

THE MICRO USER

No. 2 £1.50

Education Special

Logo on the BBC Micro

An in-depth study of
the learning language



Programs to help with
early reading · spelling · factors
punctuation · angles · simple sums

DIY DATABASE FOR JUNIORS

15 software packages reviewed

Full-length programs to
stimulate, educate and
entertain — at home
and in the classroom

9

NOW MICROVITEC PUTS YOU IN TOUCH FOR £210*

The Touchtech 501 eliminates the need to use the conventional keyboard and opens up the world of computers to everyone! – a truly major break-through.

Touchtech 501 is an intelligent device with its own on-board computer. Based on the latest infra-red scanning techniques, it permits unique and direct interaction with the computer's monitor screen!

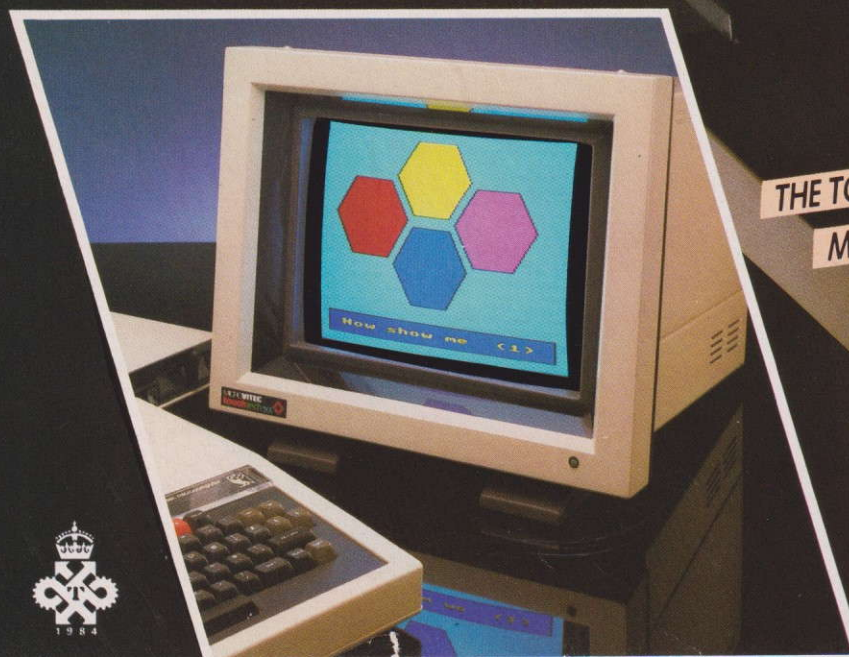
**ALL YOU DO IS TOUCH THE SCREEN!
TOUCHTECH DOES THE REST!**

Touchtech 501 has been purpose designed to operate in conjunction with Microvitec's immensely successful metal cabinet colour monitors, and the BBC computer. Not only do the units harmonise in design and colour, Touchtech 501's supports raise the monitor to the ideal angle for touch screen operation.

Although providing touch screen facilities at a fraction of the cost normally associated with this type of advanced technology, Touchtech 501 is an intelligent device in its own right, giving four point modes and four continuous modes, providing the imaginative programmer with unlimited design possibilities.

Touchtech 501 is designed to plug into the BBC B microcomputer and will be compatible with a wide range of other popular makes, including RML 480Z, Nimbus, Spectrum (with interface 1) etc.

* excluding VAT.



**THE TOUCHTECH 501 COMES COMPLETE WITH OPERATOR'S
MANUAL AND A FREE DEMONSTRATION DISK,
CONTAINING NINE STARTER PROGRAMS.
PLUS PROGRAM DEVELOPMENT SOFTWARE.**

For full details and colour brochure contact:

MICROVITEC
touchtech 501

Microvitec PLC, Futures Way, Bolling Road, Bradford, West Yorkshire
BD4 7TU Tel: (0274) 390011/726500. Telex: 517717

This equipment has been developed by Microvitec PLC in collaboration with the
Microelectronics Education Programme.

Education Special

Managing Editor
Derek Meakin

Features Editor
Mike Bibby

Production Editor
Peter Glover

Layout Design
Heather Sheldrick

Associate Editor
Alan McLachlan

Advertisement Manager
John Riding

Advertising Sales
John Snowden

Editor in Chief,
Database Publications
Peter Brameld

Tel: 061-456 8383 (Editorial)
061-456 8500 (Advertising)
061-480 0171 (Subscriptions)
Telecom Gold: MAG001
Telex: 265871 MONREF G.
(Quote Ref MAG001)
Prestel Mailbox: 614568383

Published by:
Database Publications Ltd,
Europa House, 68 Chester Road,
Hazel Grove, Stockport SK7 5NY.

ABC Member of Audit
Bureau of Circulations

The Micro User welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication will be on an all-rights basis.

© 1985 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles, listings or advertisements.

The Micro User is an independent publication and neither the BBC nor Acorn Computers Ltd are responsible for any of the articles in this issue or for any of the opinions expressed.

News trade distribution:
Europress Sales and Distribution Limited,
11 Brighton Road, Crawley, West Sussex
RH10 6AF. Circulation 0293 27053.

SPECIAL OFFER!

All the programs in this issue are available on tape or disc – and it's the best value ever in educational software.

– See coupon on
Page 73

CONTENTS

4 Case Study

We investigate Salford LGA's innovative approach to computing in the classroom.

6 Welcome to Logo

Logo is much more than Turtle graphics. Uncover some of its potential with our step by step introduction to the ideal language for educational computing.

8 Fun factors

Based on a popular arcade game this program tests the understanding of factors, at the same time providing hours of fun for the whole family.

14 Windmill

A gentle, pictorial introduction to word recognition for the very early reader.

20 Angler

Test your ability to estimate angles with this easy to play mathematical game.

24 Four Logos

Acornsoft, Logo Software, Logotron and the Open University have all produced Logo for the BBC Micro. This in depth appraisal of them all should help you decide on the most suitable for your needs.

31 Spelldroid

Is it "might" or "mite", "tea" or "tee"? Use our friendly robot in a novel way to reinforce vocabulary and spelling skills.

36 Bookshelf

Our experts cast a critical eye over six recent publications devoted to educational computing.

40 Tortal

Designed for infants, this delightful graphic program not only tests, but also teaches the simple rules of addition.

45 Software Scene

Our experts in the educational field present their detailed evaluation of some of the latest software releases.

50 Discovery

Combine strategy, skill and a little guesswork in this text game as you attempt to identify well known phrases or sayings. The ability to enter your own words extends the scope of this fascinating program.

57 Punctuation

Test and reinforce the rules of punctuation with this easy-to-use utility. Its facility for creating your own "tests" makes it suitable for a wide range of ages.

62 Logo – List processing

An easy-to-follow introduction to one of the fundamental skills in programming in Logo.

67 Junior Database

Developed and thoroughly tested in a primary school environment, this superb program introduces children to data processing in a simple and user-friendly way.

74 Chinese Takeaway

Designed to complement Tortal this graphic program teaches and tests the basic rules of subtraction.

Exploring the potential of the BBC Micro in Hilda Ogden country

The Salford Experiment

THE mere name of Salford is sufficient to conjure up mental images of Coronation Street, with Hilda Ogden in those lumpy curlers and the apparently ageless tabby among the drab back-to-backs as the credits roll.

However, it seems that the residents are increasingly objecting to being typecast as sporting flat-caps and staggering home from the taprooms every dole-day to kick wives, whippets or racing pigeons.

In fact, the local councillors have embarked on a crusade to counteract this unattractive vision of their fair city.

Visitors to Salford today will discover the bulldozers have been very busy.

Acre after acre of slum property has been reduced to rubble as the city – like the Empire on our cinema screens – fights back.

And to ensure that the next generation of ratepayers will not be labelled “underprivileged”, the Salford Experiment has been born.

Based on the BBC Micro, it is the brainchild of the city’s far-sighted education committee, which already has one other first to its credit.

That is – that it steered the authority to adopt a total nursery program, ensuring places for every two-year-old and upward in Salford.

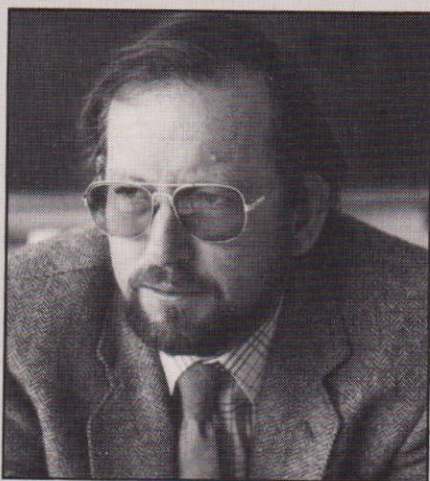
With the city’s tots happily esconced in their corporation sandpits, the education authority flexed its muscles and decided to tackle THE problem – unemployment facing school-leavers.

A radical rethink was obviously called for – and that’s what it got with the Salford Experiment seeing the light of day as the end-result.

“We figured we knew all about unemployment here in Salford”, said one committee member.

“After all, Walter Greenwood wrote Love on the Dole in our Hanky Park.

“So we decided to see if there was any practical weapon in our schools’



John Harris... “what they really should be learning about is the practical applications of micros”

armoury which wasn’t being wielded properly.

“And the more we looked, the more we realised it was the BBC Micro. We’d got it all wrong”.

The man called in by Salford to head up the experiment was the first to agree.

A former science teacher, John Harris soon realised that computer science courses were not the optimum way to enable students to benefit from computers.

“Up until then, everyone had been

thinking in terms of turning out programmers and systems analysts”, he said.

“And those students without these sort of skills were not benefiting from the micros.

“But, as in reality far more of our kids will be trying to get jobs at Tesco’s, what they really should be learning about is the practical applications of micros – such as in computerised tills”.

It was with this in mind that Salford decided to reject the concept of restricting the use of micros to computer science courses in its high schools.

Instead, it opted for a computer environment in which classes of all types could be held, using micros as essential tools.

“Whether the subject be geography or maths – or even PE – we felt there could be a practical application for the BBC Micro”, says John Harris.

To achieve this the Salford microcomputer development adviser designed a “computer suite” for each of the city’s 30 high schools.

All of these incorporate a minimum of 10 BBC Micros strategically arranged on workstations positioned on multi-level benching.

Even the seating has been carefully selected for the project. Top-quality/



Instructor John Redgate unravels the mysteries of the computer for students at one of three summer schools being held in Salford this year. The city is hoping that the renewed interest in micros will help combat unemployment.

adjustable secretarial chairs can be found in all the computer suites.

"We wanted to make sure that we gave microcomputing the status it deserves with these suites", says John Harris.

"That's why we went out of our way to make sure they didn't end up looking like the offspring of amusement arcades.

"Nor are they just converted classrooms".

And the council put its money where its mouth is on this project – paying out in the region of £12,000 per unit.

According to John Harris, the Salford Experiment is already producing positive results.

"We are at long last exploring the potential of the BBC Micro in our schools", he says.

"If you can imagine an RE class where the kids are feeding in all the variable facts about Moses crossing the Red Sea, you'll have some idea of what I mean.

"The computer, after all, can model and emulate in a way we would never have dreamed of before".

However, the Salford spokesman sees the major impact of the BBC Micros in his city's schools as helping develop decision-making skills, an ability that's becoming an essential job

requirement.

"To be brutally honest, in the past, teachers have never been very good at this, thanks to the constrictions of the curriculum", he says.

"But now society is getting more complex and there are multiple answers – shades of grey.

"The micro can solve that type of problem because it can handle so many variables".

Such is the renewed interest in micros in Salford schools that the enthusiasm has overflowed into the city's computer summer schools.

Last year only one centre was open, for a month.

This year there were three and they still could not cater for the demand.

"Once again we've been dealing with the kids who want to use computers in practical applications", says the course adviser.

"We are at last achieving what we set out to do. That is to move computers out of the field of computer junkies and give the micro a practical perspective.

"And that will undoubtedly be translated into jobs for our next wave of school-leavers.

"Perhaps when the rest of the country has witnessed this, they'll stop thinking about us in terms of Coronation Street..."

The Mini solution

THE Salford Experiment has chosen Mini Office, the best-selling business package from Database Software, as an essential teaching aid.

A finalist in two major categories in the 1985 British Microcomputing Awards, Mini Office was selected to help bridge the gap between school and working environments for the students.

"It makes working with computers just that much more realistic", says John Harris, Salford's computer adviser.

"And I'm sure it will prove the key to many of our youngsters getting jobs when they leave school".

Salford opted for Mini Office because of its four modules – word processing, database, spreadsheet and graphics – enable a BBC Micro to be turned into an inexpensive office tool.

"At its revolutionary low price of £5.95, it is a truly cost-effective way of introducing the business concepts of computers to our students", says John Harris.

"Based on our experience I feel the package will be well worth looking at by an education authority".

WELCOME TO L

DEREK RADBURN takes you through your first encounter with LOGO – and shows it's much more than Turtle graphics

WHEN you first switch on your BBC computer, with the Logo ROM or ROMs installed in the correct sockets, you will receive the message "Welcome to Logo".

This message typifies much of the philosophy which underlines Logo – that of making the machine friendly to the user. As you progress you will experience the helpfulness of Logo error messages, the consistency of its syntax and the clarity of its semantics.

Ask most computer buffs what Logo is, and it's odds-on they reply: "It's that graphics language". It is often assumed that turtle graphics is pretty well all there is to Logo. This is not so.

Logo is derived from Lisp, the premier language of the AI (artificial intelligence) world. It was the first language to lift computing beyond numbers into the symbolic representation of data by lists.

Logo has inherited much of this capacity from its parent, but with a user-friendliness not yet present in the parentheses-ridden world of Lisp.

The heart of the Logo system are words and lists. These are the only two data types which exist in most Logos. Words are the general dogsbodies of the world of Logo. They are used as commands and variables.

When you start your Logo, the system begins knowing certain words – these are known as Logo primitives.

You may then add new words of your own with definitions using these primitives. The definitions come in the form of procedures which begin with the word TO.

This tells the system that the word following the TO is the name of a new procedure, making Logo both extensible and adaptable.

A word becomes a variable when a

value is assigned to it with the MAKE command. The value can be numeric (numbers are special forms of words) or it may be another word, or a list, or the name of a primitive, procedure or variable.

It should be clear from this that Logo possesses a high degree of flexibility. If the word is the general dogsbody of Logo then the list is the matrix within which they operate.

Lists may be made up from words or other lists or a combination of both. There are special commands in the system for constructing and dismantling lists.

The syntax of Logo is highly consistent. A word is a group of characters delimited by a space. In Logo, therefore, spaces are significant. If a word appears with a quote mark " immediately in front of it, without any intervening spaces, then the system leaves it alone and treats it just as it is. For example PRINT "ACORN would result in ACORN being displayed on the screen.

If there is a colon : (dots in Logoese) in front, then the system attempts to evaluate the word by looking for any value which has been assigned to it.

Suppose you had typed:

```
MAKE "ACORN "OAK
```

then

```
PRINT :ACORN
```

would cause OAK to be displayed on the screen. If you had not assigned any value to ACORN, and attempted to refer to a non-existent value, then the system would respond with the error message "ACORN HAS NO VALUE".

If this happened in a procedure, then you would be told the name of the procedure and the line context.

Friendly, isn't it?

Finally, if a word is typed without either of the preceding two marks in front of it, then the system tries to execute it. It sees it as either a primitive or a procedure which has to be carried out.

If the word is not a primitive or has not been defined as a procedure, then you will receive the message "I DON'T KNOW HOW TO ACORN". Again, if this occurred within a procedure you would receive more information.

The informative error messages are a Logo characteristic, unequalled by imitation Logos.

The delimiters of a list are the square brackets []. Everything within them is part of the list.

It is worth briefly pointing out how Logo distinguishes between the name of something and the value attached to it.

The logical affront of the Basic assignment LET X = X + 3 is avoided.

```
MAKE "X :X + 3
```

reads *make the thing called X by taking the value attached to X with 3 added.*

Let's now play with some Logo. Let's try a trivial procedure:

```
TO SQUIGGLE  
FORWARD 150  
RIGHT 130  
FORWARD 150  
LEFT 130  
FORWARD 150  
RIGHT 130  
END
```

When you start to type this, you will notice a change from the usual question mark? Logo prompt to the caret mark >, this signifies that you are in the defining mode. That is, the computer has ceased to behave inter-actively and is storing your input.

You can escape from this by typing END on a line on its own, or by pressing Escape. If you type END, the procedure will be defined, but if you press Escape,

‘Logo possesses a high degree of flexibility’

OGO

then the definition will be aborted (in Logotron Logo you will also get the message Stopped..).

Procedures are really special forms of lists. A way of illustrating this is to look at the RUN command in Logo. RUN does not behave as it does in Basic, for in Logo no command other than the typing of a procedure's name is needed to invoke it into activity.

What RUN does is to allow a list to be run as if it had been typed in at the keyboard. Try typing this:

```
RUN [FD 150 RT 130 FD 150 LT 130 FD
150 RT 130
```

Also try:

```
MAKE "SQUIGGLE [FD 150 RT 130 FD 150
LT 130 FD 150 RT 130
```

and then follow it with

```
RUN :SQUIGGLE
```

What should become clear from this is that it is possible to have the same name shared by procedures and variables.

If you typed:

```
RUN "SQUIGGLE
```

you would get the error message: "RUN DOES NOT LIKE "SQUIGGLE AS AN INPUT".

The reason for this is that RUN expects a list as its input and you will have given it a word.

Remember, "SQUIGGLE is the word squiggle, whereas :SQUIGGLE is the value attached to SQUIGGLE, which is a list.

Logo's loop structure is the REPEAT loop. REPEAT is a primitive that needs two inputs, or arguments. The first argument must be a number, or a variable with a numeric value, to indicate how many times to repeat.

The second argument must be a list which contains what has to be repeated.

You should be able to see, here, how REPEAT and RUN are related – REPEAT really being a development of

the RUN command.

It is also possible to think of procedures as being lists which have a RUN command built-in at their beginning.

Logo has testing structures. The commonest is very similar to the one found in Basic and other languages – IF condition THEN do this ELSE do that. The difference is that, in most Logos, the THEN and ELSE do not get written.

IF is simply followed by one or two lists – the second is optional. If the condition is found to be true, then the first list is executed, if it is false, then if the second list is present, it is run.

This conditional test has to be typed as one line. Acornsoft Logo possesses another conditional structure (which is not present in Logotron) which is very useful.

This is TEST, which tests a condition and flags true or false. The complementary commands are IFTRUE and IFFALSE.

At first sight it might seem that there was little difference between TEST and IF, but in fact there is. TEST occurs on a line on its own. IFTRUE and IFFALSE have to occur on lines of their own, and most usefully they do not have to be in a set order.

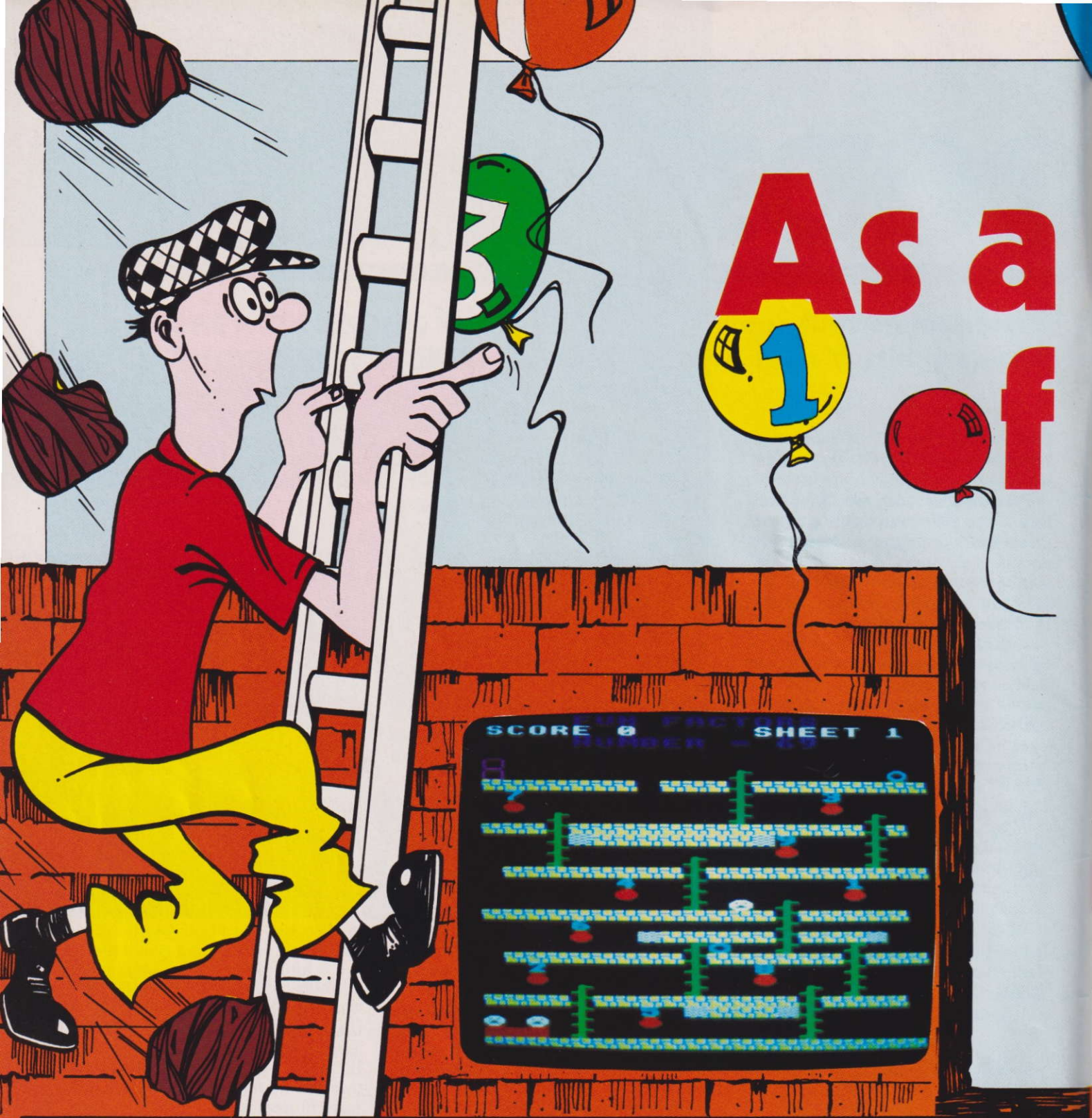
This means that it is possible to test for a false condition before a true one.

Moreover, it is possible to have each of these branches occur more than once after TEST. For example, you could have two lines beginning with IFTRUE or IFFALSE with other lines in between. From these simple conditional tests it is possible to write more.

That concludes our survey of Logo's fundamentals. As you can see, it's much more than Turtle graphics.

While Turtle graphics does unquestionably provide a good and animated vehicle for learning about Logo and its structures, hopefully this article will have revealed that Logo is considerably more than a graphics program.





PROGRAM STRUCTURE

- 30-100** Set main variables, graphics characters, envelopes and data for tune. Display the instructions and play tune.
- 110-150** Set a random number and calculate how many of the digits 1-9 are not factors of this number.
- 160-200** Select screen 1, 2 or 3 depending on score reached and display it.
- 210-240** Set initial position of moving rocks and men.
- 250-310** Check input for movement of man.
- 320-420** Check position of man and where he is to move. Check for any factors hit or for any non-factors missed.
- 430-440** Movement of man.
- 450-500** Movement of rocks.

- 510-530** All factors found. Proceed to next number and play tune.
- 540-560** Destruction of man, loss of one life.
- 570-630** Start and end of a jump, with safe landing check.
- 640-650** Fall of man.

PROCEDURES

- ladder** Display for ladder.
- screen1** Display first or second screen layout.
- screen2** Display third screen layout.
- instructions** Display rules and instructions.
- balloon** Sound for balloon bursting.
- checkjump** Check if man is in a position to jump.
- factor1** Check for digits 1-9 on screens 1 and 2.
- factor2** Check for digits 1-9 on screen 3.



matter factors

By ERIC HARPER

BASED on one of the most popular arcade-type games, **Fun Factors** is an educational game designed to give the player an entertaining way of learning about single digit factors.

The object is to spot all the factors of a random number selected by the computer, and then to demonstrate both your mathematical and playing skills.

Your man has to climb the ladders, walk along the walls, jump the gaps – at the same time avoiding the rolling rocks, which will crush him.

Scattered around the screen are nine

balloons, numbered 1 to 9.

While moving around the screen, your man has to burst all the balloons which are not factors of the number selected by the computer, by walking under them.

If he tries to walk under a balloon which is a factor, he loses a life.

He will often have to climb up one ladder and down another when a factor is blocking the way.

When only the balloons numbered with factors remain, you have to move your man to the top wall, jump the gap

and touch the large 8 at the top left of the screen.

The computer will then check that you have correctly burst all the non-factor balloons, select a new number and reset the screen.

Should you have missed any, the computer will tell you so, in which case you will have to return to these and burst them.

You score 50 points for each balloon burst, with a bonus of 100 points when you reach the top and only factors remain on the screen.

Lose three lives and your game is over.

To make the game more difficult, or more interesting – whichever you prefer – you cannot jump your man to avoid the rocks when he is under or too near the walls with chequered ends.

You can gain thinking time to work out the factors by parking your man half-way up a ladder or on sections of walls where the rocks do not roll.

There are three levels of play and you automatically progress to level 2 when your score reaches 1000 and to level 3 when it reaches 2000.

The high score of the current set of games is displayed at the beginning of each game.

To move your man, press Z to move right, X to move left, : to move up, / to move down.

Press Space to make your man jump and Z or X with Space to jump a gap.

VARIABLES

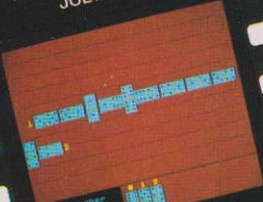
A%(n) X coordinate of rock(n).
B% X position near wall end.
B%(n) Y coordinate of rock(n).
D% Duration of note for tune.
D%(N) Direction of rock(n).
E%(n) Vertical movement of rock(n).
F% Value of digit.
F%(n) Direction of rock before falling.
G% Number of non-factors.
H% Man's horizontal movement.
I
I% } General variables for loops and counts.
J }
J% }
K }
K% Vertical position of jump.
L% Number of sheet.
M% Horizontal movement of jump.

N% Vertical movement of jump.
O% Character that the man is over.
O%(n) Colour that the rock is over.
Q% Colour that the man is over.
Q%(n) Character that the rock is over.
R% Character man is to move over.
R%(n) Character rock is to move over.
S% Player's score.
T% Colour man is to move over.
T%(n) Colour rock is to move over.
V% Man's vertical movement.
W% Number of rocks.
X% X coordinate of man.
Y% Y coordinate of man.
BA% Non-jump.
LIV% Number of lives.
PO% Point value of position man is to move to.
P% Random number.

AUTO DISC MENU
AUG 1984



FAST FILL ROUTINE
JULY 1984

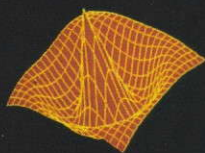


DOMINOES
MAY 1984

90130: Function Key Editor

- 1) - List keys
- 2) - Edit key
- 3) - Delete key(s)
- 4) - Compact to home
- 5) - Print key buffer space left
- 6) - Expand keys
- 7) - Update definitions
- 8) - Exit

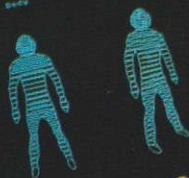
FUNCTION KEY EDITOR
MAY 1984



3D SURFACES
JAN 1985



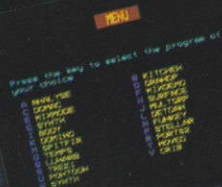
ROM READER
MARCH 1984



BODY POPPING
MARCH 1985



GRAND PRIX
JUNE 1984

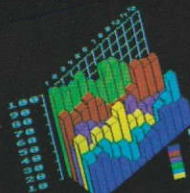


AUTO DISC MENU
AUG 1984

BEEBUG CAD PROGRAM
DEC 1983



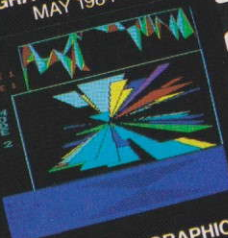
HOME ACCOUNTS PROGRAM
NOV 1984



3D BAR CHART GENERATOR
APRIL 1984



GRAPH PLOTTER
MAY 1984



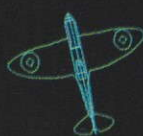
MULTI-MODE GRAPHICS
APRIL 1985



DETONATOR DAN
MAY 1984



MUSIC SYNTHESISER
AUG 1984



SPITFIRE
MAY 1984



BEEBUG CAD PROGRAM
DEC 1983

GET IN THE PICTURE WITH BEEBUG

JOIN BRITAIN'S LARGEST
COMPUTER USER GROUP
TO GET THE MOST
FROM YOUR
BBC MICRO

● JOIN BY SENDING £11.90
FOR 1 YEAR MEMBERSHIP
(10 MAGAZINES)

OR £6.40 for 6 MONTHS
MEMBERSHIP (5 MAGAZINES).

