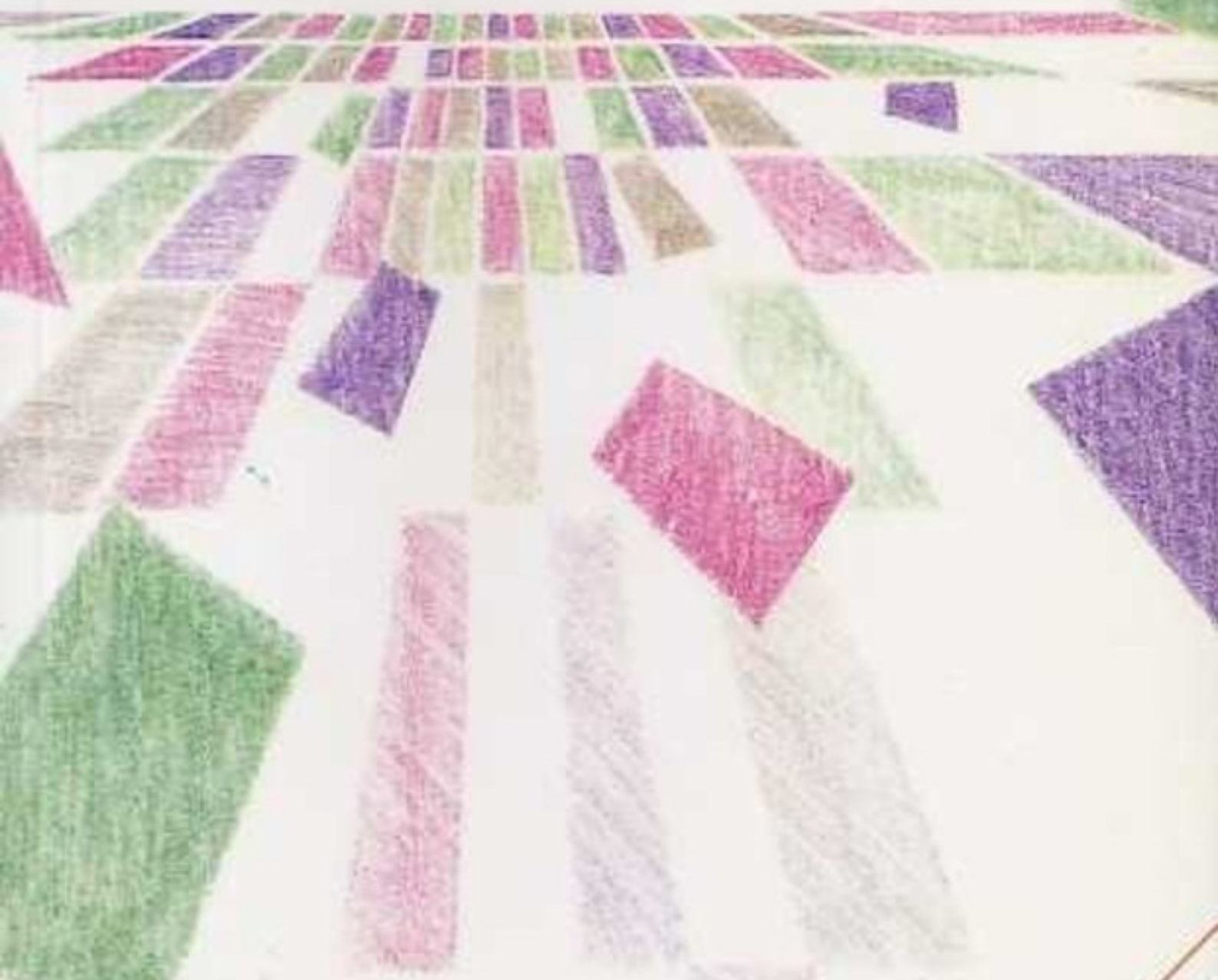
St Michael

COMPUTER PROGRAMS

START TO PROGRAM

A Beginner's Guide



START TO PROGRAM THE ZX SPECTRUM
A Beginner's Guide

Richard and David
GRAVES

St Michael

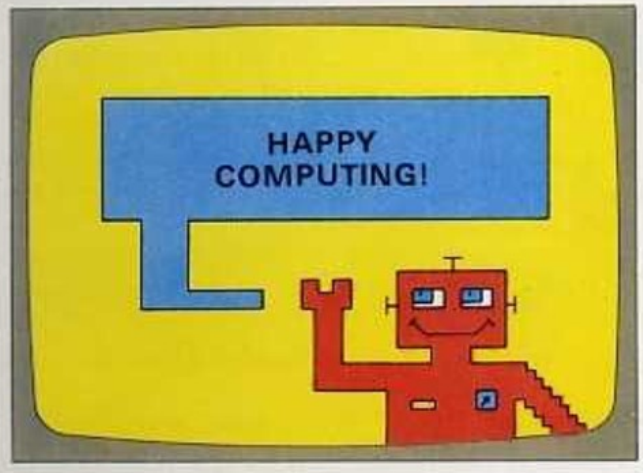
Contents

Introduction

Often people buy computers with high hopes, only to discover that many of the standard books introducing BASIC (the computer language) are dull and complicated, and that their computer becomes just an expensive toy. Now you can avoid that disappointment with this Start-to-Program Kit which includes not only a book, but two tapes with pre-recorded programs on them.

Book and tapes have been carefully designed to complement each other. Start with the tapes, and the book becomes a useful work of reference. Start with the book, and the tapes give you valuable additional experience, enabling you to see 'on screen' the effects of small changes to programs. Using this Kit, you will soon be writing your own programs.

Richard and
David Graves



1 Making a Start

How to give a simple instruction

The computer can only do what you tell it to do. You tell it to do something by typing instructions on the keyboard. Everything that you type appears on the screen. Each word, letter or symbol appears at a place on the screen where there is a flashing letter called the *cursor*. The cursor letter can be a [K], an [L], an [E], a [G] or a [C].

■ Problem 1: When the computer is switched on you cannot see the cursor, but only the words:

```
© 1982 Sinclair Research Ltd
```

near the bottom of the screen.

☑ Press the ENTER key, which is just above the BREAK SPACE key on the bottom right of the keyboard. Now you can see the flashing [K] cursor. Now, try an instruction. 'Tell me' is an instruction. How about:

```
Tell me  "Look behind you"
```

But as soon as you press the key marked T, you run into trouble. Instead of typing T on the screen, it has typed:

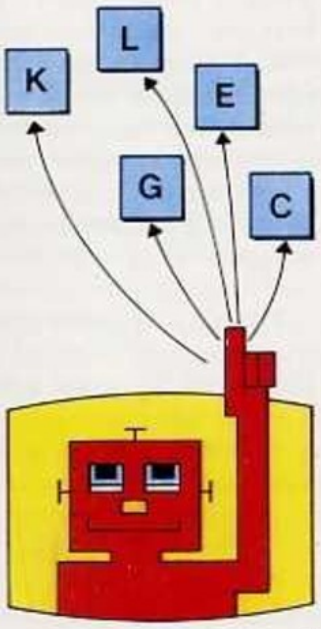
```
RANDOMIZE
```

and after it you can see the flashing [L] cursor.

■ Problem 2: It is easy to make mistakes when typing in an instruction.

☑ Look at the bottom left of the keyboard and find the key marked CAPS SHIFT. Now look at the top right of the keyboard and find the zero key. The zero key is marked 0. Do not muddle it up with the O, which is just the capital of the letter o. From now on, all zeros in this book will be shown as 0 to help you to remember. Above the 0 key you will see the word DELETE, which means to rub out or get rid of. Hold one finger on the CAPS SHIFT key. Keep holding it down while you press 0. Every time you do this, you will get rid of another word, letter or symbol to the left of the cursor. So, if you make a mistake and wish to delete it, press CAPS SHIFT and 0 until the mistake has vanished and then type in the correct instruction. Now you can delete the word RANDOMIZE, but how can you make the computer tell you to look behind you?

► *Before you can type anything into the Spectrum Computer, you need to see the cursor (which flashes on and off) on the screen. The cursor can be any one of these five letters. Each cursor letter has a different use.*



Problem 3: The computer only understands instructions when they are typed in a special language called BASIC.

Luckily, the Spectrum Computer gives you enough help to make this language fairly easy to learn. For example, it knows that at the start of an instruction there must be a *Keyword* in BASIC. So at the beginning of each new instruction, it shows you the flashing [K] cursor. (This is called being in [K] mode.) Keywords are simply words in BASIC which are often used.

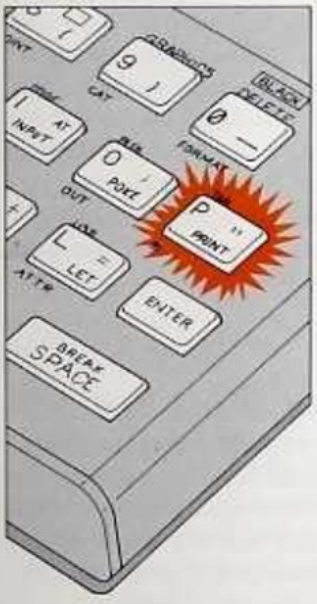
If you press a letter key when the computer is in [K] mode, you will not see that letter on the screen. Instead, you will see the Keyword which is shown *in white* on the bottom of that letter key. So when you pressed T, the computer printed the Keyword RANDOMIZE. (It just says RAND on the bottom of the T key, because there is not enough room to show the whole of the Keyword.) If you look at all the letter keys you will see that there is no TELL ME Keyword. TELL ME is not an instruction in BASIC. But on the P key there is the Keyword PRINT which you can use instead. Whenever you want the computer to print a message on the screen, you press the Keyword PRINT and then type " (inverted commas). Then you type the message. Then you type another ". So you need to type:

PRINT "Look behind you"

Look at the [K] cursor, and remember that you are in [K] mode to begin with. Now, press the P key to get the Keyword PRINT. If you look at the screen you can now see

PRINT

and after it you can see a new flashing cursor. You are

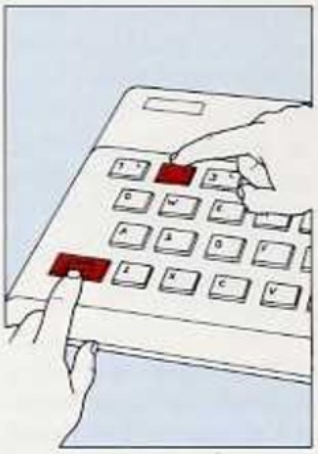


◀ *You need the instruction* PRINT *to make the computer 'print' your messages onto the screen.*

not in [K] mode any more. The computer is being helpful again! It knows that once you have begun your instruction with a Keyword, you will not want to use another one of the Keywords at the bottom of the letter keys. The new flashing cursor is an [L]. You are now in [L] mode. When you are in [L] mode, you can type *Letters*. If you want them to be lower case letters (small letters), then you simply press the letter keys. But if you want any letter to be a CAPITAL letter, then you keep one finger on the CAPS SHIFT key while you press the letter key.

(Later on in this book, you may need to type a message with all the letters in CAPITAL letters. So to avoid having to keep a finger always on the CAPS SHIFT key, you keep one finger on the CAPS SHIFT key and then press once on the number 2 key at the top of the keyboard. Above this key it says CAPS LOCK in white letters. You are now in [C] or CAPITAL mode.

There will be no change to the flashing [K] cursor, but the flashing [L] cursor will be replaced by a flashing [C] cursor. Any letters that you type in this mode will be *Capital* letters, and you can take your finger off the CAPS SHIFT key while you type your message. Now, to come back to the [L] mode, just put one finger on the CAPS SHIFT key and press once on the 2 key again. Delete any letters you may have typed while you were



◀ *If you want to type all your messages in CAPITAL letters, you keep one finger on the CAPS SHIFT key while you press once on the number 2 key.*

practising with the [C] mode, so that all that remains on the screen is the Keyword PRINT.)

When you want spaces between letters or words in a message (remember that the message is the part *inside* the sets of inverted commas), then you simply press the BREAK SPACE key at the bottom right of the keyboard. If the computer wants to have spaces between any words or symbols *outside* the inverted commas it will put them in for itself.

? Problem 4: The next thing you want to type after PRINT is not a letter, but ". You will find the inverted commas on the top right-hand corner of the P key, *in red*.

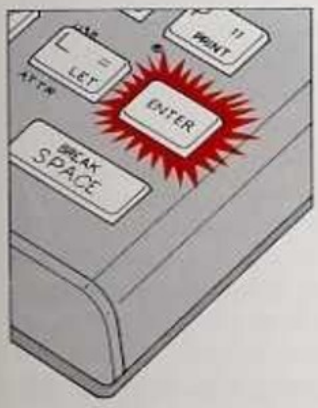
✓ When you are in [K] or in [L] (or [C]) mode, and you want to put in any of the symbols or words shown in red on the letter keys or on the number keys, then you keep one finger on the SYMBOL SHIFT key (at the bottom right of the keyboard, next to the BREAK SPACE key), while you press the letter or number key. So to type " you keep one finger on the SYMBOL SHIFT key while you press the P key. Now you know all you need to know in order to finish typing:

```
PRINT  "Look behind you"
```

Remember, if you make any mistakes you know how to delete them and try again.

When you have typed in an instruction nothing will happen until you press the key marked ENTER which is near the bottom right of the keyboard. So press ENTER. Now you can read your message near the top of the screen:

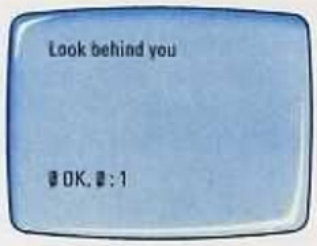
```
Look behind you
```



◀ *Each time you type an instruction, you must press the ENTER key. Then the computer will do what you have asked, and it will print a message telling you your instruction has been received.*

At the same time a message will appear, at the bottom

left of the screen, known as a report. It explains why the computer stopped doing something – either because it has finished or because something went wrong.



◀ *A 0 OK report means that everything was successfully completed. The two numbers that follow show the line and place at which the program stopped.*

This tells you that everything is okay! There is now no cursor on the screen, but you can bring it back by pressing ENTER or you can simply type a new Keyword. However, if you have made a mistake which you did not notice, and pressed ENTER when your instruction did not make sense in BASIC, then your message will not have appeared at the top of the screen. Instead, your instruction will have stayed where it is and a flashing question-mark will tell you that the computer has found a mistake. This question-mark usually appears at the place where the mistake has been made. Then you have to correct the line by deleting the mistake in the usual way and pressing ENTER again. To see what it looks like, try typing:

PRINT "Today is Thursday

and press ENTER. There should have been inverted commas at the end of the line, so the instruction has stayed where it is and there is a flashing question-mark after Thursday. This mistake is at the end of the line, so nothing needs deleting. Just add the inverted commas and the flashing question-mark will disappear. The line is now okay, and you can ENTER it.

If the mistake is a very bad one, the computer may

decide that you had better start the line all over again. Try typing:

PRINT I am well

and press ENTER. You have left out both sets of inverted commas this time, and the computer has deleted the whole line and printed a message at the bottom of the screen, saying:

2 Variable not found, 0:1

This time you must press ENTER to clear the screen and then start again. Now you have learned how to give the computer a simple instruction. Try some more instructions of your own, using the PRINT instruction. For example, tell the computer to print the name of a country you would like to visit.

