



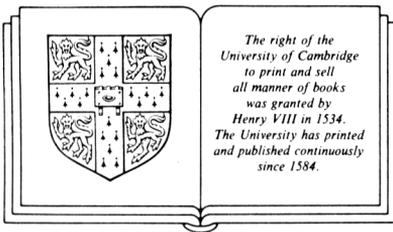
A child's guide to the **AMSTRAD MICRO**

John Dewhirst
and James Ryan



A CHILD'S GUIDE TO THE AMSTRAD MICRO

John Dewhirst
James Ryan



CAMBRIDGE UNIVERSITY PRESS
Cambridge
London New York New Rochelle
Melbourne Sydney

Published by the Press Syndicate of the University of Cambridge
The Pitt Building, Trumpington Street, Cambridge CB2 1RP
32 East 57th Street, New York, NY 10022, USA
10 Stamford Road, Oakleigh, Melbourne 3166, Australia

© Cambridge University Press 1984

First published 1984

British Library cataloguing in publication data

Dewhirst, John

A child's guide to the Amstrad micro

1. Amstrad computers – Juvenile literature

I. title II. Ryan, James

001.64'04 QA76.8.A4

ISBN 0 521 31561 1

Produced by Chancereel Publishers Ltd, 40 Tavistock Street,
London WC2.

Designed by Valerie Sargent.

Cover and cartoons by David Parkins.

Printouts by James Ryan.

Typesetting and graphic reproduction by ReproSharp Ltd, 47 Farringdon
Road, London EC1.

Printed by Richard Clay (The Chaucer Press) Ltd, Bungay, Suffolk, England.

Acknowledgements

The authors and publisher would like to thank Amstrad Consumer Electronics plc for the loan of an Amstrad microcomputer during the preparation of this book.

CONTENTS

Introduction	page 4
The keyboard	7
Control keys	
Letter keys	
Number keys	
Symbol keys	
Edit keys	
Getting started	21
The computer as a typewriter	
The computer as a calculator	
The computer memory	
Using the computer for storing information	
Using the computer to ask for information	
Special features	36
Changing the screen	
Drawing on the screen	
Moving on the screen	
Plotting points	
Changing colours	
Making music	
Writing programs	46
Writing programs	
Projects	
Finding out	81
Reference sections	
Editing and writing programs	
Using the screen	
Making sounds	
Word functions	
Number functions	
Operators	
Saving and loading programs	
General commands and functions	

INTRODUCTION

Each section of the book is written by a different person. Each person is an expert at their job and gives very useful advice. Read their advice very carefully. The advice is given at the start of their section of the book. If you follow the advice it will make the work you do much easier.

SECTIONS

1 The keyboard

by Pru Comet – typist.

Ms Comet shows you the ways in which the various types of keys work and the position of the keys on the keyboard.



2. Getting started

by P. C. Truemo – investigator.

P. C. Truemo shows you how to get started on the computer, how to make it print numbers and words, calculate answers and finally store items in its memory.



3. Special features

by Mort Puce – artist.

Mort Puce shows you how to draw coloured pictures on the screen and to make the computer play music.



4. Writing programs

by Prof. O. Crumpet – designer.

Professor O. Crumpet shows you how to copy, adapt, then finally design your own programs.



5. Finding out

by Ms O. C. Termup – librarian.

Ms O. C. Termup shows you how to use the different reference sections.



Dear Reader,

Hello there. We are the people who have written this book and you will find out more about us as you read through it. When you meet us we shall introduce ourselves.

The book has been written for people who know nothing about computers, but who want to find out for themselves. It has been written to show you not just what the computer can do, but more importantly, what *you* can make the computer do for you.

We have put together all our good ideas so that it is easy for you to find out just how the Amstrad microcomputer works. The book is written in five sections and we have each written a section. The first four sections are to be read and worked through in order, but the last section is for reference and this can be used at any time for checking.

Happy computing!

Prudence Comet

Prudence Comet

Bobby Truemo

Bobby Truemo

Mortimer Puce

Mortimer Puce

Oliver Crumpet

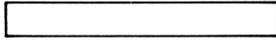
Oliver Crumpet

Olive Constance Termup

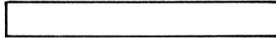
Olive Constance Termup

In each section of the book you will come across different types of diagrams. The things for you to try, that is various exercises and experiments, will all be given a 'You try' box.

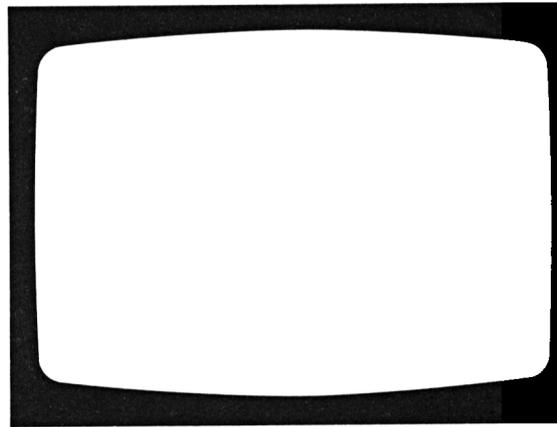
The 'You try' box looks like this.



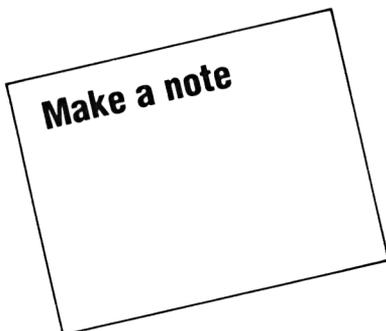
You try



The things which are displayed on the screen will all be shown in a 'Display' box. The 'Display' box looks like this.



The things which you need to make a note about and remember will all be given in a 'Make a note' box. The 'Make a note' box looks like this.



THE KEYBOARD



Pru Comet

My advice

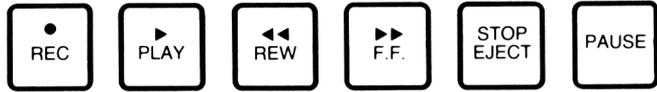
Don't be afraid of the computer. I think of it as a typewriter with a difference. Try things out. In this way you will learn quickly just what it can and cannot do. When you try things out, the computer will send messages back to you. This is the way the computer talks to you. It can make noises, change colours on the screen and do lots of other things, so you'll need a bit of time to learn about everything the computer can do. Your first job is to learn about the keyboard, find out where all the keys are and what they all do. A good idea is to make your own notes in a book of all the things you find out about the computer.

As you read through the keyboard section of the book, try things out as you go along. Don't be put off by any messages the computer sends to you. It does not realise that you are a beginner and just trying out a few things for yourself. As you work through the rest of the book you will learn the language that the computer understands, but like learning any language you can only learn a bit at a time.

Pru Comet

There are 80 keys on the Amstrad microcomputer. The six on the far right-hand side control the datacoder, and they work just like the buttons on an ordinary tape recorder.

CPC 464 User Instructions F1.1



The other 74 keys can be divided into five groups:

Control keys
Letter keys
Number keys

Symbol keys
Edit keys



Before you start looking at these keys in detail, there are a couple of things you need to know.

1. If you keep your finger on a key it will repeat itself, and instead of a you will get **aaaaaaaa**.

2. You can use **DEL** (which is short for 'delete') and **CLR** (which stands for 'clear') as rub out keys. That is, by touching **DEL** or **CLR** you can remove any unwanted characters. But beware! These keys work like the others, and if you keep your finger on them you could rub out more than you intended.

Control keys

There are ten control keys.

ESC

TAB

CAPS

LOCK

SHIFT (two keys)

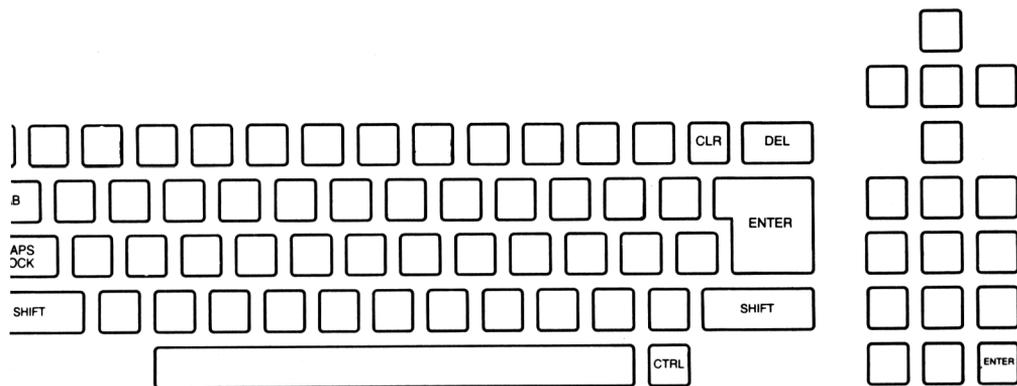
DEL

CLR

ENTER (two keys)

CTRL

CPC 464 User Instructions F2.1



ESC is coloured red. It is used to escape from what the computer is doing. If the computer is just sitting waiting for instructions, then pressing **ESC** will put this message on the screen:



If the computer is in the middle of carrying out a program, pressing **ESC** once will make it stop. If you then press any other key, it will start again from where it left off. But if you press **ESC** a second time, the computer will break out of what it is doing completely, and will be ready for you to type in more instructions.

TAB is coloured green. Pressing **TAB** will print a small arrow pointing to the right. This key does not repeat itself if you hold it down.



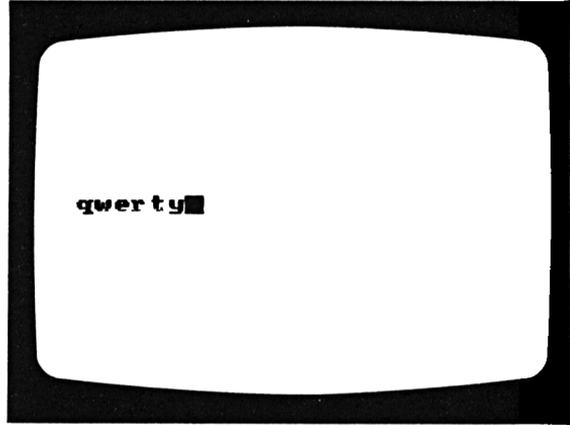
CAPS LOCK

This key is coloured green. When it has been pressed once, the letter keys print capitals and the symbol keys print the lower symbols or numbers. Pressing **CAPS LOCK** a second time switches it off, so the letter keys print small letters again. When the machine is first switched on, the **CAPS LOCK** is off.

This is what will appear on the screen

You try

Type **QWERTY**

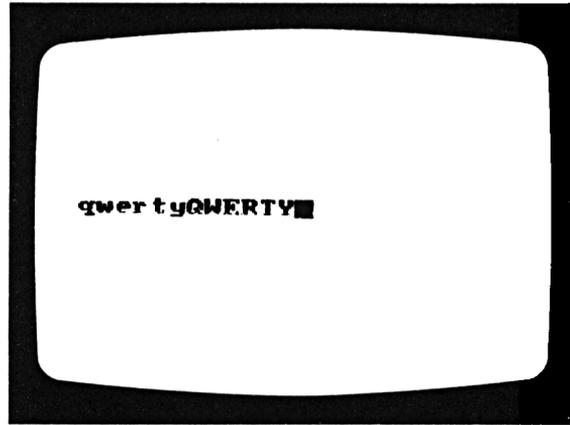


This is what will appear on the screen

You try

Press **CAPS LOCK** and type **QWERTY** again.

Now press **CAPS LOCK** a second time so that you can print small letters again.



Make a note
When **CAPS LOCK** is on, you can type capital letters.



CAPS LOCK

SHIFT These two keys are coloured green. If you hold either of them down, the letter keys print capitals and the symbol keys print the upper symbols. Once you let **SHIFT** go, you will get small letters and lower symbols or numbers again.

This is what will appear on the screen:

You try

Press in turn each of the keys on the top row of the keyboard marked **123456**



This is what will appear on the screen:

You try

Now hold down **SHIFT** and press the same keys **123456** again.



Make a note

Holding down **SHIFT** lets you type capital letters and upper symbols.

DEL is coloured green. It rubs out the character to the left of the cursor. The cursor is the solid block which shows you where the next character you type will appear on the screen.

This is what will appear on the screen

You try

Press the **DEL** key six times.



Make a note

Each time **DEL** is pressed the cursor moves one place to the left and rubs out any letter or number which is there.

CLR is coloured grey. It rubs out characters which are underneath the cursor. You can move the cursor left and right with the help of the arrow keys on either side of the **COPY** key. When the cursor is over the character that you want to rub out, press **CLR** and the character will disappear. Other characters to the right of the cursor on the same line are moved over to fill the gap.

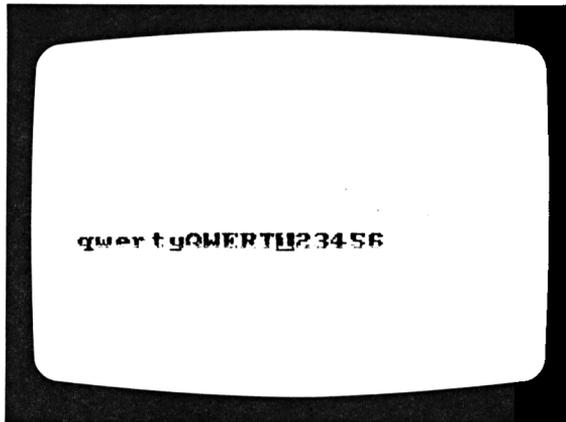


This is what will appear on the screen:

You try

Move the cursor over the letter **Y** by pressing the left arrow key seven times. Now press

CLR





ENTER There are two of these keys, a large one and a smaller one among the number keys on the right. They are both blue. **ENTER** moves the cursor to the start of the next line.

This is what will appear on the screen:

You try

Press **ENTER**
(either key will do).

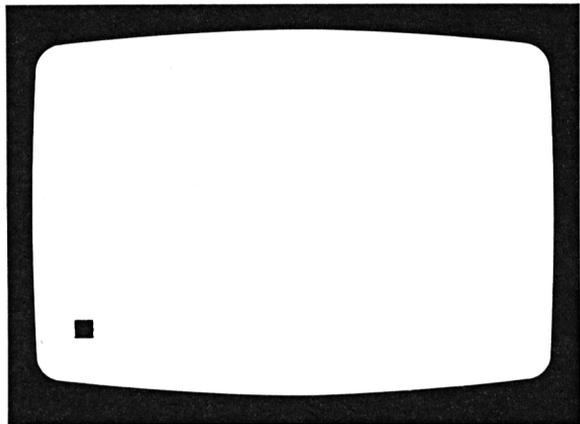


Don't worry about the message saying **Syntax error** – we will soon learn how to stop that appearing.

This is what will appear on the screen:

You try

Press **ENTER** 24 times.



Make a note

Pressing **ENTER**
moves the cursor
down one line and
scrolls the screen.

Notice that when the cursor got to the bottom of the screen, the words above it jumped up one line each time you pressed the **ENTER** key. This is called 'scrolling'.

CTRL This is coloured green. When pressed together with other keys, it carries out various special jobs. Here are two of them:

1. Hold down **CTRL** and press **CAPS LOCK** once. Now pressing the letter keys will produce capital letters, while the symbols keys will produce the upper symbols. This works just the same way as the **SHIFT** key, but it saves you the trouble of holding **SHIFT** down all the time. If you press both **CTRL** and **CAPS LOCK** a second time, the computer will return to the state it was in before – producing small letters and numbers if **CAPS LOCK** was off, and capitals and numbers if it was on.

Make a note
CTRL and **CAPS LOCK**
together have
the same effect
as holding down
SHIFT.

2. Hold down **CTRL**, press **SHIFT** and hold that down too, and finally press **ESC**. You will see the 'wake up' message appear, just as it does when you first turn the computer on.



You try

Press **CTRL**,
SHIFT and
ESC all at the
same time.

This is what will appear on the screen:



Make a note
CTRL, **SHIFT**
and **ESC** pressed
together reset the
computer. They act
like the on/off
switch.

This is called 'resetting' the computer. It makes the computer forget everything you have told it.



Letter keys

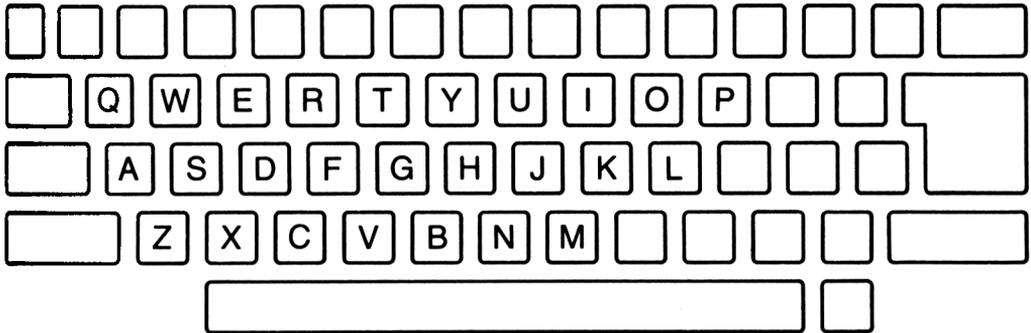
There are 26 keys which print the letters of the alphabet:
A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U,
V, W, X, Y, Z

and

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y,
z.

At the bottom of the keyboard you will also find a long key with nothing written on it. This is the space bar, and it moves the cursor one space to the right without printing anything.

On the keyboard these keys are set out in exactly the same way as you would find them on a typewriter.



You try

Type out the following. Press **ENTER** at the end of each line.

1ST ROW QWERTYUIOP

2ND ROW ASDFGHJKL

3RD ROW ZXCVBNM

You try

Type out the names of the members of your family and their birthdays. Press **ENTER** at the end of each line.

You try

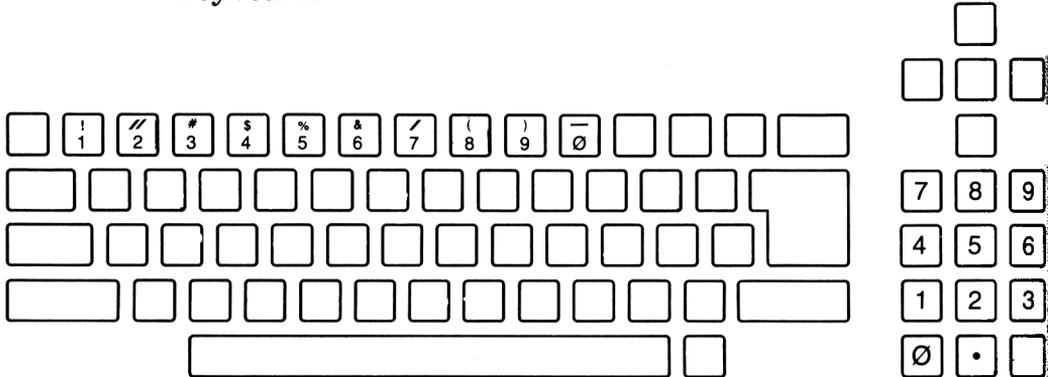
Type out: **I can use the keyboard.** Press **ENTER**.