OR SEND £1.00 FOR
A SAMPLE ISSUE (UK ONLY)

● UK ONLY —
OVERSEAS PRICES ON APPLICATION

● MAKE CHEQUES PAYABLE TO
BEEBUG
PO BOX 109 HIGH WYCOMBE
BUCKS.

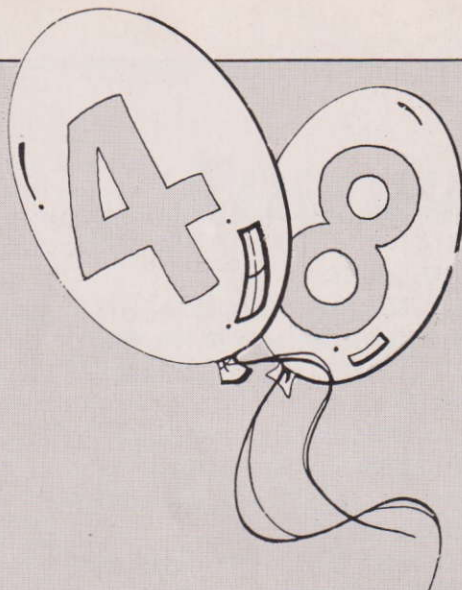
● NEWS
GAMES
REVIEWS
UTILITIES
GRAPHICS
SOFTWARE
DISCOUNTS
PROGRAMS
HINTS & TIPS
USER GROUPS
INFORMATION
APPLICATIONS
SPECIAL OFFERS
TEN MAGAZINES
PER YEAR
TECHNICAL SUPPORT

From Page 9

```

10 REM FUN FACTORS
20 REM by Eric Harper
25 REM (C) The Micro User
30 DIM A%(2),B%(2),D%(2),E%(2),F%(
2),O%(2),T%(2),R%(2),Q%(2):S%=0:TP%=0
40 VDU 23,224,17,17,241,17,159,132
,132,132,23,225,126,129,129,129,129,1
29,129,126,23,226,24,31,24,24,24,248,
24,24,23,227,60,126,255,255,255,255,1
26,60
50 VDU 23,228,28,62,99,65,65,99,62
,28,23,229,60,126,219,255,231,195,126
,60,23,230,0,0,0,0,32,118,255,23,23
1,51,204,51,204,51,204,51,204,23,255,
255,255,255,255,255,255,255,255
60 ENVELOPE 1,1,0,0,0,0,0,127,-4
,-3,-2,126,0:ENVELOPE 2,1,50,-30,10,1
00,-50,5,127,-5,-4,-2,126,0:ENVELOPE
3,2,0,-25,75,5,100,-10,127,-4,-2,-1,1
26,0:ENVELOPE 4,1,5,5,5,-2,-4,-6,127,
0,-4,-3,126,0
70 DATA 115,2,100,5,115,2,100,5,75
,2,60,3,60,2,75,3,75,2,60,3,150
80 LX=1:WX=2:MODE 7:VDU23;8202;0;0
;0;0;:PROCInstructions
90 LIVX=3:SX=0:MODE 2:VDU23;8202;0
;0;0;0;
100 FOR I = 1 TO 10:READ NX,DX:SOUN
D 1,1,NX,DX:NEXT I:RESTORE:FOR J = 1
TO 1000:NEXT J
110 IF SX<1000 THEN PX=RND(100) ELS
E IF SX>2000 THEN PX=RND(300) ELSE PX
=RND(200)
120 FX=PX+1:GX=0
130 FOR K = 1 TO 9
140 IF PX/K<INT(PX/K) THEN GX=GX+1
150 NEXT K
160 COLOUR 129:COLOUR 4:PRINT TAB(0
,30);" "
170 IF SX<2000 THEN PROCscreen1 ELS
E PROCscreen2
180 IF SX<1000 THEN LX=1 ELSE IF SX
>=1000 AND SX<2000 THEN LX=2 ELSE IF
SX>=2000 THEN LX=3
190 COLOUR 12:COLOUR 128:PRINT TAB(
4,1);"FUN FACTORS";TAB(4,3);"NUMBER =
";PX:COLOUR 7:PRINT TAB(0,2);"SCORE
";SX:TAB(12,2);"SHEET ";LX
200 VDU 5:BCOL 0,3:MOVE 0,31:PRINT
STRING$(20,CHR$(255)):BCOL 0,4:MOVE 0,
31:PRINT STRING$(20,CHR$(224)):VDU 4
210 AX(1)=19:AX(2)=0:B%(1)=6:B%(2)=
18:D%(1)=-1:D%(2)=1:COLOUR 6:FOR IX =
1 TO WX:FX(IX)=0:QX(IX)=32:RX(IX)=32
:E%(IX)=0:PRINT TAB(AX(IX),B%(IX));CH
R$(228):NEXT IX
220 XX=6:YX=30:BX=0:OX=32:JX=0
230 COLOUR 7:PRINT TAB(XX,YX);CHR$(

```



```

229:TAB(0,29);STRING$(LIVX-1,CHR$(229
+" "))
240 VDU 23,1,0;0;0;0;
250 IF JX=1 THEN GOTO 570
260 HX=0:VX=0
270 IF INKEY (-98) THEN HX=-1:VX=0
280 IF INKEY (-67) THEN HX=1:VX=0
290 IF INKEY (-105) THEN VX=1:HX=0
300 IF INKEY (-73) THEN VX=-1:HX=0
310 IF INKEY (-99) THEN PROCcheckju
mp
320 IF XX+HX<0 OR XX+HX>19 THEN HX=
0
330 IF XX+HX=3 AND YX=30 THEN HX=0
340 IF GX<>0 AND XX+HX=0 AND YX=6 T
HEN HX=0:PRINT TAB(0,4);"NON-FACTOR/S
MISSED":FOR I = 1 TO 3500:NEXT I:PRI
NT TAB(0,4);SPC (20)
350 IF POINT(XX*64+24,(32-YX)*32-16
)>7 THEN GOTO 540
360 IF HX=0 AND VX=0 THEN GOTO 450
370 RX=32:POX=POINT((XX+HX)*64+24,(
32-YX-VX)*32-4):IF VX<>0 AND POX<>2 A
ND OX=226 THEN VX=0:RX=226:TX=2 ELSE
IF VX<>0 AND POX<>2 THEN VX=0 ELSE IF
VX<>0 AND POX=2 THEN RX=226:TX=2
380 IF POX=2 THEN RX=226:TX=2
390 IF POX=5 AND GX=0 THEN GOTO 510
400 IF POINT((XX+HX)*64+24,(32-YX)*
32-36)<>4 AND POINT((XX+HX)*64+24,(32
-YX)*32-36)<>2 AND POINT((XX+HX)*64+2
4,(32-YX)*32-36)<>6 THEN GOTO 640
410 IF SX>2000 AND POINT(11*64+24,2
4*32-4)<>2 THEN PROCfactor2 ELSE PROC
factor1
420 IF PX/FX = INT(PX/FX) THEN PRIN
T TAB(0,4);"WRONG ";FX;" IS A FACTOR"
:GOTO 540
430 XX=XX+HX:YX=YX+VX:COLOUR 0X:PRI
NT TAB(XX-HX,YX-VX);CHR$(0X):COLOUR 7
:PRINT TAB(XX,YX);CHR$(229):OX=RX:QX=
TX
440 IF POINT(XX*64+24,(32-YX)*32+4)
=1 THEN SOUND 2,1,200,1:PRINT TAB(XX,
YX-1);" ";TAB(XX,YX-2);" ":6X=6X-1:SX
=SX+50:PRINT TAB(6,2);SX
450 FOR IX = 1 TO WX:IF E%(IX)=1 AN

```

```

D POINT(AX(IX)*64+24,(32-BX(IX))*32-3
6)=4 THEN SOUND 0,1,5,1:E%(IX)=0:D%(I
X)=-FX(IX):FX(IX)=0 ELSE IF E%(IX)=1
THEN GOTO 490
460 IF POINT(AX(IX)*64+24,(32-BX(IX
))*32-36)=0 THEN EX(IX)=1:FX(IX)=DX(I
X):DX(IX)=0
470 IF AX(IX)=3 AND BX(IX)=30 THEN
PRINT TAB(AX(IX),30);" ";AX(IX)=19:B
X(IX)=6:E%(IX)=0:QX(IX)=32:D%(IX)=-1
480 RX(IX)=32:IF POINT((AX(IX)+DX(I
X))*64+32,(32-BX(IX))*32-4)=2 THEN RX
(IX)=226:TX(IX)=2
490 COLOUR 0X(IX):AX(IX)=AX(IX)+DX(I
X):BX(IX)=BX(IX)+E%(IX):PRINT TAB(AX
(IX)-DX(IX),BX(IX)-E%(IX));CHR$(0X(IX
));COLOUR 6:PRINT TAB(AX(IX),BX(IX));
CHR$(228):OX(IX)=TX(IX):QX(IX)=RX(IX)
500 FOR J = 1 TO 10:NEXT J:NEXT IX:
GOTO 250
510 SOUND 3,3,100,1:SX = SX+100:PRI
NT TAB(6,2);SX:TAB(0,4);"ALL FACTORS
CORRECT":FOR I = 1 TO 3000:NEXT I
520 FOR I = 1 TO 10:READ NX,DX:SOUN
D 1,1,NX,DX:NEXT I:RESTORE
530 CLS:GOTO 110
540 SOUND 0,1,6,1:COLOUR 7:PRINT TA
B(XX,YX);" ";TAB(XX,YX);CHR$(230):VDU
19,0,4,0,0,0:FOR I = 1 TO 50:NEXT I:V
DU 20:FOR I = 1 TO 75:NEXT I:VDU 19,
0,4,0,0,0:FOR I = 1 TO 150:NEXT I:VDU
20
550 FOR I = 1 TO 4000:NEXT I:LIVX=L
IVX-1:IF LIVX=0 THEN GOTO 80
560 FOR I = 1 TO 3500:NEXT I:CLS:GO
TO 110
570 KX=KX+1:IF KX=3 THEN NX=-NX
580 RX=32:IF POINT((XX+MX)*64+24,(3
2-YX-NX)*32-4)=2 THEN RX=226:TX=2
590 IF XX+MX<0 OR XX+MX>19 THEN GOT
O 640
600 COLOUR 0X:XX=XX+MX:YX=YX+NX:PRI
NT TAB(XX-MX,YX-NX);CHR$(0X):COLOUR 7
:PRINT TAB(XX,YX);CHR$(229):OX=RX:QX=
TX:IF KX=4 THEN JX=0
610 IF JX<>0 THEN GOTO 450
620 IF POINT(XX*64+24,(32-YX)*32-36
)=0 THEN GOTO 640
630 GOTO 450
640 COLOUR 0X:PRINT TAB(XX,YX);CHR$(
0X:XX=XX+HX:COLOUR 7:IF POINT(64*XX+
24,(32-YX)*32-36)=4 THEN GOTO 540
650 REPEAT:YX=YX+1:PRINT TAB(XX,YX-
1);" ";TAB(XX,YX);CHR$(229):FOR I = 1
TO 100:NEXT I:UNTIL POINT(64*XX+24,(32
-YX)*32-36)=4:GOTO 540
660 DEF PROCcladder(x,y)
670 COLOUR 2:FOR IX = y TO y+4:PR

```


From Page 11

```

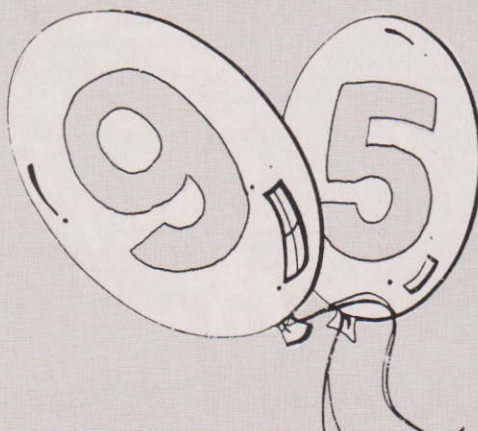
INT TAB(X%,I%)CHR$ 226:NEXT
680 ENDPROC
690 DEF PROCscreen1
700 COLOUR 131:PRINT TAB(0,7);STRIN
G$(20,CHR$ 224);TAB(0,11);STRING$(19,
CHR$ 224);TAB(4,12);CHR$ 231;STRING$(
7,CHR$ 224);CHR$ 231;TAB(1,15);STRIN
G$(19,CHR$ 224);TAB(0,19);STRING$(19,C
HR$ 224)
710 PRINT TAB(7,21);CHR$ 231;STRIN
G$(9,CHR$ 224);CHR$ 231;TAB(1,23);STRI
NG$(19,CHR$ 224);TAB(0,27);STRING$(19
,CHR$ 224);TAB(9,28);CHR$ 231;STRING$
(3,CHR$ 224);CHR$ 231
720 COLOUR 128:PROCladder(11,6);PRO
Cladder(3,10);PROCladder(17,10);PROCl
adder(9,14);PROCladder(13,18);PROClad
der(9,22);PROCladder(16,22);PROCladde
r(4,26);PROCladder(14,26)
730 PRINT TAB(7,7);" "
740 COLOUR 1:PRINT TAB(1,9);CHR$ 22
7;TAB(15,9);CHR$ 227;TAB(13,13);CHR$
227;TAB(6,17);CHR$ 227;TAB(16,17);CHR
$ 227;TAB(4,21);CHR$ 227;TAB(2,25);CH
R$ 227;TAB(12,25);CHR$ 227;TAB(7,29);
CHR$ 227
750 COLOUR 6:PRINT TAB(1,8);7;TAB(1
5,8);3;TAB(13,12);9;TAB(6,16);4;TAB(1
6,16);1;TAB(4,20);6;TAB(2,24);2;TAB(1
2,24);8;TAB(7,28);5
760 COLOUR 5:PRINT TAB(0,5);CHR$ 22
5;TAB(0,6);CHR$ 225
770 IF SX<1000 THEN GOTO 800
780 COLOUR 131:COLOUR 4:PRINT TAB(1
8,8);STRING$(2,CHR$ 224);TAB(18,9);CH
R$ 231;CHR$ 224;TAB(11,16);STRING$(5,
CHR$ 224);TAB(11,17);CHR$ 231;STRING$
(3,CHR$ 224);CHR$ 231;TAB(0,20);STRIN
G$(3,CHR$ 224);TAB(0,21);STRING$(2,CH
R$ 224);CHR$ 231
790 PRINT TAB(13,24);STRING$(3,CHR$
224);TAB(13,25);CHR$ 231;CHR$ 224;CH
R$ 231
800 ENDPROC
810 DEF PROCscreen2
820 COLOUR 131:PRINT TAB(0,7);STRIN
G$(20,CHR$ 224);TAB(0,11);STRING$(18,
CHR$ 224);TAB(5,13);CHR$ 231;STRING$(
11,CHR$ 224);TAB(1,15);STRING$(2,CHR$
224);CHR$ 231;TAB(4,17);STRING$(15,C
HR$ 224);TAB(4,18);STRING$(7,CHR$ 224
)
830 PRINT TAB(0,19);STRING$(10,CHR$
224);CHR$ 231;TAB(10,21);CHR$ 231;ST
RING$(9,CHR$ 224);TAB(1,23);STRING$(1
5,CHR$ 224);TAB(12,25);CHR$ 231;STRIN
G$(3,CHR$ 224);CHR$ 231;TAB(0,27);STR
ING$(19,CHR$ 224)

```

```

840 PRINT TAB(9,28);STRING$(5,CHR$
224);TAB(9,29);CHR$ 231;STRING$(3,CHR
$ 224);CHR$ 231
850 COLOUR 128:PROCladder(5,6);PRO
Cladder(16,6);PROCladder(2,10);PROClad
der(13,10);PROCladder(6,12);PROCladde
r(13,12)
860 PROCladder(8,16);PROCladder(17,
16);PROCladder(8,18);PROCladder(2,22)
:PROCladder(10,22);PROCladder(5,26);P
ROCladder(15,26)
870 PRINT TAB(7,7);" ";TAB(11,11);C
HR$ 231;" ";TAB(3,19);" ";TAB(7,27);"
"
880 COLOUR 1:PRINT TAB(10,9);CHR$ 2
27;TAB(17,9);CHR$ 227;TAB(1,13);CHR$
227;TAB(10,15);CHR$ 227;TAB(14,19);CH
R$ 227;TAB(5,21);CHR$ 227;TAB(4,25);C
HR$ 227;TAB(9,25);CHR$ 227;TAB(16,29)
;CHR$ 227
890 COLOUR 6:PRINT TAB(10,8);1;TAB(
17,8);8;TAB(1,12);6;TAB(10,14);5;TAB(
14,18);7;TAB(5,20);3;TAB(4,24);9;TAB(
9,24);4;TAB(16,28);2
900 COLOUR 5:PRINT TAB(0,5);CHR$ 22
5;TAB(0,6);CHR$ 225
910 ENDPROC
920 DEF PROCInstructions
930 IF SX>TP% THEN TP%=SX
940 PRINT TAB(13,0);CHR$ 129;"FUN F
ACTORS"
950 PRINT " The computer will sel
ect a number and you must decide which
of the numbers 1 to 9 are factors
(divide exactly) into the number. You
have to burst the balloons of
the non-factors by walking"
960 PRINT "under them but if you tr
y to walk under the balloon of a fact
or you will lose a life. You cannot
jump within 3 paces of a chequered
wall end."
970 PRINT " You get 50 points for
each balloon you burst and 100 poi
nts when you have found all the fact
ors and reached the 8 at the top. Y
ou will then get a new number.";

```



```

980 PRINT " You also get a new numb
er after losing a life."
990 PRINT CHR$ 133;"Score 1000-SHEE
T 2 and 2000-SHEET 3"
1000 PRINT "Press 'Z' for LEFT and
'X' for RIGHT"
1010 PRINT "Press ':' for UP and '/'
for DOWN"
1020 PRINT "Press SPACE to jump a ro
lling rock which is trying to de
stroy you and SPACE and 'Z' or 'X' to
gether to jump a gap."
1030 PRINT CHR$ 133;"YOUR SCORE = ";
SX;TAB(19);"HIGH SCORE = ";TP%
1040 PRINT TAB(8);CHR$ 131;"Press 'S
PACE' to begin"
1050 *FX15
1060 Q$=INKEY$(0);IF Q$=" " THEN END
PROC
1070 GOTO 1060
1080 DEFPROCballoon
1090 SOUND 2,1,200,1;SX=SX+50;COLOUR
3:PRINT TAB(6,2);SX
1100 ENDPROC
1110 DEF PROCcheckjump
1120 IF POINT(X%*64+24,(32-Y%)*32-36
)>4 OR POINT(X%*64+24,(32-Y%)*32+28)
=4 OR POINT(X%*64+24,(32-Y%+1)*32+28)
=4 THEN ENDPROC
1130 BAX=0;FOR BX=-3 TO 3
1140 IF POINT((X%+BX)*64+24,(32-Y%)*
32+28)=4 OR POINT((X%+BX)*64+24,(32-Y
%+1)*32+28)=4 THEN BAX=1
1150 NEXT BX
1160 IF BAX=1 THEN ENDPROC
1170 SOUND 2,4,100,1;JX=1;MX=HX;HX=0
:NX=-1;KX=0
1180 ENDPROC
1190 DEF PROCfactor1
1200 IF X%+HX=1 AND Y%=10 THEN FX=7
ELSE IF X%+HX=15 AND Y%=10 THEN FX=3
ELSE IF X%+HX=13 AND Y%=14 THEN FX=9
ELSE IF X%+HX=6 AND Y%=18 THEN FX=4
ELSE IF X%+HX=16 AND Y%=18 THEN FX=1
ELSE IF X%+HX=4 AND Y%=22 THEN FX=6
1210 IF X%+HX=2 AND Y%=26 THEN FX=2
ELSE IF X%+HX=12 AND Y%=26 THEN FX=8
ELSE IF X%+HX=7 AND Y%=30 THEN FX=5
1220 ENDPROC
1230 DEF PROCfactor2
1240 IF X%+HX=10 AND Y%=10 THEN FX=1
ELSE IF X%+HX=17 AND Y%=10 THEN FX=8
ELSE IF X%+HX=1 AND Y%=14 THEN FX=6
ELSE IF X%+HX=10 AND Y%=16 THEN FX=5
ELSE IF X%+HX=14 AND Y%=20 THEN FX=7
ELSE IF X%+HX=5 AND Y%=22 THEN FX=3
1250 IF X%+HX=4 AND Y%=26 THEN FX=9
ELSE IF X%+HX=9 AND Y%=26 THEN FX=4
ELSE IF X%+HX=16 AND Y%=30 THEN FX=2
1260 ENDPROC

```




Specialist Academic & Educational Software
for the BBC microcomputer. Main Northern
Distributor for Viglen Computer supplies.
Acorn Approved.

Resource Facilities

PANEL (8k ROM) This powerful and versatile utility has been designed for the serious assembly language and machine-code programmer. It is the ideal complement to the on-board assembler. All the options of "front-panels" on other educational micros together with many extra features, including: *Register and memory display and modification. *Single-stepping and break-points. *Intelligent block moves. *Disassembler. *Offset calculator.

ROM £20.00

MICROBA - A biology adventure game in which you can wander around the blood, respiratory system, urinary system and alimentary canal of a typical human being.

Cassette £6.00 Disc £7.00

CATB - "Computer Assisted Tables Bingo". A novel and enjoyable way of learning multiplications tables. Class set of 30 bingo cards plus program.

Cassette £8.00 Disc £9.00

REGRESS - "Lines of Best Fit" Regression analysis. A powerful graphical program. Linear/Logarithmic/Inverse scaling. Up to 58 points. Remove the worst point. Print results.

Cassette £8.00 Disc £9.00

ORDER - A graphic display of 8 major sorting routines: Bubble, Decrementing-wave Ripple, Cocktail-shaker, Straight Insertion, Delayed Replacement, Shell-Metzner, Heapsort, Quicksort. Incorporate procedures in your own programs.

Cassette £6.00 Disc £7.00

DISKIT - STAT: a detailed report on the program size and status of each surface of your discs. Clear single screen display. Print option. EZICOPY: simple copying of all programs from catalogue - menu-driven. RENAME extensions.

Disc £5.00

RESMAN - A resources management package. An ideal disc-based filing system for all your resources. Helps you keep track of what you should have.

Disc £9.00

VIEW Printer-driver: Juki 6100, Sanple 2000, Quendata DWP, Epson FX/RX/MX/DX Canon PW, KAGA Taxan KP, KDC FT-5001, Brother HR15.

Disc £5.00

16+ALG - A suite of 10 programs to assist in the teaching and learning of Algebra at O level & 16+. An excellent revision aid. Written by an experienced schoolmaster.

Disc £10.00

SIMPLE SENSORS for the A-D port. Simple unit plus notes on application and programs. Light £12.00 Sound £13.00 Temperature £11.00 Touch £11.00

N.B.

In addition to our software for the BBC microcomputer, we also stock a comprehensive range of Hardware, Media, and Accessories: Viglen disc drives, Monitors, Printers, Discs, ROMs, Expansions, Filing systems.

ALL PRICES EXCLUDE VAT & CARRIAGE. Minimum carriage & packing £1.00. (Please add carriage plus 15% VAT.)

Telephone (0422) 65935

for best prices on quality HARDWARE, FIRMWARE, SOFTWARE, MEDIA & ACCESSORIES. Education Authority Orders welcome.

Resource Facilities. Masters Lane, Halifax HX2 7DX.

FOR CONNOISSEURS OF MODERN LANGUAGE LEARNING

COMMODORE 64 • BBC (32K) • ELECTRON • SPECTRUM (48K)



Dealers, contact
Lightning, Proteus,
Centresoft, Microdeal
or Tiger.

For beginners, O-level and beyond, these best-selling programs are unique and highly successful aids to language learning. Each cassette provides a comprehensive series of vocabulary lessons and a variety of self-paced learning and test modes. All accents and special characters are clearly displayed and different colours denote masculine, feminine and neuter words to reinforce gender learning. The create command enables new lessons in vocabulary or grammar to be entered, edited as required, then saved on tape. By using this simple yet vital feature, homework lists and exam revision can be retained indefinitely and recalled on demand. Two cassettes are available for each language, covering thousands of words; Level A provides 16 lessons in general vocabulary; Level B provides a further 16 lessons including adjectives, adverbs and fully conjugated verb lists.

KOSMOS
S.O.F.T.W.A.R.E.

KOSMOS SOFTWARE LTD 1 Pilgrims Close Harlington, DUNSTABLE, Beds. LU5 6LX Tel: (05255) 3942



The French Mistress Level A @ £8.95 <input type="checkbox"/>	The French Mistress Level B @ £8.95 <input type="checkbox"/>	Computer type.....(Commodore 64/BBC/Electron/Spectrum)
The German Master Level A @ £8.95 <input type="checkbox"/>	The German Master Level B @ £8.95 <input type="checkbox"/>	Mr/Mrs/Miss
The Spanish Tutor Level A @ £8.95 <input type="checkbox"/>	The Spanish Tutor Level B @ £8.95 <input type="checkbox"/>	Address
KOSMOS SOFTWARE LTD		Postcode
1 Pilgrims Close, Harlington, DUNSTABLE, Beds. LU5 6LX.		

IN the last Education Special, we published House, a graphics program by Len Scott.

Ideal for early learners, or those with learning difficulties, it was the sort of simple program that infant teachers had been longing for.

Len has now created Windmill, another delightful program which uses exactly the same format.

Using a controlled vocabulary and straightforward keys, the child is helped to draw a picture featuring trees, fields, a van and a windmill, choosing colours as he or she wishes. A flashing cursor highlights the author's suggested colours.

Teachers using the previous program have commented quite favourably on the way the program not only reinforces elementary word-recognition but also acts as a stimulus to much valuable discussion.

First, a word of warning. The program uses almost all available memory, so be careful not to include any unnecessary spaces, particularly at the ends of lines after using the Copy key to edit.

If you are in any doubt whatsoever, this short routine will assist. In direct mode type in:

```
VDU23,32,255,255,255,255,255,255,255,
255
MODE 6
LIST
```

All spaces will be displayed as white squares, but it will be up to you to decide which are the superfluous ones and remove them.

VARIABLES

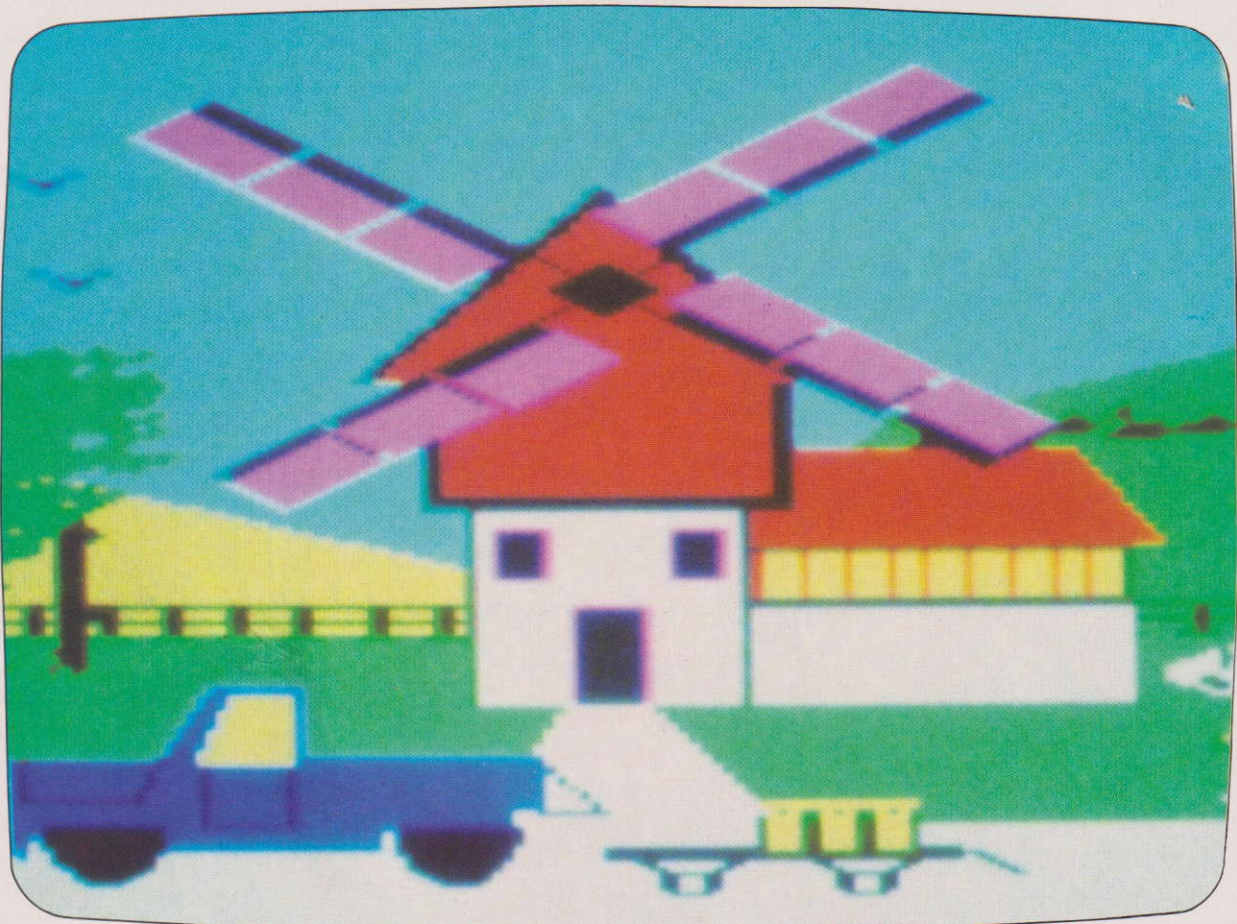
a%-h% Variable selection of either colours or numbers.
e,f,g,i Tab position of pointer.
cr% Radius of circle.
ss%,cd% Start and end degrees of the circle.
st% Steps to jump in circle program.
theta% Theta value.
xc%,yc% x and y axis of centre of circle.
x%,y% x and y axis of outer perimeter of circle.
y,z x and y axis for PLOT.
h PLOT number.
lc Colour of leaves.
ra,rb x and y location of leaves.
u,v Pitch and duration of sound.
xl,yl Start location of leaves.
xn,xm x and y location of hedge.
xws,yws x and y position of segments of windsail.
xo,xp x and y location of birds.
xi,xj x and y location of sacks.