◀ *If you try to ENTER an instruction with a bad mistake in it, the computer will rub out your line and instead, will print a message saying that something is wrong.*

How to give a list of instructions

One instruction on its own is rather boring. If you want the computer to do something more interesting, you need to give it a list of instructions.

? Problem 5: The computer obeys every instruction as soon as you press ENTER. How can you make it wait until you have given it a list of instructions?

☑ Type a number before each instruction, or line. The computer will then store each numbered line in its memory. Start with 10 for the first line, 20 for the second, and so on. You can type numbers when you are in [K] or [L] mode simply by pressing the number keys at the top of the keyboard. If you begin an instruction with a number, the computer will stay in [K] mode. It knows that a number at the beginning of an instruction is a line number and it is still waiting for a Keyword.

Now, type in this list of instructions, *and remember to press* ENTER *after each line*. If each line is correctly typed, it will then appear at the top of the screen, where your list of instructions will slowly build up. At the same time, the 'OK' message will appear at the bottom left of the screen. Press ENTER to begin with, to start with a clear screen.

Do not worry if you are typing an instruction which will not all fit onto one line of the screen. Do not press ENTER when there is no room left on the line – just carry on typing. Some of the words will go over onto the next line on the screen, but the computer will treat it all as one line, because you have not yet pressed ENTER.

The . (full stop) and ? (question-mark) are in red, on the letter M and the letter C keys; the - (hyphen) and ! (exclamation-mark) are on the letter J and the number 1 keys. You will need the SYMBOL SHIFT key for all of these.

```
10 PRINT "I am a ZX Spectrum."
20 PRINT "I hope you enjoy using me."
30 PRINT "Do you know what BASIC
means?"
40 PRINT "Beginners All-purpose"
50 PRINT "Symbolic Instruction Code!"
60 STOP
```

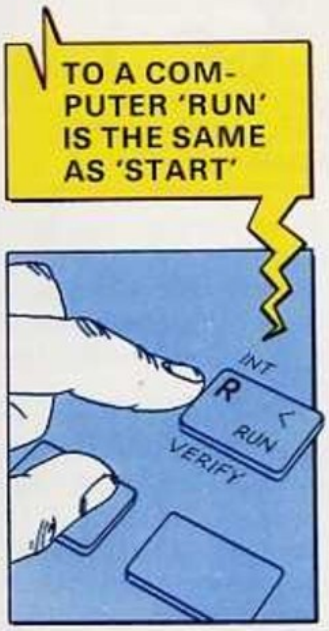

The Keyword STOP at line 60 tells the computer that the list of instructions has come to an end. It is on the A key in red, and so you must press A while keeping a finger on the SYMBOL SHIFT key.

? Problem 6: Now that you have typed in a list of instructions, how do you make the computer obey the instructions?

☑ Press the Keyword RUN (on the R key), and then press ENTER. A list of instructions is called a *program*. Now you have entered your first program and RUN it. Near the bottom of the screen you will see a message telling you that the computer has reached the end of the program at line 60:

```
9 STOP statement, 60:1
```

▶ *When you have typed in a list of instructions, press* RUN *to make the computer carry these instructions out and then press* ENTER. *You will find the* RUN *keyword on the R key.*



If you want to look at the list of instructions again, then press the Keyword LIST (on the K key) and press ENTER. If you want to run the program again, simply press RUN again, and press ENTER.

❓ Problem 7: How can you get rid of a program without switching off the computer?

☑ Press the Keyword NEW (on the A key) and press ENTER. Now, press NEW and press ENTER and write a program of your own, using the PRINT instruction. For example, you could write a program which will print the names of your favourite books. Or write a program which will print a list of five wild animals.

Three programs to show what the computer can do

Type out these programs and see what happens. They are to show you what the computer can do. (Helpful hints on finding keys are given just after each program.)

ONE

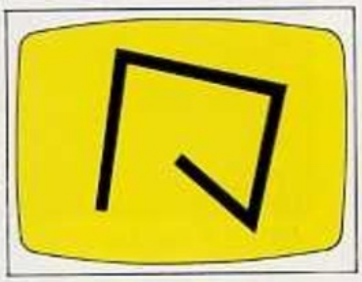
```
10 PLOT 104,48
20 DRAW 103,0
30 DRAW 0,103
40 DRAW -103,0
50 DRAW 0,-103
```

You will find the Keyword PLOT on the Q key, and the Keyword DRAW on the W key. The , (comma) is a red symbol on the N key. For the — (minus sign) use the hyphen on the J key. Now, RUN the program. If you have typed in the lines correctly, you will see a square box on the screen.

❓ Problem 8: When you run the program, if your square is not complete, or looks funny, you have probably typed in some numbers wrongly.

☑ Press LIST and look carefully at the lines. When you find a mistake, you must type out that whole line again and press ENTER. The computer will automatically replace the wrong line with the new one you have just typed. Now run the program again and, if there are no more mistakes, you will have a complete square.

► *If you get some of the numbers in program 'One' wrong, you will not get a proper square on your screen. Always be very careful to copy lines exactly.*



Now, press LIST again and add these lines to colour in the square with magenta (a light purple):

```
60 INK 3
70 FOR x=3 TO 15
80 FOR y=13 TO 25
90 PRINT AT x,y;"■"
100 NEXT y
110 NEXT x
120 STOP
```

You will find the Keyword FOR on the F key, and the Keyword NEXT on the N key. The Keyword AT is in red on the I key. The Keyword TO is in red on the F key. The = (equals) sign is a red symbol on the L key. The ; (semi-colon) sign is a red symbol on the O key. You can manage all those quite easily. But what about INK in line 60, and the black square ■ in line 90?

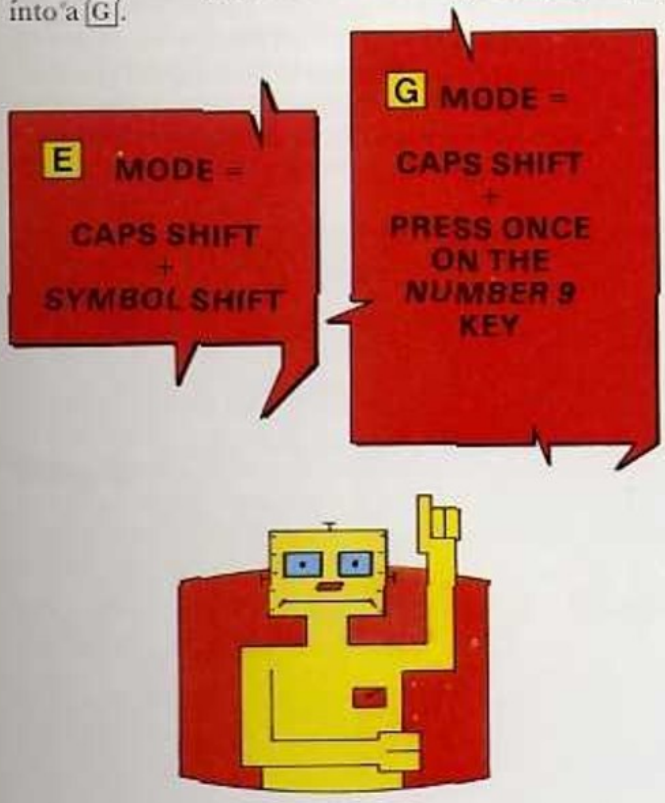
Problem 9: INK in line 60 is not on a key at all. It is shown in red, *below* the X key. How can you type it into a program?

☑ There are many *Extra* words or symbols above the letter keys (in green) and below them (in red). When you want to type one of these words or symbols, you must first change the cursor into [E] mode. You do this by keeping one finger pressed on the CAPS SHIFT key, while with another finger you press the SYMBOL SHIFT key once. Now the flashing [K] or [L] cursor changes into a flashing [E] cursor, to show that you are in [E] mode. (Often called *Extended* mode.) Now that you are in [E] mode, if you press a letter key you will type the word or symbol which you can see *in green* above it. And if you press a letter key while keeping one finger pressed on the CAPS SHIFT key, you will type the word or symbol which you can see *in red* below it. So to type INK, you must get into [E] mode, and press the X key while keeping a finger pressed on the CAPS SHIFT key. You will only stay in [E] mode for one press of a letter key. Then you will go straight to [L] mode, and if you want to get back into [E] mode you will have to begin again.

Problem 10: The square symbol ■ in line 90 is a drawing, or *Graphics* symbol. It is on the 8 key. How can you type it, or the other Graphics symbols on keys 1 to 7, into a program?

☑ To type a Graphics symbol, you must first

change into [G] mode. You do this by keeping one finger pressed on the CAPS SHIFT key, while you press once on the 9 key. To remind you that the 9 key is the one which you need in order to get into [G] mode for Graphics, you can see that the word GRAPHICS is in white just above the 9 key. As soon as you are in [G] mode, the flashing cursor will change from [K] or [L] into a [G].



Now, before you type one of the Graphics symbols, you must decide if you want to type it exactly as shown on the key, with the white areas in white and the black in black; or whether you want to type the *opposite* of what is shown, with the white areas in black

and the black areas in white. To type it as shown, simply press the correct number key. To type the opposite, keep one finger pressed on the CAPS SHIFT key while pressing the number key. So, to type ■ you must change into [G] mode, and then keep one finger pressed on the CAPS SHIFT key while pressing the 8 key. You will stay in [G] mode until you tell the computer to get out of [G] mode. To do this, keep one finger pressed on the CAPS SHIFT key while you press the 9 key again. Then you will be in [K] or [L] mode.

Now run the program. You will see that the square will fill up with colour. Press NEW and press ENTER when you want to get rid of this program. (You are probably finding that the Spectrum keyboard is quite hard work to begin with. Don't worry, it becomes much easier with use.)

◀ *You can use the INK instruction with the Graphics symbols on keys 1 to 8 to colour in parts of the screen.*

TWO

```
10  LET  x=INT (RND*22)
20  LET  y=INT (RND*32)
30  LET  z=INT (RND*7)
40  BORDER  1
50  INK  z
60  PRINT AT x,y;"■"
70  PAUSE  5
80  GOTO  10
90  STOP
```


You will find the Keyword LET on the L key, and the Keyword BORDER on the B key. The Keyword PAUSE is on the M key. GOTO is on the G key. INT is in green *above* the R key, and RND is *above* the T key, so you need to get into [E] mode to type them. The symbol * is a red symbol on the B key, and the (and) (bracket) symbols are in red on the 8 and 9 keys. (Remember to get into [E] mode and keep your finger pressed on the CAPS SHIFT key to type INK.)

This program will print an endless series of coloured squares on a white background, surrounded by a dark blue border. The first time that you run it, you will notice that no squares appear in a narrow

▲ *Because of the instruction GOTO 10, this program will go round in a loop forever, printing different coloured squares on a white background.*

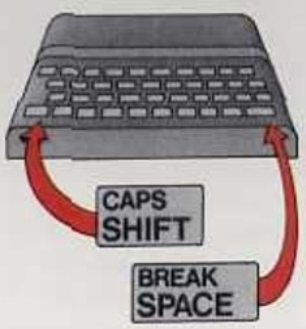
gap at the bottom of the white area. You will see a reason for this later. The program goes on and on because when it reaches line 80 the computer reads the instruction GOTO 10. This tells it to jump at once to line 10, and so it goes round and round in circles. It never reaches line 90. This is known as a *loop*.

? Problem 11: How can you stop the program running, without switching off the computer?

✓ Keeping one finger pressed on the CAPS SHIFT key, press the BREAK SPACE key. The program will stop running, and a message will appear in the gap at the bottom of the screen saying something like:

```
L BREAK into program, 20:1
```

The number 20 in this message is the number of the line which the computer had read and obeyed just before you broke into the program. It could have been one of the other line numbers. Now you can run the program again, and a different pattern will gradually appear. Or, you can go on with the same pattern. To do this, you type the Keyword CONTINUE (CONT on the C key) and press ENTER. The computer will then carry on, beginning at the line after the one mentioned in the BREAK message.

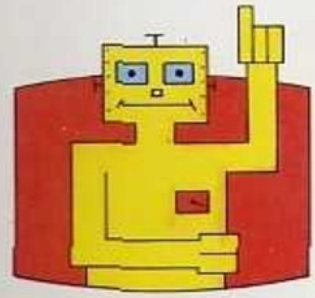


◀ *If you want to break out of a program while it is running, you press both these keys at the same time.*

You will see that as soon as you broke into the program, the narrow white gap near the bottom of the screen disappeared, so whether you run the program again, or continue it, it looks better. To get rid of this program press the CAPS SHIFT and BREAK SPACE keys, then press NEW, then ENTER.

THREE

```
10 BEEP 1,0
20 BEEP .5,2
30 BEEP .5,0
40 BEEP 1,5
50 BEEP 1,4
60 BEEP .5,2
70 BEEP .5,4
80 BEEP 1,5
90 GOTO 10
100 STOP
```



You will find the Keyword BEEP below the Z key. So you will need to get into [E] mode, and then keep one finger on the CAPS SHIFT key while you press the Z key. (Remember the full stop symbol is on the M key.)

A reminder

You have been finding your way around the keyboard. You know:

1) that you begin in K mode, and can then type the *numbers on the number keys and the Keywords in* white on the letter keys;

2) that you then go into L mode, and can then type the numbers on the number keys and the letters on the letter keys (and that if you want to type a lot of letters in CAPITALS, you can go into C mode);

3) how to type the red words or symbols *on* the letter and number keys when you are in K or L mode, by pressing them while keeping a finger pressed on the SYMBOL SHIFT key;

4) how to go into E mode by pressing the SYMBOL SHIFT key once, while keeping a finger pressed on the CAPS SHIFT key. You can then type the green words *above* the letter keys by pressing the keys; and the red words *below* them by pressing the keys while keeping a finger pressed on the CAPS SHIFT key;

5) how to get into or out of G mode, by pressing the 9 key while keeping a finger pressed on the CAPS SHIFT key; and that while you are in G mode you can type the Graphics symbols shown on the number keys by simply pressing the number keys. To get the opposite of the Graphics symbols shown on the keys, you press the number keys while keeping a finger pressed on the CAPS SHIFT key.