If you do not like the computer saying **Syntax error** then put a number at the beginning of each line.

When you are ready to move on to the next section of this book, press **CTRL**, **SHIFT** and **ESC** together to reset the computer.

Number keys

There are ten keys on the top row of the main keyboard which print the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9 and Ø. You will find the same ten numbers grouped together in the 'numeric keypad', along with a full stop (or decimal point) and the small **ENTER** key, to the right of the main keyboard.



The Ø key is used for zero so that it is not confused with the letter O. The numeric keypad is useful when you are typing in a lot of numbers. It will also still print numbers when the number keys on the main keyboard are printing symbols because you have pressed **CTRL** and **CAPS LOCK**.



You try

Type out these figures. Press **ENTER** at the end of each line.

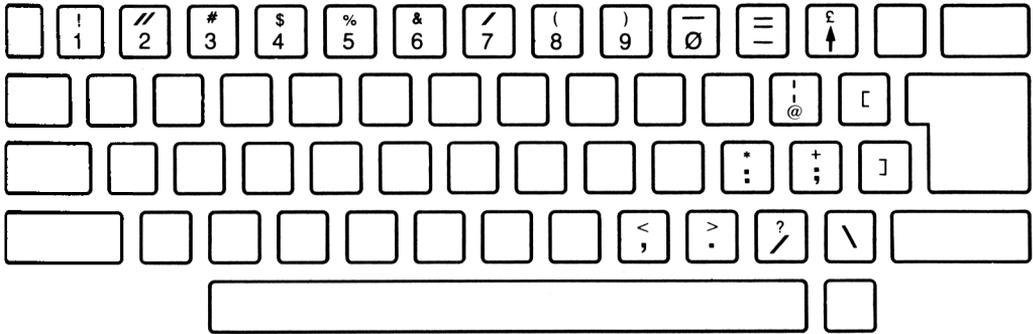
1
121
1331
14641

If you do not like the computer saying **Line does not exist** then put a letter at the end of each line before you press **ENTER**.

Reset the computer by pressing **CTRL**, **SHIFT** and **ESC** together when you are ready to move on to the next section.

Symbol keys

The 11 keys which print symbols are grouped together on the right-hand side of the main keyboard. There are also some more symbols above the numbers on the main keyboard, and to use these you must hold down one of the **SHIFT** keys or press **CTRL** and **CAPS LOCK**.



You try

Type out the following. Press **ENTER** at the end of each line.

1 !

2 "

3 #

4 \$

5 %

6 &

7 '

8 (

9)

Ø -

You try

Type out the following. Press **ENTER** at the end of each line.

1 punctuation ! " ' ; : , . ?

2 arithmetic % () + - * / ↑

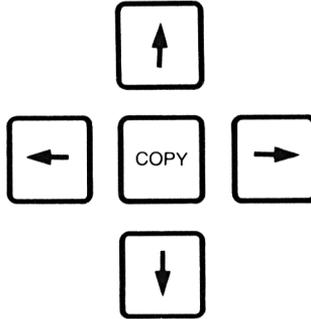
3 other \$ & @ £ [] \ ! : -

When you are ready to move on, reset the computer by pressing **CTRL**, **SHIFT** and **ESC** together.

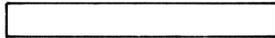
Edit keys

These are the four arrow keys and the **COPY** key, grouped together above the numeric keypad.

CPC 464 User Instructions 1.15



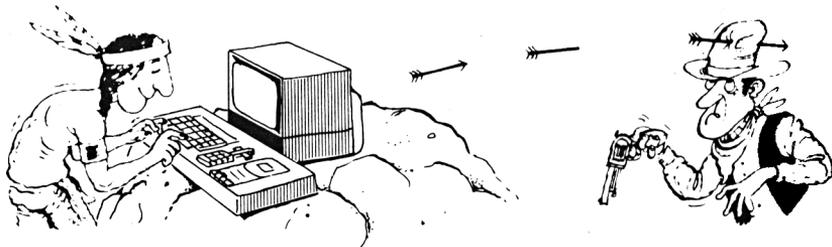
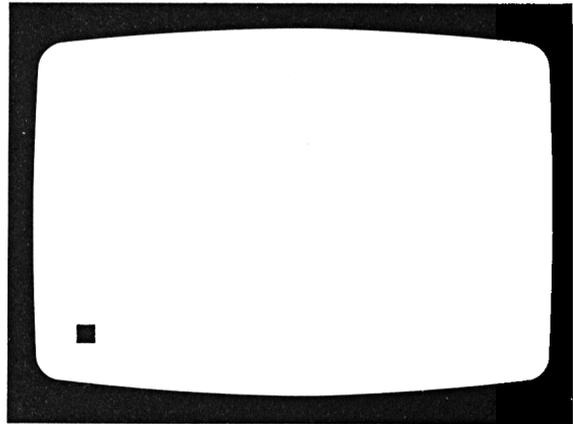
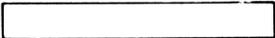
The arrow keys move the cursor one space in the direction the arrow is pointing each time they are pressed. If you keep the arrow keys pressed down you can zoom around the screen. Hold the up or down arrows long enough, and anything written on the screen will scroll out of sight.



This is what will appear on the screen:

You try

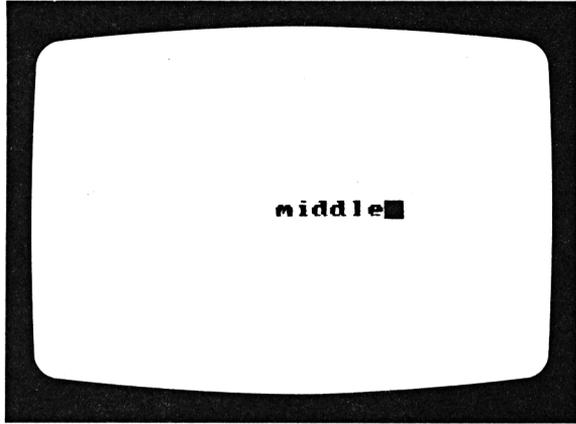
Hold down the arrow pointing down until all the writing has disappeared from the screen.



This is what will appear on the screen:

You try

Use the arrow keys to move the cursor to the middle of the screen. Now type **middle**



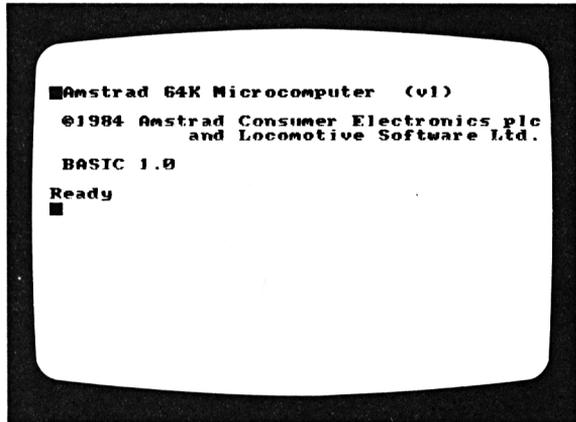
You now know how to write anywhere on the screen.

If you hold down **SHIFT** while you press an arrow key, a second cursor will appear and will move in the direction of the arrow. This is the copy cursor.

This is what will appear on the screen:

You try

Reset the computer. Hold down **SHIFT** and press the up arrow eight times.



*Find all the keys,
And type with ease.*



This is what will appear on the screen

You try

Press the **COPY** key until the copy cursor has reached the end of the words on the top line.

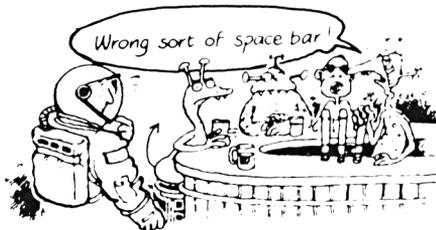
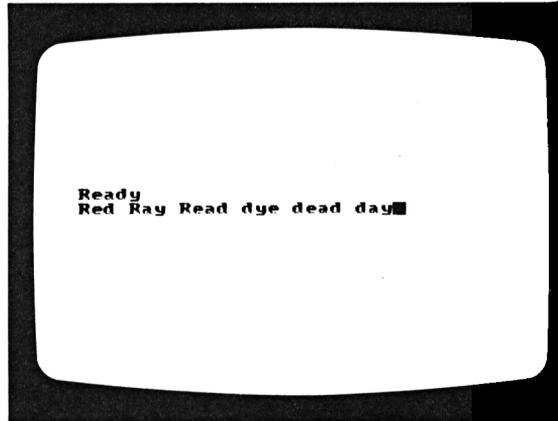


Notice how the computer copies a letter, number or space each time you press the **COPY** key. You can let go of **SHIFT** when you are using **COPY** , by the way!

This is what will appear on the screen

You try

Press **ENTER** .
 Now use **SHIFT** ,
 the arrow and
COPY keys, and
 the space bar to copy
 the following words
 from the 'Ready'
 sign: **Red Ray Read
 dye dead day**



GETTING STARTED



P. C. Truemo

My advice

When you start to make the computer do things for you always check them very carefully. The computer is very exact in the way it understands things. A missing space, a full stop in the wrong place or a wrong letter will result in the computer not understanding you.

When this happens – INVESTIGATE – track down every possible mistake until you get everything correct. It may take some time at first to track down your mistakes, but gradually you will not only become quicker at doing this, but you will also make fewer mistakes.

Make notes of the things you find useful, so that you can use them again easily whenever you want. Finally I find these points very useful.

1. Remember that the computer is very particular about where you put spaces.
2. Remember that after every instruction you must press

ENTER .

Bobby Truemo

Bobby Truemo

In this section we will find out how to use the computer in four different ways:

1. Using the computer as a typewriter.

We will discover how to print numbers and characters on the screen and how to space them out in various ways.

2. Using the computer as a calculator.

We will learn how to add, take away, multiply and divide numbers using the computer.

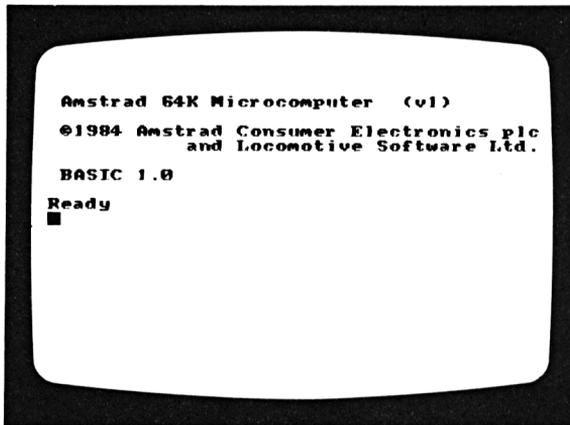
3. Using the computer for storing information.

We will find out how to put numbers into the memory of the computer.

4. Using the computer to ask for information.

We will learn how to get the computer to ask for information, which it then puts into its memory.

When switched on the Amstrad microcomputer will display the following:



A missing space, a full stop in the wrong place or a wrong letter will result in the computer not understanding you.





Amstrad Microcomputer (v1) – the make of computer and which version it is.

64K – the size of the computer's memory.

© 1984 Amstrad Consumer Electronics plc and Locomotive Software Ltd. – the copyright line. It tells you that you are not allowed to make copies of the machine or the programs that come with it unless you have the permission of Amstrad and Locomotive Software.

BASIC 1.0 – the computer language used by the Amstrad computer.

Ready – the signal that the computer is ready for you to type in instructions.

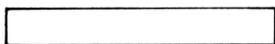
■ This is the cursor. It shows you where the letters that you type will appear on the screen.

The computer as a typewriter

To make the computer write things we use **print**. After the statement is complete, remember to press **ENTER** .

CPC 464 User Instructions 8.54

Printing numbers



This is what will appear on the screen:

You try

Type **print 5** then press **ENTER** .
You must put a space after **print**.



Make a note

To print a number, type **print** followed by a space and then the number, then press **ENTER** .



You try

Type **print 675** then press .

You try

Using the same idea, print some numbers of your own.

This is what will appear on the screen

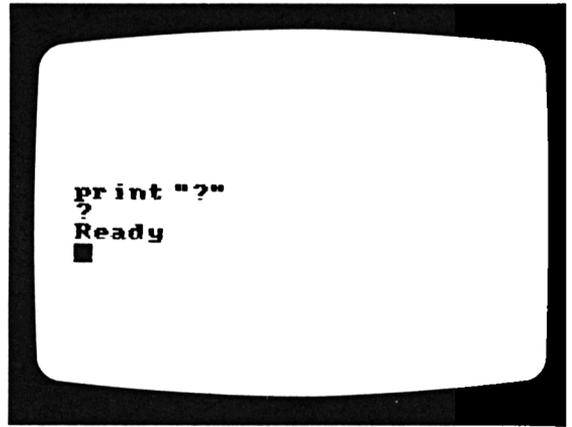


Printing symbols, letters and words

You try

Type **print "?"** then press .

This is what will appear on the screen:



Make a note

To print a symbol, letter or word type **print** followed by a space, then the symbol, letter or word inside speech marks, then the press

.



This is what will appear on the screen:

You try

Type **print "bee"**
then press .

```

print "bee"
bee
Ready
█

```

You try

Use **print** to make
the computer write
your name.



Spacing numbers

This is what will appear on the screen:

You try

Type **print 1,2,3** then
press .

```

print 1,2,3
1          2          3
Ready
█

```

This is what will appear on the screen:

You try

Type **print 1;2;3** then
press .

```

print 1;2;3
1 2 3
Ready
█

```



You try

Type **print 12,,34**
then press

You try

Using the same ideas
space out some
numbers of your
own.

This is what will appear on the screen



Make a note

The symbol
, prints spaced out
; prints numbers
with two spaces
in between
,,, prints on two
lines

*Don't stand and wait,
INVESTIGATE.*

Spacing words

You try

Type **print**
"dog", "cat" then
press

This is what will appear on the screen





You try

Type print
"dog";"cat" then
press .

This is what will appear on the screen:



You try

Type print
"dog",,"cat" then
press .

This is what will appear on the screen:



You try

Using the same idea
space out your first
and second names.

Make a note

- prints words spaced out
- ; prints words without spaces
- ,, prints words on two lines





The computer as a calculator

The **print** command can also be used to make the computer act like a calculator. It can be given a question and it will supply the answer. Remember that after the statement is complete **ENTER** should be pressed.

This is what will appear on the screen

You try

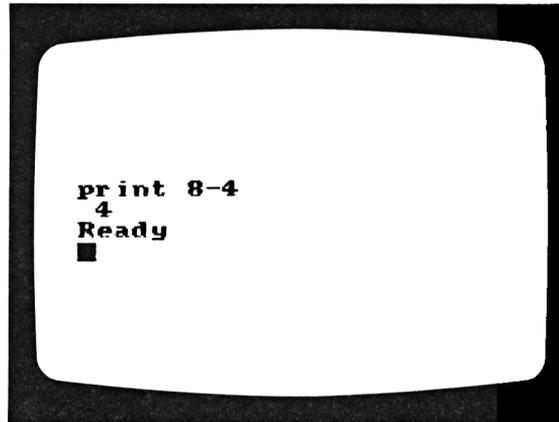
Type **print 5+7** then press **ENTER** .



This is what will appear on the screen

You try

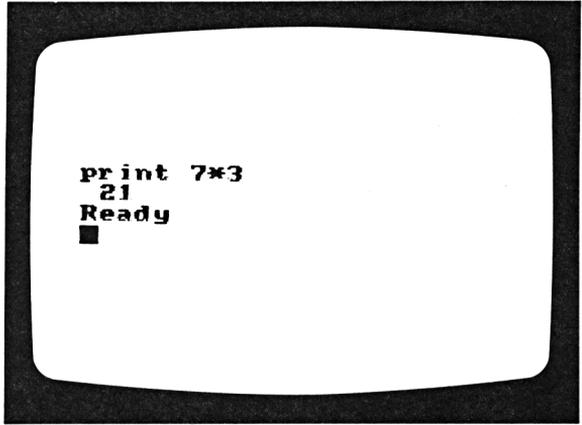
Type **print 8-4** then press **ENTER** .



This is what will appear on the screen:

You try

Type **print 7*3** then
press **ENTER** .



This is what will appear on the screen:

You try

Type **print 8/4** then
press **ENTER** .



You try

1. Using the same idea make up some of your own sums.
2. Check a shopping bill with the computer.

Make a note

The symbols used
are

- + add
- subtract
(or take away)
- * multiply
(or times)
- / divide (or share)





The computer memory

You can also use the computer to remember information. It is important to know just how the computer does this before you go on to the next part of this section.

Just as you live at one address, your friend at another, your teacher at yet another address, so the computer uses a similar system of addresses for storing information.

It stores numbers at the following addresses in its memory. The addresses can be either letters or words. For example



It stores words at addresses like these.



Make a note

The computer uses **a, b, c, d** etc. for addresses to store numbers and uses **a\$, b\$, c\$, d\$** etc. for addresses to store words.



Using the computer for storing information



The = sign is used to put numbers or words into the memory of the computer.

This is what will appear on the screen:

You try

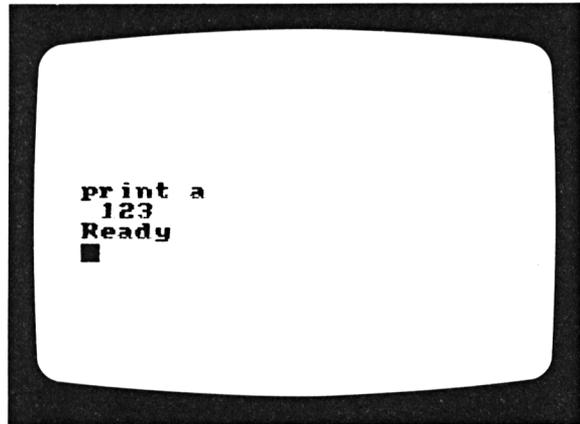
Type **a=123** then press **ENTER** .



This is what will appear on the screen:

You try

Type **print a** then press **ENTER** .