Windmill of the mind

**... a delightful reading program by
LEN SCOTT designed to make lessons
more interesting for infants – and much
easier for their teachers**

PROCEDURES

key Prints "Any key to continue" and waits for key to be pressed.
paint Displays numbers 1 to 7 in their appropriate colours on the text screen.
draw Loop to read the data statements for drawing the different shapes.
point Points to the two suggested colours.
birds Draws the birds.
start Sets up both the graphics and text screens.
windsail Draws the upward-pointing segments of windsails.
windsailb Draws the downward-pointing segments of windsails.
trunk Draws the tree trunks.
leaf Draws the leaves at random within a defined area.
hedge Draws hedge to the left of windmill.
church Draws church in centre of hedge.
sacks Draws sacks on barrow.
sound Loop to read the data statement to play tune.
man Copy of Colourful Colin from *The Micro User*.
cloud Creates white balloon above figure for flashing notes.



```

10 REM(c) The Micro User
20 MODE7:VDU23;8202;0;0;0;0;
30 PRINTTAB(4,2);CHR$(130);CHR$(14
1);"SHALL WE DRAW A PICTURE?":PRINTT
AB(4,3);CHR$(130);CHR$(141);"SHALL WE
DRAW A PICTURE?"
35 PRINTTAB(4,4);CHR$(130);CHR$(14
1);STRING$(25,"=");PRINTTAB(4,5);CHR
$(130);CHR$(141);STRING$(25,"=")
40 PRINTTAB(5,9);CHR$(131);CHR$(14
1);"FOR CHILDREN AGE 4 TO 7":PRINTTAB
(5,10);CHR$(131);CHR$(141);"FOR CHILD
REN AGE 4 TO 7"
50 PRINTTAB(9,13);CHR$(129);CHR$(1
41);"* A WINDMILL *":PRINTTAB(9,14);C
HR$(129);CHR$(141);"* A WINDMILL *"
60 PRINTTAB(8,20)"written by Len
Scott":PRINTTAB(7,24)"Press any key
to go on":A=GET
100 MODE2:VDU23;8202;0;0;0;0;
110 PROCstart:PROCpaint:PROCpoint(9
,1,15,1)
120 COLOUR3:PRINTTAB(8,3)"let us pa
int the sky":PRINTTAB(8,6)"what colou
r 4 or 6?":*FX15,0
130 aZ=GET:CLS:GCOL0,aZ+120:CL6:VDU
4:CLS
140 PROCpaint:COLOUR3:PRINTTAB(8,2)
"now paint the fields":PRINTTAB(8,4)"

```

```

what colour is corn?":*FX15,0
150 bZ=GET-48:IFbZ=3THEN60TO160ELSE
COLOUR9:PRINTTAB(4,6)"try again!":FOR
DELAY=1 TO 2000:NEXT:PRINTTAB(4,6)SP
C(11):60TO150
160 VDU5:6COL0,3:PROCcircle(0,-400,
1150,110,10,2)
170 6COL0,2:PROCcircle(1300,-400,11
75,180,5,90)
180 MOVE0,500:MOVE1300,500:PLOT85,0
,0:PLOT85,1300,0
190 6COL0,0:PROCdraw
200 FORFX=0TO1300STEP50:MOVEFX,525:
VDU108:NEXT
210 VDU4:CLS
220 COLOUR8:PRINTTAB(6,6)"2":COLOUR
7:PRINTTAB(9,6)"or":COLOUR15:PRINTTAB
(13,6)"3":COLOUR3:PRINTTAB(2,3)"how m
any trees?":*FX15,0
230 REPEAT:cZ=GET-48:UNTIL cZ=2 OR
cZ=3
240 IFcZ=2THEN60TO270ELSE60TO250
250 CLS:VDU5:MOVE175,625:PROCtrunk
260 FOReZ=0TO40:PROClcaf(2,100,620,
115,135):NEXTeZ
270 VDU5:MOVE5,600:PROCtrunk
280 MOVE100,585:PROCtrunk
290 FOReZ=0TO100:PROClcaf(2,-40,575
,150,200):NEXTeZ

```

```

300 MOVE1070,705:PROCchurch
310 PROCchedge(750,660)
320 PROCchedge(860,680)
330 PROCchedge(960,680)
340 PROCchedge(1070,690)
350 PROCchedge(1180,700)
360 PROCchedge(1180,690)
370 VDU4:CLS:PROCpaint
380 PROCpoint(6,1,18,1)
390 COLOUR3:PRINTTAB(2,2)"now colou
r the":PRINTTAB(2,4)"windmill walls":
PRINTTAB(5,6)"3 or 7?":*FX15,0
400 dZ=GET:6COL0,dZ-48:VDU5
410 PROCdraw:6COL0,0:PROCdraw:6COL0
,3:PROCdraw:6COL0,1
420 FORbx=710TO950STEP30:MOVEbx,535
:DRAMBx,565:NEXT
430 PROCdraw:6COL0,0:PROCdraw:6COL0
,1:PROCdraw:6COL0,0:MOVE540,800
440 VDU23,225,0,64,32,17,11,7,15,31
450 VDU23,226,0,2,4,136,208,224,240
,248
460 VDU23,227,31,15,7,11,17,32,64,0
470 VDU23,228,248,240,224,208,136,4
,2,0
480 VDU225,226,8,8,10,227,228:PROCd
raw

```

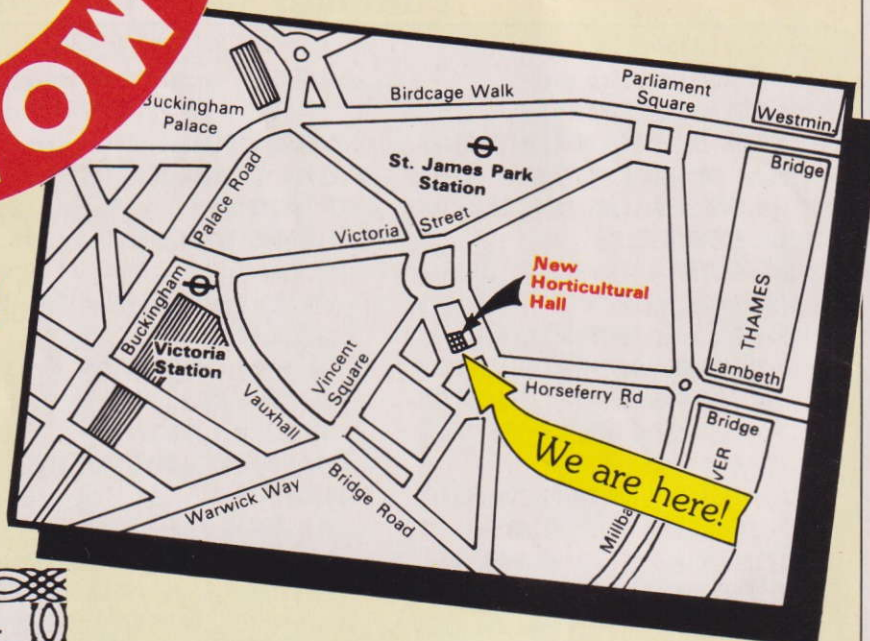

See for yourself the range of educational software planned for 1986 – at the . . .

The new generation of advanced educational programs that schools throughout Britain will be incorporating into their curriculum in 1986 will be given their first public showing at the November Electron & BBC Micro User Show.

All the leading suppliers will be there – and school parties will be welcome to see, to try and to buy.

ELECTRON & BBC MICRO USER SHOW

**Thursday to Sunday
November 14 to 17
New Horticultural Hall
Westminster,
London SW1**



By handing in this voucher at the door you save 50p off the normal admission price of £3 (adults) and £2 (children).

(Valid for a maximum of 4 people)

10am-6pm, Thursday, 14 November
10am-6pm, Friday, 15 November
10am-6pm, Saturday, 16 November
10am-4pm, Sunday, 17 November

**New Horticultural Hall
Westminster, London SW1**

Number attending: ☐ 1 ☐ 2 ☐ 3 ☐ 4

Reduced prices for School/College Groups

Entry only £1 per student if bookings are made in advance. Send your cheque (made payable to Database Publications) and SAE to:

Electron & BBC Micro User Show
68 Chester Road, Hazel Grove,
Stockport SK7 5NY.

Tel: 061-480 7863

Valid for a minimum of 10 people.

SAVE MONEY with our Special Travel Offer . . .

Discounted rail travel offer for visitors to the Electron & BBC Micro User Show. For full details write to:

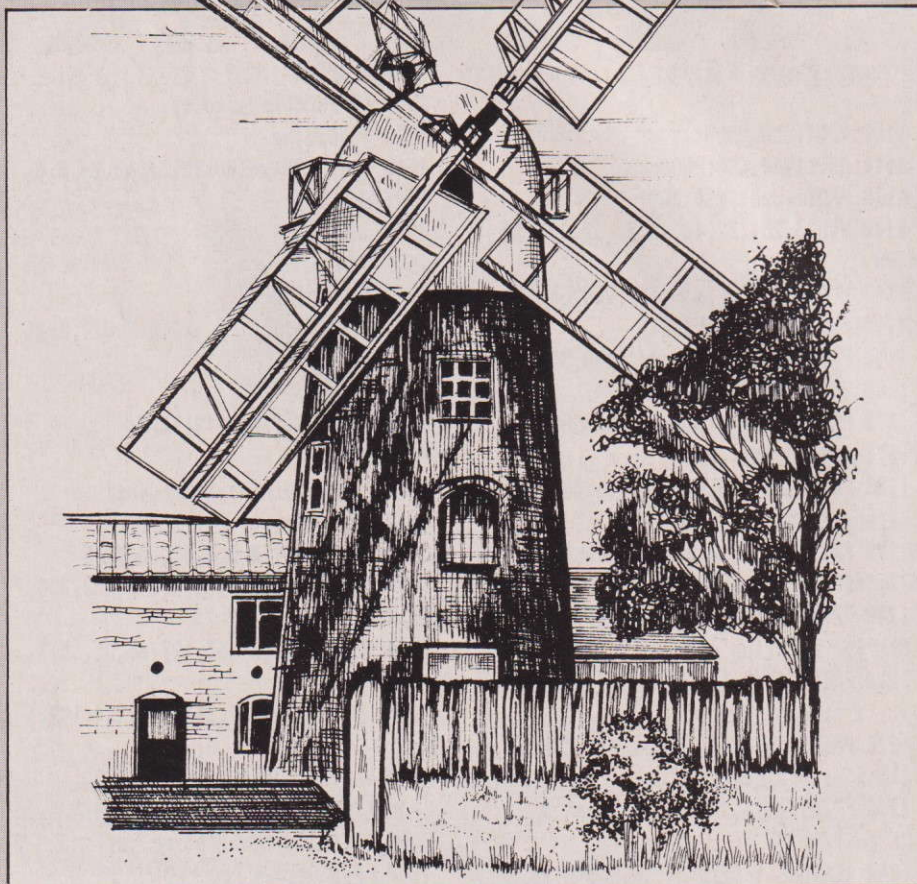
Travel Offer, P.O. Box 1, St. Albans
AL1 4ED with SAE or Telephone:
St. Albans 34475 quoting: The
Electron & BBC Micro User Show.

From Page 15

```

490 VDU4:CLS:PROCpaint
500 PROCpoint(12,1,18,1)
510 COLOUR3:PRINTTAB(2,2)"now colour
r the":PRINTTAB(5,4)"windsails":PRINT
TAB(5,6)"5 or 7 ?":*FX15,0
520 e%=GET:VDU5:GCOL0,0
530 PROCwindsail(600,860)
540 PROCwindsail(600,900)
550 PROCwindsail(760,940)
560 PROCwindsailb(640,775)
570 PROCwindsailb(725,735)
580 PROCwindsailb(800,695)
590 PROCwindsail(470,750)
600 PROCwindsail(385,710)
610 PROCwindsail(385,670)
620 PROCwindsailb(430,840)
630 PROCwindsailb(345,880)
640 PROCwindsailb(265,920)
650 GCOL0,e%-48
660 PROCwindsail(591,870)
670 PROCwindsail(670,910)
680 PROCwindsail(750,950)
690 PROCwindsail(480,740)
700 PROCwindsail(395,700)
710 PROCwindsail(315,660)
720 PROCwindsailb(650,785)
730 PROCwindsailb(735,745)
740 PROCwindsailb(810,705)
750 PROCwindsailb(415,835)
760 PROCwindsailb(330,875)
770 PROCwindsailb(255,915)
780 GCOL0,5:PROCdraw:GCOL0,0:PROCdr
aw:GCOL0,7:PROCdraw
790 PROCbirds(25,900)
800 PROCbirds(50,825)
810 PROCpoint(0,1,6,1):COLOUR3:PRIN
TTAB(0,3)"now colour the sun":PRINTTA
B(5,5)" 1 or 3 ?":*FX15,0
820 g%=GET:VDU5:GCOL0,g%-48
830 PROCcircle(1100,950,35,360,1,2)
840 PROCpoint(0,1,9,1):COLOUR3:PRIN
TTAB(0,3)"now colour the van":PRINTTA
B(5,5)" 1 or 4 ?":*FX15,0
850 f%=GET:VDU5
860 GCOL0,f%-48:PROCdraw:GCOL0,3:PR
OCdraw:GCOL0,0:PROCdraw
870 PROCcircle(225,350,45,360,1,180
)
880 PROCcircle(485,350,45,360,1,180
)
890 MOVE630,320:VDU111,9,111:MOVE66
0,335:VDU8,92,47,92,47,92
900 MOVE595,325:DRAW855,325:PLOT85,
595,335:PLOT85,855,335
910 VDU4:CLS
920 COLOUR8:PRINTTAB(6,6)"3":COLOUR

```



```

7:PRINTTAB(9,6)"or":COLOUR15:PRINTTAB
(13,6)"5":COLOUR3:PRINTTAB(3,3)"how m
any sacks ?":*FX15,0
930 REPEAT:h%=GET-48:UNTIL h%=3 OR
h%=5
940 IFh%=3THENGOTO990ELSEGOTO970
950 VDU8,92,47,92,47,92
960 MOVE595,325:DRAW855,325:PLOT85,
595,335:PLOT85,855,335
970 VDU5:PROCcsack(700,410)
980 PROCcsack(740,410)
990 VDU5:PROCcsack(680,370)
1000 PROCcsack(720,370)
1010 PROCcsack(760,370)
1020 DATA4,0,515,5,1300,515,4,0,525,
5,1300,525,0,0,0,4,700,600,4,500,600,
85,700,450,85,500,450,4,980,525,4,700
,525,85,980,450,85,700,450,0,0,0,4,98
0,575,4,700,575,85,980,525,85,700,525
,0,0,0
1030 DATA4,970,565,4,710,565,85,970,
535,85,710,535,0,0,0,4,940,645,4,700,
645,85,940,575,85,700,575,4,940,575,4
,1000,575,85,940,645,0,0,0,4,730,700,
4,470,700,85,730,600,85,470,600,4,420
,700,4,770,700,85,595,850,0,0,0
1040 DATA4,715,700,4,485,700,85,715,
615,85,485,615,4,435,700,4,755,700,85
,595,835,0,0,0,4,700,600,5,700,450,4,
500,600,5,500,450,0,0,0
1050 DATA4,575,525,4,625,525,85,575,

```

```

450,85,625,450,4,515,585,4,555,585,85
,515,545,85,555,545,4,640,585,4,685,5
85,85,640,545,85,685,545,0,0,0,4,520,
580,4,550,580,85,520,550,85,550,550,4
,650,580,4,675,580,85,650,550,85,675,
550,4,580,520
1060 DATA4,620,520,85,580,450,85,620
,450,0,0,0,4,575,445,4,625,445,85,575
,265,85,625,265,4,625,445,85,800,265,
4,575,445,4,575,265,85,450,265,4,0,35
0,4,1321,350,85,0,265,85,1300,265,0,0
,0
1070 DATA4,150,400,4,275,400,85,150,
350,85,275,350,5,150,400,4,150,350,85
,125,350,4,175,350,85,162,325,4,310,4
60,4,370,460,85,310,365,85,370,365,4,
310,460,4,310,365,85,245,365,4,370,40
5,4,550,405,85,370,365,85,550,365,4,2
60,360,4,520
1080 DATA360,85,260,350,85,520,350,4
,270,350,4,430,350,85,270,335,85,430,
335,0,0,0,4,320,455,4,360,455,85,320,
400,85,360,400,4,320,455,4,320,400,85
,300,400,0,0,0,4,300,350,5,300,395,4,
360,350,5,360,395,4,300,350,5,360,350
,4,160,400
1090 DATA5,160,370,5,250,370,5,270,4
00,4,550,360,5,595,360,5,550,400,0,0,
0

```


From Page 17

```

1100 VDU4:CLS:VDU5
1110 MOVE1040,430:PROCman
1120 PROCcloud(1000,500)
1130 VDU23,251,32,48,40,32,224,22
4,96
1140 GCOL0,8:MOVE1038,486:VDU251:6CD
L0,15:MOVE1050,490:VDU251
1150 PROCsound:PROCcloud(1000,500)
1160 VDU4:CLS:PROCkey:MODE7
1170 PRINTTAB(7,4);CHR$(131);CHR$(14
1);"thank you children"
1180 PRINTTAB(7,5);CHR$(131);CHR$(14
1);"thank you children"
1190 PRINTTAB(8,10);CHR$(131);CHR$(1
41);"for painting me"
1200 PRINTTAB(8,11);CHR$(131);CHR$(1
41);"for painting me"
1210 PRINTTAB(8,16);CHR$(131);CHR$(1
41);"a nice windmill"
1220 PRINTTAB(8,17);CHR$(131);CHR$(1
41);"a nice windmill"
1230 PRINTTAB(4,24)"do you want anot
her go Y/N ?";
1240 RESTORE:IFGET$="Y"THENGOTO100:P
ROCKey
1250 END
1260 DEFPROCkey
1270 PRINTTAB(7,24)"Any key to conti
nue";:dummy$=GET$:CLS
1280 ENDPROC
1290 DEFPROCpaint
1300 VDU4:CLS
1310 COLOUR1:PRINTTAB(0,0)"1":COLOUR
2:PRINTTAB(3,0)"2":COLOUR3:PRINTTAB(6
,0)"3":COLOUR4:PRINTTAB(9,0)"4":COLOU
R5:PRINTTAB(12,0)"5":COLOUR6:PRINTTAB
(15,0)"6":COLOUR7:PRINTTAB(18,0)"7"
1320 ENDPROC
1330 DEFPROCdraw
1340 REPEAT:READk,y,z:PLOTk,y,z:UNTI
Lz=0
1350 ENDPROC
1360 DEFPROCpoint(e,f,g,i)
1370 PROCpaint
1380 COLOUR0:PRINTTAB(e,f)"^":COLOUR
15:PRINTTAB(g,i)"^"
1390 ENDPROC
1400 DEFPROCcircle(xc%,yc%,cr%,cd%,s
t%,se%)
1410 MOVExc%+cr%,yc%
1420 FORtheta%=ss%TOcd%STEPst%*5
1430 x%=cr%*COS(RAD(theta%))
1440 y%=cr%*SIN(RAD(theta%))
1450 MOVExc%,yc%
1460 PLOT85,xc%+x%,yc%+y%
1470 NEXT

```

```

1480 ENDPROC
1490 DEFPROCbirds(xo,yp)
1500 MOVExo,yp
1510 VDU23,224,0,0,195,36,60,24,0,0
1520 GCOL0,0
1530 VDU224,9,10,224
1540 ENDPROC
1550 DEFPROCstart
1560 VDU28,0,31,19,24:CLS:VDU24,0;26
5;1279;1023;
1570 CLS
1580 VDU24,0;265;1279;1023;
1590 ENDPROC
1600 DEFPROCwindsail(xws,yws)
1610 MOVExws,yws
1620 VDU23,229,0,0,0,0,1,3,7,15
1630 VDU23,230,0,32,112,248,252,254,
255,254
1640 VDU23,231,31,63,127,63,31,15,7,
2
1650 VDU23,232,252,248,240,224,192,1
28,0,0
1660 VDU229,230,8,8,10,231,232
1670 ENDPROC
1680 DEFPROCwindsailb(xws,yws)
1690 MOVExws,yws
1700 VDU23,233,0,4,14,31,63,127,255,
127
1710 VDU23,234,0,0,0,0,128,192,224,2
40
1720 VDU23,235,63,31,15,7,3,1,0,0
1730 VDU23,236,248,252,254,252,248,2
40,224,64
1740 VDU233,234,8,8,10,235,236
1750 ENDPROC
1760 DEFPROCtrunk
1770 GCOL0,0
1780 VDU23,246,192,99,98,54,55,55,30
,28
1790 VDU23,247,28,28,28,28,28,28,28,
28
1800 VDU23,248,8,16,48,96,192,128,0,
0
1810 VDU23,249,28,28,60,60,28,60,100
,171
1820 VDU23,250,28,28,30,31,30,28,28,
28
1830 VDU246,28,8,8,10,247,8,10,250
1840 VDU8,10,247,8,10,250,0,10,249
1850 ENDPROC
1860 DEFPROCleaf(lc,xl,yl,ra,rb)
1870 VDU23,225,0,32,115,127,62,24,16
,0
1880 Y%=yl+RND(ra):X%=xl+RND(rb)
1890 MOVEX%,Y%:GCOL0,lc:VDU225
1900 ENDPROC
1910 DEFPROC hedge(xn,xm)
1920 MOVExn,xm:GCOL0,0

```

```

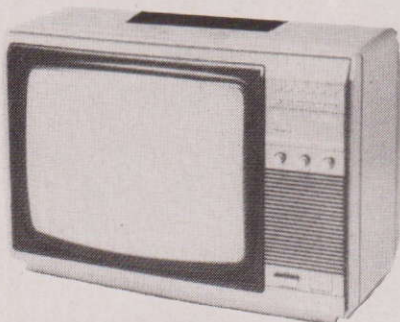
1930 VDU23,242,0,0,0,4,61,127,255,25
2
1940 VDU23,243,1,3,35,119,252,248,24
0,0
1950 VDU242,243
1960 ENDPROC
1970 DEFPROCchurch
1980 GCOL0,7
1990 VDU23,244,4,14,14,31,31,31,31,3
1
2000 VDU23,245,0,0,0,0,0,240,248,252
2010 VDU244:GCOL0,1:VDU245:GCOL0,0:M
OVE1078,705:VDU94
2020 ENDPROC
2030 DEFPROC sack(xi,xj)
2040 MOVExi,xj
2050 VDU23,237,0,0,0,17,17,17,17,17
2060 VDU23,238,31,31,10,14,14,14,14,
14
2070 VDU23,239,17,17,31,0,0,0,0,0
2080 VDU23,240,14,14,0,0,0,0,0,0
2090 GCOL0,0:VDU237,8,10,239
2100 GCOL0,3:VDU8,11,238,8,10,240
2110 ENDPROC
2120 DEFPROCsound
2130 REPEAT:READu,v:SOUND1,-9,u,v:UN
TILu=0
2140 DATA69,0,81,16,89,8,97,16,105,4
,97,4,89,16,77,8,61,16,69,4,77,4,81,1
6,69,8,69,12,65,4,69,8,77,16,65,8,49,
16,69,0,81,16,89,8,97,16,105,4,97,4,8
9,16,77,8,61,16,69,4,77,4,81,8,77,8,6
9,8,65,8,57,8,65,8,69,8,49,4,69,4,81,
4,97,4,117,16,0,0
2150 ENDPROC
2160 DEFPROCman
2170 VDU23,227,28,34,34,108,0,0,0,0
2180 VDU23,228,0,28,28,136,128,227,1
,3
2190 VDU23,229,0,0,0,0,62,28,28,28
2200 VDU23,230,28,28,62,62,0,0,0,0
2210 VDU23,231,0,0,0,0,20,20,20,0
2220 VDU23,232,0,0,0,0,0,0,0,119
2230 VDU227:GCOL0,0:VDU8,228:GCOL0,7
:VDU8,229:GCOL0,1:VDU8,10,230:GCOL0,3
:VDU8,231:GCOL0,7:VDU8,232:GCOL0,0
2240 ENDPROC
2250 DEFPROCcloud(xl,xk)
2260 MOVExl,xk:GCOL0,7
2270 VDU23,233,0,0,0,1,7,15,63,127
2280 VDU23,234,0,0,96,248,252,254,25
5,255
2290 VDU23,235,127,127,63,15,7,0,0,0
2300 VDU23,236,255,255,252,252,248,9
6,0,0
2310 VDU233,234,8,8,10,235,236
2320 ENDPROC

```


CARE ELECTRONICS

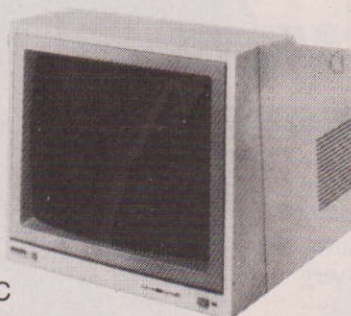
UNIT 14, PEERGLOW INDUSTRIAL ESTATE, OLD'S APPROACH, TOLPITS LANE, WATFORD, HERTS.

TELEPHONE: 0923 777155



**PHILIPS COLOUR
TV/MONITORS
80 CHARACTERS
PER LINE**

★★ FREE
LEADS SUPPLIED
FOR USE WITH BBC



★★ SUPER VALUE
MONO
ANTI-GLARE SCREEN
HIGH RESOLUTION
GREEN MODEL BM7502

@ 81.65a Inc VAT

★★ SUPER VALUE
AMBER MODEL MM06
@ 80.50a Inc VAT

MEDIUM RESOLUTION

14" MODEL 14CT2006 £230.00 Inc. VAT

We can supply up to 26" TV/Monitors –
please ring for details.

COMPUTER/PRINTER SELECTOR 6 to 1

*26 pin as BBC *Colour finished to match BBC
*1 micro to 6 printers *6 micros to 1 printer

@ 75.90d inc VAT

Dual Power Supply Units for Disc Drives

- ★ Colour finished to match BBC Micro.
- ★ Comes complete with moulded mains lead
- ★ 6 pin BBC outlet or Dual power leads for disc drives. (Please specify when ordering).

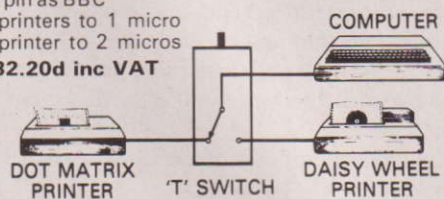
High Rated @ 33.35b Inc VAT

COMPONENTS! WIDE RANGE AVAILABLE

Acorn Disc Interface @ 81.65c Inc VAT
Eprom 2764 250ns @ 3.68c Inc VAT
Eprom 27128 250ns @ 6.90 Inc VAT

COMPUTER/PRINTER SELECTOR 'T' SWITCHER

* 26 pin as BBC
* 2 printers to 1 micro
* 1 printer to 2 micros
@ 32.20d inc VAT



READYMADE LEADS:

BBC to single disc @ 3.45c Inc VAT
BBC to Dual disc @ 4.60c Inc VAT
Disc drive single @ 6.67c Inc VAT
Disc drive dual @ 10.12c Inc VAT
Printer @ 7.82c Inc VAT
Parallel 26 pin @ 8.74c Inc VAT

HAVE YOUR 14" & 16" PHILIPS, PYE or FERGUSON TX COLOUR TV CONVERTED TO A TV/MONITOR RGB CONVERSION KIT

@ £44.85d

- ★ Resolution better than 585 x 450 pixels.
- ★ Image clarity comparable to leading monitors.
- ★ Includes RGB lead for connecting with BBC.