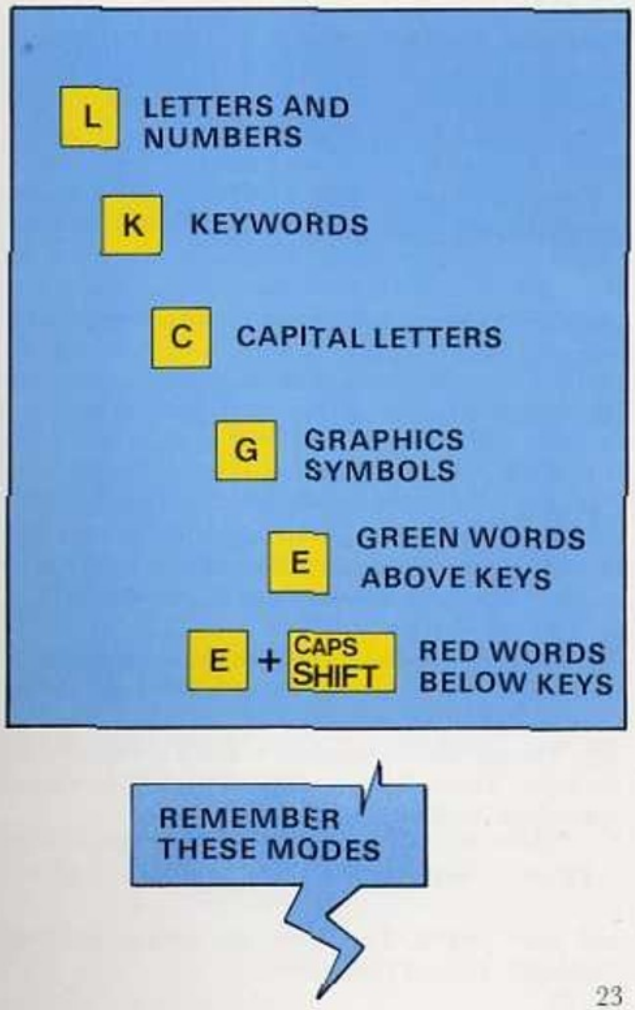
You have also learned:

6) how to use nine useful instructions: PRINT, SPACE, STOP, RUN, LIST, NEW, GOTO, BREAK and CONTINUE;

7) how to put your instructions into the computer by pressing the ENTER key;

8) how to delete mistakes, using the 0 and CAPS SHIFT keys;

9) how to write a simple program, using the PRINT instruction, and how to make the computer RUN it. (You will learn more about the instructions PLOT, DRAW, INK, LET, BORDER, INT, RND, PAUSE and BEEP later on in the book.)



2 Making the Computer Work for You

Simple mathematics

First of all, find the + (plus) and the − (minus) signs. The + is a red symbol on the K key, so remember that to type + you will need to keep one finger on the SYMBOL SHIFT key while you press the K key. The − is a red symbol on the J key.

Here is a simple sum: 3+4. If you type this on the computer and press ENTER, nothing will happen except that a flashing question-mark appears after the + sign. You will have to delete this line. But if you *give the correct instruction, the computer will do the* sum very quickly, and print the answer near the top of the screen, while giving its usual 'OK' message at the bottom. Try typing:

```
PRINT  3+4
```

and press ENTER. The answer appeared at once. But then you could have worked it out in your head just as *quickly. Here is a more difficult sum:* 4×8989898.

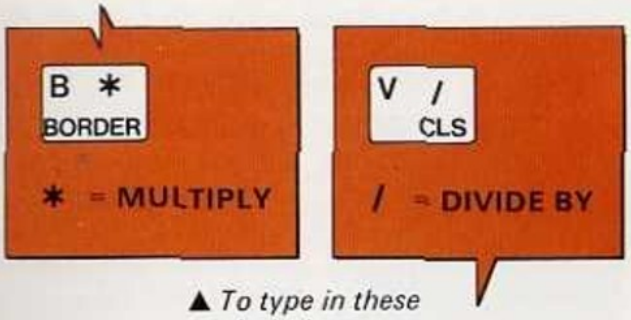
? Problem 12: It was easy to find the plus and minus signs; but where are the × (times) and the ÷ (divided by) signs?

✓ The sign for × (times) is *. It is a red symbol on the B key. The sign for ÷ (divided by) is /. It is a red symbol on the V key. Now, type:

```
PRINT  4*8989898
```

and press ENTER. In a flash, the answer appears: 35959592. Try out these sums:

5×12345678
1471365÷9



▲ *To type in these symbols, keep one finger on the SYMBOL SHIFT key while you press the B or V keys.*

If you have a sum with more than one part, such as 2×5−4÷2, the computer will work out the multiplication and division first. It would work out that 2×5=10; that 4÷2=2; and that 10−2=8, so the answer is 8. But if you place parts of a long sum inside brackets (the red symbols on the 8 and 9 keys), it will work out the parts *inside the brackets* first. So 2×(5−4)÷2 would be worked out as 5−4=1, then 2×1=2, then 2÷2=1, so the answer is 1. Try checking this out by typing the following statements, and pressing ENTER after each one:

```
PRINT  2*5-4/2
PRINT  2*(5-4)/2
```

To put a decimal point in a sum, you can use the full stop on the M key. Try:

```
PRINT  35*7.25
```

and press ENTER. The answer should be 253.75. Now try out some sums of your own. For example, divide the number of pupils in your school by the number of teachers to find out how many pupils there are to each teacher.

◀ *If you type in a sum with more than one part, the computer will work out the multiplication and division first, so the answer here is 2.*

◀ *But if you put part of the sum inside brackets, it will work out whatever is inside the brackets first, giving a new answer of 1 in this case.*

Number variables

(Doing the same type of sum, with different numbers each time.)

Imagine that your great-aunt has died and that you will be inheriting a sum of money. You have decided to spend a quarter of what you receive on a day at the races, to save a quarter, and to spend the other half on some new clothes. These fractions are fixed. But you have no idea how much money you will be given! It might be anything from £1 to £50, or even more. Because the number of pounds is *not fixed* it is called a *variable*. In the program which follows, you may call this *number variable* by the letter **a**. (And you keep it

outside the inverted commas of the PRINT statements.)

```
10 PRINT "I am given £";a
20 PRINT "For gambling I have £";a/4
30 PRINT "For saving I have £";a/4
40 PRINT "For clothes I have £";a/2
50 STOP
```

You will find that the £ symbol is in red on the X key. Notice that there needs to be a semi-colon between the inverted commas of a PRINT message and a number variable. You could also have used a comma rather than a semi-colon here, but when the program was run, a comma would have left a large gap between the £ sign and the amount of money. A semi-colon tells the computer that the next item on the screen must be moved right up to the last letter or symbol to be printed on that line. Now run this program. You will see that the computer will only print on the screen:

```
I am given £
```

and near the bottom of the screen you will see the message:

```
2 Variable not found, 10:1
```

You have put in the variable **a**, but you have not told the computer what **a** is! Or, to put it another way, you have not *defined* **a**. You could guess that you will get £20 for your inheritance, and add the line:

```
7 LET a=20
```

You will find that the Keyword LET is on the L key. (You can now see why you always begin by numbering program lines in tens. When you are

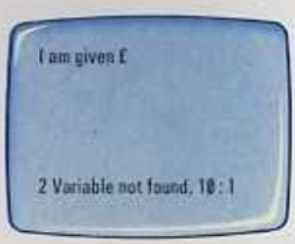
making up a program there is always plenty of space for new lines to be added in.) If you run the program now, the computer will print:

```
I am given £2Ø
For gambling I have £5
For saving I have £5
For clothes I have £1Ø
```

and below this, you will see the message:

```
9  STOP statement, 5Ø:1
```

But with this method, you will have to type a new line 7 each time you guess a new amount of money! So you had better delete (rub out) line 7, and do something else.



◀ *The computer cannot make sense of your program until you define what the variable in line 1Ø (the letter* **a***) stands for.*

❓ Problem 13: How can you delete a line in a program when it has already been typed in, and you have pressed ENTER?

☑ Supposing that you wish to delete line 7. If you wish to delete it and put nothing in its place, simply type 7 and press ENTER. If you wish to put something in its place, type 7, and the rest of the new line, and press ENTER. The computer will immediately delete the old line 7 and put the new one in its place. Now, delete line 7 and look at the screen to see that it has

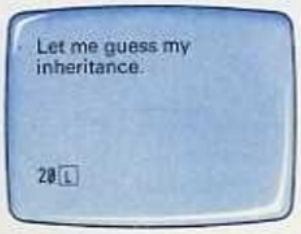
disappeared. Here are three new lines to enter:

```
5  PRINT  "Let me guess my
inheritance."
6  PRINT  "How much will it be?"
7  INPUT  a
```

You will find the INPUT keyboard on the I key. Try running the program. The computer will print:

```
Let me guess my inheritance
How much will it be?
```

Then the computer will wait and do nothing, because of your instruction INPUT **a**. The computer is waiting for you to put in, or input, whatever number you have decided to place in the store labelled **a**. The



▲ *By putting an INPUT instruction into the program, you make the computer wait after line 6 until you have typed in the amount of money you think you will receive. Only then will it work out how much you have to spend on each item.*

computer will not continue with the program until you have made an input. Try inputting 2Ø (simply type 2Ø, which will appear at the bottom of the screen, and press ENTER.) The computer will at once

go on with the program and will give you the answers that you had before. But now press run again, and press ENTER. This time, input the number 32. The answers will, of course, be different. In fact, you can now run the program again and again, guessing a different sum of money each time, and seeing how much you will have for spending, and how much for saving.

? Problem 14: Your program does not have a title.

✓ To give any program a title, you enter a line beginning with the Keyword REM (on the E key), followed by the title in ". For this program the title could be MONEY, in which case you should add:

```
3 REM "MONEY"
```

The computer takes no action when it reads the word REM; so REM lines are useful for putting lines into programs to REMind yourself what is happening when you come to look at the program days or months after first writing it. This program is now complete and working well. But before you type NEW and move on to something else, you can use the program to find out the answer to another problem.

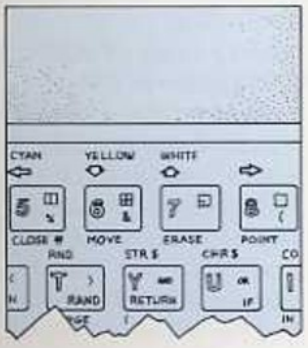
? Problem 15: If you want to make a small change to a line that has already been entered, you do not want to have to retype the whole line. Suppose, for example, that you want to see what will happen if the semi-colon in line 1Ø is changed to a comma.

✓ Type LIST and press ENTER so that you can see the "MONEY" program on the screen up in front of you. Now look at the top row of the Spectrum keyboard. Above the number keys 5, 6, 7 and 8, you can see arrows pointing left, down, up and right. You can use these arrows to move around on the screen when you are making changes to lines. (Making

changes is called *editing*.) To use the arrows you need to keep one finger pressed on the CAPS SHIFT key, while you press the number key below the arrow you want.

Try keeping one finger pressed on the CAPS SHIFT key while you press once on either the 6 or the 7 number key. Then look at the screen to see what has happened. Between the figure 3 and the word REM of the first line in the program, a pointer has appeared which looks like this: >. This pointer is called the *program cursor*. You may have noticed that whenever you are typing the lines of a program and entering them, there is always a > next to the number of the line which has just been entered.

When you want to edit a line, the first thing to do is to use the arrows to move the program cursor to that line. The down arrow will move the program cursor down one line, and the up arrow will move it up one line. You want to edit line 1Ø, so you need to keep one finger pressed on the CAPS SHIFT key while you type the number 6 key four times. Now the > should be between the 10 and the PRINT of the fifth line in the program. If you have typed the number key 6 once too often, then type the number key 7 once, to get the program cursor back to the right place.

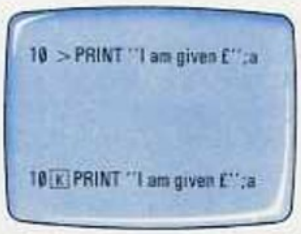


◀ *You can use the number keys 5 to 8, to move the program cursor around on the screen in the directions shown by the arrows.*

The second thing to do is to make a copy of line 10 appear on the lower part of the screen, where you can make changes to it. If you look at the top left of the keyboard you can see the word EDIT above the number 1 key. If you keep one finger pressed on CAPS SHIFT, and then press the number 1 key, this will bring down a copy of the line which has the program cursor next to it. Now, check that the program cursor is next to line 10. Then keep one finger pressed on the CAPS SHIFT key and press once on the number 1 key. At once, a copy of line 10 appears on the lower part of the screen, looking like this:

```
10[K] PRINT "I am given £";a
```

You can see that the flashing [K] cursor is just to the right of the number 10. At the point where the flashing cursor appears, you can either delete anything to the left of the cursor in the normal way, or add new things. It does not matter if you do not want to delete or add anything at the point where the flashing cursor appears. You can move the flashing cursor to the left or to the right by using the left and right arrows. And if you move it to the right, it will helpfully change from a flashing [K] cursor to a flashing [L] cursor.



▲ *To make changes to a line you have entered, you must first make a copy of that line appear at the bottom of the screen. Position the program cursor between the line number and the first instruction of the line you want. Then you keep one finger on the CAPS SHIFT key and press the 1 key.*

Suppose you want to change the semi-colon in line 10 to a comma. So you want to move the flashing cursor along to the right. Keep one finger pressed on the CAPS SHIFT key, and press the 8 key until the line on the screen looks like this:

```
10 PRINT "I am given £";[L]a
```

Now, using the CAPS SHIFT and 0 keys in the usual way, you delete the semi-colon, so that the line looks like this:

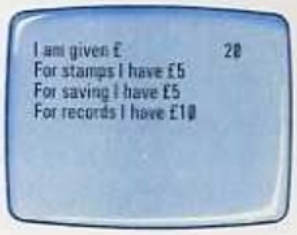
```
10 PRINT "I am given £"[L]a
```

Then you use the SYMBOL SHIFT and N keys to type in a comma, so that the line reads:

```
10 PRINT "I am given £",[L]a
```

Now you have edited the line. The flashing cursor is still in the line, but that does not matter. Simply press ENTER, and the new line 10 will disappear from the bottom of the screen and appear in the list at the top of the screen, in place of the old line 10. If you run the program, you can see how you have changed it! Now practise editing by changing lines 20, 30 and 40 in the same way.

The left and right arrows can also be used at any



► *Because you have changed the semi-colon in line 10 into a comma, there is a large gap between the £ symbol and the number of pounds in this line.*

time to edit a line *before* it has been entered. This can save a lot of time spent deleting and retyping if you make a mistake very early in a line, which you do not notice until you have typed in the rest of the line.

Other number variables

Now you can try writing a more complicated program using number variables. Although one usually see programs written out in full, with explanations afterwards, programs do not suddenly come into people's heads fully worked out! Most programs begin with just a few lines, and then the person writing the program gradually adds on more and more lines.

The first thing to do is to decide what sort of game you want to invent, or what sort of problem you want to solve. For example, perhaps you would like a program which helps you to plan your holiday spending. You might want to know how much money you will have for your holiday, and how much you can afford to spend each day. To begin with, you must decide what information needs to be input into the computer. In this case, it will need to know:

a) how much money (if any) you have already;
b) how much more money (if any) you will get during the holiday;
c) how much money (if any) you want to have left at the end of the holiday;
d) how many days you will be on holiday.

Next, you must decide how to input this into the computer. Try these lines, and remember to copy them carefully, typing in all the inverted commas exactly as shown:


```
10 PRINT "Enter the number of £s"
20 PRINT "you already have"
30 INPUT a
40 PRINT a
50 PRINT "Enter the additional money
you will get"
60 INPUT b
70 PRINT b
80 PRINT "How much money do you want
to"
90 PRINT "have left at the end of"
100 PRINT "the holiday?"
110 INPUT c
120 PRINT c
130 PRINT "Enter the number of days you"
140 PRINT "will be on holiday"
150 INPUT d
160 PRINT d
```

Remember that the INPUT Keyword is on the I key. The PRINT lines 40, 70, 120 and 160 are not needed to input information, but are there to print the numbers you choose onto the screen, so that you can see and remember them. Run the program so far, to see what it looks like.