You try

Using the = sign put the number 12 into address **b** and number 4 into address **c**. Then type **print b+c,,b-c,,b*c,,b/c** then press **ENTER** .

a=123 put the number 123 into the memory of the computer at address **a**. **print a** recalled the number which was in the memory of the computer at address **a**.



This is what will appear on the screen

You try

Type **a\$="good"**
then press



This is what will appear on the screen

You try

Type **print a\$** then
press



a\$="good" put the word **good** into the memory of the computer at address **a\$**. **print a\$** recalled the word which was in the memory of the computer at address **a\$**.

You try

Use the = sign to
put **l** in the address
b\$ and **am** in the
address **c\$**.

Type **print c\$,b\$,a\$**
and press

Type **print b\$,c\$,a\$**
then press

Using the computer to ask for information

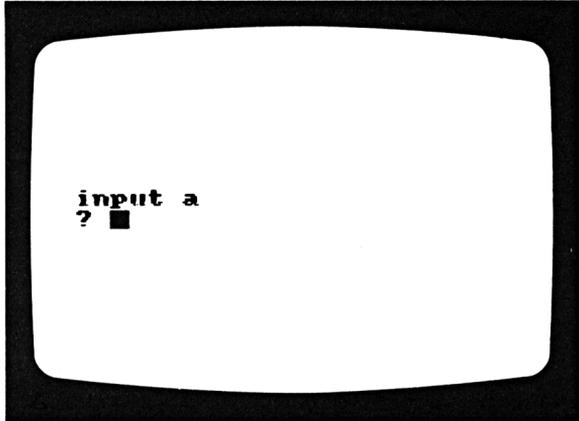
CPC 464 User Instructions F2.6

The input command is used to make the computer ask for numbers or words which it then puts into a memory address.

This is what will appear on the screen:

You try

Type **input a** then press .



This is what will appear on the screen:

You try

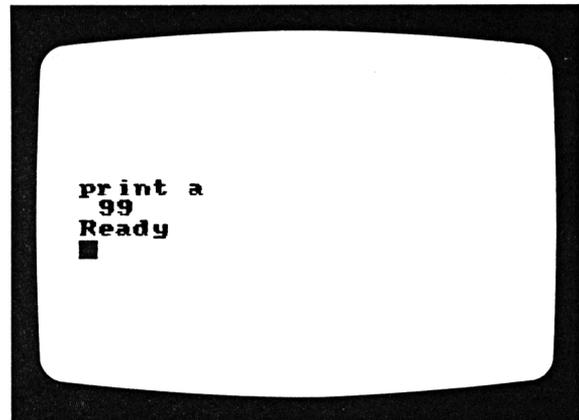
Type **99** then press .



This is what will appear on the screen:

You try

Type **print a** then press .





input a made the computer ask for a number. When the number **99** was typed, the computer put **99** into the memory address **a**. **print a** recalled the number which was in the memory of the computer at address **a**.

You try

Use the **input** statement to make the computer ask for some numbers to put into the memory addresses **b** and **c**. (? is asking for a number.) Type **print b,c,b+c,b*c** then press .

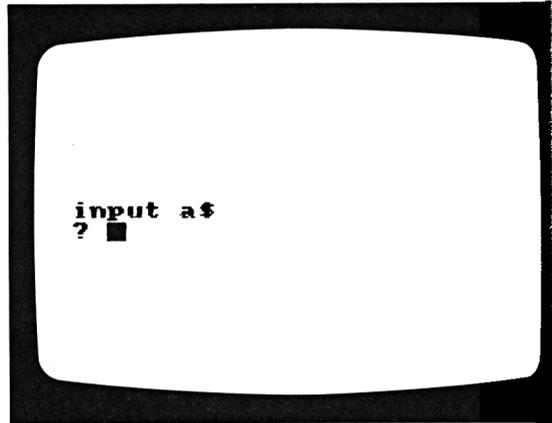
Make a note

1. **input a** makes the computer ask for a number.
2. The **?** sign is the way the computer asks for a number.
3. The number the computer is given goes into the memory address **a**.

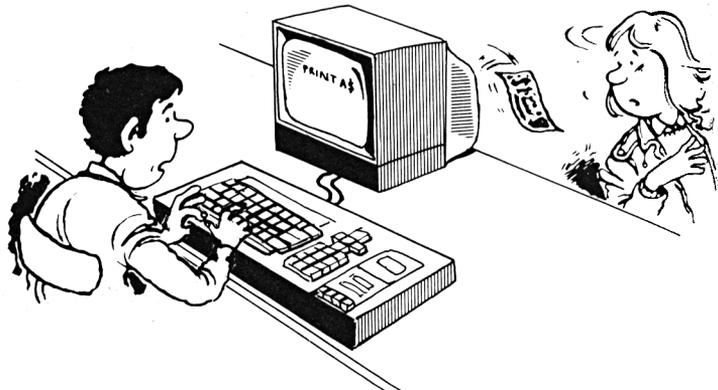
You try

Type **input a\$** then press .

This is what will appear on the screen



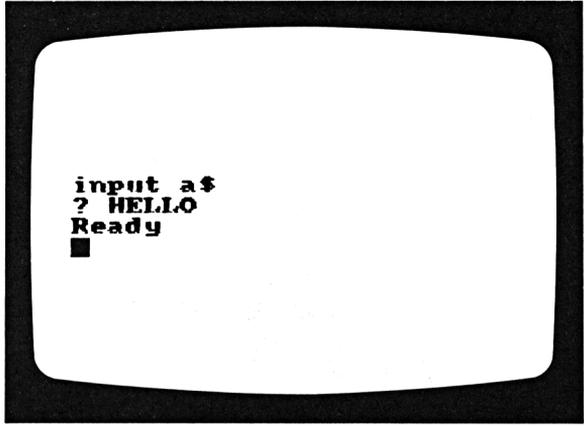
*If in doubt,
Try things out.*



This is what will appear on the screen:

You try

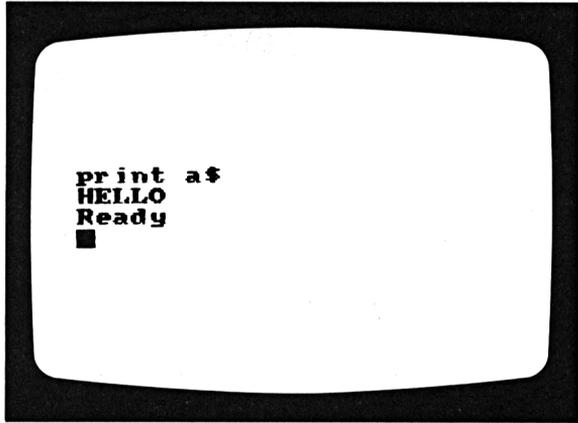
Type **HELLO** then
press **ENTER** .



This is what will appear on the screen:

You try

Type **print a\$** then
press **ENTER** .



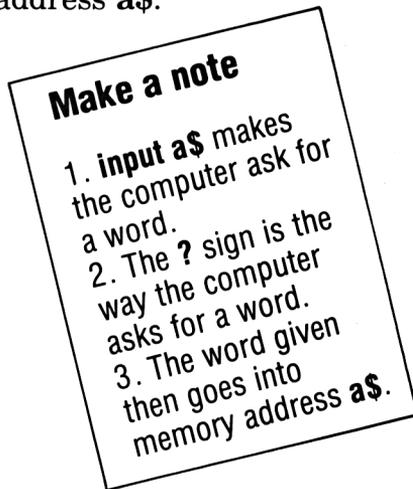
*What the computer can do,
is just up to you.*

input a\$ made the computer ask for a word. When the word **HELLO** was typed in, the computer put **HELLO** into the memory address **a\$**. **print a\$** recalled the word which was in the memory address **a\$**.

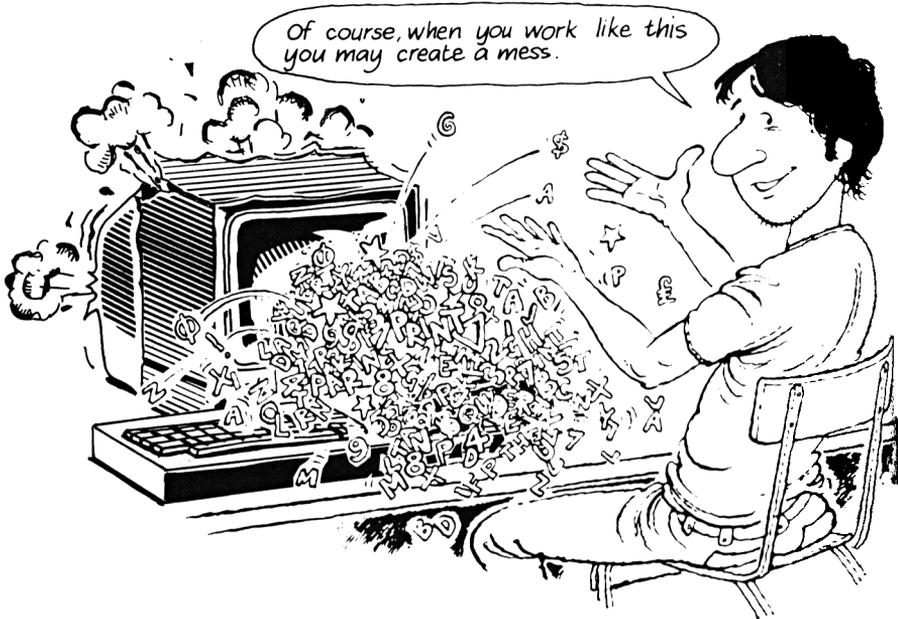


You try

Use the **input** statement to put your name into the address **b\$**. (? is asking for a word.)
Type **print a\$,b\$** then press **ENTER** .



SPECIAL FEATURES



Mort Puce

My advice

My advice is to look at new commands carefully. When you think you understand the idea, experiment and try things out. You will learn what looks right. Of course, when you work like this you may create a mess. Don't be put off: try again. Remember your successes and forget about your failures.

Try out all the ideas you can think of. Explore lines, angles, shapes, colours and even music. You can draw almost anything on the screen and colour it in different ways and then to celebrate your masterpiece compose a tune or even a complete symphony.

When you hit on a good idea make sure you write some notes about it, so that you will be able to use the idea again.

Mort Puce

Changing the screen

The way in which characters are written out on the screen depends on the mode in which the computer is operating. When you switch it on, the Amstrad microcomputer is in mode 1. In mode 1 there are 40 characters to the screen line and 25 lines on the screen. However, the computer will operate in modes 0 and 2 as well.

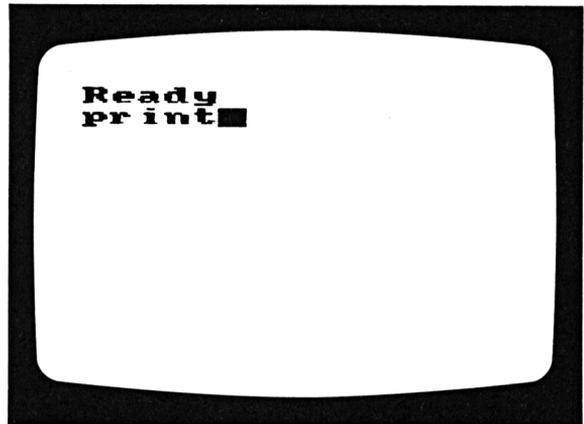
To change the mode, type **mode** followed by a space and then the number of the mode that you want. For example, **mode 0** followed by pressing **ENTER** puts the computer into mode 0. In this mode there are only 20 characters to the line. In mode 2 there are 80 characters to the line. There are always 25 lines on the screen, whichever mode you choose.

CPC 464 User Instructions 5.3

This is what will appear on the screen:

You try

Type **mode 0** then
press **ENTER**
Now type **print**
(notice the change).

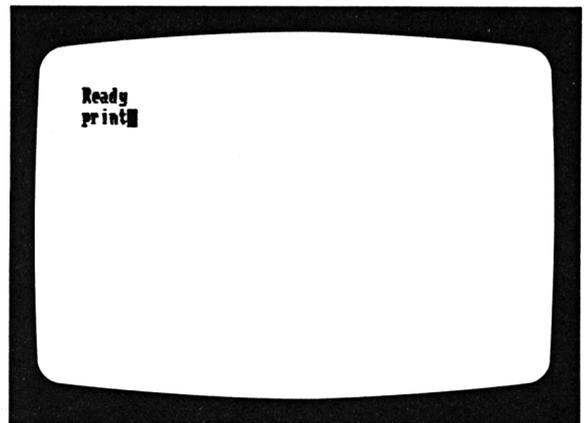


*Get on the right road,
Using the best MODE.*

This is what will appear on the screen:

You try

Type **mode 2** then
press **ENTER** .
Type **print** again.



Drawing on the screen

CPC 464 User Instructions F3.1

The **draw** command is used to draw lines and shapes. To make the computer draw, the **draw** command must be followed by a space, then two numbers separated by a comma. For example

draw 320,200

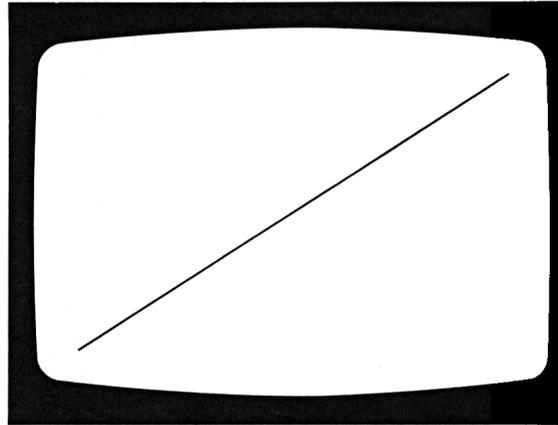
draw 639,0

Assuming that you want what you are drawing to appear on the screen, the first number can be any number from 0 to 639. The second number can be any number from 0 to 399,



The **draw** command works in a similar way to drawing on a piece of paper without lifting the pencil. The first number after **draw** gives the distance across the screen, and the second number the distance up the screen, at which the line will finish.

This is what will appear on the screen

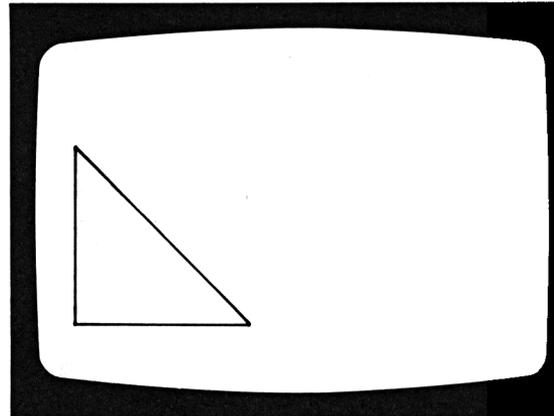


You try

Type **mode 2** then
press .

Type **draw 639,399**
then press .

This is what will appear on the screen



You try

Type **mode 2** then
press .

Type **draw 320,0**
then press .

Type **draw 0,200**
then press .

Type **draw 0,0** then
press .



You try

Type **mode 2** then
press .

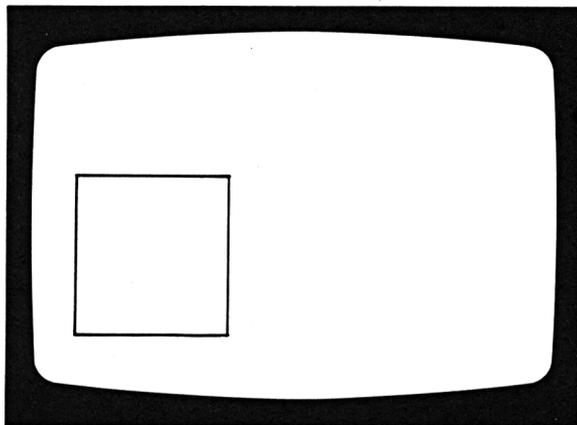
Type **draw 200,0**
then press .

Type **draw 200,200**
then press .

Type **draw 0,200**
then press .

Type **draw 0,0** then
press .

This is what will appear on the screen:



Moving on the screen

The **move** command is used to move on the screen without drawing a line. When using the **draw** command you will find from time to time that you need to move to another point on the screen without actually drawing a line. This can be done using **move**. Just as **draw** was like drawing on a piece of paper without lifting the pencil, **move** allows you to take the pencil off the paper and start to draw again at another point.

This is what will appear on the screen:

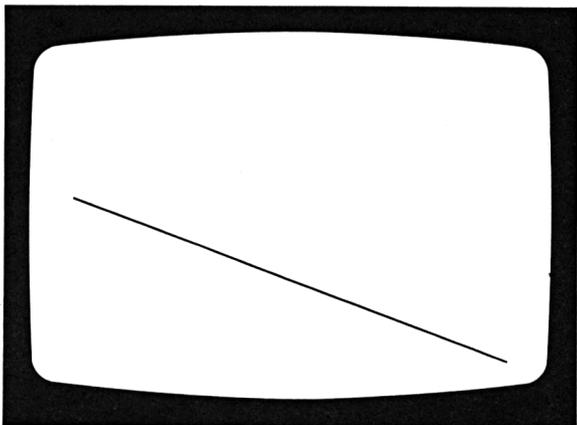


You try

Type **mode 2** then
press .

Type **move 0,200**
then press .

Type **draw 639,0**
then press .



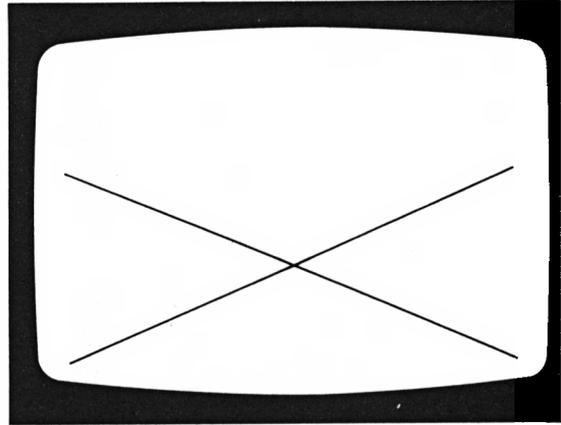
[]

This is what will appear on the screen

You try

Type **mode 2** then
 press **ENTER** .
 Type **draw 639,200**
 then press **ENTER** .
 Type **move 639,0**
 then press **ENTER** .
 Type **draw 0,200**
 then press **ENTER** .

[]



Plotting points

CPC 464 User Instructions F3.1

The **plot** command is used to move to a point on the screen and then draw a dot. Like **move**, you do not draw a line on the way to the point. The **plot** command is useful when you want to light up individual points on the screen, and when you want to draw shapes that are not made up of straight lines.