READYMADE LEADS:

RGB 6 pin DIN @ 2.99c Inc
RGB 6 pin to 7 pin DIN @ 4.14c Inc
RGB 6 pin to SCART @ 7.82c Inc

REMOTE AND STANDARD TV'S MODIFIED IN HOUSE

@ £58.65a

THE CARE LOW PROFILE ROM MODULE SYSTEM



Don't run the risk of damaging your
ROMS/EPROMS or your BBC Micro.
Whenever you want to remove ROMS/
EPROMS from your BBC. Fit our Low Profile
Rom Module System and save your collection

of ROMS/EPROMS in our specially designed
cartridges. The CARE LOW PROFILE
MODULE SYSTEM has been designed not to
take any power from your BBC Micro and
allows you unlimited ROMS/EPROMS.

★ Unlike other systems:

- ★ Won't get in the way when using the Keyboard.
- ★ Stops damage caused by shortcircuit due to metal objects e.g. paperclips falling inside.
- ★ Prevents damage caused by dust.
- ★ **Edge Card Socket** which takes all the wear, Gold Plated for extended life.

★ FOR THE AMATEUR AND PROFESSIONAL USER ★ MANUFACTURED ENTIRELY IN THE UK.

★ Each System Contains:

- ★ 1 Low Profile Assy. ★ 1 ROM Cartridge.
- ★ 1 Library Rack 6 way. ★ 5 Blank Labels.

★ Prices including VAT.

- ★ Complete Module System @ £14.95d
- ★ Spare Cartridge @ £2.76c
- ★ Library Rack to store 6 Cartridges @ £1.84c
- ★ 5 Cartridges @ £12.95c ★ 10 Cartridges @ £23.00c

HOW TO ORDER:

By Post. Enclose your Cheque/P.O. made payable to **CARE Electronics**.
Or use your ACCESS. Allow 7 days for delivery. Please add carriage.

OPEN
9am-5pm MON-THUR
9am-4pm FRI/SAT

a) @ 8.00 c) @ 1.00
b) @ 5.00 d) @ 2.00

TEL: 0923 777155



Open fire on a jelly fish to learn all about angles with JOHN SHEARD's ...

THIS two-player game develops and practises, not the skills of fishing, but the ability to estimate angles up to 90 degrees.

It has been developed with children aged 9 to 11, but has also proved very popular with older children and adults – who often find it harder and more challenging than they expect!

At the start of the program the players are asked to choose a target size (1 small, 2 medium, 3 large) and a winning score (from 2 to 10).

The program then moves on to the main screen layout with the target positioned at random each time.

The players take it in turn to estimate the angle of the target from the base line as viewed from their corner – this means the angle will usually be different for each player.

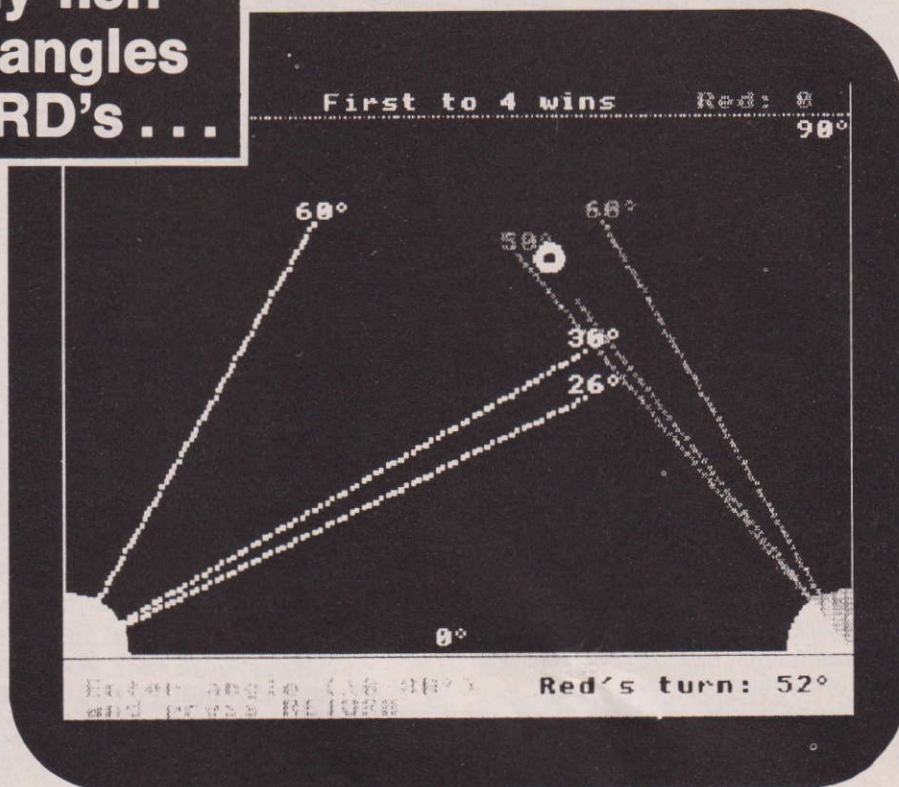
When an estimate has been entered the angle is shown graphically in that

player's corner (to reinforce the idea of angle as rotation) before a dotted line emerges and, hopefully, heads for the target.

Lines that miss the target are not deleted but are labelled and retained on the screen. This can, of course, lead to a

messy display if more than a few tries are needed for a target, but they do help the players to remember what angles they've already entered and to use them for their next estimate.

If two pairs of children are playing against each other a lot of useful



PROCEDURES

circle Used for drawing and deleting the targets, drawing the corner quadrants and drawing the angle displays. It has seven parameters:

CX%,CY% – X and Y coordinates.
CR% – radius of circle.
angle – 360 for circle, 90 for quadrant etc.
col% – colour of circle (the background colour is used for deletion).
step – defines STEP to control speed of circle-drawing.
D% – determines clockwise or anti-clockwise drawing.

setup Chooses level of difficulty and how long the game should be.

windows Defines graphics and text windows using the VDU 28 and VDU 24 statements.

axes Draws and labels the axes and draws the two corner quadrants.

target Chooses a random position for each target (within certain limitations to ensure that it is not too near a bottom corner or the axes) and prints the target using

PROCcircle.

angle Asks the two players alternately to estimate the angle of the target from the base line and enter it.

line Chosen angle is then drawn on top of the player's quadrant and a dotted line is drawn at that angle, each point being tested for a hit and to see whether the line has passed the target either horizontally or vertically.

plot Enlarges each point of the line for easier visibility.

miss Prints the line's angle and makes an appropriate noise.

hit Deletes the target and adds one point to the score of the player who hit it.

swap, red, green Set up the variables, mostly in connection with starting positions for each player's turn.

displayscore Prints current score at top of screen, together with a reminder of the winning score.

endgame Congratulates the winner, gives the joint accuracy rating and asks if another game is wanted.

ANGLER

discussion, not to mention the occasional argument, can ensue at this stage.

One point is scored for each hit, and the current score is always on display at the top of the screen. The first player to reach the pre-set winning score is the winner.

Another game of the same length and at the same level of difficulty can be started by pressing the space bar. Pressing Escape, which is disabled at line 40 during the game, returns the program to the set-up stage.

You would be well advised to REM line 40 until such time as your program is debugged.

At the end of the game, along with the final score, the players are given a joint accuracy rating – a percentage figure derived from the total number of shots both have needed to hit the targets.

This feature means that pairs of

children can compete against other pairs as well as against each other. It also means that the game can be played solo, with the player taking both parts and trying to improve the accuracy rating.

Only very modest sound effects are used. However, if even these are too much for a quiet classroom, just add the line:

```
55 *FX 210,1
```

and the program, if not the children, will be silent.

Finally, a warning! The fact that the screen is a rectangle rather than a square makes some angles deceptive to estimate.

I don't think anyone has played this game without at some stage questioning my trigonometrical ability, but if you don't believe what you see on the screen take out a protractor and check it for yourself!

VARIABLES

Most of the variables are self-explanatory, the exceptions being those that are used to work out the various mathematical calculations, for example:

TR% Target radius.

TX%, TY% X and Y coordinates of target.

H%, V% X and Y coordinates of points that make up each line of fire.

Having lowercase variables improves readability, but does make typing the program in marginally more difficult. A useful tip here, if you haven't come across it before, is to press Shift and Caps Lock together before you start typing. Thereafter, if you press Shift and type, the letters will be printed in lowercase.

```

10 REM "ANGLER"
20 REM by John Sheard
30 REM (c) The Micro User
40 ON ERROR RUN
50 MODE1
60 *FX4,1
70 *FX11,0
80 VDU19,0,4;0;
90 VDU19,3,2;0;
100 VDU 23,224,48,72,72,48,0,0,0,0
110 hit=FALSE
120 GCOLOR,128
130 PROCred
140 PROCsetup
150 random=RND(-TIME)
160 REPEAT
170 *FX200,1
180 greenscore=0:redscore=0:targets
=0:shots=0
190 PROCwindows
200 REPEAT
210 PROCaxes
220 PROCdisplayscore

230 PROCtarget
240 hit=FALSE
250 REPEAT
260 PROCswap
270 PROCangle
280 PROCline
290 UNTIL hit
300 PROC hit
310 UNTIL redscore=winscore OR gre
nscore=winscore
320 PROCendgame
330 UNTIL FALSE
340 DEFPROCsetup
350 VDU23,1,0;0;0;0;
360 CLS:PROCaxes
370 VDU5:MOVE450,1000:PRINT"A N G L
E R"
380 FOR level=1 TO 3
390 PROCcircle(level*300,752,5+(lev
el*6),360,2,10,1)
400 MOVE level*300-16,700:PRINT,lev
el
410 NEXT:VDU4:COLOUR 2

420 PRINTTAB(2,12)"Which size targe
t do you want?"
430 level=FNinput(33,12;1,3)
440 TR%=5+(level*6)
450 PRINTTAB(2,14)"What do you want
the winning score";TAB(6,16)"to be (
2 - 10)?"
460 winscore=FNinput(22,16,2,10)
470 PRINTTAB(6,19)"Press the SPACE
BAR to start"
480 PRINTTAB(2,22)"Press ESCAPE to
change these numbers"
490 REPEAT UNTIL GET=32
500 ENDPROC
510 DEFPROCaxes
520 VDU5,12,16
530 FOR corners=1 TO 2
540 PROCcircle(SX%,104,RX%,90,lineco
1,8*DX,DZ)
550 PROCswap:NEXT
560 GCOLOR,2

```


From Page 21

```

570 MOVE 4,960:PRINT"90";CHR$224
580 MOVE 1190,960:PRINT"90";CHR$224
590 MOVE 600,140:PRINT"0";CHR$224
600 MOVE 0,1023:DRAW 0,100
610 DRAW 1279,100:DRAW 1279,1023
620 MOVE 1279,975:PLOT 21,0,975
630 ENDPROC
640 DEFPROCwindows
650 VDU 28,0,31,39,29
660 VDU 24,0;100;1279;1023
670 COLOUR130
680 ENDPROC
690 DEFPROCtarget
700 targets=targets+1
710 VDU23,1,0;0;0;0;
720 REPEAT
730 TXZ=RND(1000)+140
740 TYZ=RND(624)+260
750 UNTIL SQR(TXZ^2+(TYZ-100)^2)>40
0 AND SQR((1279-TXZ)^2+(TYZ-100)^2)>4
00
760 start=1
770 PROCcircle(TXZ,TYZ,TRZ,360,2,10
,1)
780 PROCcircle(TXZ,TYZ,TRZ/3,360,0,
10,1)
790 ENDPROC
800 DEFPROCcircle(CXZ,CYZ,CRZ,angle
,colZ,step,DZ)
810 GCOL 0,colZ
820 MOVE CXZ,CYZ
830 DRAW CXZ+CRZ,CYZ
840 FOR ZZ=start TO start+(angle*DZ
) STEP step
850 IF hit THEN SOUND 0,-15,5,2
860 PLOT 85,CXZ+((CRZ*COS(RAD(ZZ)))*
DZ),CYZ+((CRZ*SIN(RAD(ZZ)))*DZ))
870 MOVE CXZ,CYZ
880 NEXT
890 ENDPROC
900 ;-----
910 DEFPROCswap
920 IF go$="Red" THEN PROCgreen ELS
E PROCred
930 ENDPROC
940 DEFPROCred
950 go$="Red"
960 linecol=1
970 IXZ=24;HXZ=1
980 DX=-1
990 SXZ=1275
1000 AIZ=-100
1010 RXZ=-100
1020 start=100

```

```

1030 ENDPROC
1040 DEFPROCgreen
1050 go$="Green"
1060 linecol=3
1070 IXZ=1;HXZ=20
1080 DX=1
1090 SXZ=4
1100 AIZ=50
1110 RXZ=100
1120 start=1
1130 ENDPROC
1140 DEFPROCangle
1150 shots=shots+1
1160 *FX21,0
1170 COLOUR 130:CLS
1180 COLOUR1:PRINTTAB(HXZ,1)"Enter a
ngle (10-80";CHR$224;");TAB(HXZ,2)"a
nd press RETURN";
1190 COLOUR 0:PRINTTAB(IXZ,1)go$;"'s
turn: ";CHR$224
1200 VDU7
1210 angleZ=FNinput(IXZ+LEN(go$)+9,1
,10,00)
1220 ENDPROC
1230 DEFPROCline
1240 PROCcircle(SXZ,104,RXZ,angleZ,li
necol+2,DZ,DZ)
1250 GCOL 0,linecol
1260 MOVE SXZ,100
1270 HZ=SXZ
1280 REPEAT
1290 HZ=HZ+(((90-angleZ)/6)*DZ)
1300 VZ=104+(ABS(SXZ-HZ)*TAN(RAD(ang
leZ)))
1310 IF POINT(HZ,VZ)=2 THEN hit=TRUE
ELSE PROCplot
1320 UNTIL hit OR ABS(SXZ-HZ)>ABS(SX
Z-TXZ)+50 OR VZ>TYZ+50
1330 IF NOT hit THEN PROCmiss
1340 ENDPROC
1350 DEFPROCplot
1360 PLOT 69,HZ,VZ
1370 PLOT 69,HZ+4,VZ
1380 PLOT 69,HZ,VZ+4
1390 PLOT 69,HZ+4,VZ+4
1400 SOUND 1,-(ABS(SXZ-HZ)+VZ)/120,1
49,1
1410 ENDPROC
1420 DEFPROCmiss
1430 VDU 5
1440 PLOT 0,-32,32:PRINT;angleZ;CHR$
224
1450 FOR SZ=149 TO 5 STEP-1
1460 SOUND 1,-10,SZ,0.6
1470 NEXT
1480 PROCcircle(SXZ,104,RXZ,90,lineco
l,8*DZ,DZ)

```

```

1490 VDU 4
1500 ENDPROC
1510 DEFPROCchit
1520 PROCcircle(TXZ,TYZ,TRZ,360,0,10
,1)
1530 IF go$="Green" THEN greenscore=
greenscore+1 ELSE redscore=redscore+1
1540 hit=FALSE
1550 ENDPROC
1560 DEFPROCdisplayscore
1570 VDU 5
1580 MOVE 50,1000:GCOL 0,3:PRINT"Gre
en: ";greenscore
1590 MOVE 420,1000:GCOL 0,2:PRINT"Fi
rst to ";winscore;" wins"
1600 MOVE 1030,1000:GCOL 0,1:PRINT"R
ed: ";redscore
1610 VDU 4
1620 ENDPROC
1630 DEFPROCendgame
1640 *FX200,0
1650 CLS:CLS:VDU 26:COLOUR 1
1660 PRINTTAB(5,3)"Final score: GRE
EN - ";greenscore;" RED - ";redscore
1670 IF redscore=winscore THEN winne
r$="R E D" ELSE winner$="G R E E N"
1680 PRINTTAB(7,6)"W E L L D O N E
";winner$;" !"
1690 COLOUR 0:PRINTTAB(3,11)"Your jo
int accuracy rating was ";COLOUR 1:P
RINT;INT(SQR(targets/shots*100)*10);"
%"
1700 PRINTTAB(2,16)"Press the SPACE
BAR for another game";TAB(15,18)"at
level ";level;TAB(15,20)"first to ";w
inscore
1710 COLOUR 0:PRINTTAB(3,23)"Press E
SCAPE to change these numbers"
1720 REPEAT UNTIL GET=32
1730 ENDPROC
1740 DEFFNinput(X,Y,A,B)
1750 VDU23,1,1;0;0;0;R=TRUE
1760 REPEAT:input$="";IF NOT R VDU7
1770 REPEAT:REPEAT
1780 PRINTTAB(X,Y)SPC2;TAB(X,Y)input
$;
1790 K=GET:UNTIL K=127 OR K=13 OR (K
)>=47 AND K<=58)
1800 IF K=127 THEN input$=LEFT$(inpu
t$,LEN(input$)-1) ELSE input$=input$+CH
R$K
1810 R=FALSE:UNTIL K=13 OR LEN(input
$)>LEN(STR$(B))
1820 UNTIL VAL(input$)>=A AND VAL(in
put$)<=B
1830 VDU23,1,0;0;0;0;
1840 =VALinput$

```


MIRRORSOFT

MICRO-LEARNING SPECIAL

MIRRORSOFT'S educational programs are equally at home in the classroom or the living room. Each one has been fully tested and evaluated for both its educational content and

its ability to motivate children. New skills, and practice in existing skills, are introduced within a game format; children are challenged to use these skills to get through the game.



First Steps With the Mr Men

Still topping the educational software charts, this delightful collection of learning games for pre-readers will captivate young children everywhere.
Cassette £8.95 Disk £11.95



Here and There with the Mr Men

Mr Tickle and his friends help young children with directions and simple route-planning in four colourful and absorbing games.
Cassette £7.95 Disk £10.95



Count with Oliver

Using simple sets plus addition and subtraction, Oliver can buy toys and icecreams from his local shop – but only if he gets his sums right!
Cassette £7.95 Disk £10.95



Quick Thinking Plus

An enhanced and expanded version of the original and highly popular QUICK THINKING!, these two space-age games test mental arithmetic skills for all ages.
Cassette £6.95 Disk £9.95



Look Sharp!

Two multi-level games to test memory and powers of observation. OLD MACDONALD'S FARM is for younger children, and S.O.R.T. is for age 8 and up.
Cassette £7.95 Disk £10.95



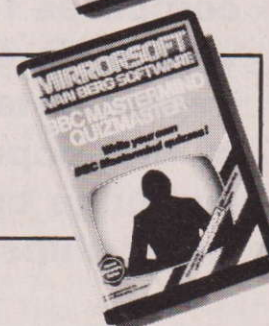
Star Seeker

A must for budding astronomers everywhere, STAR SEEKER allows you to explore the night sky – tracking stars, constellations, planets, and the path of Halley's Comet.
Cassette £9.95 Disk £12.95



BBC Mastermind/BBC Mastermind Quizmaster

Two programs, following the same format as the television quiz, which allow you to generate quizzes and tests on any subject from revision to last year's holiday.
BBC MASTERMIND Cassette £9.95 Disk £12.95
BBC MASTERMIND QUIZMASTER Cassette £5.95 Disk £8.95



MIRRORSOFT programs are available from good software stockists everywhere.

In case of difficulty or for further information, please contact:
MIRRORSOFT, FREEPOST, Bromley, Kent BR2 9UX
(No stamp needed) Tel: 01-822 3580

Four Logos for the BBC Micro

DEREK RADBURN compares four very different implementations of this popular language on ROM chips

BEFORE beginning this review I think it is right to declare my interests. In three of the implementations – Acornsoft, Logotron and Open University – I have to a greater or lesser extent been involved as a consultant; to a greater extent in the case of Logotron.

Having cleared that hurdle, I can now move to the business in hand reviewing the four Logos – Acornsoft Logo, Logo Software Logo (LSL Logo), Logotron Logo and Open University Logo.

In examining these implementations I shall be looking at the physical characteristics, the design philosophy and the performance of each on identical tasks.

One thing common to all is they come on ROM chips.

The biggest implementation is the Acornsoft, which comes in a box, with three manuals, reference card, disc and tape and uses two 16k ROMs.

The Open University implementation has two manuals and a disc and is also on two ROMs, of 16k each. Both Logotron and LSL versions are on single 16k ROMs.

The Logotron version is in a slip-case with a loose-leaf manual. LSL Logo, which is in a box, has a manual written by a primary teacher.

The Acornsoft Logo is accompanied by a disc and a cassette, which contain extensions to Logo, utilities and demonstrations. The Open University Logo also has a disc containing similar material.

Logo is made up of primitives – commands and operators which the system starts with. Acornsoft has 230 primitives listed on its reference card and this does not include the 11 commands for use in the editor.

Some of the 230 primitives are on the extension disc and have to be loaded into the machine.

LSL Logo has 122 primitives, all of which are resident in ROM. Logotron has 117, all resident in ROM. The OU Logo has 144 primitives.

The number which each implementation has should be treated with caution, for though it ought to reflect something of the comprehensiveness of the system, it does not indicate the power of individual primitives.

An illustration of this point could be to look at the primitives which deal with variables and procedures known to the system. In Acornsoft there are 27 for this purpose, whereas in Logotron there are 10, but among those 10 are two (*OPNS*, *OPPS*) which will allow the user to construct not only most of the

Acornsoft type of primitives but others as well.

The structure of Acornsoft and Logotron Logos are recognisably similar. Both are descendents of the dialect which originates from Massachusetts Institute of Technology.

Acornsoft Logo has, however, one fundamental difference when compared with the other three. It is the only one written in another language – BCPL (Basic Combined Programming Language).

This feature has a twofold significance. While it might appear that Acornsoft Logo behaves like a true Logo, it does not work like one.

This means that it is not possible to discover in a Logo way how much user space is available (this is done by typing *PR NODES* – a node is five bytes), and the system does not save its procedures as Ascii files which can be read by other Logo implementations or word processors.

While Acornsoft Logo is the only version to have property list primitives, Logotron Logo is the closest in its characteristics to the Logo implementations to be found on machines such as the Spectrum, Apple, Atari and so on.

It advertises itself as an LCSi (Logo Computer Systems Incorporated) standard Logo. Though it is not identical to the LCSi implementations (which themselves vary) found on other machines, it presents few problems to convert procedures from one machine to another.

If there is, at present, anything approaching an international standard in Logo then it is likely to be LCSi implementations. Logotron Logo is, though, radically different to previous LCSi implementations in a number of important ways.

Many primitives which are normally available in two forms – full and abbreviated – exist in only the abbreviated in Logotron. (For example *OUTPUT* is only available as *OP*.)

Though there are, in some senses similarities between the Acornsoft and Logotron versions, there are also fundamental design differences.

Acornsoft attempts to be comprehensive in its provision of facilities, while Logotron is lean and selective, expecting that the advanced user will construct many of the facilities they will need, and therefore it provides fewer, but far more generalised primitives.

Users who are used to the Logo found on the RML machines (Research Machines 380Z and 480Z computers)

will find much familiar in LSL Logo. There are differences, *REST* in RML Logo is replaced by *BUTFIRST* in LSL, and *REPEAT* has a bracketed list as its second argument in LSL Logo, but there are strong similarities in structure.

There are no *TEXT* or *DEFINE* commands in LSL Logo. *TEXT* and *DEFINE* are primitives which in most other Logos allow procedures to be turned into lists, which can be modified and then turned back into procedures. Like RML Logo, LSL Logo requires that more than one command on a line has a separator. In LSL it is the comma. It is the only version not to have a *WRAP* facility in graphics.

An inconsistency exists in the current version of LSL Logo. While it is possible to create a variable which is an empty list, it is not possible to do this with an empty word.

The OU Logo is possibly the most radical of all in its design philosophy. In all Logos hitherto it has always been possible to distinguish between procedures, names and values by syntactic markers, quote marks or colons.

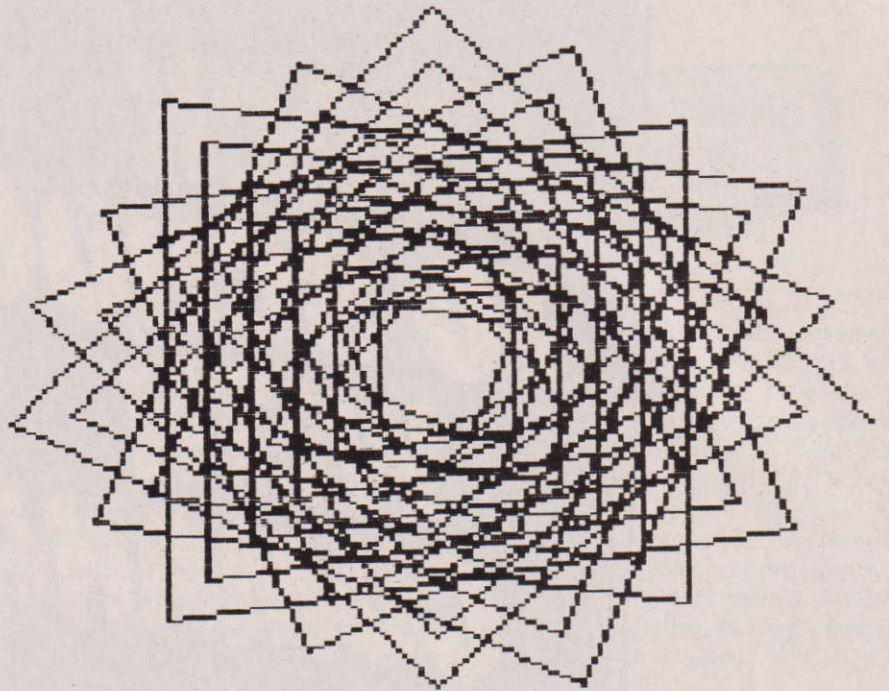
In OU Logo it is not necessary to put a colon in front of a variable's name when wanting to evaluate it. Another feature of OU Logo is dynamic lists. These are lists which evaluate the expressions which they contain.

Additional facilities of OU Logo are *DRIVE* and *PAINT*. *PAINT* is really a fill command which will cause any closed shape occupied by the turtle to be filled with the current *PEN* colour. *DRIVE* steers the turtle around the screen using the arrow cursor control keys. It is possible to get diagonals by pressing two keys together.

With two versions, LSL and OU, the defining of procedures has to take place in the editor. With LSL definition occurs by calling the editor with *EDIT* procname.

The LSL editor uses a system which is very different to the other Logo editors. It divides the screen into two sections. The upper, larger portion is where the procedure appears. The lower portion is where entries and amendments to lines are made. A line may not be longer than this space.

Editing is accomplished by a combination of function keys, cursor control keys and the Copy and Delete keys. Where the length of a Logo line exceeds the length of a screen line this is indicated by the insertion of an indicator character at the end of the screen line.



The OU editor gets invoked as soon as *TO procname* is typed. Editing takes place in a window and continuity of a line is shown by colour. Alternating colours are used to show Logo lines, with green being reserved to show the title line of the procedure. On screen help is available at all times by pressing function key f5.

Definition in Acornsoft and Logotron Logo can occur outside of the editor, in a special defining mode. This gets invoked as soon as *TO procname* is typed, and is indicated by a change from the question mark prompt to a caret.

In both cases departing from the defining mode can only be achieved by typing a final line with *END* alone on it, or by pressing Escape.

Acornsoft's editor indicates where Logo lines overflow by showing the overflow in inverse. Logotron uses a spurious indicator character at the end of the line where overflow begins. The Logotron editor has facilities which resemble those more commonly associated with word processors. Pressing f9 will invoke *FIND* and *REPLACE*.

All implementations have a trace facility of some sort to assist with debugging procedures. Logotron simply shows line by line execution of procedures, giving the appropriate inputs and outputs.

In Acornsoft Logo it is possible to *TRACE* at different levels, which are determined by the numbers which follow the *TRACE* command. *TRACE 1* shows line by line execution, *TRACE 2* traces every procedure call, and so on.

The OU implementation has *TRACE*, *WALK* and *BUG*. Here what is traced is a variable, not a procedure. Every time a variable is called then its name and inputs are shown, as too are any resultant outputs. *WALK* is the closest to the trace facility on the other two versions. This allows the execution of a procedure to be followed line by line. The function of *BUG* causes a variable to be shown every time its value is changed.

LSL Logo has similar debugging tools, though they are called by different names. An additional feature of LSL is *SPY*, which creates a file which receives a copy of all subsequent keyboard inputs. This can be accessed later to see how a particular user interacted with Logo.

I believe Logo error messages should report on what the offending item is, where it occurred, and what was wrong with it. Three of the Logos – Acornsoft, LSL, Logotron – do this to a greater or lesser degree. In the case of LSL the line itself is not shown, but the number of its position in the procedure is given.

In both the Acornsoft and the Logotron versions the actual context of the encountered error is given. OU Logo messages report the offending item and the procedure in which it has been encountered. They do not, though, provide the context of the line in which the error occurs.

All systems possessed file-handling capabilities of some sort. In all cases it

BACK TO SCHOOL WITH CUMANA

The best name in memory.