The computer now has all the information it needs. But you must tell it what to do with that information. To find out how much you have to spend, you must instruct it to add the money you already have (**a**) to the money you will get (**b**), and then take away from **a**+**b** the money you want to have left (**c**). So press LIST, and add the line:

```
170 PRINT "You have £";a+b-c;" to
spend"
```

Notice that there is a semi-colon between the

second " (inverted commas) and the number variable **a+b−c**, and a second semi-colon after the number variable and before the third ". You could have used a comma in both places instead, but this would have left too large a gap between the £ sign and the number of pounds when your program was run. Using semi-colons means that there will be no gap at all. However, you do want a gap between the number of pounds and the word *to* in the second part of the PRINT statement. So you should have pressed the BREAK SPACE key once, immediately after typing the third ". When typing program lines, it is important to copy them exactly, and to put in all the spaces correctly. If you miss one you will soon see your mistake when you run the program and you will then have to edit the line which was wrong.

From now on, to help you type in the lines correctly, the number of spaces that must be put into a PRINT statement will be shown as a dash like this _. One dash simply means put in one space.

Now you need to add some more lines to this program. To find out how much you can spend each day, you must instruct the computer to divide the amount you have to spend (**a+b−c**) by the total number of days that you are on holiday (**d**). So add the lines:

```
180 PRINT "You_will_have_£";(a+b
-c)/d
190 PRINT "to_spend_each_day"
```


(Line 190 does not need a space between the first " and the word *to*, because it is a new program line. So this line will appear as a separate line on the screen, below the words given in line 180.) This program is now ready to be run; but it has no beginning or end. You can give it a title and an end with the lines:

```
  5 REM "HOLIDAY£S"
200 STOP
```

Remember that the word STOP is in red on the A key. Now look at the program carefully.

? Problem 16: The program is now so long that you cannot see all of it on the screen!

✓ Press LIST (on the K key) and press ENTER. You will now see the first lines of the program, down to line 160. And underneath the lines, on the left-hand side, you can see the word:

```
scroll?
```

A scroll is a long roll of paper with writing on it. You look at only part of the writing at a time. Then you wind it up at one end, and unwind it at the other end and read the next part. The computer is asking whether you would like to do the same thing to the lines of the program which you have stored in its memory. If you press any of the keys (apart from the CAPS SHIFT, SYMBOL SHIFT, BREAK SPACE or N keys), it will 'wind up' the lines at the top of the screen, and 'unwind' some more lines at the bottom of the screen, so that you can read the next part of the program. When there is no more program to unwind, or 'scroll' then it will give you one of its 'OK' messages. If you do *not* want to scroll a program, but wish to keep a part of it on screen so that you can edit it, then press the BREAK SPACE key. A message will

appear at the bottom of the screen saying:

```
D  BREAK - CONT repeats, 0:1
```

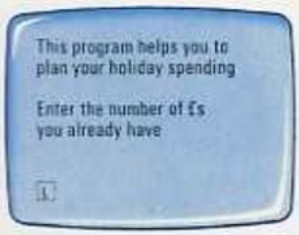
You can then add to or edit the lines in the part of the program you can see on the screen.

Now, back to the program. Perhaps it would be a good idea to explain to whoever is running the program exactly what you are trying to do. Add the lines:

```
6  PRINT  "This _ program _ helps _ you _
to"
7  PRINT  "plan _ your _ holiday _ spending"
```

Now run the program.

Problem 17: The program is working very well. But you could improve its appearance. A gap between the two lines telling you what the program is about, and the lines in which the computer asks for input, would look clearer. So would a small gap between the lines in which the computer asks for input, and the lines in which it gives answers; so would a small gap between the two answers.



◀ *You can use the PRINT instruction on its own to put blank lines on the screen, so that your program will look better when it is being run.*

☑ To leave a blank line on the screen, you simply type PRINT with nothing after it. So add:


```
8  PRINT
9  PRINT
165  PRINT
175  PRINT
```

Run the program again. It looks quite good. But programmers are hardly ever satisfied!

Problem 18: How could you make this program more interesting and useful?

☑ By using two IF . . .THEN statements. Why? Because at present, if someone enters, for example, 25 at INPUT **a** (line 30) and 27 at INPUT **b** (line 60), they might also enter 70 at INPUT **c** (line 110). Then the computer would print at line 170:

```
You have £-18 to spend
```

That would be nonsense! By using IF . . . THEN statements you can prevent something like that from happening. Add these two lines:

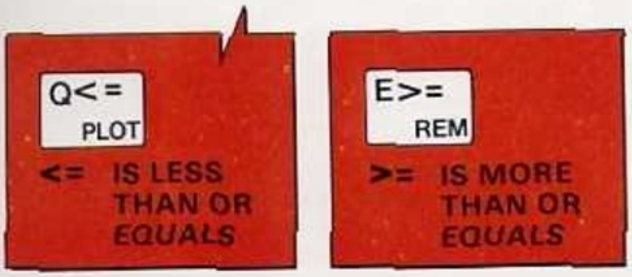
```
124  IF  a+b-c<=0 THEN PRINT
"That's _ silly! _ You _ can't _ have _ as _
little _ as _ that!"
125  IF  a+b-c<=0 THEN GOTO 80
```

You will find the IF Keyword on the U key, and the THEN Keyword in red on the G key. The <= (is less than or equals) symbol is in red on the Q key. (Never use the two separate symbols < and =, or > and = to make up <= or >=.) The ' (apostrophe) is in red on the 7 key.

Line 124 tells the computer that IF the amount of money left at the end of the holiday is £0, or less than £0, THEN it must print a message explaining that the person running the program has entered too large a number at INPUT **c** – in other words, he or she has

asked for too much money at the end of the holiday. *Line 125* tells the computer that IF the amount of money left at the end of the holiday is £0, or less than £0, THEN it must jump back to line 80. This will give *the person running* the program another chance to enter a sensible number at INPUT **c**.

After typing in these new lines, LIST the program and look at it carefully so that you can see what is happening at every line. Then run it.



▲ *To type in these symbols, keep one finger on the SYMBOL SHIFT key while you press the Q key or the E key.*

String variables

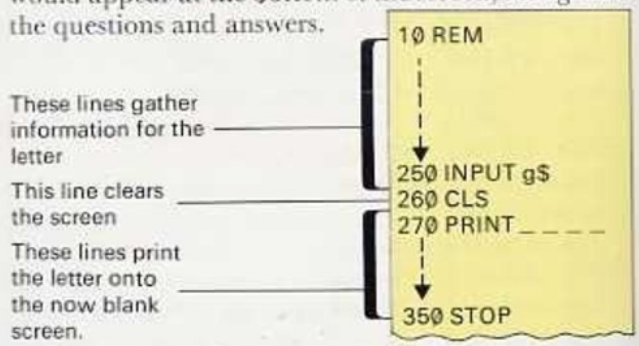
(How to give the same set of instructions, but inputting different words each time.)

For number variables, you used the letters **a, b, c** and **d.** However, variables that stand for words or letters are different – they are known as string variables. They work in just the same way except that after, for example, the letter **a,** you use the **$** ('string') sign, which is a symbol in red on the 4 key. Here is a program using string variables which you could use to write thank-you letters. (At line 270 of the program, you *should of course type in your own* address, and not 2, Housman Close.)


```
10  REM  "THANKYOU"
20  PRINT  "This_program_helps_you_
to_write"
30  PRINT  "a_thank-you_letter."
40  PRINT
50  PRINT
60  PRINT  "Enter_your_present._It_
was_a"
70  INPUT  a$
80  PRINT  a$
90  PRINT  "Choose_a_word_to_
describe_your_ _";a$
100 INPUT  b$
110 PRINT  b$
120 PRINT  "Who_gave_you_the_";a$
130 INPUT  c$
140 PRINT  c$
150 PRINT  "Choose_a_word_or_
words_to_describe_the_weather."
160 INPUT  d$
170 PRINT  d$
180 PRINT  "Choose_a_word_or_
words_to_describe_your_family."
190 INPUT  e$
200 PRINT  e$
210 PRINT  "How_will_you_sign_
your_name?"
220 INPUT  f$
230 PRINT  f$
240 PRINT  "What_is_the_date?"
250 INPUT  g$
260 CLS
270 PRINT  "_~~~~~~~~~~2,_
Housman_Close."
280 PRINT  "_~~~~~~~~~~";g$
290 PRINT
300 PRINT  "__Dear_";c$
```

```
310 PRINT "_ _ _ _ Thank _ you _ very _
much _ for _ the _ ";a$;". _ It _ is _ really _
";b$;". _ In _ fact _ it _ is _ the _ best _ ";a$;
" _ I _ have _ ever _ had."
320 PRINT "_ _ _ _ I _ hope _ that _
you _ have _ been _ having _ good _ weather. _
The _ weather _ here _ has _ been _ ";d$;". _
You _ will _ be _ interested _ to _ hear _ that _
the _ family _ are _ all _ ";e$;". _ Thank _
you _ again _ for _ the _ ";a$;"."
330 PRINT "_ _ _ _ _ _ With _ love"
340 PRINT "_ _ _ _ _ _ from _ ";f$
350 STOP
```

As you can see, the lines up to 250 are gathering information and displaying it on the screen. The lines from 270 to the end are printing the letter. At line 260 you can see the useful instruction CLS. You will find CLS on the V key. This tells the computer to CLear the Screen. Otherwise the thank-you letter would appear at the bottom of the screen, along with the questions and answers.

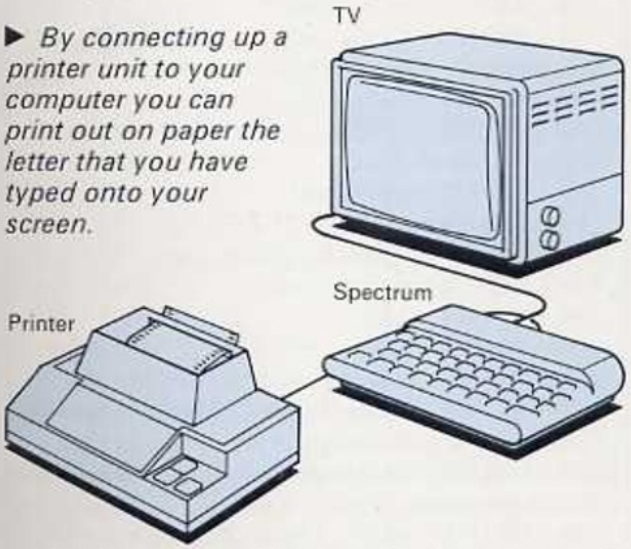


If you had a printer, you could run the program, then print the letter onto a sheet of paper; and then run it again, inputting new string variables, to write your next letter. Even without a printer, you can enjoy running this program with different variables. They can be sensible or silly! The computer will print

whatever you tell it to print. Try running the program a few times. Then make any changes which you feel will improve it.

For example, in line 90 you carefully left two gaps after the word *your*, so that when the name of your present was added to this line it would not be split up, with the first letter appearing at the end of the line and the rest of the word on the next line. But when lines 150 and 180 appear on the screen, the word *describe* in each case is split up. You could either add spaces, or divide each of these lines into two PRINT lines in the program. Can you see other places where you might want to make changes of this sort?

► *By connecting up a printer unit to your computer you can print out on paper the letter that you have typed onto your screen.*



Also, it is rather a short letter. Perhaps you would like to include a paragraph in which you mention a book which you have been reading, or a television programme which you have been watching, and say what you think of it. To do this, you must work out the form of the sentences, such as 'I have been watching a television programme called .... It stars

. . . . and . . . . I am finding it extremely . . . . I wonder what you think of it?' Then you must add lines to INPUT the information that will be needed; and a line or lines to PRINT the extra sentences in the letter.

Using string and number variables together

Here is a game which uses both number and string variables. Remember to put in all the inverted commas and semi-colons.

```
10  REM "GUESSME"
20  PRINT "This is a guessing-game for TWO players."
30  PRINT
40  PRINT "Please enter the FIRST player's name."
50  INPUT a$
60  PRINT a$
70  PRINT "Please enter the SECOND player's name."
80  INPUT b$
90  PRINT b$
100 PRINT
110 PRINT "Thank you, ";a$;" and ";b$;"."
120 PRINT b$;", please stop looking at the screen."
130 PRINT a$;", please type a number between 1 and 20 and then press ENTER."
140 INPUT a
150 PRINT "Right! Now press ENTER again to go on."
160 INPUT z$
170 CLS
```



```
180 PRINT "Now ask ";b$;" to look at the screen."
190 REM Guessing until right, loop
200 PRINT b$;", guess what the number between 1 and 20 is."
210 INPUT b
220 CLS
230 IF b<>a THEN PRINT "Hard luck! ";b;" was not correct."
240 IF b<>a THEN GOTO 190
250 PRINT "WELL DONE ";b$;", YOU ARE RIGHT! The number was ";a
260 PRINT
270 PRINT
280 PRINT "Would anyone like another turn? Enter y for yes or n for no."
290 INPUT c$
300 IF c$="y" THEN CLS
310 IF c$="y" THEN GOTO 20
320 IF c$<>"n" THEN GOTO 280
330 CLS
340 STOP
```

You will find that < > is a red symbol on the W key. The first part of this program is very easy to follow.

Line 20 explains that it is a guessing-game.

Lines 40 to 80 get information into the computer about the names of the two players, using the string variables a$ and b$.

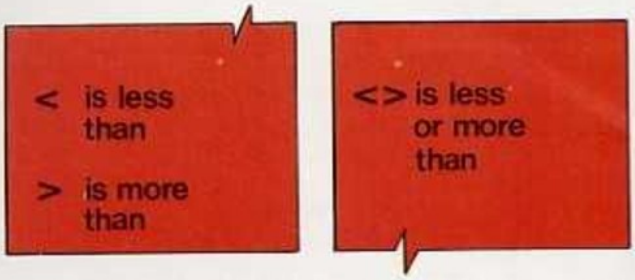
At line 140 the first player inputs a number, using the number variable **a**. (This stores the number at a place in the computer's memory labelled **'a'**.)

At line 210 the second player inputs his or her guess, using the number variable **b**. (This stores the guess at a place labelled **'b'**.)

Now the program is a little harder to follow. If the

second player has guessed right, the number in the store labelled **b** will be the same as the number in the store labelled **a**. But the second player may have guessed wrong. If he has guessed wrong, **b** will be smaller than or larger than **a**. The symbol < means *is less than*, and > means *is more than*, and you can use the symbol <> to mean *is less or more than*. (Never use the two separate symbols < and > to make <>.) So:

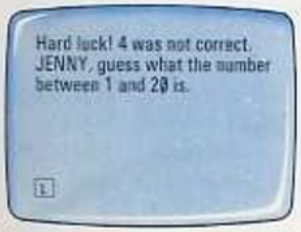
At line 230 the computer is told that IF the second player is wrong (IF b<>a) THEN it must PRINT a 'Hard luck' message.