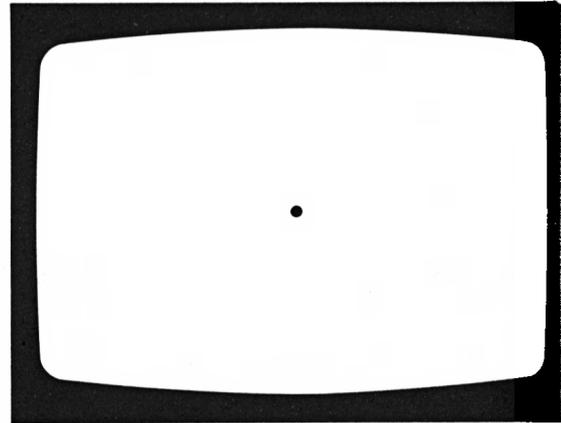
[]

This is what will appear on the screen

You try

Type **mode 0** then
 press **ENTER** .
 Type **plot 320,200**
 then press **ENTER** .
 Try plotting points in
 modes 1 and 2.

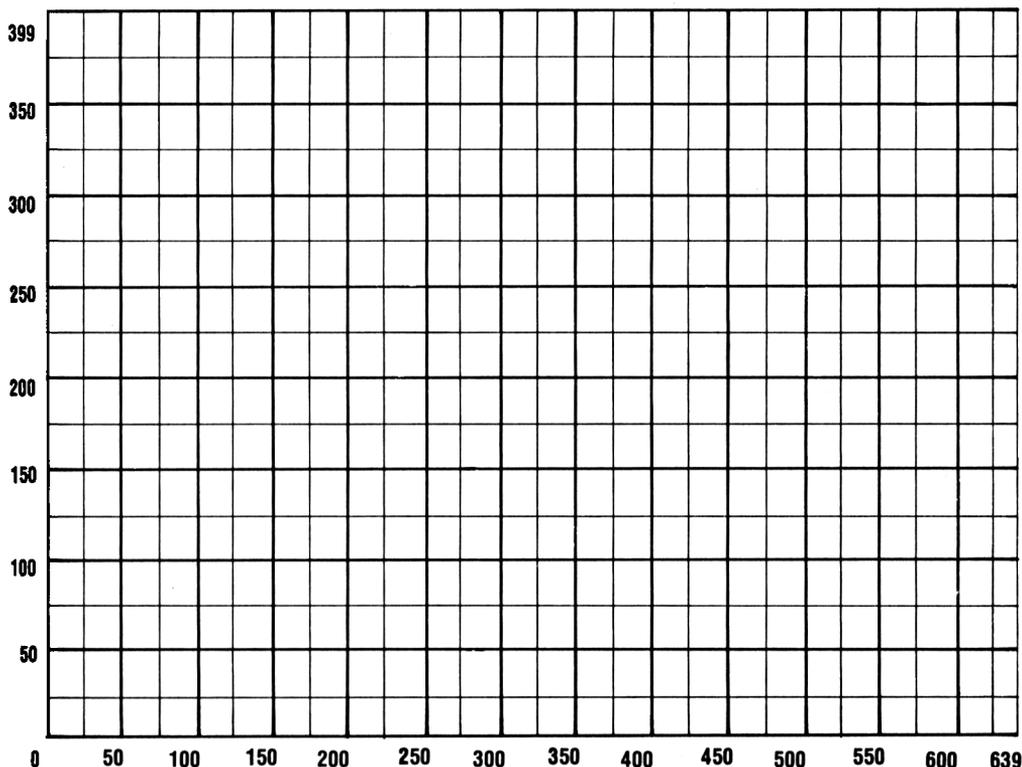
[]



Make a note

draw, **move** and **plot** must be followed by a space and then two numbers. There are 640 points across the screen and 400 up the screen. These two numbers must be separated by a comma.

To help you plan your drawings, here is a grid showing distances across and up the screen from the bottom left-hand corner:



You try

Use **draw**, **move** and **plot** commands to draw a house or a road sign.



If you are interested in exploring other ideas like **draw**, **move** and **plot**, then refer to **draw**, **mover**, **origin** and **window** in the *CPC 464 User Instructions*, and look at the 'Using the screen' reference section at the end of this book.

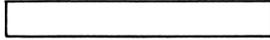


Changing colours

CPC 464 User Instructions F3.1

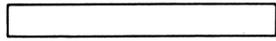
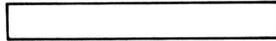
There are 27 different colours available on the Amstrad microcomputer. Of course, you will only see them if you are using a colour monitor or a colour TV. If you are using a green screen monitor, the colours will appear as different shades of green.

When you switch the computer on, it displays bright yellow characters on a dark blue background.



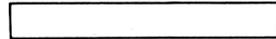
You try

Type **border 0** then press **ENTER** .
Type **border 15** then press **ENTER** .



You try

Type **border 3,4** then press **ENTER** .



The 'You try' exercise gives first a black border round the edges of the screen, and then an orange border.

The 'You try' exercise gives a border which flashes between red and magenta. When you are tired of looking at it, type **border 1** to get a steady blue border back again.

Each colour has a number, called the 'ink number'. For example, black is ink number 0 and blue is ink number 1. You will find a complete list of ink numbers in the *CPC 464 User Instructions F3.2*.

Make a note

border followed by a space and one number changes the border to a steady colour.

border followed by a space and two numbers with a comma between them gives a flashing border.



You try

Type **paper 2** then press .

Type **cls** then press

.

The 'You try' exercise changes the background within the border to bright cyan. The statement **cls** tells the computer to clear the screen.

Make a note

cls followed by clears the screen.

You try

Type **pen 3** then press .

Type **cls** then press

.

The 'You try' exercise changes the colour that the computer uses to write with from bright yellow to red. Now anything that you type will be in red.

How many colours you can show on the screen at once depends on the mode that the computer is in. When you switch on, it is in mode 1. In mode 1 you can use four colours at once, not counting the border. In mode 0 you can use up to sixteen different colours, but in mode 2 you can use only two (again, not counting the border).

Make a note

The commands **paper** and **pen** are used to choose the background and foreground colours of the screen. If your foreground and background colours are the same, you will not be able to see what you are typing.

You try

Change to mode 0 and alter the colours of the paper and pen by using ink numbers between 0 and 15. Clear the screen after altering the paper number.





You try

Reset the computer.
Type **ink 1,16** then
press **ENTER** .

You try

Type **ink 2,9** then
press **ENTER** .
Type **pen 2** then
press **ENTER** .

You try

Type **ink 3,14** then
press **ENTER** .
Type **paper 3** then
press **ENTER** .
Type **cls** then press
ENTER .

The 'You try' exercise instantly turns all the writing on the screen pink.

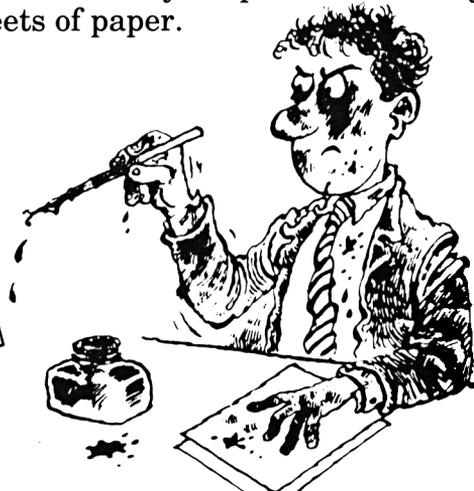
The 'You try' exercise turns everything you type after **pen 2** green. But everything before that is still pink.

The 'You try' exercise clears the screen and turns the paper colour to pastel blue.

When you switch on the computer, the four colours available for your paper and pen in mode 1 are blue, bright yellow, bright cyan and bright red. Typing **ink** followed by a space and then two numbers separated by a comma allows you to select different colours instead. Using the **ink** command is like changing the ink in your pens and writing on differently coloured sheets of paper.

Make a note

The command **ink** allows you to alter the range of colours that you can use for pen and paper.



Making music

The **sound** command is used to play musical notes. It works equally well in all three modes, so there is no need to select any particular mode. The **sound** command is followed by a space, then at least two numbers which are separated by commas. For example

sound 1,5Ø6
sound 1,6Ø,1ØØ
sound 1,239,2ØØ,2



The first number selects the channel. You only need channels other than number 1 if you want to play several notes at once.

The second number selects how high or low the sound is. The smaller the number is, the higher the note.

The third number selects the length of time the sound is played, in hundredths of a second. If you leave this number out, the note will be played for one fifth of a second.

The fourth number selects how loud the sound will be. The numbers used are usually Ø to 7, with 7 the loudest. If you leave this number out, the computer assumes you mean 4.

You try

Type **sound 1,2Ø25,3ØØ,3** then press **ENTER** .

You try

Type **sound 1,3Ø,1Ø,7** then press **ENTER** .

You try

Use the **sound** command to play some of your own notes.

The 'You try' exercise plays a low, long, quiet note.

The 'You try' exercise plays a high, short, loud note.



You can get the computer to play notes one after the other by using the **:** sign. For example,
sound 1,6Ø,1ØØ:sound 1,53,1ØØ:sound 1,47,1ØØ

WRITING PROGRAMS



Prof. O. Crumpet

My advice

You want ideas that work. You will probably need to spend some time thinking about them. If your idea does not work as you thought it would, then try to find out why. It may be that one very small alteration will make it work exactly as you intended.

However, it could be that your idea will never work, no matter how you alter it, so be prepared at times to start all over again. Sorting out ideas is not easy, but it is easier if you have some plan to work to.

Often an idea does not work quite as you wanted, but it could be used in another way. Make a note of it so that you can use it later. You must, of course, keep a very careful record of all your really good ideas.

Oliver Crumpet



The computer can do many things. Here are just a few. Try them for yourself.

`cls` (followed by) clears the screen.

`print` (followed by) gives a blank line.

`print "computer"` (followed by) gives the print out of `computer`.

`print "programmer"` (followed by) gives the print out of `programmer`.

The computer can be given many other things to do, but instead of giving them to the computer one at a time, they can be given together. When the computer is given a list of things to do in order it is called a computer program. The lines begin with line numbers to tell the computer the order in which they should be carried out. Examples of line numbers are: `10 20 30 40 50 60` etc.

The computer looks at the list and works through it `10, 20, 30, 40, 50, 60` in order until it reaches the end. Some programs have just a few line numbers, other programs may have hundreds or even thousands. An example of a program is

```
10 cls
20 print
30 print "computer"
40 print
50 print "programmer"
60 print
```

*You are making a start
At the programming art.*

This is what will appear on the screen:

You try

Type the program above, line by line. After you have written a complete line press to go to the start of the next line.



This is what will appear on the screen

You try

Now type the word
run and press

ENTER .



If you have made any mistakes you can correct them by retyping the line in which they occur. Suppose your screen showed



The mistake is in line number **30**, so by retyping the line number **30** this mistake can be corrected.

30 print "computer"
(followed by **ENTER**).

This will correct the line in the program and give you the correct display.

You may also alter lines in the program in the same way. Type in the following.

30 print "a computer"
(followed by **ENTER**).

This will alter the line in the program.

While you are writing programs it is quite useful to be able to look back at the program you have written. To do this type the word **list** followed by **ENTER** .

This is what will appear on the screen:



to find things missed,
just type LIST.

```
Ready
list
10 CLS
20 PRINT
30 PRINT "a computer"
40 PRINT
50 PRINT "programmer"
60 PRINT
Ready
█
```

Notice that some of the words are now in capitals. This shows that the computer has recognised them as BASIC commands. (It does not matter whether you type them in small letters or capitals.)

The program line numbers go up in tens so it is possible to add extra lines. For example, try typing:

15 print "I am" (followed by **ENTER**).

This puts the line into the program as follows:

```
10 cls
15 print "I am"
20 print
30 print "a computer"
40 print
50 print "programmer"
60 print
```



You can check that this has happened by typing **list** followed by **ENTER**. Type **run** followed by **ENTER**.

This is what will appear on the screen:

```
I am
a computer
programmer
Ready
█
```

Make a note

1. Number each line **10, 20, 30** etc.
2. Press **ENTER** when the line is complete.
3. Type **run** and press **ENTER** when the program is complete.
4. Correct mistakes by retyping the line.
5. Type **list** and press **ENTER** to look at the program.



Writing programs

In order to write your own programs you need to find out just what the computer can do. The exercises which follow should help you.

Before you start the next section you need to have understood the last section. If you are at all unsure about anything, work through it again.

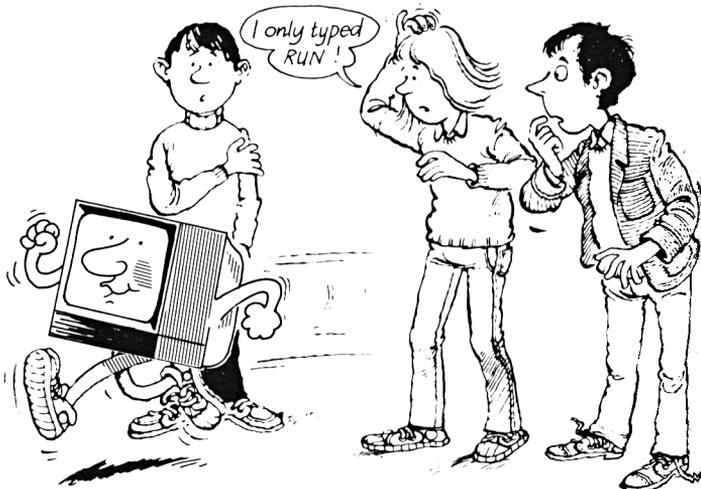
Type in these programs just as the earlier program was typed in, that is one line at a time followed by **ENTER** to get to the start of the next line. Check your program when it has all been typed in and correct any mistakes by rewriting the faulty line. When you are sure that the program is complete and correct type **run** and press **ENTER** and see what happens.

So that the computer does not get one program confused with the next, type **new** and press **ENTER** before typing in a new program. The computer then forgets the old program.

When you have copied and run the programs in each section then try the exercises. If you come up with any ideas for similar programs then try them out and see how they work.

At the end of this chapter there are five projects which are ideas for longer and more interesting programs. For most of the projects, you are helped with writing the basic program, and then you are given some ideas to improve it.

*To see what is done,
Just type RUN.*





Programs using PRINT

PRINT is used to print out lists, information, diagrams and instructions.

This is what will appear on the screen:

You try

Type **new** then press **ENTER**.

Type in the following program.

```
10 cls
20 print "Pru Comet"
30 print "P.C. Truemo"
40 print "Mort Puce"
50 print "O. Crumpet"
60 print "O.C. Termup"
```

Type **run** then press **ENTER**.



When writing a program, you must remember to start with NEW.

This is what will appear on the screen:

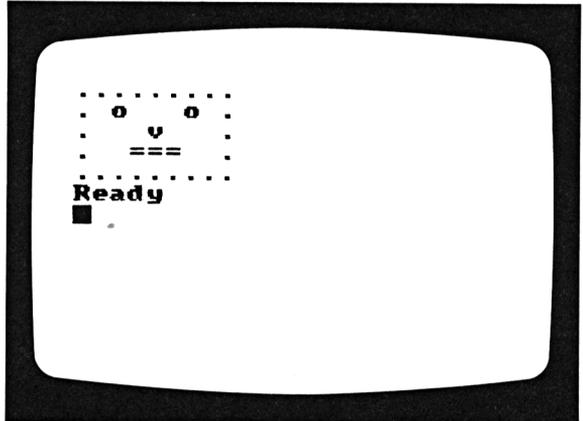
You try

Type **new** then press **ENTER**.

Type in the following program.

```
10 cls
20 print ". . . . . ."
30 print ". 0 0 ."
40 print ". v ."
50 print ". === ."
60 print ". . . . . "
```

Type **run** then press **ENTER**.



You try

Make up your own programs using **PRINT** to do the following:

1. Write out a shopping list.
2. Design a motorway sign.
3. Draw a space shuttle.

TAB is used to set out information or diagrams.

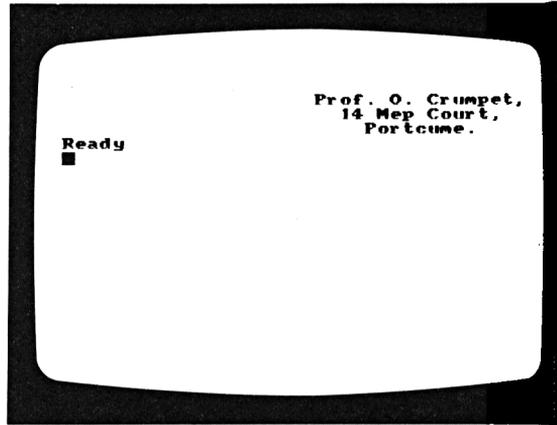
This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

```
10 cls
20 print tab(20) "Prof. O.
Crumpet,"
30 print tab(22) "14 Mep Court,"
40 print tab(24) "Portcume."
```

Type **run** then press .



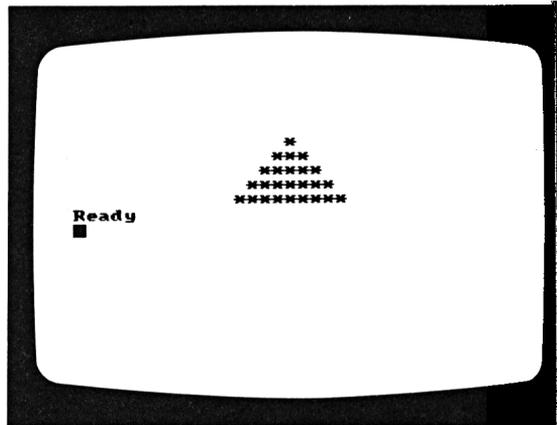
This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

```
10 cls
20 print tab(19) "*"
30 print tab(18) "***"
40 print tab(17) "*****"
50 print tab(16) "*****"
60 print tab(15) "*****"
```

Type **run** then press .



You try

Make up your own programs using **TAB** to do the following.

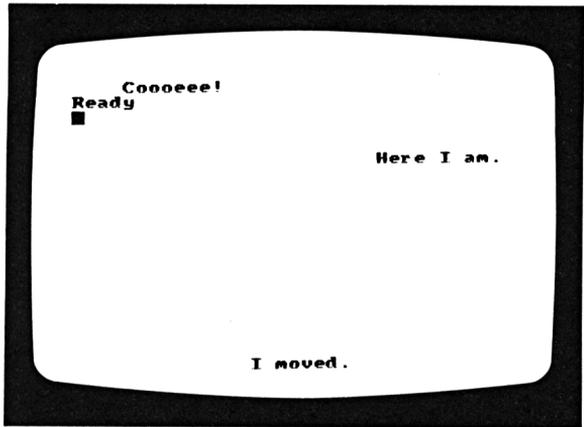
1. Draw a staircase.
2. Set out your address.
3. Draw a sloping line.