Now A Name In Educational Software

LTS Software for the BBC Micro

"A completely new range of exciting programs for schools, all with full colour graphics, and many sponsored by the MEP, are now available from Cumana, Europe's leading supplier of disk drives".

LANGUAGE DEVELOPMENT PACK

Three pack program to develop language and grammar skills

Ages 10-15 yrs £12.95

EDSELL A fast spell checking package

Ages 8-9 yrs upwards £17.25

PROBLEM SOLVER

Develops problem solving and decision making skills

Ages 10-17 yrs £10.95

MATHS TALK Two programs to develop understanding of mathematical concepts and related language skills

£7.95

PICTURE BOOK

Five programs for sentence construction & reading skills

Ages 6-12 yrs £14.95

CLOZE Invaluable language teaching aid

Ages 8-16 yrs £10.95

FRUIT PICKER Spelling and word formation program

Ages 6-11 yrs £8.95

PIED PIPER An aid to memory combining letters, numbers and musical notes

Ages 4-10 yrs £8.95

HARLEQUIN For the development of spatial skills

For younger children £14.95

JUNIOR PACK For children with learning difficulties, or younger children

£7.95

Four unique full colour adventure games – ideal stimuli for creative writing and the development of problem solving skills:-

YOUR ADVENTURE

7-13 yrs £10.95

SPACE ADVENTURE

7-14 yrs £11.95

MATHEMAGIC LAND

10-16 yrs £8.95

THE LAST ADVENTURE

7-13 yrs £17.95

SPECIAL DISCOUNT AVAILABLE ON ALL PROGRAMS FOR SCHOOLS

NOW AVAILABLE – "School Help" software programs by Micro-Draw for use with the Cumana Touch Pad

Cumana Limited,
Pines Trading Estate, Broad Street, Guildford, Surrey, GU3 3BH.
Telephone: Guildford (0483) 503121 Telex: 859380

Please send further information about Cumana's LTS Educational Software
NAME/SCHOOL _____

ADDRESS _____

TEL. NO. _____

From Page 25

was possible to *SAVE* a procedure and bring it back again from either the tape or disc to which it had been saved.

There is a difficulty with Logotron Logo. *SAVE* will not work with **TAPE* or **NET*. This can be overcome, though, as procedures can be saved by using **SPOOL*.

In the case of LSL Logo it was necessary to save each procedure individually. With the other versions there were facilities to save clusters of procedures with their associated variables. Logotron Logo allowed the saving of specific variables.

Two versions – LSL and Logotron – provided the facility for writing and reading data to and from text files. LSL Logo's facilities in this respect were the most extensive of all. All of the systems allowed the user to have access to the machine's own operating system. However LSL cautioned users against using all but a few selected operating system calls.

To give an indication of their relative performance I gave all four systems identical tasks. One was the rather trivial job of reversing a list backwards and forwards a 100 times. The procedure I used is shown in the processing speed test.

Essentially this was a test of processing speed. Table I shows the results. The next test, however, as well as testing speed tests processing power, since it involves the generation and manipulation of an exponentially growing list, which is then used to draw intricate patterns known as dragon curves. The Acornsoft listing is shown in the second test.

(Note: The higher the number following Dragon, the more processing power needed. Only Logotron Logo was able to accomplish Dragon 8. The results are summarised in Table II.)

In summary I found four very

Open University Logo... very much a dish for gourmets

different implementations. Acornsoft undoubtedly is a very full implementation, apart from the surprising absence of text file handling facilities. It comes as a lavish and stylish package and the three manuals are comprehensive in their coverage.

The accompanying software on disc and cassette allows for very useful extensions like screen dumps to printers. As a style of implementation it attempts to provide all the facilities a user might reasonably require.

Its most grievous shortcoming lies in its slowness, and its need for a second processor if anything beyond orthodox turtle graphics are to be achieved. I would see this as an appropriate implementation where there is unlikely to be extensive use of list processing facilities or where there is likely to be little desire or need to have to extend the facilities of Logo.

It is overall an easy to use implementation, with helpful error messages and plenty of good documentation.

The Open University Logo is, I think, very much a dish for gourmets. I do not believe its radical attempt to develop Logo makes it likely to have widespread appeal. For example, the absence of the colon as an indicator of evaluation is fine for computer scientists and experienced Logo users, but it is likely to lead to confusion of concept of a variable with that of a procedure for beginners, particularly children.

I find OU Logo contradictory, in that it has quite a good turn of speed in list processing, but is slow in its execution of graphics (generally, in Logos, the reverse is the case).

A particularly likeable feature, for me, was the editor, with its good use of colour and constantly available on-screen help. The debugging facilities are very full and helpful. The bug I encountered with *PUTFIRST* and *PUTLAST* – which has since been rectified – prevented me from being able to run the dragon curve program in OU Logo. I also encountered a serious problem in getting the implementation to work with a second processor – it simply wouldn't!

There is, though, a special disc or cassette version for use with the Tube and second processor.

The documentation is a rich source of ideas and information about programming. It describes some quite advanced ideas in clear, simple English, but I am not sure that easy language ensures the grasping of complex, abstract ideas.

LSL Logo is the product of a design philosophy which would make it an ideal version to be used in circumstances where RML Logo was also to be used. It attempts to be highly consistent, and generally succeeds.

There is, though, the notable exception of the way in which it treats empty words. This is not a trivial discrepancy. Being able to simply output an empty word at the end of some recursive list processing is essential. To be able to do this with LSL Logo, the user has to resort to subterfuge by creating a variable which takes its value as the *BUTFIRST* of a one character word.

I do not like having to do my

Processing speed benchmarks

```
TO REV OBJ
  IF EMPTY :OBJ [IF WORD :OBJ
  [RETURNVALUE :NULL] [RETURNVALUE []]]
  IF WORD :OBJ [RETURNVALUE JOIN LAST
  :OBJ REV BUTLAST :OBJ]
  IF LIST :OBJ [RETURNVALUE PUTLAST REV
  FIRST :OBJ REV BUTFIRST :OBJ]
```

LSL

```
TO REV :OBJ
  IF EMPTY? :OBJ [IF WORD? :OBJ [OP " ]
  [OP [ ] ]
  IF WORD? :OBJ [OP WORD LAST :OBJ REV
  BL :OBJ]
  IF LIST? :OBJ [OP LPUT REV FIRST :OBJ
  REV BF :OBJ]
  END
```

Acornsoft

Logo Version	Timing in seconds	
	Single processor	Second processor
Acornsoft	423.25	282.47
LSL	76.23	51.89
Logotron	71.77	46.43
OU	222.12	—

Table I

Processing power benchmarks

```

TO DRAGON :ORDER :POINTER
  MAKE "D [L L R]
  IF :ORDER > 0 [REPEAT :ORDER - 1 [IF
:POINTER > 0 [MAKE "POINTER 0 LLIST]
[MAKE "POINTER 1 RLIST]]]
  RUN :D
  END
TO LLIST
  MAKE "D (SE :D "L ALT :D)
  END
TO RLIST
  MAKE "D (SE :D "R ALT :D)
  END
TO ALT :L
  IF EMPTYQ :L [OP [ ] ]

```

```

  MAKE "N (((COUNT :L) - 1) / 2)
  IF ITEM (:N + 1) :L = "L [OP REPLACE
"R :N :L] [OP REPLACE "L :N :L]
  END
TO REPLACE :W :N :L
  IF EMPTYQ :L [OP [ ] ]
  IF :N = 0 [OP SE :W BF :L] [OP SE
FIRST :L REPLACE :W (:N - 1) BF :L]
  END
TO L
  REPEAT 18 [FD 0.9 LT 5]
  END
TO R
  REPEAT 18 [FD 0.9 RT 5]
  END

```

Acornsoft version of the Dragon Curve program

From Page 27

arithmetic in *ADD 2 3* form instead of *2 + 3*, still less *SUBTRACT 5 2*, but I do not think that this, or the need to use commas as separators (which, again, I do not favour) are shortcomings – they are a matter of belief and preference.

Where I find myself less happy is over the error messages. There is a “No room” message, but it does not get invoked in all the circumstances when it ought. On several occasions when attempting to get the system to do dragon curves beyond its capacity I got variable error messages once it crashed.

If I had not already validated my procedures by running them at a less demanding level, I could have spent many a happy moment looking for non-existent bugs.

I feel unhappy about the absence of the provision of a swift and simple method of getting from the graphics screen to the text screen and back again. I also do not like the restrictions imposed on Logo line length by the editor. In a version as powerful as this I believe there will be need to write longer lines to be able to tap that power.

I think that the strong characteristics of this implementation lie in its debugging and file-handling. Documentation is well produced and “The Young Person’s Guide to Logo” is clear and attractive. LSL Logo is definitely a version which should be looked at.

The Logotron version (and here I remind the reader of possible bias) proved to be the fastest and most powerful version tested.

It had helpful error messages. Its

documentation I think is particularly interesting, in that it makes use of the existing Logo literature.

Though giving an adequate introduction to turtle graphics, it does not attempt to develop the topic, but refers the reader to readily-obtainable books. Instead, there is the most extensive treatment of list processing in any manual I have so far seen.

I feel Logotron Logo does have some shortcomings, though. I think it suffers, as does LSL, from the absence of the type of extensions provided in the Acornsoft package. For example, if Logo is destined for the classroom I think it vital that it should include screen dump software.

I also regret the absence of true *TEXT* and *DEFINE* primitives. For most users this latter point will not be significant. What is, for me, beyond doubt is that Logotron Logo is the fastest and most powerful of the four.

Logo Version	Time in seconds	
	Single processor	Second processor
Acornsoft Dragon 4 1	100.67	60.14
LSL Dragon 5 1	281.26	151
Logotron Dragon 5 1	141.29	82.54
Acornsoft Dragon 6 1	incapable	245.71
LSL Dragon 6 1	565.89	301.19
Logotron Dragon 6 1	273.27	166
Logotron Dragon 8 1	incapable	668.21

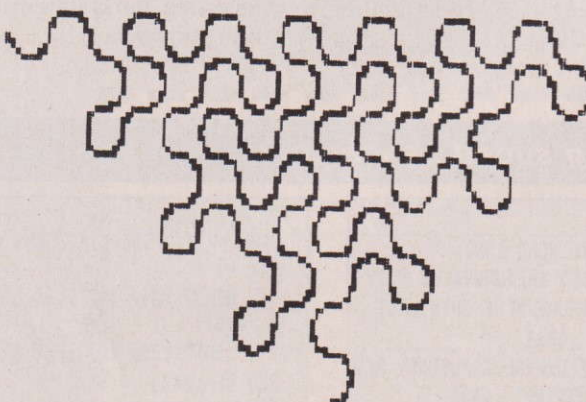
Table II

I like the idea of fewer, but more generalised and powerful commands, which allow the user to construct the commands they want. Both LSL and Logotron have plans for extensions, both in hardware and software.

In presenting this review I am very conscious that in a few words I am attempting to describe work which has taken many people months of dedicated labour to achieve.

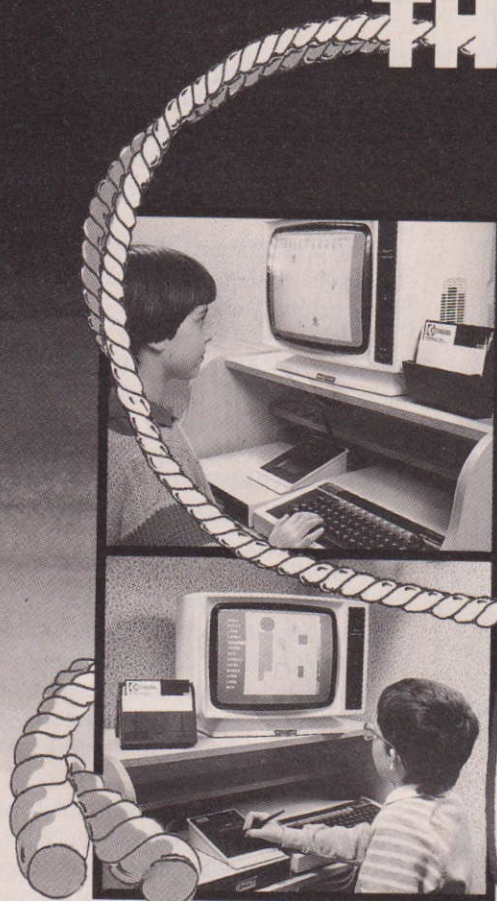
I ask the reader to be mindful of this, and if deciding to buy a Logo, to try it first. Readers may be also interested in the British Logo User Group, details of which may be obtained from Roger Knott, c/o CAMET, University of Technology, Loughborough, Leics. LE11 3TU.

I would, finally, like to express gratitude for the generosity of Acornsoft, who loaned me a computer system with a second processor upon which to conduct this review.



Typical output of the Dragon Curve program

WHAT MAKES OUR DRIVES THE SAFEST?



What makes our drives the safest? This does! As the computer market and the computer peripheral market in particular, has grown, one company above all others has invested in safety. That company is Cumana, already the best name in memory, and now reaching new standards for safety. The British Standard specifications lay down quite firmly, standards in design that are your guarantee of electrical safety. Their famous "seal of approval" has now been awarded to Cumana disk drives.

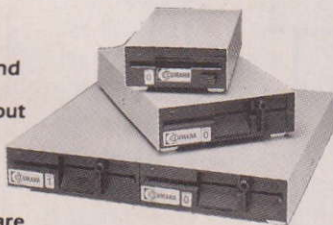
- * Each component part is thoroughly checked
- * Prying fingers are safe, as Cumana drives need no ventilation slots
- * Moulded mains plug with half-insulated plugs
- * All terminals on the main switch are insulated
- * 500 ma in line fuse holder, completely insulated
- * Power supply PCB covered by perforated screen
- * Rigorous quality control to BS415 standard
- * Plus many other manufacturing and quality control tests, in fact no component can be changed without it being approved by the British Electrical Approvals Board.

Cumana disk drives are not only competitively priced, but also the reference standard by which others are judged.

Be safe not sorry - insist on... Cumana, the best name in memory.

**BE SAFE
NOT SORRY-**

Insist on



CUMANA®
The best name in memory

Cumana Ltd., Pines Trading Estate, Broad Street,
Guildford, Surrey, England, GU3 3BH.
Tel: Guildford (0483) 503121. Telex: 859380.

For further information about Cumana disk drives please complete and return this coupon.
Interests:

Home Use ☐ Education ☐ Dealer ☐

Business ☐

Name

Address

Tel. No.

Note: If dealer, please attach this form to your letterheading

SCORE 2

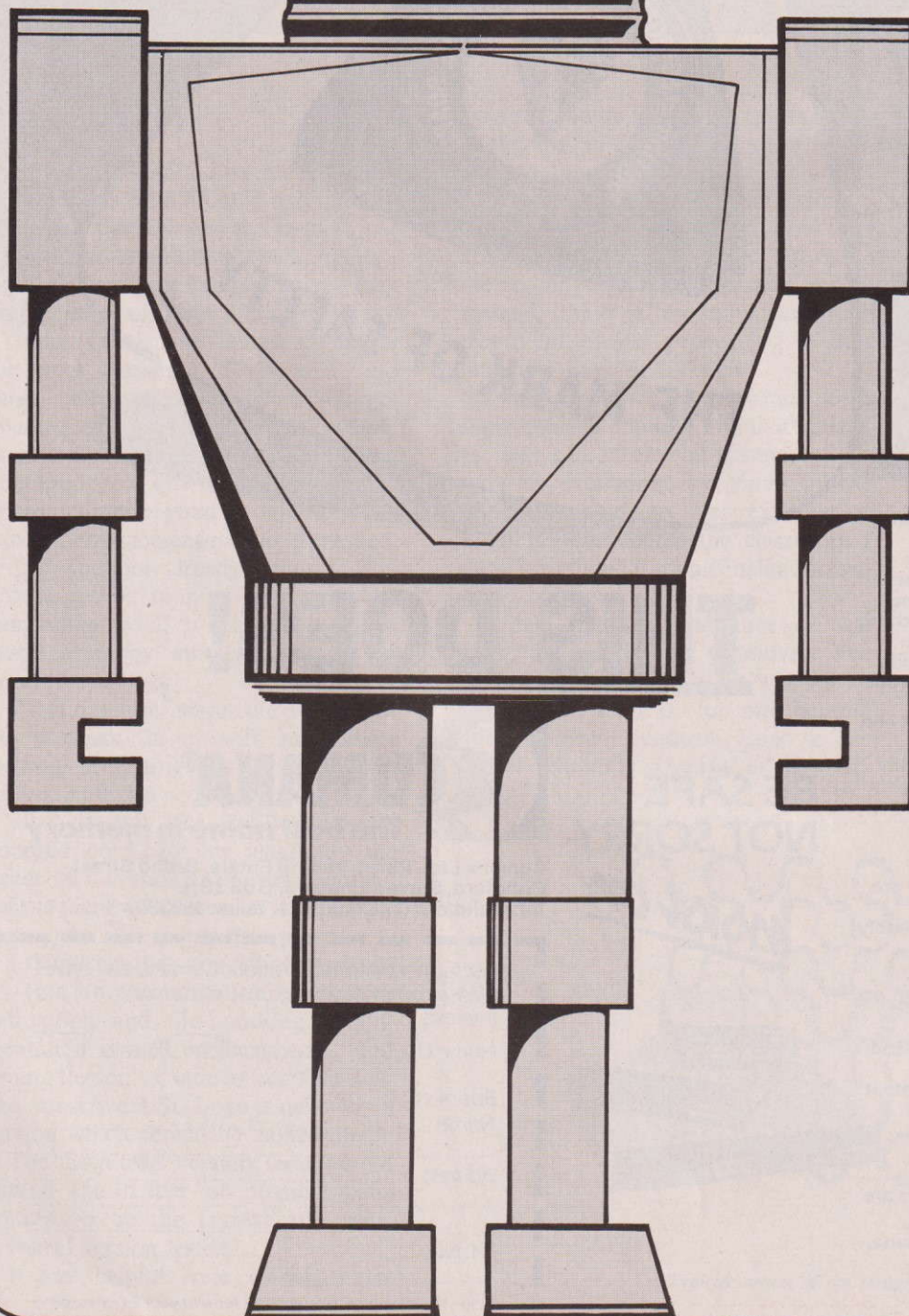
RATING Knight

might

mite

CORRECT

I might come out to play.



SPE

THIS is a spelling program for children of around eight years and over. As the program is designed for use by all ability levels in a class, on-screen instructions and keyboard inputs have been kept to a minimum.

The presentation is attractive without being distracting, with colour and sound used helpfully throughout.

Mode 1 is used, a fair compromise between having lots of colours available and being able to present a simple sentence legibly on the screen.

The friendly spelling robot Spelldroid introduces himself, discovers his operator's name and how confident he is of his spelling ability.

This done, a pair of words are presented, one in each of the robot's "eyes", together with a simple sentence which provides a context example.

The task is simply to choose which of the two words is the one correctly spelt – the one to the left or the one to the right.

Selection is made via the cursor keys, pressing left cursor key for the left-hand word and the right cursor to choose the one on the right.

If no selection is made within a certain time, then Spelldroid assumes his operator doesn't know the answer and carries on to the next pair of words.

A tour of the listing will explain how the program works.

Early on the Break key is reprogrammed (line 40) so that teacher can rest easy knowing that he won't have the disaster of having to re-load the program from tape in the middle of a session.

After initialising the screen and main variables the title page is called. The choice of sound is offered, as is a look at a short page of instructions.

PROCinstr contains two little drawing procedures – PROCkeyl and PROCkeyr – which represent the cursor keys graphically.

PROCinit introduces the program, elicits the pupil's name and his level of spelling competence.

Most of the input validation required by the program resides in this procedure, so it might be worth a closer look.

In line 730 the pupil's name is

LLDROID

Spelling becomes fun with this colourful program by PHILIP DUNN

prompted for and a loop entered if Return only is pressed.

After a friendly THANK YOU from the robot, it prompts for the level of difficulty. Again null strings are rejected and multiple digits (caused, say, by holding down a number key too long) are trimmed to the first input.

Entry of a word instead of a number is trapped at line 800 and a message issued.

As a result of the input a rating is assigned (kinder than a level) which determines the set of data to be used and the time limit set for each answer.

In lines 140 to 160 the robot's face is drawn containing the four text windows defined by PROCtext(P).

Next, the arrays used are dimensioned.

W\$() will be used to store the words output in the report. R\$() and WRONG\$() store the right and wrong spellings.

SAMPLE\$() and SAMPLE2\$() are the first and second parts of the example sentence. W() holds the pseudo-random value of P which determines which of the first two windows will display the correctly-spelled word. These P values are held in the very last data statement.

Line 380 directs the incorrect word into the other window.

Window 4 is used to display the robot's verdict on the answer and Window 3 is always used for the sample sentence, which is always completed correctly after a choice has been made or the time allowed has expired.

PROCcheck tests for a correct answer and PROCgood and PROCbad generate the robot's response and increment the score accordingly.

In PROCend use is made of the command VDU26 to return the text screen to its initial default value enabling the final score and comments to be written outside any of the four windows used in the main program.

In this procedure an unprompted

on-screen report can be called for.

PROCprint will output a hard copy of all the words the named pupil should have chosen but didn't! This could serve as a record or as a basis for further practice.

Where a printer is not available PROCreport(b) asks if a printer is connected, and generates either hard copy or a screen display of words which the pupil should learn.

The screen presentation was made simpler and quicker to program by employing the multi-parameter box-drawing procedures PROCboxz (SIZE,X,Y,Z,SCALE) and PROCrect (SIZE,X,Y,Z,SCALE).

X and Y are the screen coordinates and Z the colour to be used in plotting the box.

SIZE and SCALE determine the length of the side of the box and the ratio of its sides.

The variables S and T are used with the functions FNpad(N) and FNplace(N). FNpad(N) returns the length of the word required to complete the sentence correctly.

This value is assigned to the variable S, and SPC(S) is used to print a gap of the correct size in the sample sentence.

FNplace(N) returns the length of the first part of the sample sentence and in line 430 this value is assigned to the variable T. In line 450 this is used to position the right answer in the example sentence.

Lastly, in any educational program it is important that the teacher can change the data easily so that the differing needs of his pupils can be catered for.

The program has a very simple structure, all the data being grouped at the end and the early REM statements show the form in which the data should be entered.

This simple program provides quite a flexible basis for further language programs. Change the colour scheme, the PROCbox parameters and the text window shapes and hey, presto! Puncdroid or Grammdroid!

```
10 REM SPELLDROID - SPELLING GAME
20 REM WITH ROBOT HELPER
30 REM CHOOSE CORRECT WORD FROM PA
IR USING LEFT CURSOR FOR LEFT SIDE AN
D RIGHT CURSOR FOR RIGHT
40 *KEY 10 OLDIM RUNIM
50 REM(c) The Micro User
60 REM INPUT DATA IN ORDER <RIGHT
WORD>,<WRONG WORD>,<SAMPLE SENTENCE>,
(<SAMPLE SENTENCE PART TWO IF NEEDED>
)
70 MODE1:VDU23;8202;0;0;0;0;0;0;SCORE
=0;RATING$="";VDU26:CLS
80 B$="PRESS A KEY TO GO ON"
90 PROCpage
100 PROCinit
110 VDU23;11,0;0;0;0
120 PROCtune:PROCtune:PROCtune
130 PROCpic
140 X=60:A=0:B=0
150 COLOUR131:FORX=1TO4:PROCtext(X)
:CLS:NEXT
160 VDU19,3,6,0,0,0

170 REM SET UP ARRAY FOR WORDS
180 COLOUR0
190 DIMW$(20)
200 DIMR$(20)
210 DIMWRONG$(20)
220 DIMSAMPLE$(20)
230 DIMSAMPLE2$(20)
240 DIMW(20)
250 REM FILL ARRAYS
260 FOR N=1 TO 20:READ R$(N),WRONG$(N),SAMPLE$(N),SAMPLE2$(N)
270 NEXT
280 *FX4,1
290 REM NOW PRINT IN APPROPRIATE WI
NDOW
300 RESTORE 1700
310 FOR N=1TO20
320 READ W(N)
330 NEXT
340 FOR N=1 TO 20 :P=W(N)
350 PROCtext(P):PRINTTAB(1,2)R$(N)
```



```

720 VDU26:COLOUR128:CLS:VDU23;11,25
5;0;0;0
730 COLOUR1:PRINTTAB(15,8)"Hello fr
iend";TAB(10,12)"I am a spelling droi
d.":COLOUR3:PRINTTAB(12,14)"What is y
our name?":COLOUR1
740 PROCrect(100,320,128,3,7):INPUT
TAB(12,26)C$:IF C$="" GOTO740
750 COLOUR 3:PRINTTAB(12,18)"THANK
YOU ";C$:DUMMY=INKEY(300):CLS
760 COLOUR2:PRINTTAB(5,10)"How good
are you at spelling?"
770 COLOUR1:PRINTTAB(5,14)"1...Not
very good";TAB(5,16)"2...O.K.";TAB(
5,18)"3...Good";TAB(5,20)"4...Brill
iant"
780 PROCrect(50,150,540,3,1):PROCRe
ct(50,150,474,3,1):PROCrect(50,150,40
8,3,1):PROCrect(50,150,342,3,1):PROCRe
ct(70,300,152,3,1):INPUTTAB(10,26)A$
790 A$=LEFT$(A$,1):IF A$=""GOTO780
800 IF INSTR("1234",A$)=0 PRINTTAB(
10,26)"":COLOUR3:PRINTTA
B(5,30)"Press 1,2,3 or 4 please.":COL
OUR1:GOTO780
810 IF A$="1" RATING$="Servant"
820 IF A$="2" RATING$="Squire"
830 IF A$="3" RATING$="Knight"
840 IF A$="4" RATING$="King"
850 IF A$="1" TX=200:RESTORE1500
860 IF A$="2" TX=150
870 IF A$="3" TX=100
880 IF A$="4" TX=50
890 CLS
900 ENDPROC
910 DEFPROCend
920 COLOUR3:COLOUR128:
930 PROCtext(4):CLS:PRINTTAB(4,2)"E
ND"
940 VDU26:PRINTTAB(5,17)"You scored
";TAB(5,19);" ";SCORE;"/";N-1
950 IF SCORE<10 AND A$<>"1" THEN PR
INTTAB(25,17)"Not as good";TAB(25,19)
"as you";TAB(25,21)"thought !"
960 IF SCORE>=10 AND SCORE<15 THEN
PRINTTAB(25,17)"Ok. level";TAB(25,19)
"2 next time"
970 IF SCORE>=15 AND SCORE<18THEN

```

32 MICRO USER EDUCATION SPECIAL


```

1370 PROCrect(120,110,80,3,9)
1380 FOR Y=780TO830 STEP 5:PLOT78,X,
Y:NEXT
1390 PROCbox(600,650,43,3,1)
1400 PROCbox(600,48,43,3,1)
1410 PROCrect(20,60,740,3,59)
1420 PROCbox(590,55,50,3,1)
1430 PROCbox(590,660,50,3,1)
1440 PROCrect(200,55,660,3,6)
1450 PROCrect(20,60,670,3,59)
1460 PROCrect(20,60,690,3,59)
1470 PROCrect(60,90,880,1,8)
1480 PROCrect(60,720,880,1,8)
1490 PRINTTAB(5,3)"SCORE ";SCORE;TAB
(23,3)"RATING ";RATING$
1500 COLOUR131:FORX=1TO4:PROCtext(X)
:CLS:NEXT:ENDPROC
1510 DEFPROCinstr:COLOUR128:VDU26:CL
S:PROCbox(160,128,1200,1,1):PRINTTAB(
12,6)"INSTRUCTIONS":PRINTTAB(5,10)"In
this game you must choose";TAB(5,12)
"which word to put in the sentence."
1520 PRINTTAB(5,16)"Use the and
CURSOR KEYS";TAB(5,20)"to choo
se the word in the ";TAB(5,22)"left o
r right eye of the robot.";TAB(5,26)"

```

```

Always read the sentence carefully";T
AB(5,28)"before you choose.":PROCkey1
:PROCkeyr
1530 COLOUR3:DUMMY=INKEY(500):PRINTT
AB(10,30)B$:X=GET:ENDPROC
1540 DEFPROCkey1:PROCbox(100,430,450
,1,1):MOVE500,500:DRAW450,500:MOVE470
,490:PLOT85,470,510:ENDPROC
1550 DEFPROCkeyr:PROCbox(100,730,450
,1,1):MOVE750,500:DRAW800,500:MOVE780
,490:PLOT85,780,510:ENDPROC
1560 DEF FNpad(N)=LEN(R$(N))
1570 DEF FNplace(N)=LEN(SAMPLE$(N))
1580 DATA to,two,Take this money,Dad
1590 DATA said,sed,Johnny,you were l
ate.
1600 DATA might,mite,I,come out to p
lay.
1610 DATA slipping,sliping,I keep,on
the ice.
1620 DATA seat,seet,Sit down on that
,
1630 DATA tea,tee,Do you like,to dri
nk?
1640 DATA poor,pore,My friend is ver

```

```

y,.
1650 DATA door,dore,Shut that,.
1660 DATA floor,flore,My cat sat on
the,.
1670 DATA meat,meet,We get,from a bu
tcher.
1680 DATA feet,feat,John has got two
,.
1690 DATA street,street,We walk down
the,.
1700 DATA wheat,weet,Flour is made f
rom,.
1710 DATA when,wen,See you,you get b
ack.
1720 DATA wait,wate,I'll,for you at
school.
1730 DATA use,yous,I,a saw to cut th
ings.
1740 DATA worm,wurm,The bird caught
a,.
1750 DATA square,squair,My paper is,
.
1760 DATA cake,cack,The,was lovely.
1770 DATA soup,soop,My,was very hot.
1780 DATA 1,2,1,1,2,2,2,1,2,1,1,2,
1,2,1,2,1,1,2

```

Announcing a new full-feature word processor ROM (16K) which won't burn a hole in your pocket

For £27.50 (inc. VAT, p&p) You will receive:

- 1) The "ProWord" word processor ROM
- 2) A full manual (with introductory and reference sections)
- 3) The function key strip (4 functions per key)

FEATURES INCLUDE:

- Tape, disc & 2nd processor 6502 compatible. (Special "HiWord" 2nd processor only version available for same price).
- All formatting on screen, including 4 justification styles: left, centre, right, pad.
- On screen fonts: normal, bold, underline, italic, superscript, subscript, inverse.
- Editor features include: find & replace, block move, variable margins & tabs, backup text, 16 cursor functions, etc.
- 40 or 80 column display, with 25 or 32 lines. Text and background colour may be altered to suit your own taste.
- Up to 160 column formatting (using automatic left to right scrolling display).
- Full printer support, including microspacing, page headers & footers, page numbers (with Roman numeral option), double line spacing, alternate page printing (for double sided printing), overprinting etc.
- In-ROM printer setup program allows you to configure this system to suit your particular printer without having to buy costly add-on printer drivers.
- Large documents may be created over several files, which are printed as if they were one large file. This means that you are not limited by the BBC Micro's small memory.
- For Disc users: Inbuilt Mailing list & Mail Merge system (allows you to create standard letters etc.) Mailing list system may also be used separately as a mini Database.