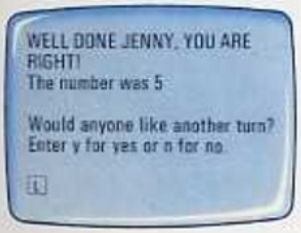
At line 240 the computer is told that IF the second player is wrong (IF b<>a) THEN it must GOTO (jump back to) line 190 to give the second player another chance to guess the right number. So the lines from 190 to 240 form a loop, and the computer will go round and round in circles until the right answer has been given at INPUT **b**.

You have made this loop as clear as possible to understand by putting a REM line at line 190, explaining that the next section of the program is a 'Guessing until right, loop'. A REM line which is used inside the program to help you REMember which part of the program is doing what, and *not* as a program title, does not need inverted commas, and it can have gaps between words and be as long as you like.

When the correct number has been guessed, that is **b**=**a**, the computer will take no notice of lines 230 and 240, and will go straight to line 250, where it is told to print a 'Well done' message. When the guess is right, the computer will soon reach line 280. Then you come to some more IF . . . THEN statements. IF anyone inputs **y**, THEN the computer will CLear the Screen, and GOTO line 20, to start a new game.



◀ *If the player's guess is more than or less than the right answer, the computer will print a 'Hard luck' message and offer the player another chance.*



◀ *When the player's guess is equal to the right number, the computer prints a 'Well done' message and offers another game.*

❓ Problem 19: What if someone inputs the wrong letter?

☑ If the letter is **n**, then the computer will ignore lines 300, 310 and 320 and carry on to line 330, the screen will clear and the program will STOP. If the letter is another letter, because by mistake the player had not typed either **y** or **n**, or they have typed a capital Y or N, then the computer will ignore lines 300 and 310. But at line 320 it will jump back to line 280 to give the player another chance to press **y** or **n**.

❓ Problem 20: How can you stop the computer moving through a program too fast?

☑ Lines 150 and 160 deal with this problem. The computer can always be stopped for as long as you like by an INPUT line. When there is an INPUT line the computer has to wait until an input has been made before it can continue. Pressing the ENTER key is one form of input. So at line 150 you told the player that if the game was to continue, the ENTER key must be pressed. Pressing the ENTER key gave the computer the input for which it was looking at line 160 and it immediately went on with the program.

Ways to improve "GUESSME"

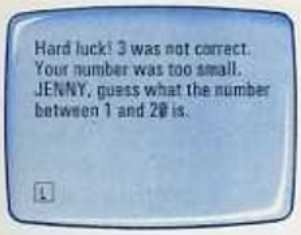
1) See if you can work out some ways to improve the *appearance* of the program on the screen. A few more blank PRINT lines might be useful.

2) At the moment, there is something wrong, not just with the appearance of the game, but with the way it is made up (its *structure*.) When you have made a guess, you have no idea whether your guess is larger or smaller than the right answer. It would make the game more interesting if you added a line instructing the computer to print a message saying either 'Your number was too small' or 'Your number was too big'. To do this, you could use more IF . . . THEN statements. Try these:

```
233 IF b<a THEN PRINT "Your_
number_was_too_small."
235 IF b>a THEN PRINT "Your_
number_was_too_big."
```

The separate symbols for < (is less than) and > (is more than) are in red on the R and T keys.

If you like this game and want to SAVE it on tape so that you can play it at some other time in future, see page 87.



◀ *A program can be made to look as if it is replying to a player's answers and giving clues to the right number. You use IF . . THEN instructions to do this.*

A reminder

You have now learned:

1) how to use the computer for simple maths;
2) how to use number and string variables;
3) how to use a semi-colon to make your programs look better on the screen when they are being RUN;
4) how to delete lines once they have been entered, and put in new ones;
5) how to use REM to give your program a title;
6) how to edit programs using the CAPS SHIFT key, together with the arrows above keys 5, 6, 7 and 8, and with EDIT above key 1;
7) how to scroll a long program so that you can read all of it;
8) how to use PRINT to leave blank lines on the screen;
9) how to use IF . . . THEN statements;
10) how to use CLS to clear the screen;
11) how to make it easier to see what is happening when there is a loop in a program by putting in a REM line at the beginning of the loop;
12) how to use the ENTER key as an INPUT.

3 Making Sounds

You can instruct the ZX Spectrum Computer to make strange noises and to produce musical sounds! When you are putting sound into a program, one of the lines might look like this:

```
10 BEEP 1,0
```

10 is of course the number of the line.
BEEP is the Keyword, or instruction, for sound.
1, (first number): You can make the sound last for a short or a long time, by using numbers from 1 to 10. The first number, 1, tells the computer to make the sound for one second. The number, 2, would have told it to make the sound for 2 seconds, or 3 for 3 seconds, and so on. Even if you do not usually use decimal numbers, you will know that .5 is the same as a half, so 1.5 would tell the computer to make the sound for 1½ seconds; 2.5 for 2½, and so on. (If you are used to dealing with the decimal point in numbers, you may like to know that, for the length of sound, you can use all the numbers between .01 and 10.49.)
0, (second number): The sound itself (the note) is chosen by the second number, which can be from 0 up to 69 or down to −60. The second number, 0, tells the computer to play the note 'Middle C'. For musical sounds, 50 is about the highest that is useful, and −18 about the lowest.

Making strange noises

If you want the computer to make strange noises, you must use notes between 51 and 69, or between −19 and −60. Try this program:


```
10 REM "NIGHT"
20 BEEP 1,-40
30 BEEP 1,-30
40 BEEP 1,67
50 BEEP 1,-50
60 BEEP 1,-30
70 BEEP 1,-40
80 STOP
```

Remember that BEEP is in red beneath the Z key, so you need to go into [E] mode and then press Z while one finger is pressing on the CAPS SHIFT key. To repeat the sound after you have run the program, just press RUN again.

Making up tunes

If you want to make up tunes, you must use notes between 50 and −18. Try this:

```
10  REM  "SOLDIER"
20  BEEP .5,7
30  BEEP .5,4
40  BEEP .5,0
50  BEEP .5,9
60  BEEP 1,7
70  BEEP .5,5
80  BEEP .5,2
90  BEEP .5,-1
100 BEEP .5,-4
110 BEEP 1,0
120 BEEP .5,7
130 BEEP .5,4
140 BEEP .5,0
150 BEEP .5,9
160 BEEP 1,7
170 BEEP .5,5
180 BEEP .5,7
190 BEEP .5,9
200 BEEP .5,11
210 BEEP 1,12
220 STOP
```

Using a group of lines more than once

You may have noticed that in the "SOLDIER" program, lines 20 to 70 and lines 120 to 170 are exactly the same. Whenever you are writing a program and have a group of lines which you want to use more than once, it is a good idea to *write them only once*, after the STOP line in the program, and to put a special instruction into the program which will call them up whenever you want them. The special instruction is GOSUB (on the H key), followed by the number of the first line in the group of lines which you wish to call up. Now write the "SOLDIER" program again, putting in GOSUB instead of lines 20 to 70 and

lines 120 to 170. The program will look like this:

```
10  REM  "SOLDIER"
20  GOSUB 1000
30  BEEP .5,2
40  BEEP .5,-1
50  BEEP .5,-4
60  BEEP 1,0
70  GOSUB 1000
80  BEEP .5,7
90  BEEP .5,9
100 BEEP .5,11
110 BEEP 1,12
120 STOP
```

Now, after the STOP line, you must enter the lines which your GOSUB will call up. To make these lines easy to see and understand in a program, it is a good idea to give them a high number which sets them apart from other lines. That is why you have given the instruction GOSUB 1000 instead of GOSUB 130. It also helps if you make the first of the GOSUB lines a REM line, REMinding you what is happening in the lines which follow. The last of the GOSUB group of lines must always be the Keyword RETURN, which you will find on the Y key, and which makes the computer jump back into the main program again. So you can add:

```
1000 REM  Six notes
1010 BEEP .5,7
1020 BEEP .5,4
1030 BEEP .5,0
1040 BEEP .5,9
1050 BEEP 1,7
1060 BEEP .5,5
1070 RETURN
```

In this case, you have only saved two lines. But if there had been more lines in the GOSUB, or if you had wanted to use it more than twice, you could have saved a great deal of space. More importantly, the use of GOSUBs can make a program very much easier to follow. When a program is easy to follow, it is said to be 'well organized' or, better still, 'well structured.' Good structure is the mark of a good programmer!

Now, try making up some tunes of your own, or instruct the computer to play a tune which you know already. You will find the section on sound on pages 135–138 of the *ZX Spectrum BASIC Programming* book very helpful. You can also try experimenting with strange noises. Can you make a sound like a kettle whistling as it boils? Or like a door creaking?

◀ *When you want to use a group of lines more than once in the same program, you type them out as a separate group after the end of the program. And you use the instruction GOSUB, followed by the first line number of the group of lines, to send the computer to pick them up whenever they are needed.*

A reminder

You have now learned how to make tunes and strange noises using the instruction BEEP. You have also learned how to make programs easier to follow, and sometimes shorter as well, by using GOSUBs.

4 Colouring Words and Backgrounds

The ZX Spectrum Computer usually shows black *text* (words) on a white screen. But you can learn how to colour both the words and the screen. The screen is divided into two areas. The outside edge, all round the screen, is called the *border* area. To change the colour of the border is easy. You simply press the Keyword BORDER, which you will find on the B key, followed by a number. You can see which number to choose, because some of the number keys have the names of colours above them. So you can see that 1 gives dark blue, 2 gives red, 3 gives magenta (a light purple), 4 gives green, 5 gives cyan (a light blue), 6 gives yellow, 7 gives white and 0 gives black. So BORDER 1 would give you a dark blue border to the screen. Try typing in that instruction as one of the first lines of this program:

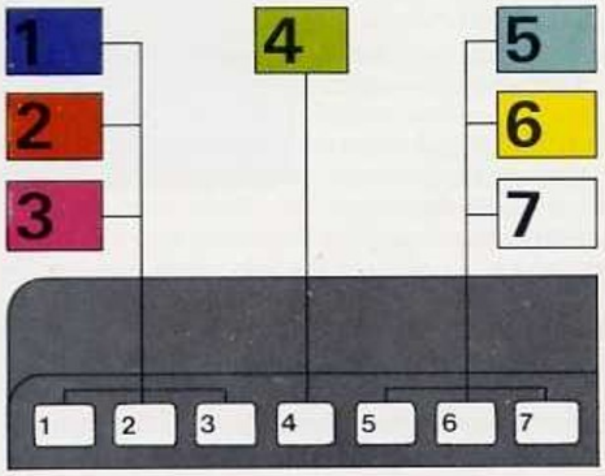
```
10 REM "COLOURWORD"
20 BORDER 1
30 PRINT "Now_you_can_see_a_
blue_border."
```

Run this program, and the blue border appears. The area inside the border is called the *paper* area. This is the part of the screen on which your programs appear. The colour of the text in this area is called the *ink* colour. To change the ink colour you use the Keyword INK, which you will find in red beneath the X key. (Look back at Chapter 1 if you want to refresh your memory about how to type the words marked in red *beneath* the letter keys.) After the Keyword INK you use one of the numbers at the top of the keyboard, just as you did after the Keyword BORDER.

As well as changing the ink colour, you can change the paper colour, so that you can, for example, print in red 'ink' on yellow 'paper'. To change the paper colour beneath each letter that is printed, use the Keyword PAPER (in red beneath the C key) and follow it with one of the numbers at the top of the keyboard. So, INK 2 would change the colour of the

▲ *The ZX Spectrum can colour in a border around the edge of the screen.*

▲ *It can print coloured letters and numbers against a differently coloured background.*



text to red, and PAPER 6 would change the background of each letter that is printed to yellow. Try adding these lines to your program. (You will see that before you ENTER your instructions, they appear in white at the bottom of the blue border.)

```
40 PRINT "Press_ENTER_to_continue."
50 INPUT z$
60 INK 2
70 PAPER 6
80 PRINT "Now_you_can_see_red_
letters_on_a_yellow_background."
```

If you want your program to make the whole of the paper area yellow, and not just the part on which something is being printed, then after the PAPER instruction you need to give the instruction CLS. As you know, this clears the screen. But there are two interesting things about this instruction which you could not notice before, because the paper and border colours were set to white. The first is that it only clears the *paper* area of the screen. The second is that it clears the *whole* of the paper area to match whatever paper colour has been set. Try adding the following lines to your program. (As soon as you enter the first line, you will notice that all of the paper area becomes yellow and all of the program lines appear in red. Don't worry!)

```
 90 PRINT
100 PRINT "Press_ENTER_to_
continue."
110 INPUT z$
120 CLS
130 PRINT "Now_the_whole_paper_
area_is_yellow."
```

Now, to get everything back to normal, you need to

change the border and paper areas to white, and the ink to black. So add these lines too:

```
140 PRINT
150 PRINT "Press ENTER to
continue."
160 INPUT z$
170 PAPER 7
180 CLS
190 BORDER 7
200 INK 0
210 PRINT "Back to normal."
```

When you run your program, the first part will look different. Don't worry, just run it again. You will find that the first part of the program will then be the same as before.

▲ *You can fill the whole of the centre area of the screen with colour.*

▲ *And you can print different parts of sentences in more than one colour.*

Using different coloured letters in one line

As well as changing the colour of *all* the text which is shown on the screen, you can give instructions which only change the ink colour of a particular PRINT

statement – or even just a part of a PRINT statement. Try adding these lines to your program:

```
220 PRINT
230 PRINT "Press ENTER to
continue."
240 INPUT z$
250 CLS
260 PRINT INK 2;"My house is red,";
INK 1;" but the roof is blue."
270 PRINT INK 4;"Grass is green,";
INK 3;" not magenta!!!"
```

When you run the program, you will see that after each INK instruction in a line, the words change colour. Remember that parts of a PRINT statement have to be joined with commas or semi-colons.

Making letters flash

To make letters flash you use the Keyword FLASH (in red beneath the V key), followed by the number 1. Try adding these lines:

```
280 PRINT
290 PRINT FLASH 1;"Stop telling me
things I know!"
```

Or, if you want the line to be in red as well as flashing, change it to:

```
290 PRINT INK 2; FLASH 1;"Stop
telling me things I know!"
```

The FLASH instruction can be switched off by FLASH followed by the number 0. Try typing in the following lines:

```
300  PRINT
310  PRINT  INK 3; FLASH 1;"STOP_IT";
FLASH 0;"_YOU_IDIOT!"
```

And, you can make the background brighter, by using the Keyword BRIGHT (in red beneath the B key), followed by the number 1. Add these lines:

```
320  PRINT
330  PRINT  INK 2; FLASH 1; BRIGHT
1;"I_SAID_STOP_IT!"
```

As the words 'I SAID STOP IT!' flash, you can see that the white is very much brighter than the normal, rather dull, white which is usually on the screen. You can use BRIGHT 0 to switch off the brightness in the same way that you used FLASH 0 in line 310.