Programs using LOCATE

LOCATE is used to print anywhere on the screen.

This is what will appear on the screen:



You try

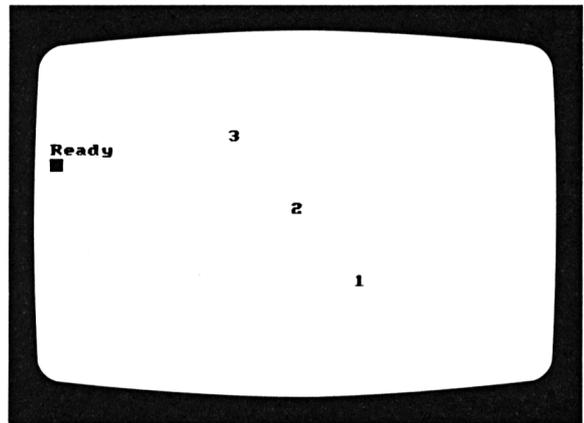
Type **new** then press .

Type in the following program.

```
10 cls
20 locate 25,5
30 print "Here I am."
40 locate 15,23
50 print "I moved."
60 locate 5,1
70 print "Cooooee!"
```

Type **run** then press .

This is what will appear on the screen:



You try

Type **new** then press .

Type in the following program.

```
10 cls
20 locate 25,17
30 print 1
40 locate 20,12
50 print 2
60 locate 15,7
70 print 3
```

Type **run** then press .

You try

Make up your own programs using **LOCATE** to do the following.

1. Print messages on different parts of the screen.
2. Draw a wavy line.



Programs using =

The = sign is used to put a number into an address.

This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

10 cls

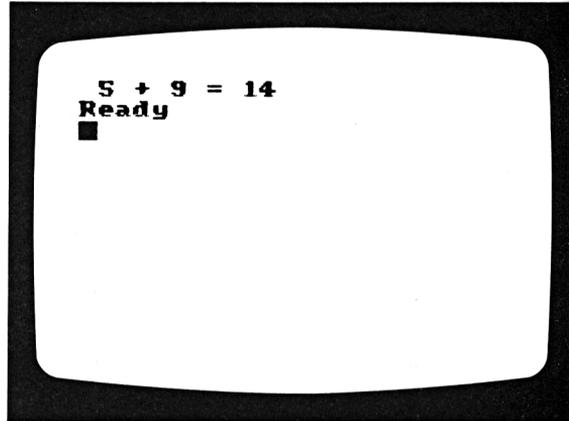
20 a=5

30 b=9

40 sum=a+b

50 print a;"+";b;"=";sum

Type **run** then press .



This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

10 cls

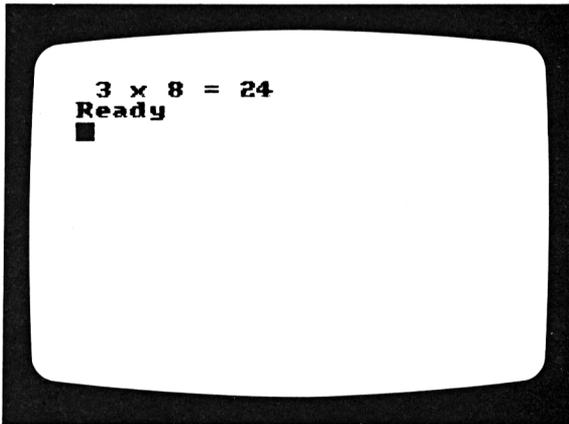
20 e=3

30 f=8

40 product =e*f

50 print e;"x";f;"=";product

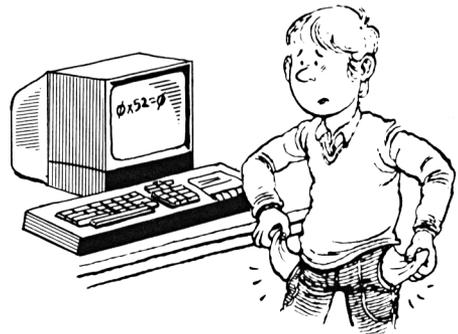
Type **run** then press .



You try

Make up your own programs using = to do the following.

1. Add two other numbers.
2. Multiply two other numbers.
3. Subtract or divide numbers.



The = sign is also used to put words into addresses.

This is what will appear on the screen:

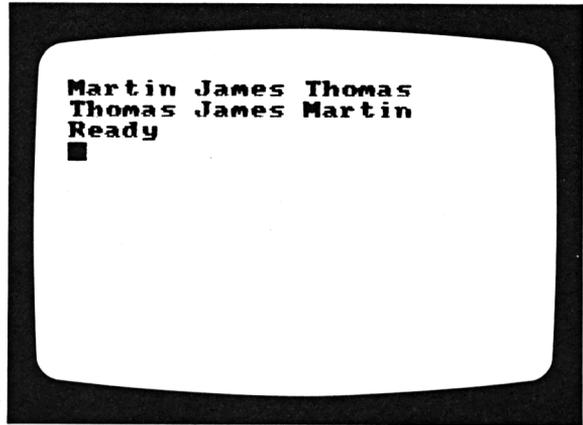
You try

Type **new** then press .

Type in the following program.

```
10 cls
20 a$="Martin"
30 b$="James"
40 c$="Thomas"
50 print a$ " "b$ " "c$
60 print c$ " "b$ " "a$
```

Type **run** then press .



You try

Type **new** then press .

Type in the following program.

```
10 cls
20 a$="can"
30 b$="you"
40 c$="go"
50 print a$ " "b$ " "c$
60 print b$ " "a$ " "c$
```

Type **run** then press .

This is what will appear on the screen:



You try

Make up your own programs using = to do the following.

1. Make a\$="in", b\$="on", c\$="set", d\$="side", e\$="to", f\$="up", g\$="wards".

By putting these words together, see how many longer words you can print.

2. Make sentences from words.

3. Select items from a menu.

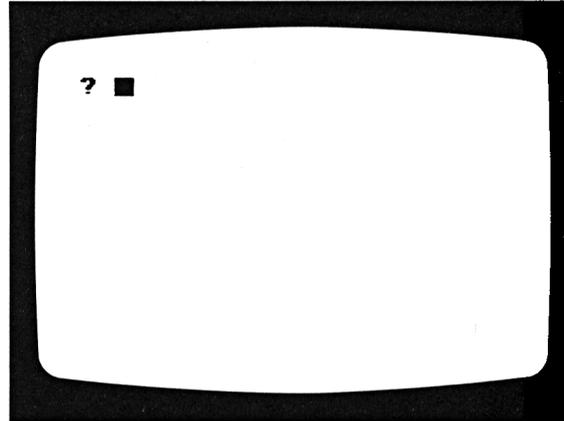
INPUT is used to put numbers into programs.

This is what will appear on the screen

You try

Type **new** then press **ENTER** .
Type in the following program.

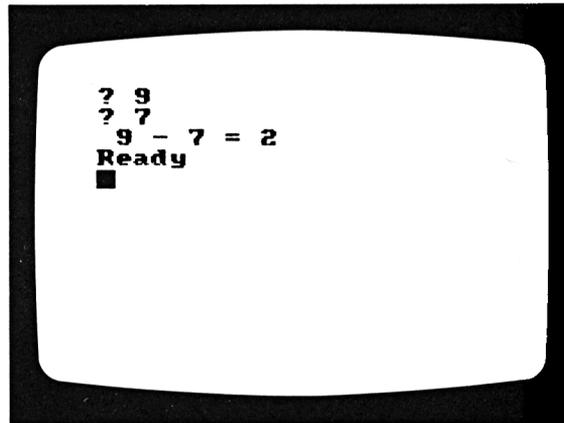
```
10 cls
20 input a
30 input b
40 difference=a-b
50 print a;"-";b;"=";difference
Type run then press ENTER .
```



This is what will appear on the screen

You try

Type in a number,
for example **9**, then
press **ENTER** .
Type in a number,
for example **7**, then
press **ENTER** .



You try

Now put your own numbers into
the program.



This is what will appear on the screen:

You try

Type **new** then press .

Type in the following program.

10 cls

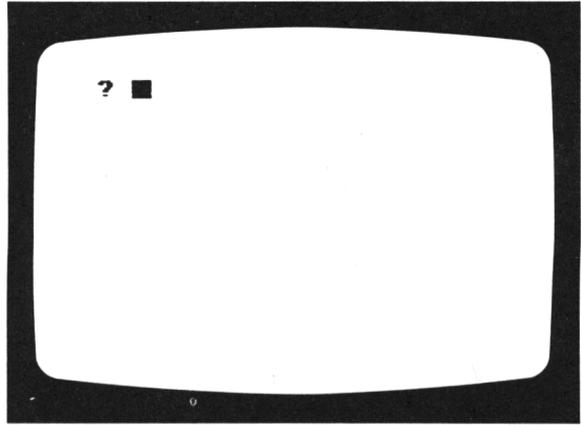
20 input n

30 input d

40 quotient=n/d

50 print n;"/";d;"="";quotient

Type **run** then press .

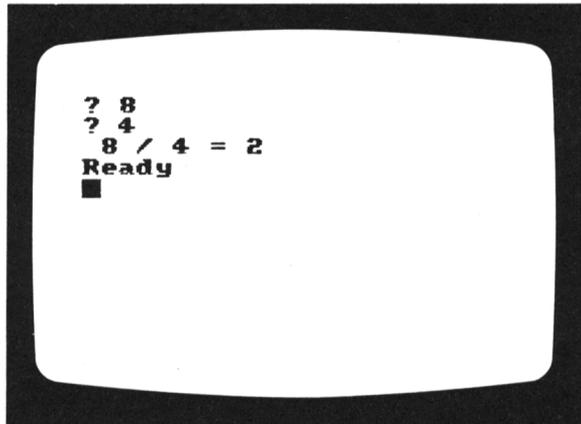


This is what will appear on the screen:

You try

Type in a number, for example **8**, then press .

Type in a number, for example **4**, then press .



You try

Type **run** then press .

Now put your own numbers into the program.

You try

Make up your own programs using **INPUT** to do the following.

1. Add two numbers together.
2. Multiply two numbers.
3. Ask for the amount of pocket money you receive in a week and multiply this by 52 to give the amount you receive in a year.



INPUT is also used to put words into programs.

This is what will appear on the screen

You try

Type **new** then press .

Type in the following program.

10 **cls**

20 **print "Ann,Eve,Kate,Mary"**

30 **print "Find the palindrome."**

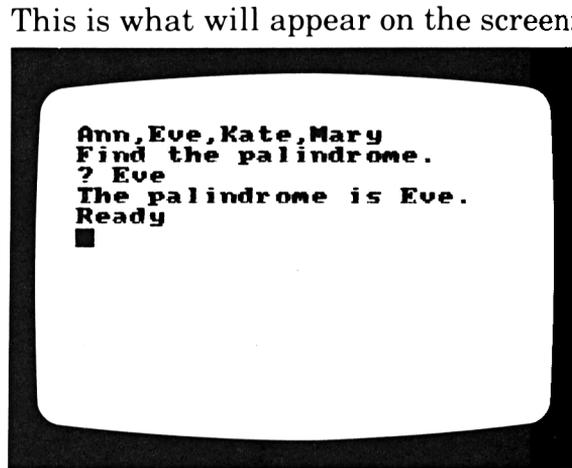
40 **input answer\$**

50 **print "The palindrome is
Eve."**

Type **run** then press .

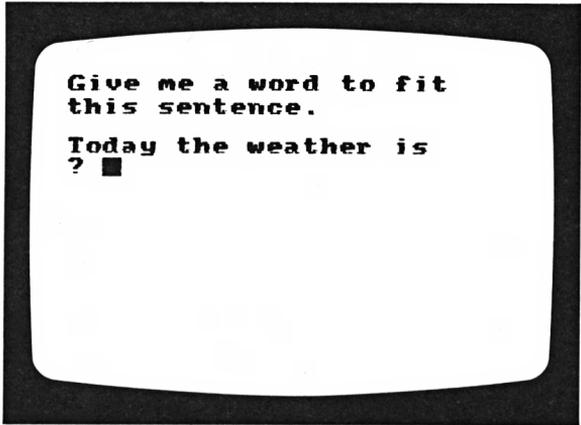
You try

Type in the palindrome **Eve** then
press .



(A palindrome is a word which is spelt the same way backwards as forwards.)

This is what will appear on the screen:

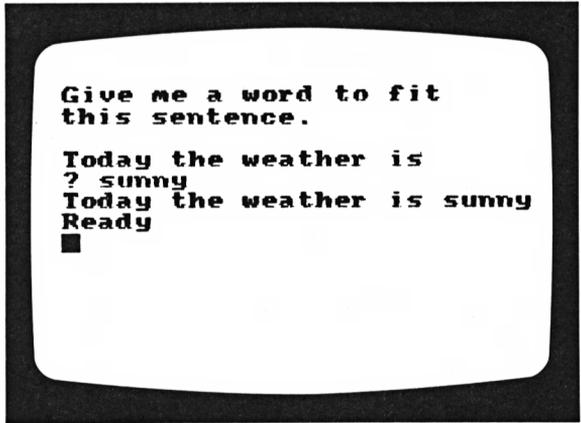


You try

Type **new** then press **ENTER** .
Type in the following program.
10 cls
**20 print "Give me a word to fit
this sentence."**
30 print "Today the weather is"
40 input answer\$
**50 print "Today the weather
is";answer\$**

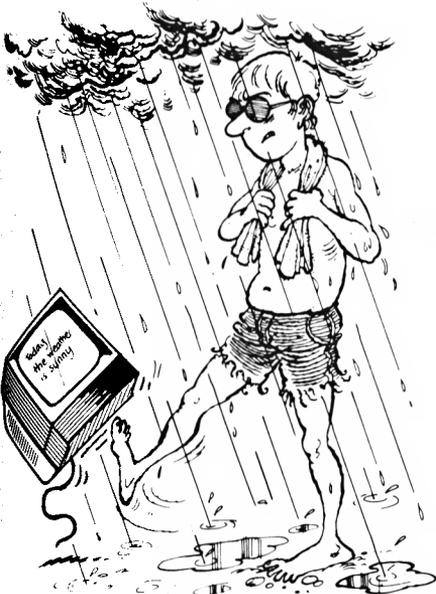
When a line is too long to fit on the screen just keep typing normally and the computer will understand you even though a word is on two lines. Do not press **ENTER** until you have reached the end of the instruction.
Type **run** then press **ENTER** .

This is what will appear on the screen:



You try

Type in **sunny** (or another word to describe the weather) then press **ENTER** .



You try

Make up your own programs using **INPUT** to do the following.

1. Find a missing word.
2. Answer a simple question.
3. Make the computer have a conversation with you.

Programs using FOR/TO/STEP/NEXT

CPC 464 User Instructions 8.18

FOR/TO/STEP/NEXT is used to repeat the same lines in turn for a set of numbers.

This is what will appear on the screen:

You try

Type **new** then press **ENTER** .

Type in the following program.

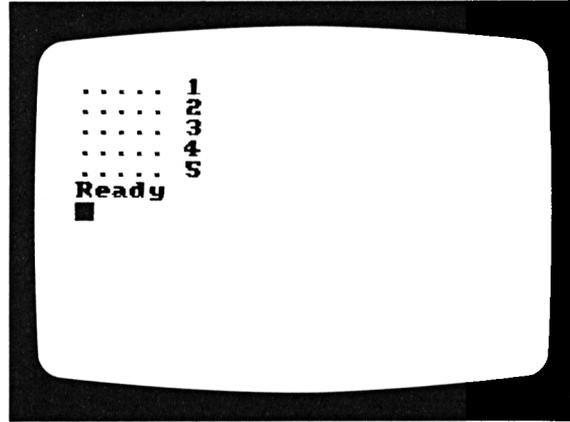
```
10 cls
```

```
20 for count=1 to 5 step 1
```

```
30 print ". . . . .";count
```

```
40 next count
```

Type **run** then press **ENTER** .



For each value of **count** between 1 and 5, the row of dots and value of **count** was printed.

This is what will appear on the screen:

You try

Type **new** then press **ENTER** .

Type in the following program.

```
10 cls
```

```
20 for n=1 to 9 step 2
```

```
30 print n
```

```
40 next n
```

Type **run** then press **ENTER** .



You try

Make up your own programs using **FOR/TO/STEP/NEXT** to do the following.

1. Print out the numbers 1 to 20.
2. Print out your name and address three times.
3. Print out the three times table.

If you do not tell the computer the size of the step it assumes **step 1**, so we could have left this out in the first example.



Programs using DATA/READ

CPC 464 User Instructions 8.9, 8.37

DATA/READ is used to put data (information) into the computer and read (recall) it when it is needed.

This is what may appear on the screen:

You try

Type **new** then press .

Type in the following program.

```

10 cls
20 data is,clue,a,here
30 read w$,x$,y$,z$
40 print z$," ";w$," ";y$," ";x$
50 print y$," ";x$," ";w$," ";z$
Type run then press .

```



This is what will appear on the screen:

You try

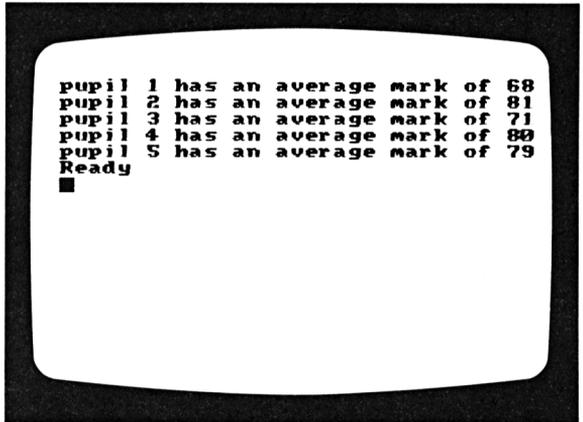
Type **new** then press .

Type in the following program which, for each of five pupils, reads in their maths mark and English mark and works out the average.

```

10 cls
20 data 75,61,83,79,
64,78,93,67,86,72
30 for pupil = 1 to 5
40 read maths, English
50 average =(maths +
English)/2
60 print "pupil";pupil;" has an
average mark of";average
70 next pupil
Type run then press .