FOR MORE INFORMATION (INCLUDING SAMPLE PRINTOUT) PLEASE SEND A STAMPED, SELF-ADDRESSED ENVELOPE.

Available only from:

**T. Hall, 31 Wethersfield Road, Noctorum, Wirral,
Merseyside L43 9YF.
Telephone: 051-652 7560**

Make cheque or P.O. payable to **T. Hall**. Allow 14 days for delivery. Please specify if "HiWord" ROM is required.
(Written by the author of the Number 1 selling (April '84) Adventure game "Twin Kingdom Valley")

Dealer enquiries welcome.

Educational orders welcome.

Can your supplier live up to our standards?

Example

Do you buy 40 track 100K single sided disc drives?

Are you using the CSX100 or the CS100 disc drive?

Viglen can supply you with a 40 track 200K double sided drive containing the same make of mechanism as the CSX100 and CS100 at the same or lower price than you are currently paying for your 100K drive.

- TWICE THE CAPACITY
- NO COMPATIBILITY PROBLEMS
- LOWER PRICE!

Interested?

If you are already committed to buying a particular make of disc drive, we at Viglen will supply a disc drive with the identical mechanism. AV Departments will have no cause to worry about compatibility of spare parts etc.

Viglen, specialist producers of disc drives for the BBC micro, and manufacturers of the best-selling ROM Cartridge Kit, have become one of the leading suppliers of BBC-related equipment to Education Departments. Before you commit your hard-won capitation, consider the benefits that only a large company like Viglen can offer.

Competitive Prices

Viglen have a deserved reputation for leading the field in offering top quality products at the lowest possible prices. We supply you direct from our factory at truly wholesale prices.

Safety Tested

Our disc drives have been thoroughly tested and recommended by the prestigious Test Bureau. Copies of their reports available on request.

Thoroughly Reliable

All Viglen products are individually tested at our factory before despatch. Our disc drives contain only top-quality Japanese mechanisms as used by leading disc drive suppliers. All products carry a full 12-month warranty – extended warranty available at nominal cost.

After-Sales Service

What happens if the equipment should fail? As soon as we receive notification from an Educational Establishment of any faulty equipment, we arrange for our carriers to collect the item (at our expense) and return it to us. It is then replaced or repaired and returned to you, usually within 7 days.

Rapid Delivery

We carry a huge range and stock of hardware and software items, including over 40 different types of disc drive. We always aim to deliver within seven days of receipt of order.

Technical Advice

Our highly-trained staff will be pleased to help you with software and technical advice, and Viglen offers free technical training for AV Departments and can supply them with Service Manuals, Alignment Diskettes and spare parts.

Evaluation

Products are available to Advisers, AV Departments and Teachers' Centres for evaluation.

Tenders

Viglen are able to fulfil Educational Contractual requirements on a wide range of products. Why not let us tender for your next Education Contract and check our prices and service capabilities.

Personal Contact

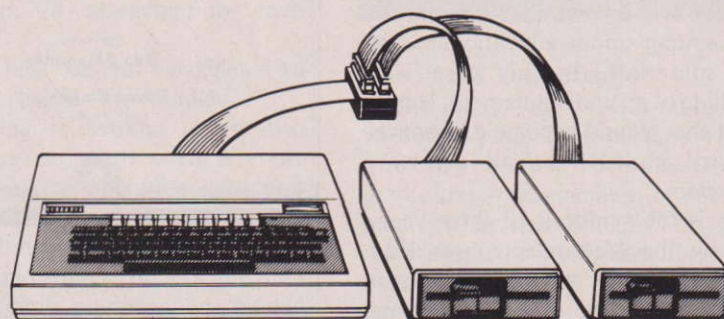
Viglen are expert in providing friendly, helpful, personal service. For further information, contact our Education Department and ask for Ray Davis, Education Marketing Manager, or Alison Batchelor, Education Marketing Co-ordinator.



**Viglen, Unit 7, Trumpers Way,
Hanwell, London W7 2QA.
Telephone: 01-843 9903 (8 lines)**

DUAL DRIVES FOR LESS THAN £15?

**Convert your single disc drives
into duals for only £14.75 + VAT**



**With the Viglen Dual Connector
anyone can convert single drives
to duals, instantly!**

- *Specially designed circuitry eliminates the need for any tools.*
- *No need to take disc drives apart to change links.*
- *No soldering. ● No mess.*
- *Simply plug in for instant dualling.*

Just plug the Viglen Dual Connector into the BBC micro and any two single disc drives to create a dual disc drive. Drives will be configured Drive 0 and Drive 1 automatically.

- *Fast Back-Up*
- *Fast Copying of Files*

The Viglen Dual Connector will dual-up any make or type of single disc drives *with PSU* (power supply unit). If you have disc drives *without* built-in PSU, then you will require a dual power connector for the BBC micro - available from Viglen for only **£9.90 + VAT**.

- *Postage and packing £1.90 + VAT.*
- *Official Orders from Government and Educational Establishments welcomed.*

Essential reading for software developers

**Educational Software –
A creator's handbook.**

*Edited by Ken Alexander
and Diana Blanchard
MEP Software Unit.
Tecmedia Production.*

QUALITY is a word that I rarely use when writing about educational computer materials. In this case it is something of an understatement. Here is a book that should become a, if not *the* standard source for all software production.

This is the synthesis of three years' work by the Microelectronics Education Programme and makes full use of its wide knowledge of techniques and approaches.

It is an amalgam of many contributors' work with each chapter edited by someone with broad experience in that particular area.

The 276 pages are crammed with information and ideas.

If you are seriously considering producing software, you quickly realise that this is essential reading.

Here are helpful guidelines, in-depth studies of development issues, and many possible solutions to the vast range of problems facing the software developer.

The general aim of the book is to appeal to producers of software, as opposed to programmers and systems analysts.

The editors consider that there are two types of producers who should be catered for.

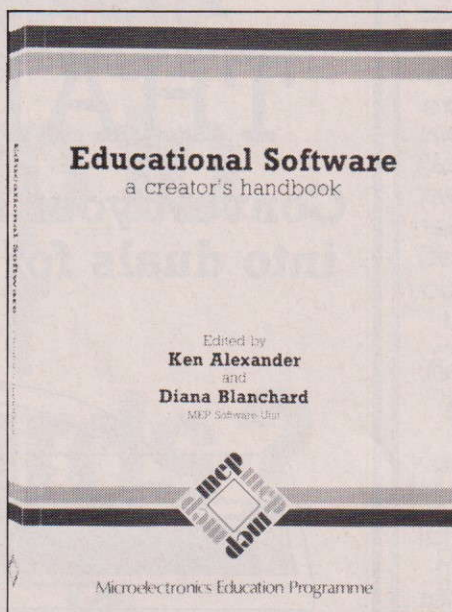
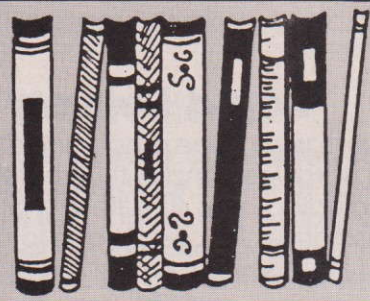
First of all there are those who know little of the technicalities, concentrating on educational aims but needing to appreciate the major issues and processes of software development.

The other main group are the technical producers who need to understand the educational aims involved in the particular project.

The book is organised in a systematic way, dealing with software production from initiation to publication.

This logical framework can be

BOOK SHELF



appreciated with chapter headings including Origination and Design, Coding, Field Trialling, Software Support, Publication and then Software Evaluation.

In each of these chapters the basic issues are dealt with in meticulous detail.

As a result of this detail, it is a book that you would read in stages, leaving time to assimilate the wisdom before continuing.

A closer look at one of the chapters shows the type of detail that is included.

The chapter on origination and design consists of eight sub-sections – a) developing the concept, b) working party structure, c) project management, d) program design, e) user-machine interface considerations, f) software interfacing, g) further design considerations, and h) presentation and appearance.

Each of these is further subdivided and, for example, in part b) consideration is given to such factors in meetings as group size, breadth of function, level of expectation, level of commitment, relative status and the skills and knowledge required.

Information and assistance of this type can only improve the general standard and approach.

As you would expect, the MEP stresses how vital it is for teachers to be

involved in software production.

It also suggests that teachers pool their ideas and, by applying good management techniques, they should eliminate individual teacher eccentricities and so produce more generally useful software.

The point is also raised that if software is to be exploited to the full, professional help should be sought.

It is also good to note the suggestion that the software houses need to produce software which can be used by any teacher without machine specific training.

One of its many perceptive conclusions is that in the final analysis the crucial factor is the quality of the educational ideas behind the program.

The chapter on coding includes details for the BBC Micro and also considers the RML 380Z and 480Z.

Structured programming is strongly advocated and examples are given.

Fundamentally, what they try to do is to set up ground rules for program development that may be considered good practice.

One of the lines of emphasis is to standardise the subroutines that are used in programming and mention here the Computers in the Curriculum Library at Chelsea College, whose package MINLIB contains many useful items.

The appendix contains examples of the types of material available.

If you are interested in the actual coding of programs, this chapter illustrates many useful lines of development and also considers many of the common pitfalls.

A little later in the book the point is stressed about listing problems using O instead of 0.

If you are not concerned with producing software but would like to use programs, the chapter on software evaluation is well worth looking at.

The general considerations are broken down into technical and educational aspects, instruments of evaluation, financial considerations and issues relating to reviewing and selecting software.

The book is completed by an

extensive list of addresses, very detailed appendices and a useful glossary.

I cannot over-emphasize the importance of this book.

Its guidance on technical and educational issues is well thought out and backed up by a wide range of experiences.

It is very carefully presented with interesting illustrations.

At £25 for a softback, you may not buy it yourself, but get the library on to it as soon as you can!

Norman Parr

Introduction to data management

Databases in the Classroom

By Derrick Daines

Castle House Publications, 1984

IF you are interested in any form of data management, you will find this book an excellent introduction.

It is written against the background of the "deluge of data" that society has accumulated and will continue to gather at an increasing rate.

If you accept that present knowledge is only three per cent of what will be available in 50 years, time, then we will be increasingly reliant on the computer as the only effective management tool.

There are only 116 pages, but they contain a wide variety of basic concepts and skills necessary to understand, build

and manipulate a database.

The various chapter headings show the logical organisation of the book.

They start with data management including sections on data capture, input, organisation, storage and retrieval.

Each of these sections is short and to the point and this general philosophy is carried throughout the book and is a big plus-mark.

Other chapters are computer skills, computer models, introduction to classification, tape and discs as storage media, serial and random access and finally a chapter on listings.

This latter section contains very clear instructions on how to build up the fundamentals of a database on either tape or disc.

There can be up to 10 fields with a maximum of 20 characters for each string field.

String, integer or real variables with up to 1000 records are handled.

The listing is broken down into separate parts, each with a clear description of the operation, structure and the problems/extensions.

If you have limited programming experience and want to understand data access, especially on disc, then you will find this very informative.

Overall this is a very useful and well-written book from someone with practical experience at the grass roots level.

It starts at a very fundamental level and provides an excellent, and occasionally amusing, introduction to an area with much present and future educational potential.

Norman Parr

One for the DOL scheme

Handbook of Primary Education and Computing

By D.W.W. Ellingham

Castle House Publications

"... WHEN the total learning situation is considered, the computer will only be used when it is the best medium for the task in hand ..."

"... Computer programs are nothing more than resources for the teacher's consideration. They are not teacher substitutes ..."

ANY book on education and computers which includes statements such as these has got to be good. Handbook of Primary Education and Computing

HANDBOOK OF PRIMARY EDUCATION AND COMPUTING

D.W.W. Ellingham



does – and is excellent!

The author begins with an overview of the computer in the primary school – and moves on to examine the day-to-day management of the computer within the school and classroom, the creation of a software library, and the supervision of the children and equipment.

Most of the rest of the book is devoted to the description of five different categories of educational programs and the examination of 15 particular programs and their use within the infant, lower junior and upper junior age groups.

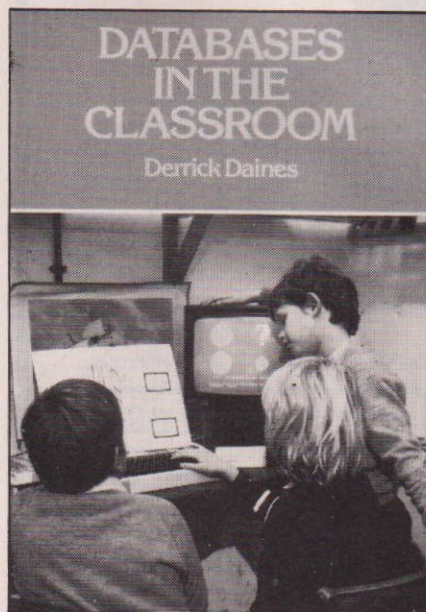
Some of the programs are already well known to many primary-school teachers (Granny's Garden, Animal, Crash, and Computer Concepts' Logo II, for example.)

But the detailed descriptions, suggestions for use, ideas on how to organise the class, and lists of supporting materials will help many teachers – both those with experience and those relatively new to computers in schools – to get a great deal more out of the programs.

For those teachers who are looking for new programs, a chapter on evaluating software is included with a very useful check list.

The book is a handbook of primary education and computing, and if I had to make a criticism it would be that in his efforts to make teachers feel at ease with the computer the author has omitted to include an adequate description of the available hardware and the opportunities it offers.

This is a pity as more hardware, such





DRAGON WORLD

Beyond GRANNY'S GARDEN is:

For children aged 6+ Dragon World will provide the basis for a whole-class project to last for many weeks encouraging activities right across the curriculum.

The package includes:-

- an audio tape of the exciting story and original music for dance/drama
- a two-part adventure
- four supplementary programs including a music synthesizer
- an illustrated story book
- a comprehensive User's Manual
- a Book of Ideas containing 100+ suggestions for class activities
- illustrated record sheets
- Concept keyboard overlay

BBC B version

Available during March

£17.65 + VAT for Disc

RML version in preparation

£16.00 + VAT for Cassette

Add £3.50 for 3" Disc.

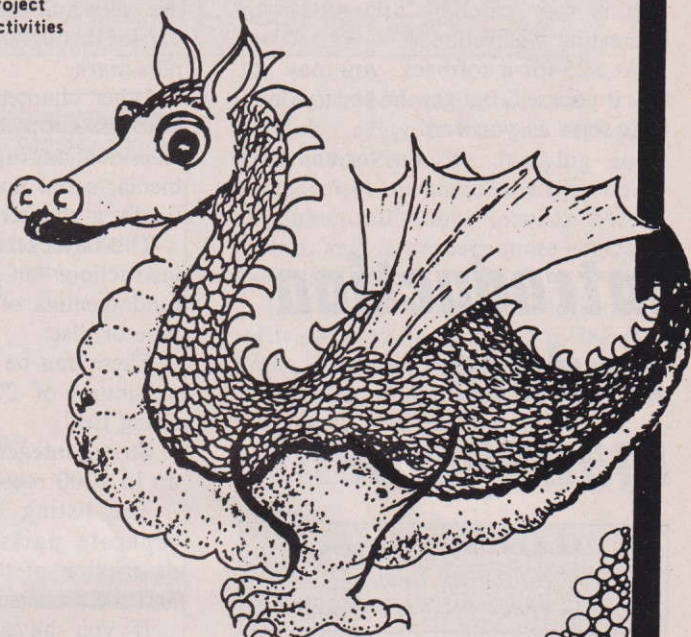
AMATION EDUCATIONAL RESOURCES

LINDEN LEA

ROCK PARK

BARNSTAPLE N.DEVON

EX32 9AQ TEL: (0271) 45566



FLOWERS OF CRYSTAL

Two part adventure, introductory dramatised audio story, work sheet, utility, map, superbly-illustrated story booklet, teacher's manual.

Flowers of Crystal, our No. 1 of 1984, was judged to be certainly original, very well presented, marvellously adaptable and unique in its scope.

*Educational Computing
Jan. '85*

TELE-BOOK

An extremely flexible menu-driven program which allows 'books' of up to 15 pages of text and graphics to be created on the BBC B micro. Ideal for primary school use.

ALSO

Our two well established children's adventures:

**GRANNY'S
GARDEN

and

SPACEX**



LINDEN LEA ROCK PARK
BARNSTAPLE DEVON EX32 9AQ

TEL: (0271) 45566

SEND A4 SAE FOR FULL DETAILS

	BBC Disc	BBC Cassette	Spectrum Cassette	RML Disc
Flowers of Crystal	£17.65	£16.00	—	Available Later
Tele-Book	£17.65	£15.00	—	—
Granny's Garden	£12.00	£10.00	£10.00	£12.00
Spacex	£10.40	£8.65	£8.65	—

Prices include P & P.

Add 1. VAT on all items.

2. £3.50 for 3" disc.

(£7.00 for TELE-BOOK 3" Disc)

From Page 37

as disc drives, becomes available for use in primary schools.

It is obvious that the author, David Ellingham, a headmaster for 17 years and now curriculum development coordinator at a primary education computer centre, has his feet firmly planted on the ground when it comes to computers in the primary school.

And there is no doubt that he writes from first-hand experience.

His honest comments on teachers and computers, software and classroom organisation make refreshing reading.

The book is excellent value and should form a part of the staff library in every primary school.

In fact the DOI scheme would have well benefited from the inclusion of this publication in its primary school package.

Frank Jukes

This will stimulate...

Writing Educational Programs for the BBC and Electron

By Dave Carlos and Tim Harrison
Macmillan (Papermac Computer Library)

THIS is the most stimulating book I have seen on writing educational software.

Indeed, if many of the "professionals" in this field were to heed the advice it offers, those of us who review their products would spend that much less time grumbling about deplorable standards.

The book is not aimed, however, at so-called experts, but rather at parents and teachers with some programming experience who feel they might do better themselves.

The basic approach has therefore been twofold:

First, to explore the differences between good and bad programming, and the knock-on educational value of doing things properly.

Secondly, to provide a library of adaptable program modules. These time-saving building blocks range from input and error-trapping procedures to graphics and other screen routines.

They are all carefully documented, with advice not only on how they can best be modified and extended, but – equally essential – when to avoid them for fear of lapsing into "bottom-up"

programming habits, and losing sight of the program's overall structure.

There are also two chapters covering some aspects of educational theory and its practical application in software design. There are excellent sections on debugging, testing, refinement, and byte-packing, and some sample programs with detailed explanations of the techniques incorporated.

In so many books on programming, such information tends to be either dreadfully boring – except to fanatics – or irritatingly pepped-up with a facile humour masquerading as "friendliness".

Not here. The authors of this book achieve a balance between a high density of information on the one hand, and readability and interest on the other.

I have only two criticisms. There is no index, despite the promise of one in the list of contents – a typographical error, the publishers tell me. And the Glossary of Terms seems to have been drawn up quite haphazardly, something which contrasts strangely with the attention to detail found in the rest of the book.

But such things hardly detract from its importance in the area of educational software, not only for Acorn machines but, by extension, for other micros as well.

And at £8.95, its 266 pages represent exceptionally good value for money.

Gabriel Jacobs

Lacking in BBC lore

Microcomputers in Education 2

By E. Ramsden (editor)
Ellis Horwood Ltd.

WHENEVER I receive a book for review, I generally flick through a few pages to try to gauge the style and content of the book, before the thorough read-through that each book obviously deserves.

Oh boy! This is not the kind of book you can skip through, as it requires a great deal of attention on the part of the reader.

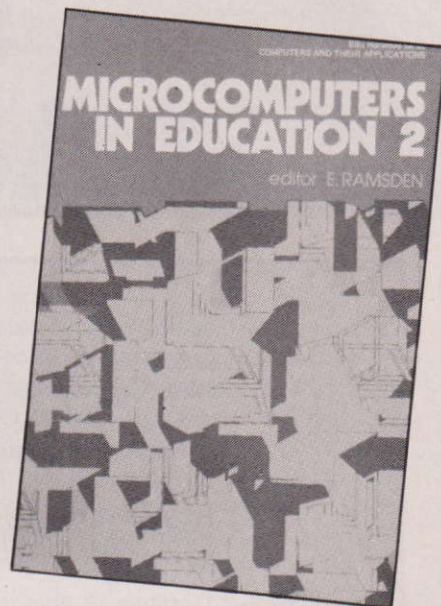
Not a little intelligence is called for also, with such phrases as "In a recursive manner, the project consists in the elaboration of an environment..." littered around the book, and just waiting to trip off the tongue.

The aim of the book is reasonably simple – to show how newer and more sophisticated languages such as Logo,

Comal and Prolog are better tools in education than Basic, education being defined by this book from the age of eleven to adult.

While few people would question the fact that Basic has several faults, the cornerstones of the arguments put forward in this book are a little fallacious.

The various authors who have each written their own chapter all have considerable experience on various Commodore machines, as were in vogue years ago, but show a rather



alarming lack of any familiarity with the BBC Micro, which surely has the lion's share of the educational field now.

I could cite many examples of this, but will limit myself to just one. Much emphasis is placed on the "fact" that only the first two letters of a variable name are relevant, so that *day* and *date* would be confused.

Perfectly true oh Commodore machines, of course, so go and buy a BBC!

Indeed, I found the absolutely endless references to Commodore machines to be monotonous in their despair.

For all that, it is a well-written and knowledgeable volume, written as it is by many people who were around computers before computers were fashionable.

A little back-slapping is evident, but their experience remains relevant.

Listings and screen dumps are well in evidence, and help the more casual reader.

Perhaps the area where this book would be most useful would be to teachers and lecturers in secondary, further and adult education. I cannot see the average home user or student gaining much from it.

Phil Tayler

TORTAL



By
**DAVID
EARLE**

Racing to success in the Addition Stakes

VARIABLES

- | | | | |
|------------------|--|----------------|--|
| A% | Setting of accumulator in PROChigh. | N% | Loop variable in PROChigh. |
| AS | Answer to question <i>J%</i> as a string. | Note% | Loop variable for number of notes in PROctune. |
| BS | String of spaces used for deletion of symbols and figures in PROCexercise. | P% | Number of steps for green tortoise in PROCexercise. |
| C% | Number of steps for red tortoise in PROCexercise. | Pitch% | Pitch of notes in PROctune. |
| D% | Horizontal coordinate of green tortoise. | QS | Question string formed using <i>F\$</i> and <i>SS</i> in PROCexercise and sent as a parameter to PROChigh. |
| D | Ascii code of string component used in PROChigh. | R% | Numerical value of correct answer to question in PROCexercise. |
| Duration% | Duration of note in PROctune. | S% | Number of steps taken in PROCexercise. |
| E% | Colour number passed as a parameter for the shell to PROCleft and PROCright. | SS | String containing the second number of the question in PROCexercise. |
| Envelope% | Envelope number used to produce notes in PROctune. | SBS | String containing the number of symbols equal to the second number in the question used if necessary in PROCexercise. |
| F% | General loop variable used to produce grass and sky and so on in PROCscreen. | Sec | Number of seconds delay in PROCdelay. |
| F\$ | String containing first number in the question in PROCexercise. | TS | Temporary string in PROChigh. |
| FBS | String containing the number of symbols equal to the first number in the question used if necessary in PROCexercise. | T% | Counter for number of times a sum has been displayed using symbols. |
| G% | Horizontal coordinate of printing in PROChigh. | U% | Vertical coordinate of tortoise in PROCleft and PROCright. |
| H% | Vertical coordinate of printing in PROC-high. | US | Temporary string containing single characters of <i>QS</i> in PROChigh. |
| H\$ | String to be converted to double height lettering in PROChigh. | V% | Flag set to 0 or 1 to determine whether or not symbolic display is operative and governing the value of <i>W%</i> . |
| I% | Pitch of sound for tortoise steps in PROCexercise. | Volume% | Loop variable for volume of sound in PROctitle. |
| J% | Numerical value of answer input in PROCexercise. | W% | Number of seconds the program will wait for a key press in PROCexercise and controlled by whether display is symbolic or figures only. |
| K% | Colour number passed as a parameter for the body to PROCleft and PROCright. | X% | Setting of X register in PROChigh. |
| L% | Horizontal coordinate of red tortoise. | Y% | Setting of Y register in PROChigh. |
| M% | Vertical coordinate of flower head in PROCending. | Z% | Value of horizontal TAB position of sum displayed during symbolic display and calculated in PROCexercise. |
| Munch% | Loop variable for munches taken by tortoise in PROChigh. | | |

THIS program was written in response to a request from an infants' school headmistress for a very simple program to allow children to practice newly learned addition skills.

It will be necessary to explain verbally what the children have to do as written instructions would probably require a higher reading age than the group the program is designed for.

The screen shows a tortoise track with two tortoises preparing to sprint(?) towards two poppies. An addition sum is displayed in white beneath the track in double height figures.

The program will wait for a predetermined time for a key press and, if no answer is entered, the program will proceed with no answer. This time can be altered to suit individual requirements.

As the correct answer cannot exceed nine a single key press will display a numerical answer in the correct place as a double height figure. If a key other than 1 to 9 is pressed it is assumed to be no answer.

A correct answer will turn the figures in the sum and the answer green and cause the green tortoise to move forward two steps, with the red tortoise moving only one.

An incorrect answer will turn the figures red and give the red tortoise two steps and the green tortoise one.

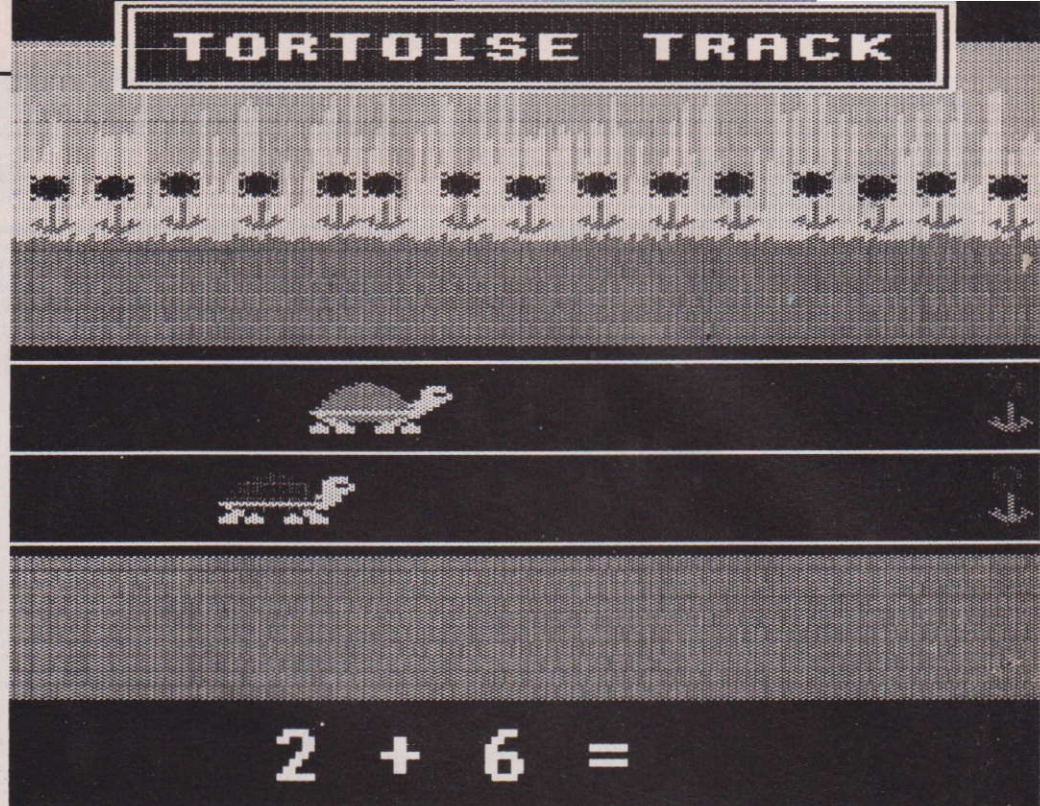
No tortoise can establish a very large lead, so constant attention is required to win, while a few wrong answers do not mean that the race cannot be won!