A different way of colouring text

It is possible to colour text simply by typing it straight into the program lines in colour! This is how you do it. Supposing that in the line:

```
340  PRINT  "I_have_stopped_now."
```

you want the word 'I' to be in dark blue, 'have' in red, 'stopped' in green, and 'now' in cyan. First, type:

```
340  PRINT  "
```

as normal. Then go into [E] mode by pressing the SYMBOL SHIFT key while you keep one finger pressed on the CAPS SHIFT key. Then press the number 1 key (the key for dark blue), while keeping a finger pressed on the CAPS SHIFT key. The flashing [E] cursor will now have become a blue flashing [L] cursor. Now

type the following letter:

```
I
```

You will see that it appears on the screen in blue. Now type a space. Then go into [E] mode again, and press the number 2 key (the key for red), while keeping a finger on the CAPS SHIFT key. You will now have a red flashing [L] cursor, and can type:

```
have
```

which will appear in red on the screen. In the same way, type 'stopped' when you have a green flashing [L] cursor, and 'now.' when you have a cyan flashing [L] cursor. When you get to the inverted commas at the end of the PRINT statement, make sure that they appear on the screen in black, by changing into [E] mode, and then getting a black flashing [L] cursor by using the number key 0 with CAPS SHIFT. If both inverted commas are *not* in black, you could find it confusing, because although what appears on the screen when you RUN the program will not change, you may alter the colour of the lines when you LIST the program. For example, see what happens to the present LIST if you change the final inverted commas in line 290 to red! Finally, to bring this program on colour to an end, add the line:

```
350  STOP
```

Run the whole program a few times, carefully comparing what happens on the screen with the list of instructions in this book. Now, try your own experiments with coloured text and backgrounds. The use of colour always makes programs look more cheerful and exciting, so why not go back to the "GUESSME" program and put some of the words

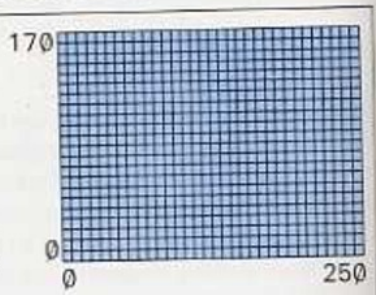
which will appear on the screen into colour? If you saved the final version on a tape, you will only have to LOAD it back into the computer (see page 89), and then you can get to work at once.

▶ *You can type different coloured letters straight into your program lines by changing the [L] cursor into a coloured [L] cursor.*

A reminder

You have now learned how to colour words (text) and backgrounds using the Keywords: BORDER, PAPER and INK.
You have also learned how to make letters flash and how to make the screen brighter using the Keywords: FLASH and BRIGHT.
You have also learned a way of putting colour directly into programs, by typing in coloured letters between the black inverted commas of PRINT statements within the lines of a listing.

▶ *When you want to draw pictures on your screen (see the next chapter), think of the screen as being a grid. The grid is 170 units from the bottom to the top and 250 units from left to right.*



5 Making the Computer Draw

If you want to draw things on the screen, you can use the paper area as a drawing-board. On the opposite page is a diagram of the paper area of the screen. You must think of this area as being about 250 units wide, and about 170 units tall. If you want to describe a point on a blank screen, you first have to say how far it is from the *left-hand side* of the paper area (for example, as shown in the first diagram on page 64, *Point A* on the screen is 27 units from the left), and then you say how far it is from the *bottom* of the paper area (*Point A* is 40 units from the bottom). This is called 'giving coordinates'. Starting with a blank screen try moving to Point A using PLOT, followed by the coordinates, with a comma between the two numbers. Type in this line:

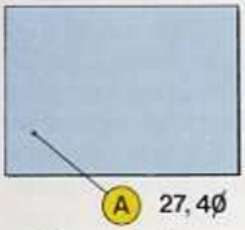
```
PLOT  27,40
```

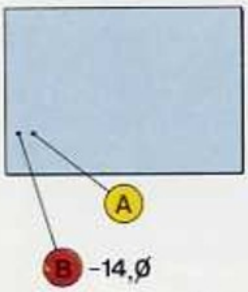
You will find the PLOT Keyword on the Q key. When you press ENTER, a small dot appears on the screen. You can now begin to draw, using the instruction DRAW, followed by the coordinates of another point on the screen. Now that you are 'on' the screen, the coordinates are given from the point you have reached. The first of the coordinates must say how far to the *left or right* you are going, and the second must say how far *up or down* you are going. As shown in the second diagram, *Point B* is 14 units to the left of Point A, and no units up or down. To go left or down, use a minus sign. Now, draw the line to Point B:

```
DRAW  -14,0
```

The DRAW Keyword is on the W key. When you press ENTER, the computer draws a line from where you

were before (Point A) to the new point (Point B). You have already drawn the first line in a simple picture of a motor-car. On the next page is a picture showing the next ten lines, with coordinates for all the important points, starting from A and going to L.





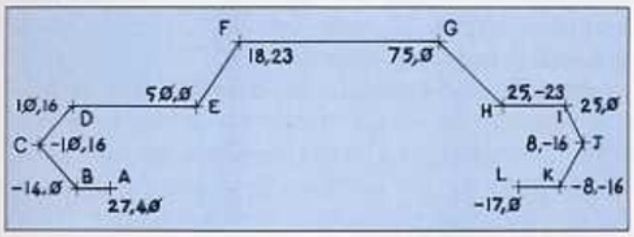
▲ *You use the numbers on your grid to give the computer coordinates for where to draw the lines. You start from a specific point, such as Point A and then count along from A to your next point (Point B).*

Now it is time to start turning all this into a program. Type in these lines:

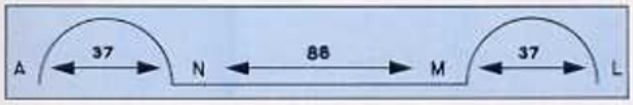
```
1Ø  REM  "CAR"
2Ø  REM  The body
3Ø  PLOT  27,4Ø
4Ø  DRAW  -14,Ø
5Ø  DRAW  -1Ø,16
6Ø  DRAW  1Ø,16
7Ø  DRAW  5Ø,Ø
8Ø  DRAW  18,23
9Ø  DRAW  75,Ø
1ØØ  DRAW  25,-23
11Ø  DRAW  25,Ø
12Ø  DRAW  8,-16
13Ø  DRAW  -8,-16
14Ø  DRAW  -17,Ø
```


Try running this to check that your drawing looks right. So far, this has all been very easy. But now you want to draw something more difficult. Between Points L and A, you want something like the second diagram below.

M is 37 units to the left of L; N is 88 units to the left of M; A is 37 units to the left of N. Once you have



▲ *These are all the coordinates that you will need to make the computer draw this outline of a car.*



▼ *You will need two semi-circles above the wheels on the base of the car to continue your drawing.*

reached M, the straight line between M and N will be easy enough. But between L and M you want to draw a semi-circle, and when you have reached N you want to draw another semi-circle between N and A.

How to draw semi-circles

So that it is easy to see how to draw semi-circles, clear the screen by pressing CLS and then pressing ENTER. This will not mean that you lose the lines which you have already typed in. (They will stay in the

computer's memory until either you type in new lines beginning with those line numbers, or you type NEW and press ENTER, or switch off the power supply to the computer.) Now, starting from rest, move to a new point on the screen by typing:

```
PLOT  50,100
```

and press ENTER. Suppose that you want to draw a semi-circle between this point, and a point 150 units to the right, and 0 units up. You *begin* the instruction as if you were drawing a straight line between the two points, like this (but make sure you do not press ENTER yet):

```
DRAW  150,0
```

But then you add a comma, and the Keyword PI, which you will find in green above the M key. So type in:

```
DRAW  150,0,PI
```

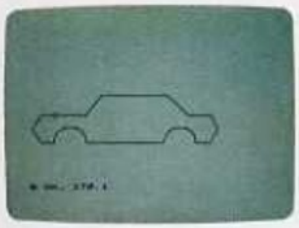
and press ENTER. You will see that you have drawn a semi-circle between the two points, starting from the first one, and moving in an *anti-clockwise* direction. If you wanted the semi-circle to be drawn in a *clockwise* direction, you would have used ,−PI instead of ,PI. Now you can return to the program. Press CLS and ENTER, then LIST and ENTER.

To get from L to M you want a semi-circle which moves in an anti-clockwise direction to a point 37 units to the left, and no units up or down. So try typing this line:

```
150  DRAW  −37,0,PI
```

The next two lines will now be easy:


```
160  DRAW  −88,0
170  DRAW  −37,0,PI
```



◀ *When you have drawn in the base of the car, the outline should look like this.*

You have now returned to Point A, from which you started. To draw the windows on the car, you will need to draw two more lines, and in each case you will need to move to a new position, using a PLOT instruction, before you can draw the line. Try adding these lines:

```
180  REM  The windows
190  PLOT  63,72
200  DRAW  118,0
210  PLOT  122,72
220  DRAW  0,23
```

Lines 190 to 200 draw a line from E to H. You can work out the PLOT position of E by starting with Point A, which was 27,40, and adding up all the numbers *less* the minus numbers which you needed to get to Point E. (So for the first number, you have 27−14−10+10+50, which equals 63; and for the second number you have 40+0+16+16+0, which equals 72.) To work out your instruction for drawing to H, add up the numbers less the minus amounts which you needed to get from E to H. (So for the first number you have 18+75+25, which equals 118, and for the second number you have 23+0−23, which equals 0.) Lines 210 and 220 draw a line from

half-way between E and H to half-way between F and G. See if you can work out why you were right when you used the numbers 122,72 and Ø,23. Next, you have the wheels to draw. For these, you need to know how to draw circles.

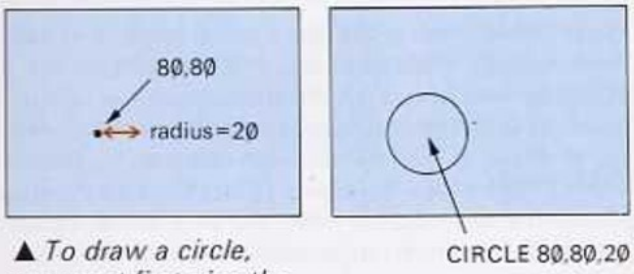
How to draw circles

Clear the screen by typing CLS and pressing ENTER. If you want to draw a circle, you first type the Keyword CIRCLE, which you will find in red beneath the H key. So your instruction begins:

```
CIRCLE
```

Next you type in the coordinates of the position of the *centre* of the circle, just as if you were giving the coordinates in a PLOT instruction. So, if you want the centre of the circle to be at 8Ø,8Ø, your line will now read:

```
CIRCLE  8Ø,8Ø
```



▲ *To draw a circle, you must first give the coordinates for the centre of the circle (such as 8Ø,8Ø). Then you must tell the computer how big (what radius) you want the circle to be (such as 2Ø).*

Next you decide how long the distance from the centre of the circle to the edge of the circle (the *radius*) is to be. Suppose you want the radius to be 2Ø, your line will now read:

```
CIRCLE  8Ø,8Ø,2Ø
```

Type this in and press ENTER, and you will see the circle on the screen. Now you can return to the program. Try adding these lines:

```
23Ø  REM  The wheels
24Ø  CIRCLE  45.5,4Ø,14
25Ø  CIRCLE  17Ø.5,4Ø,14
26Ø  STOP
```

Line 24Ø draws the wheel on the left. The centre of the circle was found by deciding that the centre of the wheel would be half-way between A and N. You know that the coordinates for A are 27,4Ø; and N is 37 to the right of A. So you need to add half of 37, which equals 18.5, to the number 27. This gives you 45.5 for the first number; and the second number will be the same, because the point is not further up or down than A. The size of the radius, 14, was found by trying several numbers until one of the circles looked just right. Line 25Ø draws the wheel on the right. See if you can work out why you were correct when you used the numbers 17Ø.5,4Ø,14. (To do this, simply work out the coordinates for M, and remember that the centre of the wheel is half way between M and L.)

Colouring the lines

Now you have drawn the car. But why not do your drawing in coloured lines, using INK instructions? The car could be red, its wheels black. Try:

```
 25 INK 2
235 INK 0
```

All the lines after line 25 will now be drawn in red. Then, after line 235, all the lines will be black again. You could easily add a coloured border to your drawing, as well. Try:

```
24 BORDER 6
```

◀ Once you have completed your drawing, you can make the computer draw the lines in colour. You could also add a coloured border.

Making the program shorter

You now have a program of 29 lines, many of which are extremely short. But you are allowed to put more than one instruction on the same line, provided that you separate the instructions by a : (colon), which you will find in red on the Z key. Try to do this so that the program is still easy to follow. For example, changing the colour of the lines is something which you may want to stand out clearly, so you may leave it on its own line. Also, remember that the computer ignores everything in a line after a REM, so it is no good having a REM instruction which you want the computer to obey! You might re-arrange the "CAR" program like this:

```
 10 REM "THECAR": REM The body
 20 BORDER 6
 30 INK 2
 40 PLOT 27,40: DRAW -14,0: DRAW
-10,16: DRAW 10,16
 50 DRAW 50,0: DRAW 18,23: DRAW 75,0:
DRAW 25,-23
 60 DRAW 25,0: DRAW 8,-16: DRAW
-8,-16: DRAW -17,0
 70 DRAW -37,0,PI: DRAW -88,0: DRAW
-37,0,PI
 80 REM The windows
 90 PLOT 63,72: DRAW 118,0
100 PLOT 122,72: DRAW 0,23
110 REM The wheels
120 INK 0
130 CIRCLE 45.5,40,14: CIRCLE 170.5,
40,14
140 STOP
```

Now you have a program of only 14 lines, which is just as easy to follow, and which takes up less space in the computer's memory. Now try experimenting with drawings of your own. How about a butterfly? Or a scene from outer space? Or a tank? Remember that although it is fun just to try things out, you will get better results if you work out your picture in advance on a piece of paper – preferably graph paper. It will also help to look at the measurements and diagram on page 102 of the *ZX Spectrum BASIC Programming* book.

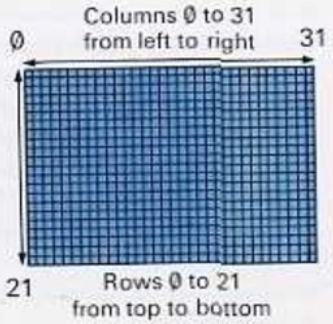
A reminder

You have now learned how to make the computer draw straight lines, semi-circles and circles, using the Keywords: PLOT, DRAW, PI and CIRCLE.
You have also learned how to use a : (colon) to place more than one instruction on the same line.

6 Drawing with Blocks of Colour

You have learned how to draw with lines. But sometimes you want to be able to *fill* an area with colour. To make this easier, the paper area of the screen is divided up by the ZX Spectrum Computer into 22 rows (from row Ø at the top to row 21 at the bottom). Each row has 32 squares (from square Ø at the left to square 31 at the right). (Remember that in this case, the lowest numbers are at the *top* of the paper area and the highest at the *bottom*. When you were drawing with lines, the numbers were lowest at the bottom of the paper area.) You can decide on the colour of the ink (what is printed) and the paper (the background) in each of these squares. More importantly, each of these squares can be exactly filled by any one of the GRAPHICS symbols shown on the number keys.