```



You try

Make up your own programs using **READ/DATA** to do the following:

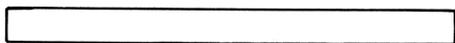
1. Rearrange a list of items.
2. Make a list of months giving the number of days in each month.



Programs using LEFT\$, MID\$ and RIGHT\$

CPC 464 User Instructions 8.23, 8.27, 8.4

The word functions **LEFT\$**, **MID\$** and **RIGHT\$** can be used to pick out parts of words.



This is what will appear on the screen

You try

Type **new** then press **ENTER** .

Type in the following program.

10 cls

20 word\$="composer"

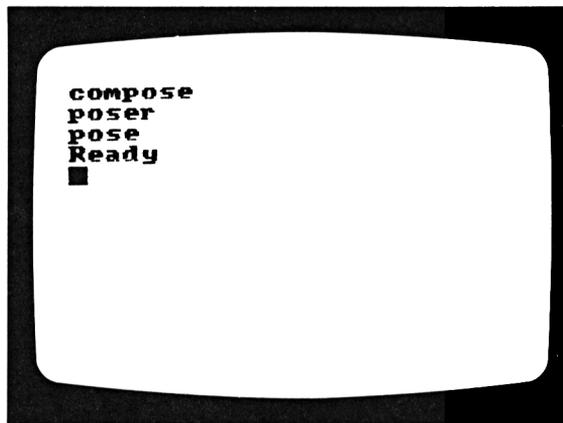
30 print left\$(word\$,7)

40 print right\$(word\$,5)

50 print mid\$(word\$,4,4)

Type **run** then press **ENTER** .

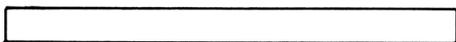
Use **LEFT\$**, **MID\$** and **RIGHT\$** to print as many words as you can if word\$="another".



Programs using INT and RND

CPC 464 User Instructions 8.22, 8.4

INT gives the whole number part of a number. For example, **INT(2.4)=2**. **RND** chooses a random number between 0 and 1.



You try

Type **new** then press **ENTER** .

Type in the following program.

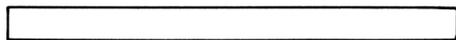
10 cls

20 for n=1 to 20

30 print n/4, int(n/4)

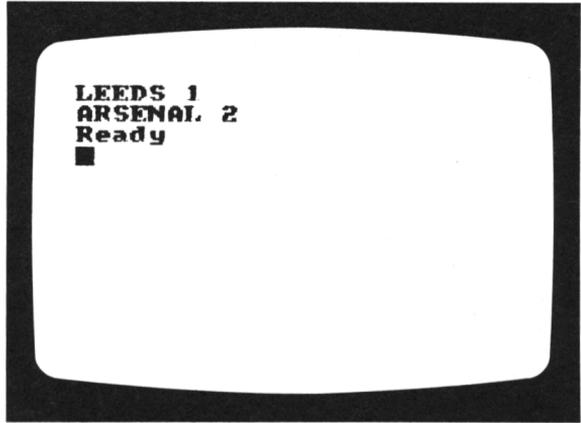
40 next n

Type **run** then press **ENTER** .



This program shows how **INT** works. We need to use **INT** with **RND** when we want the answers to be whole numbers. **RND** chooses a number between 0 and 1 (but not equal to 1), so **RND*4** chooses a number between 0 and 4 (but not equal to 4). **INT(RND*4)** chooses one of the numbers 0, 1, 2, or 3.

This is what may appear on the screen:



You try

Type **new** then press .
Type in the following program.

```
10 cls
20 Leeds=int(rnd*4)
30 Arsenal=int(rnd*4)
40 print "LEEDS";Leeds
50 print "ARSENAL";Arsenal
```

Type **run** then press .

Since the computer chooses numbers at random it is unlikely that the numbers in this display or the next one will be the same as yours.

This is what may appear on the screen:



You try

Type **new** then press .
Type in the following program.

```
10 cls
20 first=1+int(rnd*6)
30 second=1+int(rnd*6)
40 print "First dice";first
50 print "Second dice";second
```

Type **run** then press .

The instruction **first=1+int(rnd*6)** chooses one of the numbers 1, 2, 3, 4, 5 or 6 at random and puts the chosen number into the address **first**. Try running the program several times and different numbers will probably be chosen.

You try

Make up your own programs using **INT** and **RND** to do the following.

1. Choose a winning raffle ticket from 1000 tickets.

2. Select numbers for a game of bingo (numbers 1 to 90).
3. Make up sums and print the correct answer when you have had a go.

Programs using IF/THEN/ELSE

CPC 464 User Instructions 8.19

IF/THEN/ELSE is used to introduce alternatives.

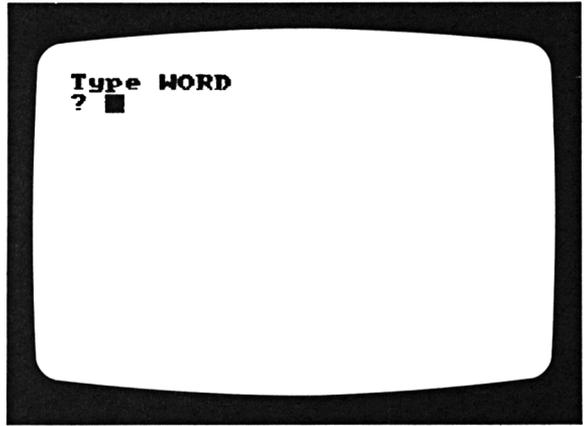
This is what will appear on the screen:

You try

Type **new** then press .

Type in the following program.

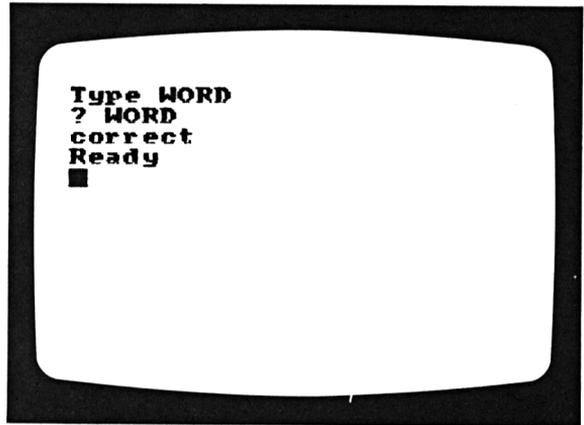
```
10 cls
20 print "Type WORD"
30 input word$
40 if word$="WORD" then print
"correct" else print "wrong"
Type run then press .
```



This is what will appear on the screen:

You try

Type **WORD** then press .



Notice that you must type **WORD** in capitals for this to work. Run the program again.

This is what will appear on the screen:

You try

Type **WIRD** then press .



This is what will appear on the screen:

You try

Type **new** then press .

Type in the following program.

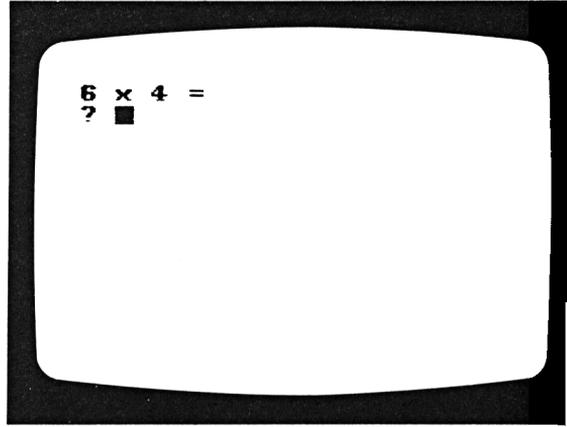
10 cls

20 print "6 x 4 ="

30 input answer

40 if answer=24 then print
"correct" else print "wrong"

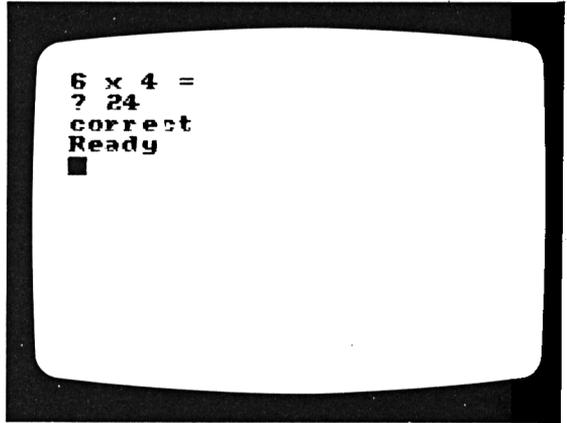
Type **run** then press .



This is what will appear on the screen:

You try

Type **24** then press .

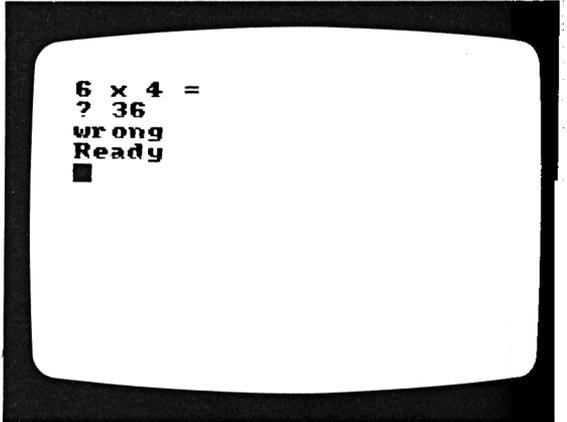


Run the program again.

This is what will appear on the screen:

You try

Type **36** then press .



You try

Make up your own programs using **IF/THEN/ELSE** to do the following.

1. Check an answer to a simple question.
2. Make the computer ask if you are well and give a suitable reply to your answer.

Programs using GOTO

GOTO is used to jump to a different line in the program.

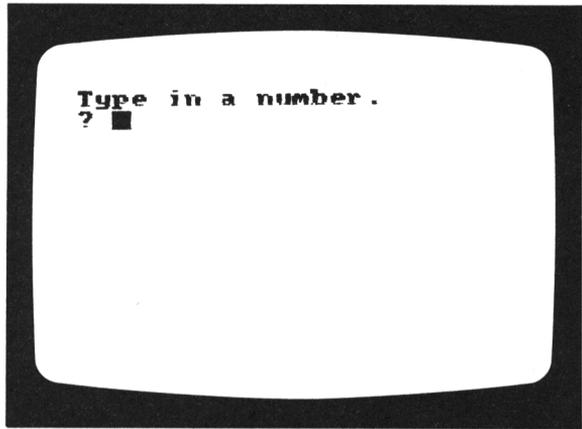
[]

This is what will appear on the screen:

You try

Type **new** then press .
Type in the following program.
10 cls
20 print "Type in a number."
30 input number
40 if number < 100 then goto 70
50 print number "is more than
10."
60 end
70 print number "is less than
100."
Type **run** then press .

[]



[]

This is what will appear on the screen:

You try

Type in a number, for example
71, then press .

[]



This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

10 **cls**

20 **print "7 x 7 ="**

30 **input answer**

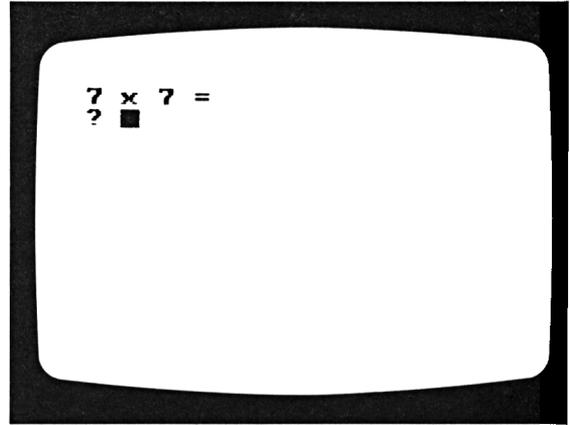
40 **if answer=49 then goto 70**

50 **print "wrong"**

60 **goto 20**

70 **print "correct"**

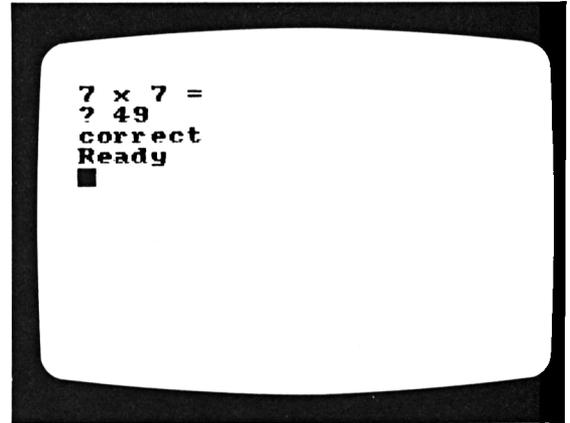
Type **run** then press .



This is what will appear on the screen

You try

Type in your answer then press .



You try

Make up your own programs using **GOTO** to do the following.

1. Print **Yes** or **No** in answer to a question.
2. Print a word in small or in capital letters.
3. Input the cost of five items of shopping and print the total cost. You will need to make the computer keep count of the number of items input so far.



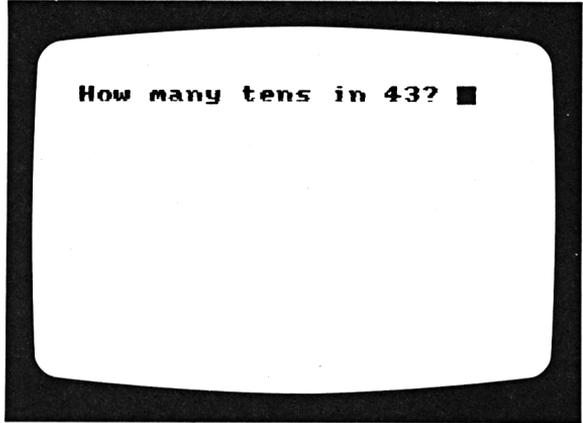
Programs using AND

AND is used to check two conditions.

This is what will appear on the screen:

You try

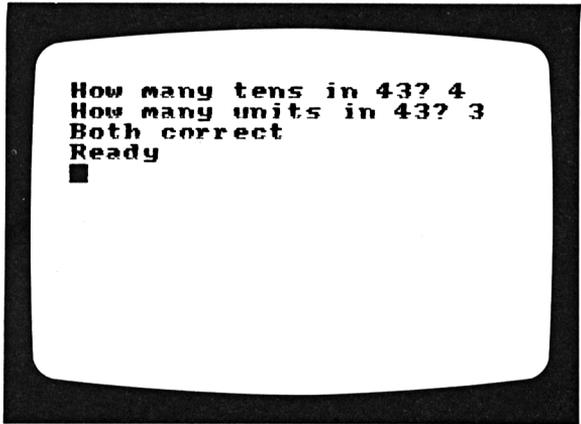
Type new then press .
Type in the following program.
10 cls
20 print "How many tens in 43";
30 input tens
40 print "How many units in 43";
50 input units
60 if tens=4 and units=3 then
print "Both correct" else goto 20
Type run then press .



This is what will appear on the screen:

You try

Type 4 then press .
Type 3 then press .



You try

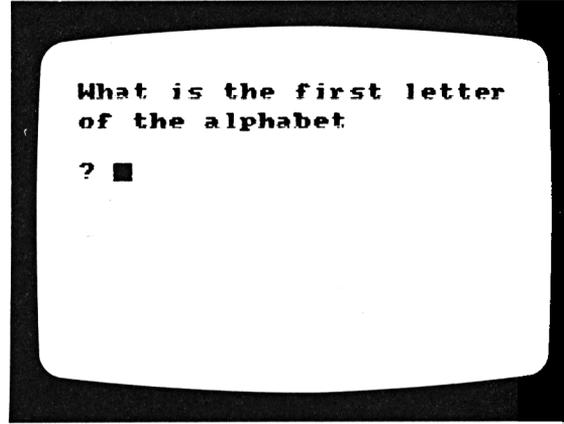
Run the program again.
Type in one or two incorrect
answers and notice what happens.

This is what will appear on the screen

You try

Type **new** then press .
Type in the following program.

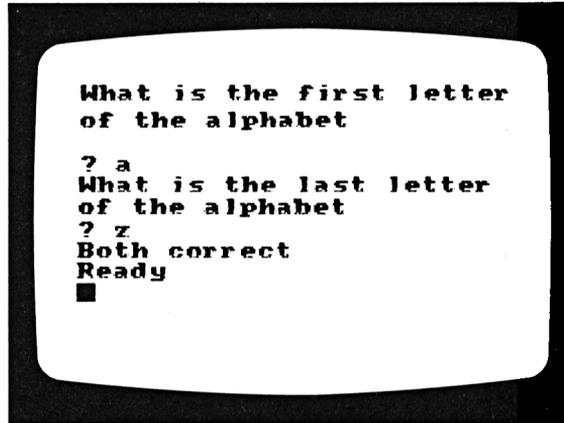
```
10 cls
20 print "What is the first letter
of the alphabet"
30 input a$
40 print "What is the last letter of
the alphabet"
50 input z$
60 if a$="a" and z$="z"
then print "Both correct" else
goto 20
Type run then press  .
```



You try

Type **a** then press .
Type **z** then press .

This is what will appear on the screen



You try

Run the program again.
Type in one or two incorrect letters
and notice what happens.

You try

Make up your own programs using **AND** to
do the following:

1. Check for two possible answers to a question.
2. Select two words from a list of words.
3. Select two numbers from a list of numbers.

Programs using OR

OR is used to check for one of two conditions.

This is what will appear on the screen:

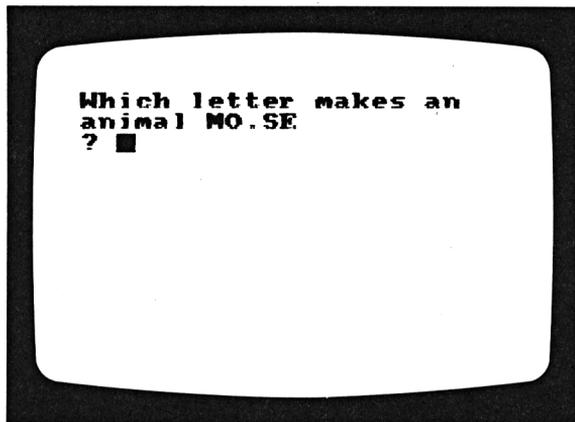
You try

Type **new** then press .