If an incorrect answer is given the sum is displayed again after the movement of the tortoises with the appropriate number of symbols above the figures to give the child a chance to actually count the symbols on the screen. The number of times the same sum is displayed can be altered to suit individual requirements.

The reward for winning the race is a flashing "YOU WON!" in the grass and a tune played. The result of losing is comparatively unattractive thus, hopefully, minimising the temptation to enter incorrect answers on purpose! This particular problem has been noted by some teachers and is obviously undesirable.

There are 33 questions in DATA at lines 1280 to 1360 which, together with a random RESTORE at line 170 ensure that the same set of questions are not normally used for consecutive attempts.

Escape is trapped and Break will only cause a re-run of the program. The program is deliberately simple yet holds



the interest of the age group it is intended for.

While entering the program it is a nuisance to have Escape and Break trapped, auto repeat off and the cursor keys disabled. To avoid this enter a line

55 GOTO 100

before setting AUTO to enter the program. This will give you full control for editing purposes but remember to remove the line when you have the program working properly.

The length of time the program will wait for a key press is controlled at line

290. The value of *W%* is the number of seconds the program will wait and there are two figures involved. The value of 30 in line 290 gives a waiting time of 30 seconds when the question is displayed with symbols and figures and the value of 10 gives a waiting time of 10 seconds then the question is displayed with figures only. Either or both of these figures can be altered.

The number of times a question is displayed with symbols can be controlled in line 600 by altering the value *T%* is compared with.

PROCEDURES

delay	Produces a delay of the number of seconds passed as the parameter.
define	Defines characters and envelope used in the program.
ending	Produces ending of race in accordance with which tortoise has won.
exercise	The main procedure which actually controls the tortoise race calling PROCleft and PROCright as required.
high	Produces double height display of string passed as a parameter at the TAB positions sent as parameters. Used in PROCexercise and PROctitle.
left	Produces tortoise graphics with left legs together in colours and at position passed as parameters. Used to blot out graphics by sending background colour as both colour parameters. Used in PROCexercise.
right	Produces tortoise graphics with right legs together. Otherwise as PROCleft.
screen	Produces main screen display.
title	Produces title display, calling PROctune when required.
tune	Plays tune in PROctitle and PROCending.



From Page 41

```
10 REM TORTAL
20 REM by DAVID EARLE from an idea
by DAVID WARD
30 REM (C) Micro User 1985
40
50 MODE2:VDU23;8202;0;0;0;
60 *FX11,0
70 *FX4,1
80 *FX220,0
90 *KEY10 OLD!MRUN!M
100 PROCdefine:PROCTitle:PROCscreen
110
120 REPEAT
130 PROCexercise:PROCending
140 UNTILFALSE
150
160 DEFPROCexercise
170 RESTORE(1200+RND(8)*10):VZ=0
180 LX=32:DX=32:IX=140
190 PROCright(1,3,LX,480):PROCleft(
2,3,DX,580)
200 REPEAT
210 COLOUR9:TX=0
220 READF$,S$,RX
230 Q$=F$+" "+S$+" = ":IFQ$="2 +
1 = "THENRESTORE1200
240 VDU19,9,0,0,0,0
250 VDU4:COLOUR120
260 PROChigh(B$,0,26):PROChigh(Q$,5
,26)
270 VDU19,9,7,0,0,0
280 *FX15,1
290 IFVZ=1THENWX=30ELSEWZ=10
300 JZ=INKEY(WZ*100):IFJZ<40 THEN J
Z=JZ+16
310 COLOUR130:PRINTTAB(0,23)B$:COLO
UR120
320 JZ=JZ-48
330 IFJZ>0 AND JZ<10 THEN A$=STR$(J
Z) ELSE A$=" "
340 IF VZ=1 THEN 360
350 PROChigh(A$,13,26):GOTO370
360 PROChigh(F$+" "+S$+" = "+A$,Z
X,26):VZ=0
370 IFJZ=RXSOUND1,1,150,1:SOUND2,1,
150,1 ELSE SOUND1,2,300,20:SOUND2,2,3
00,20:SOUND3,2,300,20
380 IFJZ=RXCX=1ELSECX=2
390 IFJZ=RXPX=2ELSEPX=1
400 IFJZ<>RXVDU19,9,1,0,0,0 ELSEVDU
19,9,2,0,0,0
410 SZ=0
420 REPEAT
430 VDU5:PROCdelay(1)
```

```
440 PROCright(0,0,LX,480):PROCleft(
0,0,DX,580)
450 LX=LX+56
460 DX=DX+56
470 IFLX-DX>56THENPX=2
480 IFDX-LX>56THENCX=2
490 PROCleft(1,3,LX,480):PROCright(
2,3,DX,580)
500 SOUND0,-15,3,5:SOUND1,0,IX+10,5
510 SX=SX+1:PROCdelay(0.4)
520 PROCleft(0,0,LX,480):PROCright(
0,0,DX,580)
530 IFSX<CXLX=LX+56
540 IFSX<PDX=DX+56
550 PROCright(1,3,LX,480):PROCleft(
2,3,DX,580)
560 SOUND0,-15,3,5:SOUND1,0,IX,5
570 IX=IX+2
580 UNTILSZ=1
590 IFLX>960 OR DX>960 THEN 780
600 IFJZ=RX OR TX>2 THEN 780
610
620 VX=1:ZX=VAL(F$)*2-1:TX=TX+1
630 VDU4:PROChigh(B$,0,26)
640 VDU19,9,7,0,0,0
650 FB$=STRING$(VAL(F$)," "+CHR$237
)
660 SB$=STRING$(VAL(S$),CHR$237+" "
)
670 COLOUR130:COLOUR0
680 PRINTTAB(0,23)FB$
690 COLOUR120:COLOUR9:PROChigh(F$,Z
X,26)
700 PROCdelay(1)
710 PROChigh(F$+" "+ZX,26)
720 PROCdelay(1)
730 COLOUR130:COLOUR0
740 PRINTTAB(0,23)FB$+" "+SB$
750 COLOUR120:COLOUR9
760 PROChigh(F$+" "+S$+" = ",ZX,26
)
770 GOTO280
780 UNTILLX>960 ORDZ>960
790 IFDX=LXTHENPROCright(0,0,LX,480
):PROCleft(0,0,DX,580):DX=1032:LX=103
2:PROCright(1,3,LX,480):PROCleft(2,3,
DX,580)
800 IFDX>LXTHENPROCright(1,3,LX,480
):PROCleft(0,0,DX,580):DX=1032:PROCle
ft(2,3,DX,580)
810 IFLX>DXTHENPROCright(0,0,LX,480
):PROCleft(2,3,DX,580):LX=1032:PROCri
ght(1,3,LX,480)
820 ENDPROC
830
840 DEFPROCleft(EX,KX,LX,UX)
850 MOVEZX,UX:GCOL0,EX
```

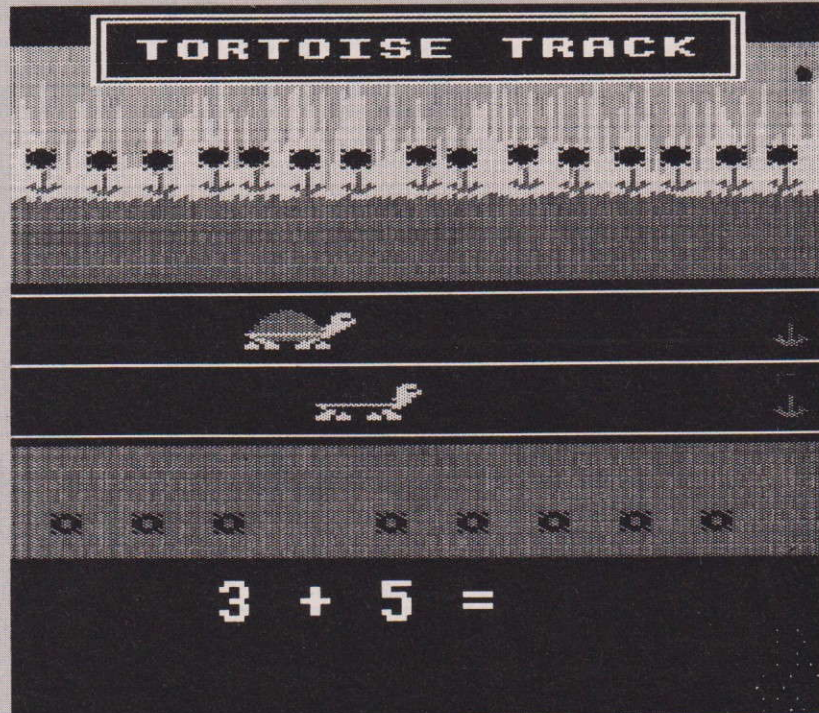
```
860 VDU224,225,10,0,0,226,227
870 GCOL0,KX
880 VDU8,8,232,233,235,11,0,234
890 ENDPROC
900
910 DEFPROCright(EX,KX,LX,UX)
920 MOVEZX,UX:GCOL0,EX
930 VDU224,225,10,0,0,226,227
940 GCOL0,KX
950 VDU8,8,228,229,231,11,0,230
960 ENDPROC
970
980 DEFPROCdefine
990 VDU23,224,&0000;&0701;&1F0F;&3F
1F;
1000 VDU23,225,&0000;&F0C0;&FCFB;&FE
FC;
1010 VDU23,226,&7F3F;&3F00;&000F;&00
00;
1020 VDU23,227,&FFFE;&FE00;&00FB;&00
00;
1030 VDU23,228,&0000;&40FF;&7430;&EE
EE;
1040 VDU23,229,&0001;&01FF;&1706;&3B
3B;
1050 VDU23,230,&0000;&7000;&C8F0;&F0
F0;
1060 VDU23,231,&E0E0;&80C0;&0000;&80
80;
1070 VDU23,232,&0000;&40FF;&5C30;&EE
EE;
1080 VDU23,233,&0001;&01FF;&1D06;&3B
3B;
1090 VDU23,234,&0000;&3800;&6478;&F8
7C;
1100 VDU23,235,&E0F0;&80C0;&0000;&80
80;
1110 VDU23,236,&1010;&1010;&D654;&3B
7C;
1120 VDU23,237,&78B4;&CCCC;&78CC;&00
B4;
1130 VDU23,238,&0000;&3030;&0030;&00
80;
1140 ENVELOPE1,4,0,0,0,0,0,126,-5,
-3,-1,126,110
1150 ENVELOPE2,3,-4,-1,2,6,6,28,81,-
4,-5,-1,126,63
1160 B$=STRING$(20," ")
1170 ENDPROC
1180
1190 DEFPROChigh(H$,GZ,HX)
1200 XZ=0:VZ=13:AZ=10:D=&D00
1210 T$=CHR$(240)+CHR$8+CHR$10+CHR$(
241)
1220 FORNZ=1 TOLEN(H$)
1230 U$=MID$(H$,NZ,1):?D=ASC(U$):CAL
L&FFF1
```




```

1240 VDU23,240,D71,D71,D72,D72,D73,D
?3,D74,D74,23,241,D75,D75,D76,D76,D77
,D77,D78,D78
1250 PRINTTAB(6X+MX-1,HZ)T$:NEXT
1260 ENDPROC
1270
1280 DATA2,2,4,3,3,6,4,4,8,1,3,4
1290 DATA5,4,9,1,8,9,6,2,8,4,2,6
1300 DATA1,2,3,2,3,5,3,4,7,2,4,6
1310 DATA4,5,9,5,3,8,1,1,2,6,3,9
1320 DATA1,3,4,3,5,8,4,3,7,1,4,5
1330 DATA2,5,7,3,6,9,1,5,6,2,6,8
1340 DATA4,1,5,1,6,7,5,1,6,2,7,9
1350 DATA3,2,5,3,1,4,6,1,7,1,7,8
1360 DATA2,1,3
1370
1380 DEFPROCscreen
1390 GCOL0,7
1400 MOVE0,496:DRAW1280,496
1410 MOVE0,396:DRAW1280,396
1420 MOVE0,596:DRAW1280,596
1430 GCOL0,2
1440 FORFX=380 TO224 STEP-4:MOVE0,FX
:DRAW1280,FX:NEXT
1450 FORFX=618 TO738 STEP4:MOVE0,FX
:DRAW1280,FX:NEXT
1460 GCOL0,6
1470 FORFX=742 TO952 STEP4:MOVE0,FX
:DRAW1280,FX:NEXT
1480 MOVE0,742
1490 GCOL0,3
1500 FORFX=0 TO1280 STEP8:DRAWFX,742
+RND(160):DRAWFX+4,742-RND(10):NEXT
1510 FX=12
1520 VDU5
1530 REPEAT
1540 MOVEFX+RND(32),776-RND(8)
1550 FX=FX+84
1560 VDU18,0,2,236,11,8,18,0,1,237,8
,18,0,0,238
1570 UNTILFX>=1240
1580 GCOL0,4:MOVE128,992:MOVE128,896
:PLOT85,1152,992:PLOT85,1152,896
1590 GCOL0,7:MOVE128,992:DRAW128,896
:DRAW1152,896:DRAW1152,992:DRAW128,99
2
1600 MOVE144,984:DRAW144,984:DRAW113
6,984:DRAW1136,984:DRAW144,984
1610 COLOUR7:COLOUR132:VDU4
1620 PRINTTAB(3,2)"TORTOISE TRACK"
1630 VDU5:MOVE1280,448:VDU18,0,2,236
,11,8,18,0,1,237:MOVE1280,548:VDU18,0
,2,236,11,8,18,0,1,237
1640 VDU19,8,2,0,0,0
1650 VDU4:COLOUR8:COLOUR130:PRINTTAB
(6,11)"YOU WON!":VDU5
1660 VDU19,9,7,0,0,0:COLOUR128:COLOU

```



```

R9
1670 PROCdelay(1)
1680 ENDPROC
1690
1700 DEFPROCending
1710 VDU4:COLOUR130:PRINTTAB(0,23)B$
:COLOUR128:PROChigh(B$,0,26):VDU5
1720 IFLX=DXTHEN1830
1730 FORMunchX=0TO1
1740 SOUND0,-15,5,2
1750 SOUND0,-15,4,1
1760 SOUND0,0,0,5
1770 NEXT
1780 IFLX>DXTHENMX=480 ELSEMX=580
1790 MOVE1280,MX:VDU18,0,0,237
1800 IFDX>LXTHEN1810ELSE1830
1810 VDU19,8,8,0,0,0
1820 PROCtune
1830 PROCdelay(3):VDU19,8,2,0,0,0
1840 PROCright(0,0,LX,480):PROCleft(
0,0,DX,580)
1850 MOVE1280,448:VDU18,0,2,236,11,8
,18,0,1,237:MOVE1280,548:VDU18,0,2,23
6,11,8,18,0,1,237
1860 ENDPROC
1870
1880 DEFPROCtune
1890 RESTORE1940
1900 FORNoteX=0TO20
1910 READEnvelopeX,PitchX,DurationX
1920 SOUND1,EnvelopeX,PitchX,Duratio
nX

```

```

1930 NEXT
1940 DATA1,73,8,0,0,2,1,53,8,0,0,2,1
,73,8,0,0,2,1,53,8,0,0,2,1,81,8,1,89,
4,1,93,4,1,89,4,1,81,4,1,73,16,0,0,2,
1,81,8,1,89,4,1,93,4,1,89,4,1,81,4,1,
73,16
1950 ENDPROC
1960
1970 DEFPROCtitle
1980 VDU24,128,448,1072,768;
1990 GCOL0,132:GCOL0,7:CL0:COLOUR7:C
OLOUR132
2000 MOVE128,448:DRAW128,768:DRAW107
2,768:DRAW1072,448:DRAW128,448
2010 MOVE152,464:DRAW152,752:DRAW104
8,752:DRAW1048,464:DRAW152,464
2020 PROChigh("T O R T A L",4,12)
2030 PROCtune:PROCdelay(2)
2040 FORVolumeX=0 TO15 STEP-1
2050 SOUND0,VolumeX,4,1
2060 NEXT
2070 MOVE272,500:DRAW768,500
2080 MOVE776,560:VDU5,224,225,10,8,8
,226,227,8,8,232,233,235,11,8,234
2090 PROCdelay(2):VDU26:GCOL0,128:CL
6
2100 ENDPROC
2110
2120 DEFPROCdelay(Sec)
2130 TIME=0
2140 REPEATUNTILTIME=Sec+100
2150 ENDPROC

```


EARLY WORDS BBC & Electron £7.95 tape £9.95 disc

A package of 6 colourful and entertaining programs on one tape or disc. The 6 programs cover a series of topics which each include full colour scenes to teach the names and spelling of a large number of common objects in an easy, attractive and enjoyable way.

Age guide 3-6 years

EARLY MATHS BBC & Electron £7.95 tape £9.95 disc

A package of 4 programs on one tape or disc which cover basic numeracy.

The 4 programs introduce early learning and understanding of simple addition, subtraction, multiplication and division using a variety of animated routines to help the child in an instructive and entertaining way.

Addition & Subtraction
Multiplication & Division

Age guide 4-7 years
Age guide 5-8 years

TYPING TUTOR BBC Model B only £12.95 tape £14.95 disc

A complete Typing Tutor – for novices or those wishing to improve their skills.

Ideal as an introduction to typewriting or word processing.

Learn to touch-type easily and gain confidence in the use of the keyboard.

On-screen display of both keyboard and lesson with speeds – accuracy – time taken – full self-test facilities.

Comprehensive Book of Instructions with hints and additional information.

*(Electron version due in May) *

Typing Tutor



Willow Software BBC B

The programs are available from many good software shops or send cheque/P.O. to:
(Send for full list of programs available)
(Dealer and Distributor enquiries welcome)

Willow software

The Willows, Wrington Lane
Congresbury, Bristol
(0934) 834056

MONKEY

AS SEEN ON TV

Monkey is full of fun and magic. Monkey provides the basis for whole-class project. Monkey encouraging class activities.

For children age 7+

The package includes:

A two-part adventure program + story.

An illustrated story book.

A comprehensive teacher manual + a lot of suggestions for class activities.

A Monkey poster.

BBC B Version now available

£16 + VAT for Disc (Price includes P&P)

£15 + VAT for Tape (Price includes P&P)

RML and Commodore 64 version in preparation.

C & C Co.

9 Ettrick Place, Ayr KA8 9HH

Tel: 0292 289470 or

Tel: 0292 74241.

We also welcome official government or school orders.

Mail Order please fill in coupon with your remittance.



To: C & C Co.

9 Ettrick Place, Ayr KA8 9HH

Please send me _____ copies of Monkey Disc Package at £16 + VAT (£18.40)

Please send me _____ copies of Monkey Tape Package at £15 + VAT (£17.25)

I enclose a cheque/Postal Order for £ _____

Name _____

Address _____

_____ P.Code _____

Sig. _____ Date _____

EC

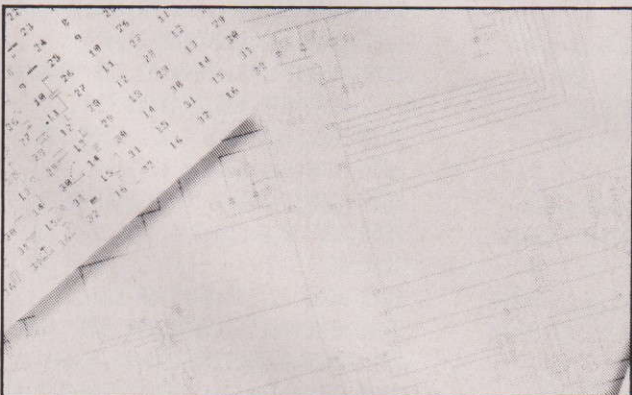
Datapen

CIRKWIK

Datapen

SCHEMATIC DRAWING ON THE BBC MICRO

A CAD package orientated to the production of schematic drawings, such as circuit diagrams, flow charts, pipework diagrams, fluid logic diagrams and many similar professional and engineering applications.



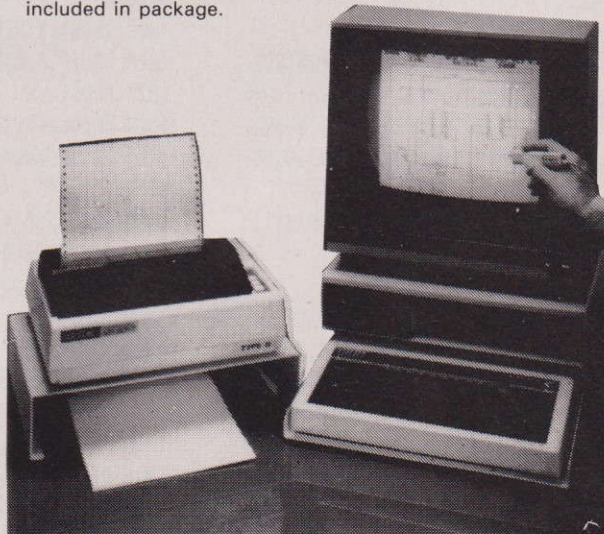
CIRKWIK Version for lightpen only **£19.95**
CIRKWIK Version for trackerball or lightpen **£24.95**
CIRKWIK Version for grafpad or lightpen **£24.95**

HARDWARE:

DATAPEN Lightpen (introductory programs included) **£25.00**
MARCONI Trackerball (Micro-Draw program included) **£59.50**
or with new icon based program included **£69.50**
ATARI Trak-Ball c/w with user port interface **£33.00**
User port/Atari Track-Ball interface only **£13.00**

Our prices already include VAT & Delivery
S.A.E. for details of lightpen, CIRKWIK and other programs.

- ★ Lightpen or trackerball driven.
- ★ Virtual screen 8 x the BBC's mode 4.
- ★ Uses standard dot matrix printer in dual-density graphics mode to produce excellent quality diagrams.
- ★ Automatic parts list generation.
- ★ Up to 640 different symbols may be in use in any one diagram.
- ★ Total symbol library unlimited in size.
- ★ Create your own symbols on a highly magnified scale.
- ★ Many electronic symbols already included in package.



DATAPEN MICROTECHNOLOGY LTD. Dept ED, Kingsclere Road, Overton, Hants RG25 3JB Telephone: (0256) 770488

DRAW THE EASY WAY

TWO programs, **Technical Drawing**, from Jabtech, and **MICAD**, from Ellis Horwood, have very similar areas of application – the production of simple “wireframe” line drawings of three-dimensional objects, displayable in either pictorial form or in orthographic projection (front, side and plan views).

Technical Drawing has been developed from a BBC National Schools Software Competition prizewinner and enables the user to simulate, at a simple level, the process of “Solid Modelling”.

This is a Computer Aided Design (CAD) process used on industrial mainframe computers which allows the designer to build a complicated object up from a number of simpler solids such as cubes, prisms and pyramids.

Unfortunately there are no provisions for curved surfaces like cylinders or spheres.

The program has a number of menus to step through (some of which, rather irritatingly, repeat themselves) and makes extensive use of the function keys.

The screen is laid out in an imaginative manner, framed, with a quarter-screen column on the right which gives relevant information and incorporates an isometric drawing of a cube.

This is a representation of the permitted three-dimensional working area.

The object that is being drawn is shown in miniature inside this cube, allowing the user to position drawn objects relative to others.

The big snag is that actually drawing items is not easy.

The user of this program has to be very familiar with its idiosyncracies and must keep a close eye on X,Y,Z coordinates.

Drawing is by specifying a direction and length – no “rubber band” trial and error here.

It is easy to delete a line immediately, but impossible to edit after completion.

However, when completed, it is a simple matter to reposition objects, copy them and change their size or attitude.

The example that comes with the program is a jet plane – and the wings, tail and tailplanes are all derived in this way from the same simple wedge shape.

Both detail and assembly

drawings are saved on to a library disc and finding them is very easy indeed.

The use of a cursor steps through the name and drawing number of each object on the screen, until selection is made by Return.

Once the drawing is loaded into memory all sorts of things can be done.

Pictorial projection is in isometric only, although the object can be rotated in ninety-degree stages around any of the three X,Y,Z axes.

It can be displayed as single

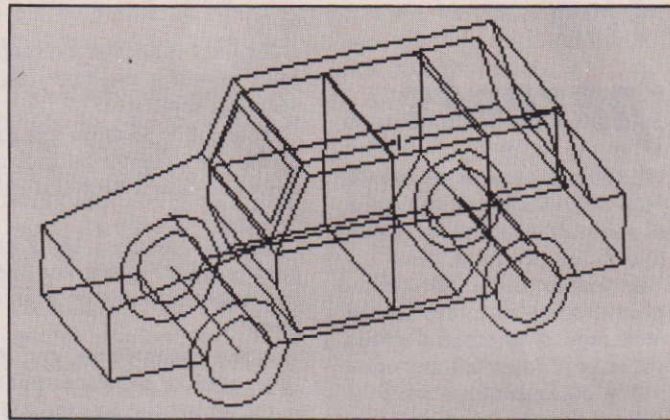
simple but very effective.

It allows a variety of line types (all to BS 308), circles and arcs in four different modes, text insertion and a grid background.

The best bits are that it can automatically put dimensions on a drawing, hatch cross-sections and “zoom enlarge” areas for fine drawing.

Output from both parts is by dot matrix screen dump or, I am pleased to see, via Penman or Hewlett Packard plotters.

MICAD 3 is very much easier to draw with than Technical Drawing and allows any of the



A car – using the MICAD program

front or side elevations, or as a plan view only.

It can also be displayed as a traditional three-part orthographic drawing with choice of first or third angle projections (complete with relevant British Standard 308 symbols). This is especially useful for teaching as the small cube still displays the object in isometric.

Technical Drawing is a very clever piece of programming and I can see a number of uses for it in the teaching situation.

It is, however, rather fiddly and takes a lot of getting familiar with – it is not very user-friendly, unlike MICAD.

MICAD comes in two parts, MICAD 2 and MICAD 3 – the numbers refer to two and three dimensional drawing.

MICAD 2 is a straightforward drafting package for drawing orthographic plans and elevations.

As these programs go, it is

four elevations or the plan to be displayed individually.

These are all called by their own function keys as are also three useful, different pictorials.

Otherwise the user has to specify particular views in terms

of angular rotation around the three axes in succession – and these are difficult to predict.

Drawing is very simple, using the cursor keys for X and Y axes and the Q and A keys for the Z axis.

Lines are “rubber banded” allowing accurate estimation and there is a very useful “Extrusion” command.

This enables the user to draw a shape in one plane, say the side of a car, and then to move “outwards” at right angles.

The command E then causes the shape to be repeated in a plane parallel to the original and all points to be joined between them.

This allows very quick wireframe “solids” to be drawn.

Editing is simple: Circles can be represented albeit rather crudely, and zoom can be employed as in the drafting package.

If the user gets “lost” in three dimensions – and this is easy with six cursor buttons – the simplest thing is to switch to an orthographic elevation and reorientate.

The drawing process can be carried on in any of the various display angles.

Apart from the fact that it does not allow objects to be built up from simple detail parts, it is an excellent program.

All in all, these are two very interesting programs, each with individual strong points.

There are some basic similarities and if I were forced to choose between them I would go for the MICAD package for its double use and its simple user-friendliness. **Terry Hallard**

Failure on two fronts

SORTING data is a common task performed by computers and any attempt to make sorting more understandable to the amateur must be welcomed.

Order, by Resource Facilities, is an investigative program which will allow you to incorporate BBC Basic listings of sort algorithms into your own programs.

The program allows you to investigate the number of exchanges carried out while sorting data that either you

choose or is chosen for you by the program.

It will measure how long sorting for a particular set of data took, displaying comparable times for any or all of the eight sort algorithms in a bar chart.

Finally it will provide a BBC Basic listing of the sort algorithm.

The program compares bubble sorting with ripple,

From Page 45

insertion, Shell/Metzner, heap and quick sorting.

Educationally it is inadequate in its use of graphics to show exactly what happens in a sort.

No difference is made between comparisons and exchanges.

The author's "illustration of switches" is merely the dumping of the partially sorted array for any pass, with the exchanged data highlighted.

The alpha sort on the BBC Welcome tape is more visually instructive.

The listings are accurate, but you would need to be a competent programmer to understand and incorporate the densely-packed code into your own programs.