▶ *To print blocks of colour on your screen, think of the screen as rows of squares. There are 32 squares from left to right, and 22 squares from top to bottom.*



Colouring a single square

Type in:

```
PRINT AT 15,5; INK 3; PAPER 6;"■"
```


PRINT AT is the instruction you use when you wish to print something in a special place on the screen. Remember that AT is in red on the I key. The numbers, 15,5 tell the computer that the square in which you wish to print is in row number 15 (the sixteenth row from the top), and is square number 5 in that row (the sixth square from the left). Notice the position of the semi-colons. When you get to the Graphics symbol ■, remember that you need to get into [G] mode. As you know, you get in or out of [G] mode by keeping one finger pressed on the CAPS SHIFT key, while you press once on the 9 key. If you then press any one of the keys between 1 and 8, a Graphics symbol will be typed. The black area on the symbol will be the area filled with the chosen *ink* colour, and the white area with the *paper* colour. If you want these colours the other way round, then when you type the Graphics symbol you need to keep one finger on the CAPS SHIFT key, this will reverse the appearance of the black and white areas.

Remember to get back into [L] mode to put in the inverted commas at the end of the line. Then press ENTER, and you should see a square with magenta along the top, and yellow along the bottom, near the bottom left of the paper area.

Colouring a row of squares

Imagine that you want to colour a row of squares starting at 3,Ø and ending at 3,31. The first lines *could* look like this – but do not bother to type them in!

```
10 PRINT AT 3,Ø; INK 2; PAPER 5;"■"
20 PRINT AT 3,1; INK 2; PAPER 5;"■"
30 PRINT AT 3,2; INK 2; PAPER 5;"■"
40 PRINT AT 3,3; INK 2; PAPER 5;"■"
```

and so on. At this rate, it would take another 28 instructions to finish the row. Luckily, there is a way to make this much faster and easier, by using a FOR . . . NEXT loop. FOR is on the F key and NEXT is on the N key. Try typing these lines (remember that TO is in red on the F key):

```
10  FOR y=0 TO 31
20  PRINT AT 3,y; INK 2; PAPER 5;"■"
30  NEXT y
40  STOP
```

When you run this program, the row of colour appears very quickly. This is what is happening. The lines from 10 to 30 are a FOR . . . NEXT loop. The first time the computer reaches line 10, it is told that **y** has been given values all the way from 0 to 31. Because it is the first time it has reached the line, it has to choose the first value, which is 0. So at line 20 it prints the Graphics symbol at the square 3,0. Then, when it reaches line 30, it is told to go back to find the NEXT value of **y**. At line 10, it discovers that the next value of **y** is 1. So at line 20 it prints the Graphics symbol at 3,1 . . . and so on, until it has printed the symbol at 3,31. Then it finds no more values of **y**, and so it goes on to line 40 at last, and STOPs.

Colouring several rows of squares

Supposing that you wanted to colour *not only* the row of squares from 3,0 to 3,31, *but also* the five rows of squares beneath it (from 4,0 to 4,31; from 5,0 to 5,31; from 6,0 to 6,31; from 7,0 to 7,31; and from 8,0 to 8,31). You could do this by doing six lots of FOR . . . NEXT loops of three lines each, beginning like this – have a look at the following lines but do not type them in:


```
10  FOR y=0 TO 31
20  PRINT AT 3,y; INK 2; PAPER 5;"■"
30  NEXT y
40  FOR y=0 TO 31
50  PRINT AT 4,y; INK 2; PAPER 5;"■"
60  NEXT y
```

and so on. At this rate, it would take another 12 instructions to finish the block of colour. Again, there is a way to make this much faster and easier, by using one FOR . . . NEXT loop inside another FOR . . . NEXT loop. Try typing these lines:

```
10  FOR x=3 TO 8
20  FOR y=0 TO 31
30  PRINT AT x,y; INK 2; PAPER 5;"■"
40  NEXT y
50  NEXT x
60  STOP
```

When you run this program, the complete block of colour gradually appears. This is what is happening. The lines from 10 to 50 are a FOR . . . NEXT loop. The first time the computer reaches line 10, it is told that **x** has been given values all the way from 3 to 8. Because it is the first time it has reached the line, it has to choose the first value, which is 3. So **x** now equals 3. But before it can go on with this FOR . . . NEXT loop, it finds at line 20 that it has reached another FOR . . . NEXT loop, which runs from lines 20 to 40. So, with **x** at the value of 3, it has to get all the way through this FOR . . . NEXT loop, which is telling it to colour in row number 3. When it has done that, it reaches line 50, and is told to go back to find the NEXT value of **x**. Then, with **x** at the value of 4, it will colour in row number 4; and so on, until it has reached row 8, square 31. Then it finds no more values of **x**, and so it goes to line 60 at last, and STOPs.

Printing more than one block of colour in a program

Imagine that you wanted to use several different blocks of colour in one program. You can use the same set of instructions for every block, by making some of the items in those instructions *variable* items. You can then use this set of instructions with a GOSUB instruction, as you did in the "SOLDIER" program on page 53. Look at the lines to print a block of colour again. Six of the items in this set of instructions can be represented by a number variable, and one by a string variable. You can LET:

a = the first row on which squares are to be printed;
b = the last row on which squares are to be printed;
c = the first square that you want in each row;
d = the last square that you want in each row;
e = the number after INK;
f = the number after PAPER: and
g$ = the Graphics symbol between " "

You can now write out the set of instructions like this:

```
1010 FOR x=a TO b
1020 FOR y=c TO d
1030 PRINT AT x,y; INK e; PAPER f;g$
1040 NEXT y
1050 NEXT x
```

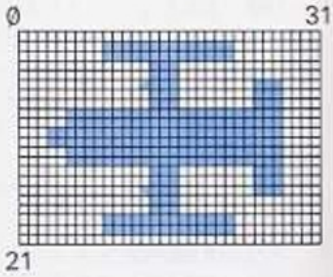
Now, whenever you want to put a block of colour in a program, you simply need a line to define the variables, and then a line to send a GOSUB to this set of instructions. Try typing out this program, which has seven blocks of colour, and two single squares filled in with colour (see page 73 for instructions on the way to reverse the Graphics symbols):

```
10 REM "SPACESHIP"
20 REM Colour main hull
30 LET a=8: LET b=13: LET c=5: LET d=24: LET e=5: LET f=3: LET g$="■"
40 GOSUB 1000
50 REM Stabilizers at wing ends
60 LET a=1: LET b=2: LET c=9: LET d=21
70 GOSUB 1000
80 LET a=19: LET b=20
90 GOSUB 1000
100 REM Wings
110 LET a=3: LET b=7: LET c=14: LET d=16: LET e=2: LET f=7: LET g$="■"
120 GOSUB 1000
130 LET a=14: LET b=18
140 GOSUB 1000
150 REM Tail section
160 LET a=5: LET b=16: LET c=25: LET d=27
170 GOSUB 1000
180 REM The nose
190 LET a=10: LET b=11: LET c=3: LET d=4
200 GOSUB 1000
210 REM The guns
220 PRINT AT 5,13; INK 1;"■"
230 PRINT AT 16,13; INK 1;"■"
240 STOP
1000 REM A block of colour
1010 FOR x=a TO b
1020 FOR y=c TO d
1030 PRINT AT x,y; INK e; PAPER f;g$
1040 NEXT y
1050 NEXT x
1060 RETURN
```

Press RUN to check that it is working properly. If not, then press BREAK SPACE and press LIST and check your program carefully.

Now look at the program again. In line 30 you had to define all the variables. But you will notice that in line 60 you only had to define some of them. This was because the others were the same as they had been in line 30, and did not need defining again. Now try experimenting with blocks of colour to make your own pictures. Remember that although it is fun just to try things out, you will get better results if you work out your picture on graph paper first. The diagram on page 102 of the *ZX Spectrum BASIC Programming* book shows the full layout on your television screen of the area in which you can work.

▶ *When you are planning any drawings of your own, you will find it a lot easier if you plot them out first, like this, on a piece of graph paper.*



A reminder

You have now learned the PRINT AT instruction, and the use of FOR ... NEXT loops.

7 "GHOSTHUNT"

This is a very long program for what is basically a guessing-game. There is a haunted house with a ghost in one of the rooms, and you have to guess which room it is in. The nub of the game is that the ghost appears in a different room each time. You will also find that there are two new ways of doing things in this program.

Use of long variable names for number variables

Until now you have used single letters in statements such as:

```
LET a=20
```

But you can use names instead, which sometimes make it easier to see what is happening. For example, you could type:

```
LET My height=5
```

Or:

```
LET wealth=20000
```

In "GHOSTHUNT", you will find at line 30 the variable *Ghostroom*, and at line 40 the variable *Score*. The only rule is that these long variable names must begin with a letter.

Use of RND

This is a very important instruction, which means

that you can invent games with an element of chance in them. You will find RND in green above the T key. You will also need to find the Keyword INT, which is in green above the R key. The following statement would make the computer print, at RaNDom, any *whole* number between 1 and 10:

```
PRINT  1+INT (RND*10)
```

Or:

```
PRINT  1+INT (RND*8)
```

will make the computer choose a number between 1 and 8; and so on. In this program, there is a house with eight rooms; and the room with the ghost in it is called the Ghostroom. At line 30 the instruction:

```
LET  Ghostroom=1+INT (RND*8)
```

means that the computer chooses a number between 1 and 8; and, until a new game begins, that number is the number of the Ghostroom. Later in the program, at line 2270, the player has to guess that number. If the player guesses correctly, the game soon comes to an end, and the computer will choose another number between 1 and 8 for the Ghostroom in the next game.

If the guess is wrong, the player has a chance to guess again. If the player guesses right first time, the Score will be 6. Every wrong guess puts the Score down by 1 point. If you get down to 0 points, the ghost will eat you! So you only have six chances to guess the right number of the Ghostroom.

Now, study the program carefully to work out the order in which things are going to happen. You will see that many GOSUBs have been used, which have split the program up into chunks, each of which should be reasonably easy to understand. The main body of the program ends at line 130, and then there are all the GOSUBs.

(As this is a very long program, you might want to enter it in stages. It's worth remembering that you can SAVE part of your program and then LOAD it back into the computer later, when you want to continue. How to SAVE and LOAD programs is explained in the next chapter.)

Now, type in the program, and run it:

```
10  REM  "GHOSTHUNT"
20  GOSUB 1000
30  LET Ghostroom=1+INT (RND*8)
40  LET Score=6
50  GOSUB 2000
60  IF a<>Ghostroom THEN GOSUB 3000
70  IF Score=0 THEN GOTO 100
80  IF a <>Ghostroom THEN GOTO 50
90  GOSUB 4000
100 GOSUB 7000
110 IF b$="y" THEN GOTO 20
120 IF b$<>"n" THEN GOTO 100
130 STOP

1000 REM Introduction
1010 BORDER 4
1020 PRINT:PRINT:PRINT
1030 PRINT INK 2; FLASH 1;"_ _ _ _ _
_ _ _ _ _GHOSTHUNT_ _ _ _ _ _ _
_ _ _ _ _ _ _"
1040 GOSUB 6500
1050 PRINT:PRINT:PRINT
1060 PRINT INK 1;"_ _ _ _ _ _ _ _ _ _ _ _
_ _by"
1070 PRINT
1080 PRINT INK 1;"_ _ _The_Graves_
```

```
Family _ (1984)"
1090  PRINT:PRINT
1100  PRINT "What _ is _ your _ name?"
1110  INPUT a$: PRINT a$
1120  GOSUB 6000
1130  PRINT:PRINT
1140  PRINT INK 4;"Soon, _ ";a$;" _ you _
will _ see"
1150  PRINT INK 4; FLASH 1;" _ _ _ _ _ _
_ _ A _ HAUNTED _ HOUSE _ _ _ _ _ _ _
_ _ "
1160  PRINT:PRINT:PRINT
1170  GOSUB 6500
1180  PRINT INK 3;"Press _ ENTER _ to _
continue"
1190  INPUT z$: CLS : RETURN
2000  REM Ghosthouse and Guess
2010  REM Main house
2020  LET a=8: LET b=19: LET c=4: LET
d=27: LET e=6: LET f=7: LET g$="■":
GOSUB 5000
2030  REM The roof
2040  LET a=4: LET b=7: LET c=8: LET
d=23: LET e=2: LET g$="▀": GOSUB 5000
2050  LET a=6: LET c=6: LET d=7:
GOSUB 5000
2060  LET c=24: LET d=25: GOSUB 5000
2070  PRINT AT 7,5; INK 2;"▀": PRINT AT
5,7; INK 2;"▀"
2080  PRINT AT 7,4; INK 2;"▗": PRINT AT
6,5; INK 2;"▗": PRINT AT 5,6; INK 2;"▗":
PRINT AT 4,7; INK 2;"▗"
2090  PRINT AT 7,26; INK 2;"▀": PRINT AT
5,24; INK 2;"▀"
2100  REM Upstairs windows
2110  LET a=9: LET b=12: LET c=5: LET
d=8: LET g$="■": GOSUB 5000
2120  LET c=11: LET d=14: GOSUB 5000
2130  LET c=17: LET d=20: GOSUB 5000
2140  LET c=23: LET d=26: GOSUB 5000
2150  REM Downstairs windows
2160  LET a=15: LET b=17: LET c=5: LET
d=7: GOSUB 5000
2170  LET c=10: LET d=12: GOSUB 5000
2180  LET c=19: LET d=21: GOSUB 5000
2190  LET c=24: LET d=26: GOSUB 5000
2200  REM The door
2210  LET b=19: LET c=14: LET d=17: LET
e=3: LET g$="■": GOSUB 5000
2220  PRINT AT 0,0; INK 4; FLASH 1;" _ _ _
_ HERE _ IS _ THE _ HAUNTED _ HOUSE _
_ _ "
2230  PRINT INK 4;"with _ 8 _ rooms. _
Where _ is _ the _ GHOST"
2240  PRINT AT 2,8; INK 1;a$;" _ ?"
2250  GOSUB 6000
2260  PRINT AT 21,0; INK 2;"NOW _
ENTER _ YOUR _ GUESS _ FROM _ 1 _
TO _ 8"
2270  INPUT a: CLS : RETURN
3000  REM Score after wrong answer
3010  PRINT
3020  LET Score=Score-1
3030  PRINT INK 2;"BAD _ LUCK _ ";a$
3040  PRINT:PRINT:PRINT:PRINT
3050  PRINT INK 4;"The _ GHOST _ was _
not _ in _ room _ "; FLASH 1;a
3060  PRINT:PRINT:PRINT:PRINT
3070  IF Score=0 THEN PRINT FLASH 1;
BRIGHT 1; INK 2;"OOPS! _ YOU _ HAVE _
BEEN _ EATEN _ BY _ THE _ GHOST"
3080  IF Score=0 THEN PRINT:PRINT
3090  IF Score=0 THEN GOSUB 6500
3100  IF Score=0 THEN PRINT INK 1;"Please
_ press _ ENTER"
3110  IF Score=0 THEN GOTO 3160
```

```
3120  PRINT INK 1;"Your _ possible _ score _
is _ now _ down _ from _ 6 _ to _ "; INK 2;Score
3130  PRINT:PRINT:PRINT:PRINT
3140  GOSUB 6500
3150  PRINT INK 1;"Press _ ENTER _ when _
you _ are _ "; INK 2;"brave _ "; INK 1;" _
enough _ to _ have _ another _ try!!!"
3160  IF Score=0 THEN LET a=Ghostroom
3170  INPUT z$: CLS : RETURN

4000  REM Score after right answer
4010  PRINT PAPER 2; INK 6; FLASH 1;
BRIGHT 1;" _ _ _ _ WELL _ DONE _ _ _ _ ";
FLASH 0;" _ _ ";a$: GOSUB 6000
4020  PRINT
4030  PRINT INK 0; BRIGHT 1;" _ _ _ The _
GHOST _ was _ in _ room _ "; FLASH 1;a;
FLASH 0;" _ _ _ _ _ _ "
4040  REM The GHOST
4050  REM GHOST's mouth
4060  INK 4
4070  PLOT 103,47
4080  DRAW 52,0: DRAW 0,16: DRAW -8,-8:
DRAW -9,8: DRAW -9,-8: DRAW -9,8:
DRAW -9,-8: DRAW -8,8: DRAW 0,-16
4090  REM GHOST's eyes
4100  INK 0
4110  CIRCLE 110, 90,7: CIRCLE 146, 90,7
4120  REM GHOST's outline
4130  PLOT 79,25: DRAW 0,60: DRAW 104,0,
-PI: DRAW 0,-60
4140  GOSUB 6500
4150  PRINT:PRINT:PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:
PRINT:PRINT:PRINT:PRINT:PRINT
4160  PRINT INK 1;"Your _ score _ this _ time _
is _ ";FLASH 1; BRIGHT 1;Score
4170  PRINT INK 1;"Thanks _ for _ the _
game, _ ";a$
4180  GOSUB 6000
4190  PRINT INK 1;"Now, _ press _ ENTER."
4200  INPUT z$: CLS : RETURN

5000  REM A block of colour
5010  FOR x=a TO b
5020  FOR y=c TO d
5030  PRINT AT x,y; INK e; PAPER f;g$
5040  NEXT y
5050  NEXT x
5060  RETURN

6000  REM Cheerful tune
6010  BEEP 1,5: BEEP .5,9: BEEP .5,0: BEEP 1,11
6020  BEEP 1,5: BEEP .5,0: BEEP .5,5: BEEP .5,9:
BEEP .5,11: BEEP 1,12
6030  RETURN

6500  REM Sinister tune
6510  BEEP 1,5: BEEP .5,9: BEEP .5,0: BEEP 1,11
6520  BEEP 1, -15: BEEP .5, -13: BEEP .5, -15:
BEEP .5, -17: BEEP .5, -15: BEEP 1, -13
6530  RETURN

7000  REM New game or stop
7010  BORDER 5
7020  PRINT:PRINT:PRINT:PRINT:PRINT:
PRINT
7030  PRINT INK 3;"If _ you _ want _ another _
game _ of"
7040  PRINT
7050  PRINT INK 2;" _ _ _ _ _ _ _ _ _ _
GHOSTHUNT"
7060  PRINT
7070  PRINT INK 3;a$;", _ then _ enter _ y _
for _ YES"
7080  PRINT
```

```
7090  PRINT INK 3;"_ _ _ _ _ _ _ _ _or_n_
for_NO."
7100  INPUT b$: CLS : RETURN
```