Type in the following program.

```
10 cls
20 print "Which letter makes an"
30 print "animal MO.SE"
40 input letter$
50 if letter$="U" or
letter$="O" then print "correct"
else goto 20
```

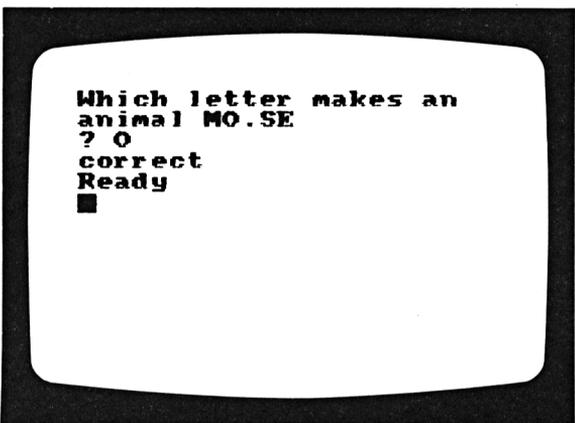
Type **run** then press .



This is what will appear on the screen:

You try

Type **O** then press .

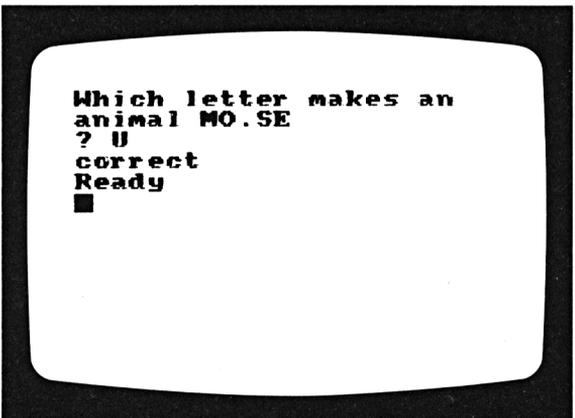


Run the program again.

This is what will appear on the screen:

You try

Type **U** then press .



You try

Type in incorrect answers and notice what happens.

You try

Type **new** then press .
Type in the following program.

10 cls

20 print "Type in the odd letter"

30 print "A E H I O U"

40 input letter\$

50 if letter\$="H" or letter\$="h"
then print "Correct" else goto 20

Type **run** then press .

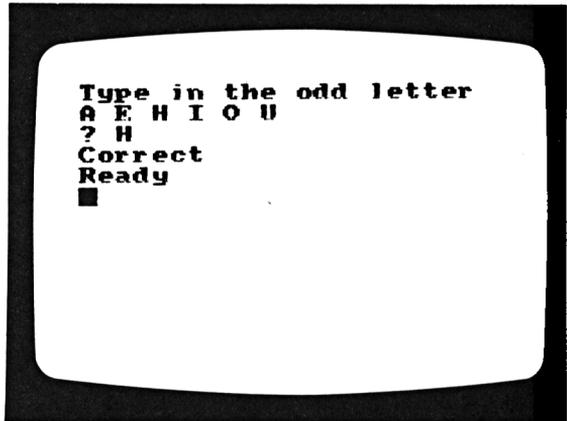
This is what will appear on the screen:



You try

Type **H** then press .

This is what will appear on the screen:

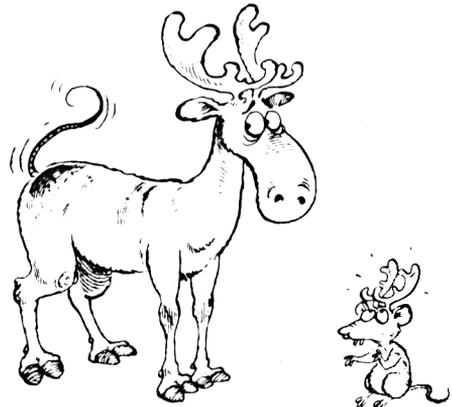


You try

Run the program again and try typing in **h**. Run it once more and try typing in other letters.

Make up your own programs using **OR** to do the following:

1. Check for one of two possible answers to a question.
2. Check a numerical answer which may be one of two numbers.





Project 1 – Knock, knock

The aim of this program is to make the computer tell knock, knock jokes. The main structure is a large **WHILE . . . WEND** loop, but first it is necessary to set up some starting values and give some data. The main steps are given below.

1. Make **count=0**. (We are going to count the number of jokes so that we know when we have reached the end. This is to start the count. We shall increase the count by one each time a joke is told.)

Make **jokes=6**. (This gives the number of jokes the computer can tell.)

DATA Ivor, sore hand from knocking on your door, Howard, I know, Mr, last bus home, Amos, quito, Lettuce, in and you'll find out, Ken, I come in

2. Now we can set up the **WHILE . . . WEND** loop. Start with the instruction **WHILE count<jokes**, then inside the loop the computer should be made to

(a) ask if you want to hear a knock, knock joke

(b) **INPUT** the answer

(c) if the answer is **no** then **END**

(d) otherwise **READ name\$, lastline\$**

(e) **PRINT** Knock, knock

Who's there?

name\$

name\$ who?

name\$;" ";lastline\$

(This will take several **PRINT** statements.)

(f) Make **count=count+1** (increasing count by one).

(g) Put in **WEND** to end the loop.

Improvements and variations

A. Add some more jokes.

B. Make the telling of the jokes into a conversation with the computer. Then the computer will print **Knock, knock**, you will input **Who's there?** etc.

C. Make the program end if the answer to **Do you want to hear another joke?** is **no** or **NO**.

D. Make the computer print a message when it has run out of jokes to tell.



E. Include a pause between telling one joke and asking if you wish to hear another. This can be done by inserting the line **FOR n=1 TO 2000:NEXT n**

This makes the computer go round and round a loop which does nothing. Using a number greater than 2000 would make the pause longer.



Project 2 – Drawing

There is a great deal that you can do using **DRAW** and **MOVE**. Here are some ideas.

A. Try drawing a picture of a space rocket, a church, a TV set, a castle, a ship, or a telephone kiosk. It is easier if you draw the picture first on a copy of the grid on page 41.

B. Draw a small star (using the grid on page 41 to help you). Try adding the same number to all the distances across (the first number after **DRAW** and **MOVE**). See where the star is drawn now. Then try to draw the star further up the screen.

By using a **FOR...NEXT** loop, **INT** and **RND** try drawing many stars on the screen at random positions (so that it is different each time you run the program). If you have a colour monitor or TV set you can try making the stars different colours (you will need to use the **INK** command to do this).

C. Try using

```
FOR row=1 TO 390 STEP 30: MOVE 0, row: DRAW 639, row: NEXT row
```

D. You can make a solid square by trying

```
FOR row=150 TO 250 STEP 2: MOVE 150, row: DRAW 639, row: NEXT row
```

Actually, it is a bit taller than it is wide. Try to put this right. Try to make a solid rectangle which is wider than it is tall.



Project 3 – Countdown

In this project you will program the computer to play a game with you. In this game you start with 20 counters. You and the computer take it in turns to remove 1, 2, 3 or 4 counters. The one to remove the last counters so that none are left is the winner. The program can be written as follows.

1. Make the computer explain the game using **PRINT**.
2. Set up the number of counters to be 20.
3. Have your turn, which consists of the following steps.
 - (a) Make the computer ask for the number of counters you wish to remove and **INPUT** the answer.
 - (b) Take away your answer from the number of counters to give the number of counters remaining and **PRINT** the result.
 - (c) Check whether the number of counters is zero. If it is then make the computer tell you that you have won and **END** the game.
4. Make the computer have a turn by doing the following steps.
 - (a) If the number of counters is less than five (<5), then the computer can choose to remove all the counters. Otherwise the computer chooses to remove a random number of counters between 1 and 4 (use **INT(RND*4)+1**). (Use **IF/THEN/ELSE** for this part.)
 - (b) Calculate the new number of counters and **PRINT** the result.
 - (c) Check whether the number of counters is zero. If it is then the computer says that it has won and ends the game, otherwise it returns to your turn (step 3). Make sure that the name for the number of counters is the same as at the beginning of step 3.

Improvements and variations

A. The computer could ask whether you want the first turn, and if not have the first turn itself.

B. Instead of having the computer write down the number of counters, make it draw the counters (use **PRINT** and the letter **O** as a counter).



C. Make the screen clearer by using **CLS**, **PRINT** on its own for a blank line, and extra spaces.

D. It would be a good idea to make the computer check that you are not cheating by trying to remove too many counters or none at all.

E. You could vary the number of counters at the start and the number which may be picked up at each turn.

F. When you have played the game a few times and know how to beat the computer, try making the computer play more intelligently.

Project 4 – Music

The object of this program is to play the tune Brother Joseph (Frère Jacques). The English words, with notes above, are given below (C' etc. mean that the note comes from Octave 1, above middle C).

G A B G G A B G
Brother Joseph, Brother Joseph,

B C' D' B C' D'
Wake up now! Wake up now!

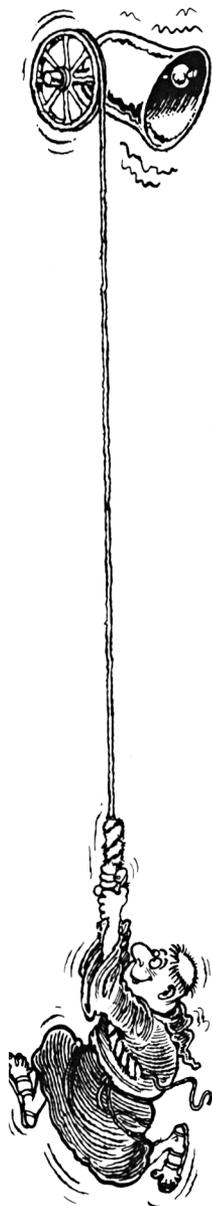
D' E' D' C' B G D' E' D' C' B G
Go and pull the bell rope. Go and pull the bell rope.

G D G G D G
Ding, dang, dong. Ding, dang, dong.

Each note can be given to the computer as a command of the form **SOUND, 1, tone, duration**. The **tone** says how high or low the note is, and **duration** gives its length. The numbers for the tones are

D	E	F	G	A	B	C'	D'	E'
426	379	358	319	284	253	239	213	190

The duration should be 25 for the short notes, 50 for the medium-length notes, and 100 for the long notes. If you find it difficult to decide on the timing, make the duration 50 for all the notes at first, and make the changes after listening a few times.





Here are some suggestions for writing out this program.

1. Make **d=25** at the beginning of the program (**d** standing for duration), and then you can write the length of the notes as **d**, **d*2** or **d*4** depending on how long you want them to be. Then you can easily change the speed at which the tune is played by changing the value of **d**.

2. You can also try using **DATA/READ** in a similar way to their use in the Knock, knock project. This is a better way to program, especially since it needs less typing. It also makes it easier to change the key in which the tune is played, that is, to play it higher or lower.

If you are programming in this way, you should put pairs of numbers giving the tone and duration (such as **319,d*2** for the first note) into **DATA** statements at the beginning of the program. Count how many notes there are altogether. Then tell the computer

```
FOR note=1 TO however many it is  
READ tone,duration  
SOUND 1,tone,duration  
NEXT note
```



Improvements and variations

A. The tune probably sounds slightly wrong because there is no gap between the notes. If two notes of the same tone are played one after the other, they slide into each other and sound like one note. You can get over this by using

```
SOUND 1,Ø,5
```

to put a short pause where it is needed.

B. You could make the words come up on the screen as the tune is being played by using **PRINT**. This could also help to make it easier to see where in the tune you have reached when you are trying to correct the program.



Project 5 – Lions and antelopes

The purpose of this program is to set up a situation like the one faced by the warden of a game park in Africa. We shall look at the lions and antelopes in the game park. The lions kill the antelopes for food (about 200 each in a year), and the warden has to choose how many lions to kill to keep the numbers under control.

The structure of the program is as follows.

1. Make the computer explain the situation for someone new coming to the program, using **PRINT**.
2. Give starting values for the number of lions and the number of antelopes. It is sensible to start with about 15 lions and 10000 antelopes.
3. Make the computer ask the number of lions to be killed this year and **INPUT** the reply. This number is called the cull.
4. Work out the number of lions and antelopes for the following year using the equations below.

$$\text{lions} = \text{INT}(\text{lions} - \text{cull} + \text{lions} * (\text{antelopes} - 400 * \text{lions}) / 10000)$$

$$\text{antelopes} = \text{INT}(\text{antelopes} - 200 * \text{lions} + \text{antelopes} * (20000 - \text{antelopes}) / 30000)$$

The equation for the lions says that the new number of lions is the number from the previous year, minus the number killed, plus a number of births and deaths depending on the number of lions available to breed and the number of antelopes available as food. The equation for the antelopes is the number from the previous year, minus the number killed by lions, plus the number of births and deaths. Make the computer **PRINT** the new number of lions and antelopes.

5. Make the computer ask if you want another go, and if so then go to step 3.



Improvements and variations

- A. Alter the number of lions and antelopes at the start either by using **INPUT** so that the user chooses them, or by using **INT** and **RND** so that the computer chooses them.
- B. Make the computer keep a count of the year and **PRINT** the year number each time round. The computer could then ask if you wish to continue after every ten years, instead of every year.
- C. Write the program in such a way that it only runs while there are still some lions and some antelopes left alive.
- D. Include messages of congratulations if the number of lions and antelopes become large and scold the user if they become small.
- E. Make the writing on the screen clearer by using **CLS** (to clear the screen), **PRINT** on its own (to give a blank line), and put in extra spaces between words so that no words are split between lines.
- F. If you feel ambitious you could try drawing a picture of the park, and using symbols for lions and antelopes (say one symbol to represent a lion and another to represent 100 antelopes).





Other ideas for computer programs

1. A game of chance which uses random numbers
2. A spelling test
3. A list of questions on a particular subject
4. A musical scale
5. A questionnaire
6. A coloured diagram
7. A science experiment
8. A clock
9. A cash register
10. An anagram finder
11. A language translator
12. A passage in which missing words have to be filled in
13. A pattern designer
14. A list of names and addresses
15. A plan or scale drawing
16. A word game
17. An alphabetical sorter
18. A number game with questions and answers
19. A foreign money exchange
20. A cartoon
21. A maze
22. A rocket launcher
23. A TV advert
24. A guessing game
25. A tune

You may need to look back at some of the programs in the book or ones which you have written yourself. It may also be helpful to check certain things in the reference sections from time to time.

FINDING OUT



Ms O.C. Termup

My advice

Always check the terms you use just as you would check the meaning of unusual words with a dictionary. Most of the terms are fairly easy to understand, but there are a lot of them. If you are unsure, then check the meaning and use. All the terms you are likely to need are contained in the eight parts of the reference section. So first find out which section the term is in and then look it up in that section.

The sections are

- Editing and writing programs
- Using the screen
- Making sounds
- Word functions
- Number functions
- Operators
- Saving and loading programs
- General commands and functions

It would be very useful if you made your own reference book to contain all the terms you come across. You need not put them in separate sections, but it will make things much easier if they are in alphabetical order.

Olive C. Termup



Reference sections

The terms which you have used in the book will all be found in the reference sections, but you will also find some new terms. These are included because they may be useful when you are writing your own programs.

If you are writing a program which examines words, then you might find some of the word function terms very useful. In a mathematics program some of the number function terms will be very useful.

The reference sections only give a brief outline of the meaning and use of the terms and if you wish to find out more about them you will have to look them up in other books. The most relevant book is

Amstrad CPC 464 User Instructions
Amstrad Consumer Electronics plc

which comes with the computer. More technical information can be found in

Amstrad BASIC: The Complete Technical Specification
Amsoft
ISBN 1 85084 001 6

There are also a number of computer magazines which can be very helpful. Quite regularly they list programs which you might find interesting and you may well be able to adapt these programs to your own needs. Some suitable magazines are

Computer and Video Games
CPC 464 User Club Magazine
Personal Computer News
Personal Computing Today
Your Computer



Here is a list of all the terms that the Amstrad CPC 464 computer understands. The ones in heavy type are those which are likely to be most useful to you. These are the ones described more fully in the following pages. Wait to use the other terms until you have learnt more about BASIC.

**Editing and
writing
programs**

AUTO
CONT
DELETE
EDIT
ERL
ERR
ERROR
FRE
HIMEM
LIST
MEMORY
NEW

ON ERROR GOTO
REM
RENUM
RESUME
RUN
STOP
TRON
TROFF

**Using the
screen**

BORDER
CLG
CLS
DRAW
DRAWR

INK
LOCATE
MODE
MOVE
MOVER
ORIGIN
PAPER
PEN
PLOT
PLOTR
POS
PRINT
SPEED INK
SYMBOL
SYMBOL AFTER
TAB
TAG

TAGOFF
TEST
TESTR
VPOS
WINDOW
WINDOW SWAP
WRITE
XPOS
YPOS
ZONE

Making sounds

ENT
ENV
ON SQ GOSUB
RELEASE
SOUND
SQ

Word functions

ASC
BIN\$
CHR\$
INKEY\$
INSTR
LEFT\$
LEN
LOWER\$
MID\$
RIGHT\$
SPACE\$
STR\$
STRING\$
UPPER\$
VAL

Number functions

ABS
ATN
CINT

COS
CREAL
DEG
EXP
FIX
HEX\$
INKEY
INT
LOG
LOG10
MAX
MIN
MOD
PI
RAD
RANDOMIZE
RND
ROUND
SGN
SIN
SQR
TAN
UNT

Operators

AND
NOT
OR
XOR

Saving and loading programs

CAT
CHAIN
CHAIN MERGE
CLOSEIN
CLOSEOUT
EOF
LOAD
MERGE
OPENIN
OPENOUT
SAVE
SPEED WRITE

General commands and functions

AFTER
CALL
CLEAR
DATA
DEF FN
DEFINT
DEFREAL
DEFSTR
DI
DIM
EI
END
ERASE
EVERY
FOR/TO/STEP
GOSUB
GOTO
IF/THEN/ELSE
INP
INPUT
JOY
KEY
KEY DEF
LET
LINE INPUT
NEXT
ON/BREAK
ON/GOSUB
ON/GOTO
OUT
PEEK
POKE
READ
REMAIN
RESTORE
RETURN
SPEED KEY
TIME
WAIT
WEND
WHILE
WIDTH

Editing and writing programs

AUTO enters line numbers without first typing in the number of the line. It can be ended by pressing **ESC** .

AUTO gives 10, 20, 30 . . . etc.

AUTO 120 gives 120, 130, 140 . . . etc.

AUTO 120,4 gives 120, 124, 128 . . . etc.

AUTO ,2 gives 10, 12, 14, 16 . . . etc.

CPC 464 User Instructions 8.4

CONT is used to start a program running again from where it was stopped by **ESC** being pressed twice, or by **STOP** or **END** in the program itself.