Run the program a few times to enjoy playing the game. Now think about whether you can improve it:
1) Can you improve the interest of the game? For example, you might want to add lines which will tell you if your guess was more or less than the right number, as you did for the earlier program, "GUESSME". You could fit this in between lines 3030 and 3050, where at the moment there are four blank lines on the screen.
2) Can you improve the small details of the program? For example, you might want to change some of the sounds, or the colours of the house.
3) Can you improve the overall structure of the game? For example, you might like the picture of the ghost to appear more than once. The lines from 4040 to 4130 which draw the ghost could become some GOSUB lines starting at 8000. Then you could call up the ghost lines between lines 3020 and 3030, and follow them by a few bars of spooky music, so that they stayed on the screen for a few seconds. Then you could clear the screen, and go on to the 'Bad luck' message in line 3030.

A reminder

You have learned to use long variable names for number variables. More importantly, you have learned how to use the instruction 1+INT (RND*(a number)). This is one of the most useful instructions when you are inventing games. (You could now turn back to the "GUESSME" game and try to turn it into a game for *one* player, by making the computer choose a number between 1 and 20 at random!)

Saving and Loading Programs

Saving programs

You have learned to give each program a title in the first line, using not more than ten letters and/or numbers, between sets of inverted commas. For example, the first line of the program for thank-you letters was:

```
10  REM "THANKYOU"
```

Spaces between words *can* be included in the title, provided that, when added together, the total number of letters, numbers and spaces is not more than ten. To save the "THANKYOU" program on tape, you must:
1) Type the list of program lines for "THANKYOU" into the computer.
2) Type:

```
SAVE "THANKYOU"
```

You will find the SAVE Keyword on the S key.
3) From the back of the ZX Spectrum keyboard, pull out the plug from the *EAR* socket.
4) Press ENTER, and at once the following message should appear:

```
Start tape, then press any key.
```

If you have tried to save a title with a total of *more* than ten numbers, letters and spaces, then the following message will appear instead, and you will have to begin again:

```
F Invalid file name, Ø:1
```

5) Press the RECORD lever on your tape recorder (on some recorders you have to press PLAY as well).
6) Press any key, *except* the CAPS SHIFT or SYMBOL SHIFT keys.
7) Watch the screen. In the border area, you will see wide red and blue stripes, then very narrow blue and yellow stripes, then a blank screen, then more wide red and blue stripes, then more very narrow blue and yellow stripes. Then the stripes should disappear and the computer should print a message near the bottom of the screen, saying:

```
Ø OK, Ø:1
```

Checking that a program has been properly saved

1) Replace the plug in the *EAR* socket at the back of the keyboard.
2) Rewind the tape to just before the start of where you saved the "THANKYOU" program.
3) Type:

VERIFY "THANKYOU"

You will find the VERIFY Keyword in red beneath the R key.
4) Press ENTER, and at once the border area of the screen will go blank.
5) Press the PLAY lever on your tape recorder. The border area will change between light blue and red, until the computer finds that it is listening to the beginning of the "THANKYOU" program. Then you will see wide red and blue stripes, then very

narrow blue and yellow stripes, then on the screen you will see:

```
Program: THANKYOU
```

Then there will be a red border, then more very narrow blue and yellow stripes. Then the stripes should disappear, and the computer should print a message near the bottom of the screen, saying:

```
Ø OK, Ø:1
```

Your program has been saved on tape, and you can switch off the computer without losing it.

Loading programs

To load your program back into the computer:
1) Type:

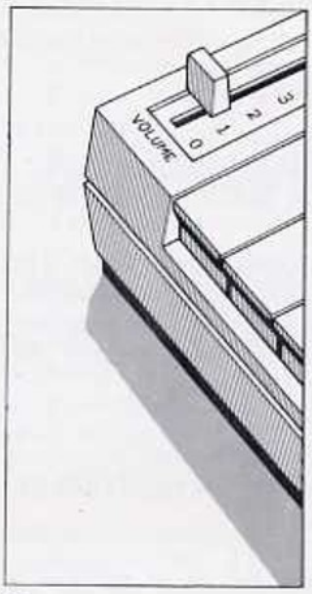
LOAD "THANKYOU"

You will find the LOAD Keyword on the J key.
2) Press ENTER, and the border area goes blank.
3) Rewind the tape to just before the start of the "THANKYOU" program.
4) Press the PLAY lever on your tape recorder. The screen will look exactly as it did after you pressed the PLAY lever when you were verifying your program, and when the 'OK' message has appeared the program is loaded and ready to be run.

Some common problems with verifying or loading programs

1) The tape recorder may be set at too low a

volume, in which case turn up the volume.
2) The recorder may be dirty, in which case play a cleaning tape.
3) The batteries may be running low, in which case put in new ones – unless you are on mains power.
4) You may have started the tape in the middle of the program, instead of just before the beginning of the program; so rewind the tape and try again.
5) You may have made a mistake in typing the name of the program which you are trying to verify or load. If this seems possible, try again.
6) You may have forgotten to put the plug back in the *EAR* socket. If so, start again. (Or you may have forgotten to take it out when saving the program. To check this, unplug the computer leads from the tape recorder and play the section of tape on which you think the program has been saved. If it was saved, you will hear some very loud noises!)



◀ *If the volume control on your tape recorder is not adjusted properly, you may have trouble loading your programs into your computer. You may need to adjust the volume control a number of times before you find the right position for your tape recorder.*

9 A List of Programming Words

Here you will find all the programming words which you have learned, together with a brief reminder about how they are used and what they do. Look at this chapter carefully because you will also find a few new ways of doing things.

BEEP (in red beneath Z key) This instruction is used for creating sounds.

BORDER (on B key) This instruction, followed by one of the number keys which control colour, will change the colour of the area all around the edge of the screen. So:

BORDER 1

will give dark blue, and so on.

BRIGHT (in red beneath B key) There are two levels of brightness on the screen. After the instruction:

BRIGHT 1

in a program, everything that is printed on the screen will be brighter.

BRIGHT 0

returns the brightness to its normal level. BRIGHT can be used within a PRINT statement. For example:

PRINT BRIGHT 1;
"Happy";
BRIGHT 0; , _sad."

CIRCLE (in red beneath H key) This instruction will draw a circle. It is followed by three numbers like this:

CIRCLE 50,60,25

The first two numbers give the coordinates of the centre of the circle, and the third number gives the radius of the circle.

CLS (on V key) This instruction clears the central area of the screen of any text or graphics, and colours it with whatever paper colour has previously been set. (If no colour is set, this will be white.)

DRAW (on W key) This instruction will draw a straight line from the last point in a program to a new point. For example:

DRAW 30,60

tells the computer to draw a line 30 points to the right and 60 points up from the present

position. If no starting position is given, the computer draws from the bottom left-hand area of the screen. This instruction can also draw a semi-circle if used with PI (in green above M key). Thus:

DRAW 30,60,PI

will draw a semi-circle in an anti-clockwise direction from the last point reached in the program to a point which is 30 points to the right and 60 points up. (Putting a — (minus) sign before PI would have made the computer draw the semi-circle in a clockwise direction.)

FLASH (in red beneath V key) After the instruction:

FLASH 1

in a program, everything that is then printed on the screen will 'flash' as the ink and paper colours switch back and forth. The instruction:

FLASH 0

stops the flashing. FLASH can be used within a PRINT statement. For example, you might have:

PRINT FLASH 1;
"HELP!"

FOR (on F key) This is the first instruction in a FOR . . . NEXT loop. It makes the computer carry out a task a particular number of times. Thus:

```
10  FOR a=1 TO 6
20  PRINT a
30  NEXT a
```

will print (on separate lines) the numbers 1, 2, 3, 4, 5 and 6.

GOSUB (on H key) GOSUBs are a very useful way of keeping a program tidy and easy to follow. For example:

GOSUB 1000

sends the computer to line 1000 and the following lines, until the instruction RETURN is reached, when it jumps back to the line immediately after the GOSUB 1000 from which it set out. This means that a single GOSUB line in the main program, in this case GOSUB 1000, can be used to call up the same set of lines again and again.

GOTO (on G key) This makes the computer jump to a certain line in the program.

IF (on U key) This is used at the start of an IF . . . THEN statement to say that IF something is true, THEN the computer must do a certain thing. For example:

IF a=b THEN PRINT "Right!"

INK (in red beneath X key) Is used with a number key to change the colour of what is printed or drawn on the screen, and also the ink part of the Graphics symbols. So:

INK 3

will mean that words or line drawings, or the ink part of Graphics symbols, will all appear on the screen in magenta. INK can be used within PRINT statements. Thus:

PRINT INK 3;"How_are_you,_";INK 1;"Robin?"

INPUT (on the I key) This instruction is used to put information into the computer:

1) It can enter numbers. For example:

```
50  PRINT "Enter_the_spending_money_you_get"
60  INPUT b
```

When you type in the amount, for example, 15, and then press ENTER the number 15 will be placed in a store marked **b**; and if in future you give the instruction:

PRINT b

then the number 15 will appear on the screen as soon as you have pressed ENTER.

2) It can also enter words:

```
210  PRINT "How_will_you_sign_your_name?"
220  INPUT f$
```

When you type in your name, for example, Janet, then the word Janet will be placed in a store marked f**$**, and if in future you give the instruction:

PRINT f**$**

then the word Janet will appear on the screen as soon as you have pressed ENTER.

INT (above the R key in green) See RND below.

LET (on L key) This instruction is used to store information in the computer;

1) It can store numbers. For example, on page 27, you used:

LET a=20

When you did that, the number 20 was stored in the computer under the label **a**.

2) It can store words. For example, you might have a line saying:

LET a**$**="pocketwatch"

This statement enters the word 'pocketwatch' in a store labelled a**$**.

LIST (on K key) A useful statement used to make the computer list out whatever program it has in its memory.

LOAD (on J key) This

instruction is used when loading a program into the computer.

NEW (on A key) When you type NEW and press ENTER, any program lines stored in the computer's memory will be rubbed out.

NEXT (on N key) See FOR above.

PAPER (in red beneath C key) This instruction, followed by one of the number keys which control colour, will change the colour of the background behind what is printed on the screen. It will also change the colour of the paper part of the Graphics symbols.

PAUSE (on M key) The instruction:

PAUSE 50

stops the program running for exactly one second (PAUSE 100 would give two seconds, and so on).

PI (above M key in green) See DRAW above.

PLOT (on Q key) This instruction places a dot on the screen. For example:

PLOT 20,100

places a dot at the co-ordinates given, in this case, 20 units to the right, and 100 units up.

PRINT (on P key) This instruction is usually used to print something on the screen.

REM (on E key) The computer ignores everything in a line after a REM statement, so REM statements are very useful for REMinding yourself what a program is about.

RETURN (on Y key) See GOSUB above.

RND (above the T key in green) This instruction makes the computer choose a number at RaNDom.

PRINT RND

will print a random number between 0 and 1 (but never reaches 1).

RUN (on the R key) This instruction tells the computer to work its way through the program which is in its memory.

SAVE (on S key) This is used when saving a program onto tape.

STOP (in red on A key) This statement marks the place where a program ends.

THEN (in red on G key) See IF above.

VERIFY (in red beneath R key) This is used to check that a program has been properly saved onto tape.

Index

Some words such as PRINT are on many pages. So as not to waste your time, this Index lists only the most useful examples.

Editorial: Christopher Maynard
Design: Ben White
Covers: Pinpoint Design

Published for Marks and Spencer p.l.c.
by Grisewood & Dempsey Computer Publishing
40 Bowling Green Lane, London EC1 0NE

START TO PROGRAM

is a beginner's guide to writing computer programs in BASIC. The pack contains a colourfully illustrated manual for absolute beginners and two cassettes of do-it-yourself software.

Cassette 1 includes five short programs for you to alter by playing with different shapes, colours, graphics and sound commands. Cassette 2 has *GHOSTHUNT* — an adventure puzzle which can be rewritten from start to finish (if you wish) with the help of a friendly computer tutor that almost never laughs at your mistakes.