CPC 464 User Instructions 8.8

DELETE deletes lines from a program.

DELETE 22-30 deletes all lines between 20 and 30 inclusive.

DELETE -30 deletes all lines up to and including line 30.

DELETE 30- deletes all lines from (and including) line 30.

DELETE 40 deletes line 40. This could also be done just by typing 40 and pressing **ENTER** .

CPC 464 User Instructions 8.11

EDIT is used to edit lines of a program.

EDIT 40 prints line 40 on the screen with the main cursor over the first character on the line. Changes can then be made by moving the main cursor along the line with the cursor keys, and typing in extra characters or rubbing characters out with the **DEL** and **CLR** keys.

CPC 464 User Instructions 1.16

LIST lists a program. (See page 48.)

LIST lists the entire program.

LIST 10 lists line 10 only.

LIST -50 lists all lines up to and including line 50.

LIST 50- lists all lines from (and including) line 50 to the end of the program.

LIST 50-100 lists all lines between 50 and 100 inclusive.

CPC 464 User Instructions 8.24

NEW removes any existing program from the computer's memory. The computer is then ready for a new program to be typed in. (See page 50.)

CPC 464 User Instructions 8.28

REM is used for putting a comment or REMark into a program to help you remember what the program is doing. The computer ignores anything written on the line after **REM**.

40 REM The computer will ignore this.

CPC 464 User Instructions 8.38

RENUM renumbers the lines of a program.

RENUM gives 10, 20, 30... etc.

RENUM 100 gives 100, 110, 120... etc.

RENUM 100,50 leaves all lines before 50 as they are, but then gives 100, 110, 120... etc.

RENUM 100,50,5 leaves all lines before 50 as they are, but then gives 100, 105, 110... etc.

RENUM 100,,5 gives 100, 105, 110... etc.

CPC 464 User Instructions 8.39



RUN tells the computer to run through a program. (See page 50.)

RUN 100 tells the computer to run through a program, starting at line 100.

RUN "name" tells the computer to load a program called "name" from cassette, and then run it.

CPC 464 User Instructions 8.41

STOP ends the running of a program. The program can be restarted with **CONT**.

CPC 464 User Instructions 8.45

TRON makes the computer print out the line number of each line that it comes to as a program is running. This is useful when you are trying to track down mistakes in a program. To turn off, use **TROFF**.

CPC 464 User Instructions 8.49

Using the screen

BORDER is used to change the colour of the screen border. (See page 42.)

BORDER 9 makes the border green.

BORDER 9,10 makes the border flash alternately green and cyan.

CPC 464 User Instructions F3.2

CLG clears the graphics screen. The graphics screen is normally the whole of the screen within the border, but may be altered by using **ORIGIN**.



CLG clears the graphics screen to the colour last specified in a **CLG** command, or ink \emptyset if none has been specified.

CLG 2 clears the graphics screen to colour 2.

CPC 464 User Instructions 8.7

CLS clears the text screen. The text screen is normally the whole of the screen within the border, but up to eight text windows can be defined using the **WINDOW** command.

CLS clears text window \emptyset (the whole screen unless you have defined it otherwise) to its current paper colour.

CLS 3 clears text window 3 to its current paper colour.

CPC 464 User Instructions 8.8

DRAW draws a line on screen. (See page 38.)

DRAW 200,300 draws a line to the point 200,300 from point \emptyset, \emptyset . Any further instruction draws a line from 200,300 to the next point. The first number after **DRAW** gives the distance across the screen, and the second number gives the distance up the screen. Point \emptyset, \emptyset is the bottom left-hand corner, unless this has been altered with the **ORIGIN** command.

DRAW 200,300,13 draws a line to point 200,300 in the colour currently defined by 13.

CPC 464 User Instructions 8.12

DRAWR works in the same way as **DRAW**, except that it draws relative to the current position of the graphics cursor.

DRAWR 20,30,13 draws a line in colour 13 to a point 20 units to the right and 30 units up from the last point visited.

CPC 464 User Instructions 8.12

INK is used to select the range of colours which can be used for pen, paper and graphics. (See page 44.)

INK 1,8 would make paper 1 and pen 1 bright magenta.

INK 1,7,8 would make paper 1 and pen 1 flash between purple and bright magenta.

CPC 464 User Instructions F3.2, F3.5

LOCATE is used to move the text cursor to a new point on the screen. Text will then be printed beginning from that point. (See page 53.)

LOCATE 20,10 will move the text cursor to a point 20 columns across from the left margin, and 10 rows down from the top of the screen.

LOCATE 2,20,10 will move the text cursor 20 columns



across and 10 rows down from the top left corner of text window 2.

CPC 464 User Instructions 8.25

MODE is used to select the way graphics and writing appear on the screen. (See page 37.)

MODE 0 gives up to 16 colours in the display, 20 characters per line and 25 lines on the screen.

MODE 1 gives up to 4 colours, 40 characters per line and 25 lines on the screen.

MODE 2 gives 2 colours, 80 characters per line and 25 lines on the screen.

CPC 464 User Instructions 5.3

MOVE moves the graphics cursor to a point on the screen without marking the screen as it does so. (See page 39.)

MOVE 200,300 moves the cursor to the point 200,300 from point 0,0. The first number after **MOVE** gives the distance across the screen, and the second number gives the distance up the screen. Point 0,0 is the bottom left-hand corner of the screen, unless this has been altered with the **ORIGIN** command.

CPC 464 User Instructions 8.28

MOVER works in the same way as **MOVE**, except that it moves the graphics cursor relative to the current position of the graphics cursor.

MOVER 20,30 moves the cursor to a point 20 units to the right and 30 units up from the last point visited.

CPC 464 User Instructions 8.28

ORIGIN is used to choose the starting point for the graphics cursor. **ORIGIN** can also be used to define a graphics window.

ORIGIN 320,200 places the point 0,0 in the centre of the screen.

ORIGIN 0,0,40,600,360,40 defines a graphics window 40 units in from the screen border on all sides.

CPC 464 User Instructions 8.32



PAPER is used to choose the background colour of the screen within the border. (See page 43.)

PAPER 1 sets the background to the colour currently specified by number 1.

PAPER 3,1 set the background of window 3 to the colour currently specified by number 1.

CPC 464 User Instructions F3.2, 8.33

TESTR works in the same way as **TEST**, except that it tests the colour of a point relative to the current position of the graphics cursor.

TESTR(30,20) will find the number of the colour of the point 30 units to the right and 20 units above the last point visited.

CPC 464 User Instructions 8.48

WINDOW is used to divide the screen into different areas, called windows. These windows can then be given different paper and pen colours, and characters can be printed in the window specified in a **PRINT** command.

The **WINDOW** command takes the form **WINDOW**

window number, left column, right column, top row, bottom row.

10 **MODE** 1

20 **WINDOW** #3,7,17,7,12

30 **PAPER** #3,2

40 **PEN** #3,3

50 **CLS** #3

60 **PRINT** #3, "Hello"

CPC 464 User Instructions 5.10



Making sounds

SOUND is used for playing notes. (See page 45.)

SOUND 1,478 plays middle C (tone period 478) on channel 1. The computer assumes you want to play the note for one fifth of a second at medium loudness.

SOUND 1,478,100 plays the same note for one second (100 times 0.01 sec.)

SOUND 1,478,100,7 plays the same note at maximum loudness (7 out of 7).

CPC 464 User Instructions F3.16

Word functions

INKEY\$ scans the keyboard to see if a key has been pressed.

10 **a\$=INKEY\$**

20 **IF a\$="" GOTO 10**

30 **IF a\$="y" THEN PRINT "Yes":END**

40 **IF a\$="n" THEN PRINT "No" ELSE GOTO 10**

CPC 464 User Instructions 8.20



INSTR searches a string (a series of characters) to see whether another string can be found in it.

x=INSTR (a\$,b\$) looks for the first position of **b\$** in **a\$** and stores the answer in **x**.

x=INSTR (n,a\$,b\$) does the same, but starts looking from the **n**th position.

PRINT INSTR("CAMPBELL","M") prints 3.

PRINT INSTR("CAMPBELL","L") prints 7.

PRINT INSTR(3,"CENTRE","E") prints 6.

CPC 464 User Instructions 8.22

LEFT\$ copies the left-hand part of a word. (See page 62.)

1Ø a\$="FELLOW"

2Ø b\$=LEFT\$(a\$,4)

3Ø PRINT b\$

prints **FELL**.

CPC 464 User Instructions 8.23

LEN gives the number of characters in a word. (See page 64.)

1Ø K=LEN("BEAUMONT")

2Ø PRINT K

prints 8.

CPC 464 User Instructions 8.24

LOWER\$ changes the letters in a word from capitals to small letters.

PRINT LOWER\$("HELLO") prints **hello**.

CPC 464 User Instructions 8.26

MID\$ copies the middle part of a word. (See page 62.)

1Ø a\$="TRIPLET"

2Ø b\$=MID\$(a\$,2,3)

3Ø PRINT b\$

prints **RIP**.

CPC 464 User Instructions 8.27

RIGHT\$ copies the right-hand part of a word. (See page 62.)

1Ø a\$="CRUMPET"

2Ø b\$=RIGHT\$(a\$,3)

3Ø PRINT b\$

prints **PET**.

CPC 464 User Instructions 8.4Ø



SPACE\$ prints spaces.

PRINT SPACE\$(12) prints a row of 12 spaces.

CPC 464 User Instructions 8.43



STRING\$ repeats a character a given number of times.
PRINT STRING\$(7,"*") prints *****.
CPC 464 User Instructions 8.46

UPPER\$ changes the letters in a word from small letters to capitals.
PRINT UPPER\$("goodbye") prints **GOODBYE**.
CPC 464 User Instructions 8.49

Number functions

ABS gives the positive value of a number.
PRINT ABS(-7) gives 7.
CPC 464 User Instructions 8.3

ATN gives the angle whose tangent is known. The result will be in radians, unless the computer has been told otherwise by the **DEG** command.
CPC 464 User Instructions 8.4

CINT rounds up or down to the nearest whole number.
PRINT CINT(35.76) gives 36.
CPC 464 User Instructions 8.6

COS gives the cosine of an angle. The computer will assume the measurement is in radians unless told otherwise by the **DEG** command.
CPC 464 User Instructions 8.8

DEG tells the computer to calculate the values of functions like **COS** in degrees rather than radians.
CPC 464 User Instructions 8.10

EXP gives e (approximately 2.7183) to the power specified.
CPC 464 User Instructions 8.17

INT gives the whole number part of a number. (See page 62.)
INT(45.61) gives 45.
INT(-17.22) gives -18.
CPC 464 User Instructions 8.22

LOG gives the natural logarithms of numbers.
CPC 464 User Instructions 8.25

LOG10 gives logarithms of numbers to base 10.

CPC 464 User Instructions 8.26

MOD gives the remainder after division.

PRINT 11 MOD 4 gives 3.

PRINT 14 MOD 3 gives 2.

PI is a constant. Its value is 3.14159265. The circumference of a circle of radius r is $2*PI*r$.

CPC 464 User Instructions 8.34

RAD tells the computer to calculate the value of functions such as **COS** in radians. **RAD** cancels out the command **DEG**.

CPC 464 User Instructions 8.37

RND gives a random number between 0 and 1 (but not 1). It must be used with **INT** to produce whole numbers. (See pages 62 and 63 for examples.)

CPC 464 User Instructions 8.40

SGN gives the sign of a number.

1 for positive

-1 for negative

0 for zero.

CPC 464 User Instructions 8.42

SIN gives the sine of an angle. The computer will assume the measurement is in radians unless told otherwise by the **DEG** command.

CPC 464 User Instructions 8.42

SQR gives the square root of a number.

PRINT SQR(16) prints 4.

CPC 464 User Instructions 8.45

TAN gives the tangent of an angle. The computer will assume the measurement is in radians unless told otherwise by the **DEG** command.

CPC 464 User Instructions 8.48



Operators

AND is used to check whether two conditions hold. (See page 69.)

IF x>=0 AND x<10 THEN PRINT "digit"

CPC 464 User Instructions 4.18

NOT is used with **IF** and **THEN** to test if something is not true.

IF NOT a=2 THEN PRINT "no" will print **no** if **a** does not equal **2**.

CPC 464 User Instructions 4.18

OR is used to check whether one or another condition holds. (See page 71.)

IF a=3 OR b=6 THEN PRINT "yes" will print **yes** if either **a=3** or **b=6** (or both **a=3** and **b=6**).

CPC 464 User Instructions 4.18

XOR is used to check whether one or another condition holds (but not both).

IF person1=girl XOR person2=girl THEN PRINT "Marriage allowed".

CPC 464 User Instructions 4.18



Saving and loading programs

CAT is used to read out the contents of a cassette tape.

CPC 464 User Instructions 2.7

LOAD loads a program from a cassette.

LOAD "WATER" will load a program called **WATER** into the computer's memory, but the program will not be run: compare **RUN** on page 86.

CPC 464 User Instructions 8.25

SAVE saves a program onto a cassette.

SAVE "MISSILE" saves a program and calls it **MISSILE**.

CPC 464 User Instructions 2.6

SPEED WRITE is used to set the speed at which programs and data are saved onto cassette. The computer will normally save at 1000 baud, but **SPEED WRITE 1** will alter the saving speed to 2000 baud. The faster speed is less reliable.

CPC 464 User Instructions 2.6

General commands and functions

CLEAR clears all the variables written in a program.

CPC 464 User Instructions 8.7

DATA is used with **READ** to supply data for a program.
(See page 61.)

DATA 1,2,3,4,5

DATA apple,banana,carrot,date

CPC 464 User Instructions 4.14

DEF FN is used to define a function.

CPC 464 User Instructions 8.10

DIM allows groups of words or numbers to be put into the computer.

DIM x(20) allows numbers to be put into the computer with addresses **x(0)**, **x(1)**, **x(2)** ... **x(20)**.

DIM x\$(20) allows words to be put into the computer with addresses **x\$(0)**, **x\$(1)**, **x\$(2)** ... **x\$(20)**.

CPC 464 User Instructions 4.13

END tells the computer that the program has reached the end.

CPC 464 User Instructions 8.13

FOR/TO/STEP is used with **NEXT** to repeat the same lines many times. (See page 60.)

CPC 464 User Instructions 8.18

GOSUB is a program instruction to go to a subroutine which starts at a given number.

GOSUB 550 tells the program to go to a subroutine which starts at line number 550. **RETURN** is used to mark the end of the subroutine.

CPC 464 User Instructions 8.18

GOTO is a program instruction to go to a specific line number, skipping out any lines in between. (See page 67.)

GOTO 410 tells the program to go to line 410.

CPC 464 User Instructions F2.5





IF/THEN/ELSE tells the computer to do different things depending on which of various conditions holds true. (See page 65.)

IF a\$="GOOD" THEN GOTO 600 ELSE GOTO 750
CPC 464 User Instructions 8.19

INPUT allows words or numbers to be put into the computer. (See page 56.)

INPUT a for numbers.

INPUT a\$ for words.

CPC 464 User Instructions F2.6

NEXT is used with **FOR/TO/STEP**.

CPC 464 User Instructions 8.29

ON/GOSUB and **ON/GOTO** are used for options.

ON x GOSUB 100,200,300 means that if **x=1** the program goes to the subroutine starting at line 100, if **x=2** it goes to the subroutine starting at line 200, etc.

ON x GOTO 100,200,300 works in the same way.

CPC 464 User Instructions 8.29

READ is used with **DATA** to read information. (See page 61.)

10 READ A\$,B\$,C\$,D\$,E\$

20 DATA red,green,orange,blue,pink

30 PRINT B\$

prints green.

CPC 464 User Instructions 4.14



RETURN is used with **GOSUB** in order to return to the main body of a program.

CPC 464 User Instructions 8.40

WEND is used to end a loop beginning with **WHILE**. (See page 64.)

CPC 464 User Instructions 8.51

WHILE, used with **WEND**, sets up a loop so that the computer repeatedly executes the program lines within the loop while a particular condition holds true. (See page 64.)

CPC 464 User Instructions 8.51

A child's guide to the

AMSTRAD MICRO

Here are five friendly experts, who'll help you to use your Amstrad microcomputer. See how to write your own programs – draw pictures, play tunes, tell jokes, even run a game park!

P.C. Truemo

"I'll help you make sure the computer understands you – and investigate why, when it doesn't."



Pru Comet

"I'll help you understand the keyboard and learn the computer's language."



Ms O.C. Termup

"I'm in charge of the reference section of this book. I'll help you look up terms and make your own reference file."



Mort Puce

"I'm an artist and I'll show you how to draw patterns, change colours and make music with your computer."



Prof. O. Crumpet

"Ah! What a beautiful thing is a well written program! I'll show you how to plan it, write it, and run it."



THE UNIVERSITY OF CHICAGO PRESS

50 East Erie Street, Chicago, IL 60611, USA
Tel: +1 773 702 7000 Fax: +1 773 702 9232
www.press.uchicago.edu

9780226083000 HARDCOVER \$100.00
9780226083017 PAPERBACK \$35.00

9780226083024 PAPERBACK \$35.00
9780226083031 PAPERBACK \$35.00

9780226083048 PAPERBACK \$35.00
9780226083055 PAPERBACK \$35.00

9780226083062 PAPERBACK \$35.00
9780226083079 PAPERBACK \$35.00

9780226083086 PAPERBACK \$35.00
9780226083093 PAPERBACK \$35.00

9780226083100 PAPERBACK \$35.00
9780226083117 PAPERBACK \$35.00

9780226083124 PAPERBACK \$35.00
9780226083131 PAPERBACK \$35.00

9780226083148 PAPERBACK \$35.00
9780226083155 PAPERBACK \$35.00

9780226083162 PAPERBACK \$35.00
9780226083179 PAPERBACK \$35.00

9780226083186 PAPERBACK \$35.00
9780226083193 PAPERBACK \$35.00

9780226083200 PAPERBACK \$35.00
9780226083217 PAPERBACK \$35.00

9780226083224 PAPERBACK \$35.00
9780226083231 PAPERBACK \$35.00

9780226083248 PAPERBACK \$35.00
9780226083255 PAPERBACK \$35.00

9780226083262 PAPERBACK \$35.00
9780226083279 PAPERBACK \$35.00

9780226083286 PAPERBACK \$35.00
9780226083293 PAPERBACK \$35.00

9780226083300 PAPERBACK \$35.00
9780226083317 PAPERBACK \$35.00



Document numérisé avec amour par

AMSTRAD

CPC 

MÉMOIRE ÉCRITE



<https://acpc.me/>