The documentation provides no assistance at all. It does discuss the theory behind the sorts, but is woeful in its presentation and candidly will only be understood by the highly-gifted amateur.

It concentrates on the timed efficiency of sorts, but fails to discuss the relationship between the degree of initial order in a set of data and the time taken to sort it.

The "Merge" sort and "Radix" sort are mentioned but not dealt with because they "do not lend themselves to a display as used in option 1 of the menu".

A pity, as the BBC file handling system makes the two-way merge sort a sensible and viable option for certain data.

The Tree sort is not mentioned at all, though for keyboard entry of data combined with immediate retrieval of data it provides the speediest sort of all. Superficially,

Resource Facilities has produced a useful utility and educational program. In practice, only few will understand it.

Visually it is no substitute for the MICE Sorting Package (admittedly produced for the RML machines). Its scanty documentation fails to educate.

Either Resource Facilities must produce a genuine sort utility library package or an educational tool with full use of graphics.

This package fails because it is neither.

Ian Murray

Mr Men involves the parents

HERE and There with the Mr Men, from Mirrorsoft, is a suite of programs intended for use by four-to-five-year-olds.

It makes use of the ever-popular Mr Men to explain the meanings of direction words such as up, down, left and right.

The documentation, which is colourful and thorough, suggests, correctly, that these are programs in which parents should get involved.

The first program is a kind of jigsaw in which the aim is to move doorways so that parts of Mr Tickle can be slid through them. The reward is a complete, animated picture of Mr Tickle.

A drawback is that Mr Tickle has sixty parts - too many for

children at the younger end of the age range. Ten would have been plenty and it is a pity that no option was included for this.

The program, like the other three, runs very smoothly, if rather slowly.

The second game involves steering Mr Tickle's arm around walls and through doorways so that he may tickle Mr Grumpy.

It's great fun and has amused people from my own four-year-old daughter through teenagers to adults.

The moves are very simple being L(ef), R(igh), U(p) or D(own).

The cursor keys may also be used, but young fingers tend to stray to the Break key.

The instructions may be given one at a time, or entered as a sequence, so the program can involve forward planning. One might almost call it programming.

Mr Lazy is another charming game in which the hero sleeps under an apple tree.

Your job is to move a worm up the tree so that it can pick an apple and drop it down on to Mr Lazy.

If you succeed, Mr Lazy gobbles up the apple and then falls asleep again under a larger tree.

Again, instructions can be single steps or sequences and as in the other programs there is a sensible sound on/off option.

The final game is played on a chess board and is a variant on Fox and Geese.

You have control over four Mr Men and your aim is to trap the villainous Mr Tickle.

He's moved by the computer, but even so he seems fairly limited in brainpower and my seven-year-old can easily trap

him. He tends to beat my four-year-old.

There seem to be few programs aimed at the very young. My daughter has been desperate to play computer games, but Pacman and the rest are just too fast.

Mirrorsoft have tried hard here to fill the gap, with bright, cheerful fun games, which will also run on the Electron.

Although they are not all perfect, the package as a whole is thoroughly recommended.

Sue Frost

A DATA HELP FOR TEACHERS

REGRESSION, from Resource Facilities, is a specialist program with a limited market, but extremely useful to anyone involved in collecting and analysing experimental or statistical data.

It could also be a useful resource for schools and colleges teaching pupils and students how to interpret and process data.

This menu-driven program will accept the coordinates of up to 58 points and automatically scale and draw a graph containing them.

The information relating to each point is numbered and up to seven characters are allowed for each coordinate entry.

All entries can be edited at any stage in the program and the axes named.

When data is complete the

This is an easy retriever

RESMAN, an information-retrieval system from Resource Facilities, comes supplied on disc for £9.

The program loads on Shift+Break whereupon you are faced with a few seconds of title followed by a menu.

The first item of the list is Create File and this is used to build up the structure of the records and format the file on the disc.

Very simple prompts request

the number of fields and the length of their future contents.

This is limited to 31 characters including the field title, but entries can be split over a number of fields.

This would complicate the search facility, however.

A good feature was the ability to set default values into the records at this stage.

A blank disc is now required and will be configured as a file across one entire surface.

Reading a record in the file is a very simple affair, in line with the rest of the design concept, although some people like myself might find that there are too many prompts and verifications.

The rest of the menu deals with reading, printing and modifying the stored records.

All the facilities are very easy to use, though they suffer the usual problem of having to step through too many menus to get

at the stored data.

Saving and loading files from the computer to the disc are achieved automatically and thus it is possible to have very long files unlimited by the BBC microcomputer's memory.

All in all, this is a reasonable retrieval package for the price. The presentation of the information is tidy and this product might find its niche in schools or in the home.

Tim Craig

Simple-and super

program will draw the graph, draw the best fit line through the points and calculate the slope and intercepts of this line so that its equation can be found.

This best fit line can be switched on and off as desired and is just one of many useful aids in this program. All experimenters will admit to making a good guess rather than going to the trouble of calculating the slope and position of the best fit line!

The program also enables the user to remove the worst points from the data, but this cannot be reinstated.

Data can be plotted as the inverse or common log of the coordinate values on one or both axes.

The original coordinate values or the plotted values – that is, the coordinate, inverse or log – can be inspected at any time during the program run.

One or two problems surfaced while using the program.

Certain types of data values, caused a "divide by zero" error to occur when log values of the data were plotted.

Also, a call for a plot of the log of coordinate values sometimes resulted in a situation where a return to the full plot of the original values could not be obtained.

All in all, the package has a limited appeal for the private buyer, but could be a valuable teaching aid.

John Daddy

HOOP-LA, by Soft Centre, is one of those lovely, simple ideas which the best ideas often are.

A square is shown on the screen, which can be selected from a menu of four.

There are 0-100 and 0-1000, which keep to whole numbers, 0-10 which moves on to tenths, and 0-1 which include hundredths.

The highest and lowest values of these ranges are shown at the appropriate corners of the square so formed, and the middle value is also included.

Thus we might have a 1000 x 1000 square, with 0,500 and

1000 labelled along each side.

A demonstration mode can be viewed, if required, but the game is actually quite straightforward to use.

A point is placed on the screen, and the child has to supply the coordinates of the point marked.

Obviously pinpoint accuracy is unlikely (to begin with, at least), so there is a graded system of awarding points for the guess.

These range from 10 for Spot On, down to three points for ONLY JUST.

When the shot has been made,

standard abbreviations, so that a novice can quickly see how they should be played by typing in their names directly from sheet music.

Two function-key strips – one for the BBC, the other for the Electron – make input very simple, and there are only four commands to remember: (S)ound, (I)nversions, (R)eplay sequence and (P)rogression mode.

The simplicity of the whole program is demonstrated by the fact that the instruction booklet, though modest in size, is perfectly adequate.

Given its scope and ease of use, then, Chord Tutor ought to be looking forward to a respectable future in a small but discernible niche of the market.

At £14.95, however, it has been priced far too high, especially for a cassette-only package.

Furthermore, you'll be involved in the expense of a footswitch if you want to make full use of the Replay and Progression facilities, because you need to keep both hands free for your instrument while prompting the program for the next chord in a sequence.

To compound that, it's surprising that such a costly package is marred by trivial programming gaffes, some of which I suspect, may be due to the fact that a ZX Spectrum version (which is also available) has been too hastily adapted for the BBC Micro.

The program won't run, for instance, under BBC Basic I because semi-colons instead of

commas have been used in input statements.

There are other oversights, too. Lower-case characters are not recognised.

Repeat has not been disabled on the spacebar, causing sound and display to go out of sync in Replay mode if you hold it down for too long. The cursor often appears when it shouldn't.

I could go on...

The last few months have seen fierce competition in music software, with companies striving to wring every drop from the BBC's sound and graphics capabilities, while keeping the price down.

Chord Tutor has potential, but users have now come to expect better value for money.

Gabriel Jacobs

by entering the coordinates, a small box fits around the actual value, and the exact coordinates are shown as a direct comparison with one's own guesses.

It becomes a little easier as the ten goes are used, as each new point can be related to those previously displayed, and I can see a very real educational value in this technique.

The use of colour is modest, while the sound is not spectacular.

But there is a good-enough idea here to overcome those minor criticisms.

Phil Taylor

from a store of more than 700.

The player is then required to make up a sentence from these, which must not only make sense but must also be grammatically correct.

The program also provides a bank of common words, punctuation marks and "add ons" (such as -ing) which you can use to help create your sentence.

Once you have assembled your masterpiece, you are invited to read it to the other players.

This can often be followed by lengthy and sometimes heated arguments as to the validity of your sentence.

The computer, of course, has no say in the matter and will accept any gibberish.

The fun is in trying to persuade your opponents to agree that your offering is acceptable.

If your sentence is accepted the program awards you points depending on its length and grammatical complexity.

A time-limit option, of one to five minutes, can be selected and up to 11 extra points can be obtained depending on the speed at which you complete your task.

A further option allows you to increase the number of words from which the random selection is made by up to 12 words of your own choice.

The screen display is excellent. It is split into three sections, the top containing the random words, the bottom containing the standard bank of common words and punctuation, leaving

Discords have crept in

KEYBOARD Chord Tutor, from Aaby Associates, generates the finger positions for 144 chords and their inversions, and all major and minor scales, on a piano-keyboard screen display.

A companion program for guitarists, showing chord shapes across six frets, is available separately.

The program can sound arpeggios, has a Replay memory for chord sequences, and accesses a ready-made database of 98 common progressions.

Chords are identified by

Learning the lingo is fun!

THE attractive packaging for **Lingo** makes the promise: "The game that makes fun of the English language" – and this certainly proved to be the case in this excellent multiplayer game from Complete Microcomputer Services.

Lingo is a game, for up to four players, in which the computer provides each player in turn with 12 words chosen at random

From Page 47

the middle of the screen as your workspace.

The game is simply controlled using just two cursor keys, the spacebar, the Return key and the C key which cancels the last word entered.

The program is professionally presented and supplied with a clear, informative and well laid-out instruction booklet.

It is a very entertaining and educational game that seems to have been written for the computer, rather than adapted from an existing game.

Alan Cockman

The way to the stars...

STAR Seeker and **Solar System** (Mirrorsoft) are designed as an introduction to astronomy and as an aid and reference guide for

amateur astronomers.

Star Seeker accurately plots the most prominent 308 stars from 58 constellations in the night sky from any position on Earth, at any time in this and the next century.

You can obtain information on any individual star: its name, constellation, accurate position, brightness, distance from the Earth, and the times at which it rises and sets.

You can draw constellation maps, list those plotted and simulate the rotation of the Earth through the night.

You can even obtain a close-up of a small region of sky, to increase separation of clustered stars.

Solar System provides 16 options for the planets and Halley's Comet.

You can obtain information on their position in the sky, distances from the Sun and Earth, brightness, rise and set times.

You can also display their positions at a choice of scales, plot their orbital movements and draw the phases of the Moon.

Both programs are menu-

driven and easy to use.

You simply input the latitude and longitude of the point from which observations are to be made, the time and date, and choose your option.

Each program contains a screen dump routine for Epson FX and RX printers.

The manual is comprehensive and contains clear descriptions of the various options, instructions for use and a large number of practice examples.

Sometimes it is difficult to follow the instructions and examples because the print size is rather small.

This package is ideal for amateur astronomers, containing a wealth of useful information which is easily accessible.

It also has considerable potential as an educational visual aid in several curricular areas.

For example, it can be used to demonstrate fundamental geographical concepts to nine-year-olds, or provide a source of research data for geography or physics students.

Overall, this is an excellent product and I thoroughly

recommend it to any person with the slightest interest in the Solar System and stars.

Jim McHugh

Music for rockers

A COMBINED book and data cassette tape, **Introducing the Dots**, from Blandford Press, aims to teach rock instrumentalists to read and write music.

The starting-point for the package has rightly been that musicians who can only play by ear are at a disadvantage, not only in communicating with fellow-players, but also in understanding how music really works.

The author of the book, Dave Stewart, is a successful musician who last year reached Number 1 in the charts.

The market is clear. Stewart concentrates on chords and instrumentation specifically applicable to rock bands.

And though his approach to the nitty-gritty of notation is standard, he sugars the pill with wit calculated to appeal to young people.

Example: "... sheet music - a term given by the French to music they dislike".

In general, however, the book's serious purpose is never lost, and it is well-structured and nicely produced.

The back cover of the package proclaims an imaginative use of computer graphics and sound. I assume this means you have to use your own imagination, since the program barely scratches the surface of the BBC's capabilities.

The cassette roughly follows the model of the book but makes insufficient use of the computer as a tool in its own right, relying too much on screenfuls of text.

What's more, there are irritating format errors and unacceptable bugs, at least in the review copy I received.

I even had to break in and re-write some missing lines in two out of the five modules, just to get them running.

The book on its own can be recommended, but priced with its cell-mate at £14.95, it's hardly a steal.

Gabriel Jacobs

Appealing maths

I MUST say I learned a couple of very quick lessons from **Mastermaths Software**, from OUP.

First of all, read the instructions, but don't necessarily do exactly what they tell you without thinking carefully. And, secondly, don't always trust first impressions.

The loading instructions tell the user to "Wait until the menu appears. The program is then loaded. Loading time is about three minutes. Clear the computer memory by switching off for a few seconds".

The memory will certainly clear if you do things in that order!

The software comes on cassette as a set of four programs, one for each of the four rules of number - addition, subtraction, multiplication and division.

Each game is quite different from the others. In Addition: Buried Treasure, treasure has to be collected and the monster raced to the treasure chest at the bottom of the mine.

Balls must be caught in cups

and the joker is knocked over by their well-timed release in Subtraction: Catch the Joker.

Multiplication: Tangled Web has a fly escaping from a web while avoiding a tarantula and spider.

And finally balloons are released and burst on an arrow in Division: Air Balloons. (But don't burst the blue ones!)

The first game I looked at, Addition: Buried Treasure, was the weakest.

From the child's point of view it appeared to be not very fair as the reward rarely matched the difficulty of the problem.

There are also one or two minor bugs in the set.

First impressions were not encouraging. However, the programs were all quite entertaining and I eventually grew to like them more.

They were of the "drill and practice" or "reinforcement" type, but this should not put teachers off as they are reasonably well graded in difficulty and quite appealing to the children.

Correct answers are rewarded and incorrect answers produce help in the form of the correct answer or a "clue".

The problem is then presented again for the correct answer to be entered.

Unfortunately this routine can continue for as long as wrong answers are entered.

The graphics in the programs were clear, but the movements produced by the keys were sometimes a little too fast for younger children to cope with.

Sound is used to good effect as a prompt and - a nice touch here - the volume could be altered by the cursor keys.

The four programs do their job reasonably well but are not for home use.

The level of difficulty (six or seven levels and a demonstration, in each program) must be carefully matched to the child and a proper explanation of the way the numbers are manipulated must be given.

The programs are very much for classroom practice with fun.

F. Jukes

Ace down the line.

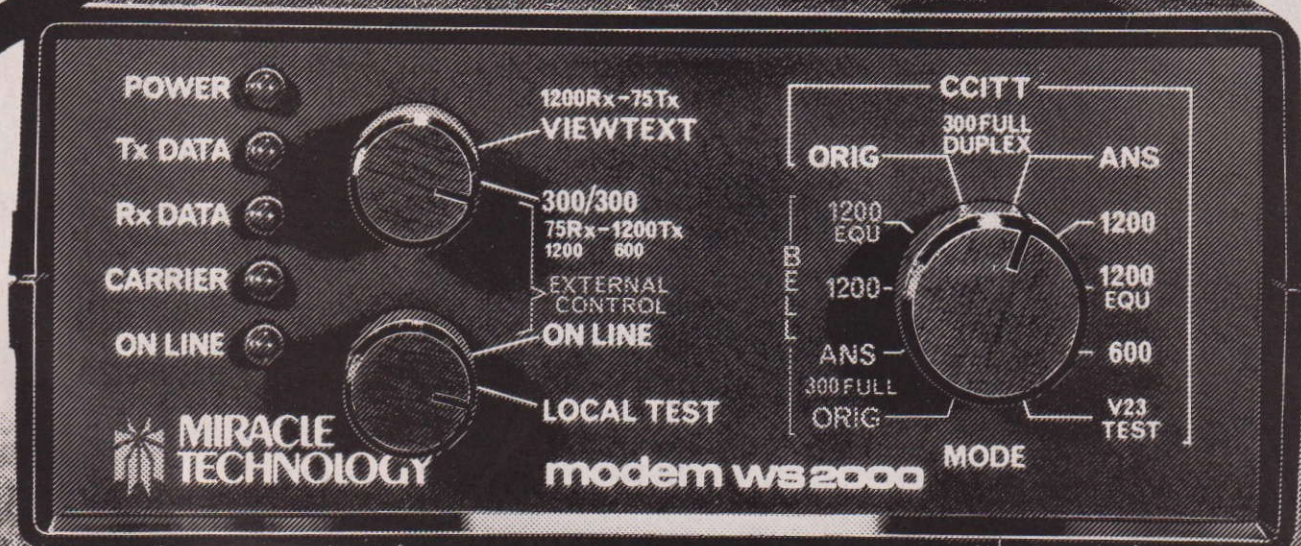
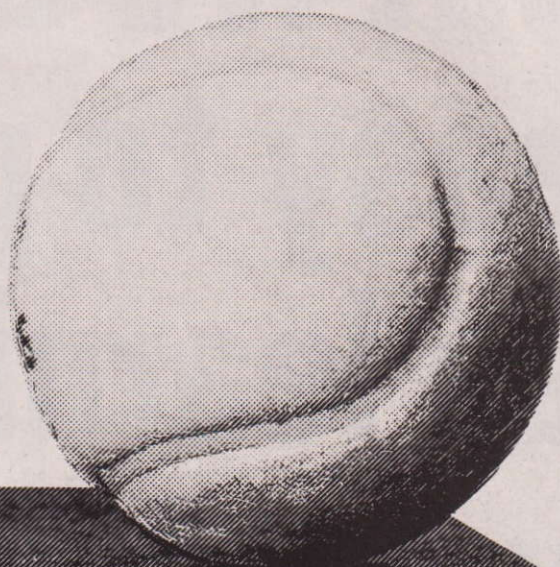
Punch data down the line with Miracle Technology's Modem WS2000.

WS2000 links your computer into the world-wide telephone network — the world of PRESTEL, Telecom Gold, Telex, MICRONET, databases and bulletin boards.

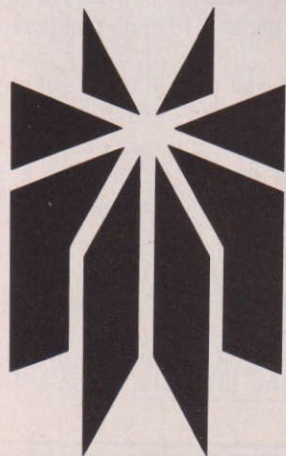
Flexible, versatile, high quality WS2000 wins every point, as over 11,000 business and home users know. MICRONET recommended, PCN Best Buy, British Microcomputing Awards Finalist — WS2000 plays its rivals off the court!

Our match winner, with BT telephone lead, mains power supply and comprehensive manual, costs £129.95 ex. (£154.73 inc. VAT & UK delivery). We can also supply the necessary leads and software for most computers — an unbeatable service!

Plus Free
Quarter's
Micronet Subscription



Prestel is a Registered Trade Mark of British Telecommunications PLC
Micronet 800 is the Trading Style of Telemap Ltd and British Telecommunications PLC



MIRACLE TECHNOLOGY

MIRACLE TECHNOLOGY (UK) LTD ST PETERS STREET IPSWICH IP1 1XB ENGLAND
☎ (0473) 216141 6 LINES TELECOM GOLD 79: KEY 001 ☒ 946240 CWEASY G 19002985



FINALIST
MICRO
BRITISH
COMPUTING
AWARDS
1985

It's all done by guess but it's as easy to play as learning your A-B-C!

By **DAVID PARR**

DISCOVERY is a new text game for the BBC Micro with OS 1.2 and either Basic I or II and should appeal to all ages. Starting with only the punctuation, the aim of the game is to reveal a piece of text by either purchasing letters or by making (educated?) guesses – with as little expenditure as possible, and a correspondingly higher score.

Although only one piece of text is included with the program (others may be added), you may create any number of additional ones and then save them to tape or disc for later use.

The text may vary from simple well-known nursery rhymes to more complicated pieces to suit all ages.

Full instructions are included with the program, but, if desired, these may be omitted by not typing lines 2930-3620 and by omitting lines 590, 880 and 1590 line 1110 should then be altered to:

1110 FOR I%=11 to 12

If no text is loaded, then the program will use the text included in the data statements.

You may add up to eight extra texts in data statements but you must alter the value of the variable *texts* in line 520 to be equal to the total number of data texts.

Each set of data should start with the number of the text – if the next text to be added is the second it should start with **DATA "2"** – and end with **DATA "-99"**.

Each extra text can have a maximum of 23 lines, with up to 37 characters each – any extra will be ignored.

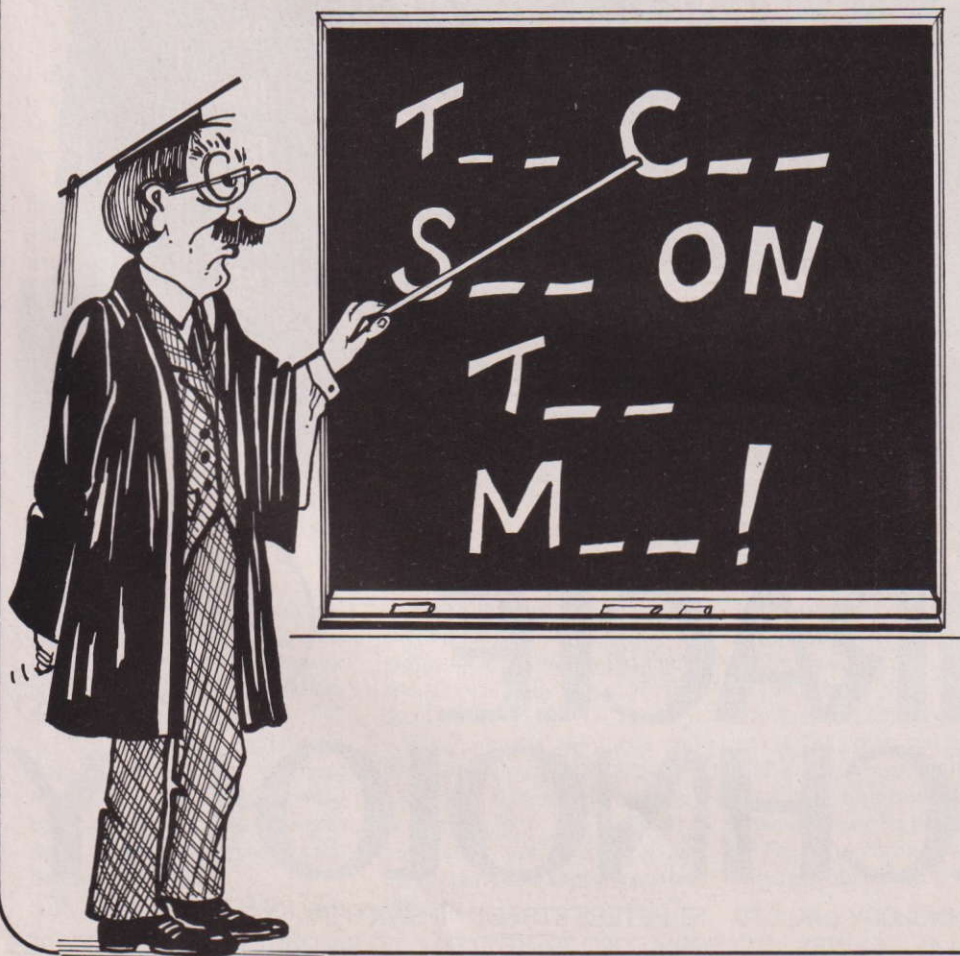
You start with a (limited) amount of money for the purchase of letters (£50).

As a vowel costs £18 and other letters less (down to £2 for a Z), care must be exercised in order to gain the maximum assistance from letters purchased.

There is also the option of purchasing all the initial letters for £20 but it is hoped that only younger players will need to use this!

To predict a letter the cursor keys are used to move to the desired position and the letter is typed in.

Upper or lower case letters may be entered – the program doesn't dis-



swork...

tinguish between them for matching purposes.

Any number of letters may be entered at a time. They are not scored until the Return key is pressed.

Wrong predictions cost -5 points each whereas correct predictions earn a variable score (1 to 9) depending on the letter and £1 in stake money for further purchases.

If a player wishes to give up, the Escape key can be pressed and he is offered a chance to see the complete text.

The program is, it is hoped, well structured with meaningful names for PROCs and variables which should make it easy to follow.

If desired the amount available for purchases may be amended to increase/decrease the difficulty by altering the value of *stake%* in line 510.

The score values for correct predictions of letters are contained in the data statement in line 840 in alphabetical order.

The purchase price is calculated in line 2140 by taking the score value (1 to 9) from 10 and then doubling it.

It results in a vowel costing a lot to buy (as it is very helpful) but not scoring a lot as it is relatively easy to predict.

Byte arrays are used to store both the text and the score values. An 852 byte array is used for the text (23*37 characters +1 for a "number of lines" byte).

To indicate a character is to be displayed it is stored in the array with a value 128 above its normal Ascii value by ORing with &80, as in line 1930.

An OSBYTE call with A containing 135 is used in line 1880 to return the character entered on the screen by the player.

Lines 1250-1280 allow the arrow keys to move the cursor around the screen having previously set the keys to return Ascii codes in line 530.

A word of warning: Anyone typing in the program should REM lines 100 and 140 until such time as they have removed all typing errors.

PROCEDURES

end Prompts for another go and exits if not.
escape Traps Escape key.
init Initialises variables.
write_text Offers chance to create new text.
title Prints title in double height.
input_text Main routine for input of text.
save_text Saves text file.
load_text Loads previously saved text.
FN_yn Converts input to upper case and exits if 'Y' or 'N'.
score Checks text on screen and calculates score and stake.
buy_or_predict Offers choice of buying or predicting a letter.
initials Offers chance to buy all initial letters.
screen Sets up the screen.
screen_window Sets up screen window.
score_window Sets up top line for score and stake display.
message Sets up bottom line to display message.
oscli Passes a command line to MOS.
free_ride Displays all letters similar to correct guess.
well_done Congratulatory message if text all revealed.
create_instructions Instructions for the creation of a new text.
load_instructions Instructions for loading text.
text_instructions Instructions for the game.

VARIABLES

I%,J%,XX%,YY% Loop counters.
X,Y Input
C,chr,ch%,prechr,input_chr,screen_chr,rightchr Character on screen or in array.
array Byte array holding text.
cmd Text command line for OSCLI.
finished Indicates game over.
gap% Number of lines at top of screen before text window.
initials Indicates whether initial letters have been purchased.
line% Used to count number of text lines.
Lines Last element of array to indicate number of lines.
loaded Indicates whether text has been loaded.
price Letter price.
right Number of correct characters after guess.
score Byte array holding score.
score% User's score.
stake% User's stake.
texts Number of texts in data statements.
CS Colour of line on screen.
MS Message.
name\$ Name of text file.

From Page 51

```

10 REM *****
20 REM *****
30 REM *      Discovery      *
40 REM *      by D Parr      *
50 REM *      *              *
60 REM * (c) The Micro User *
70 REM *****
80 REM *****
90 REM *****
100 ON ERROR GOTO350
110 MODE 7
115 VDU23;8202;0;0;0;0;
120 REPEAT
130 PROC_init
140 ON ERROR GOTO340
150 VDU26,12
160 REPEAT
170 PROC_screen(CHR$131)
180 PROC_score
190 PROC_buy_or_predict
200 PROC_input_text
210 UNTIL finished
220 PROC_well_done
230 PROC_end
240 UNTIL FALSE
250 END
260 REM=====
270 DEF PROC_end
280 PROC_message("Another go ?")
290 IF FN_yn="Y" THEN ENDPROC
300 VDU26,12:FX4
310 PRINTTAB(13,10)"GOODBYE"
320 END
330 REM=====
340 VDU7:IF ERR=17 THEN PROC_escape
350 VDU26,12:FX4
360 REPORT:PRINT" at line ";ERL
370 END
380 REM=====
390 DEF PROC_escape
400 PROC_message("Do you wish to see the text ?")
410 IF FN_yn="N" THEN PROC_end:RUN
420 PROC_message(STRING$(12," ")+"PLEASE WAIT")
430 FOR I%=0 TO 850:array?I%=(array?I% OR &80):NEXT
440 PROC_score:PROC_end:RUN
450 REM=====
460 DEF PROC_init
470 DIM array 852
480 DIM cmd 20
490 DIM score 26
500 RESTORE 840:FOR I%=1 TO 26:READ X:score?I%=X:NEXT
510 score%=0:stake%=50:finished=FALSE:initials=FALSE
520 texts=1:lines=array+852:?lines=
23

```

```

530 *FX4,1
540 PROC_write_text
550 PROC_load_text
560 IF loaded THEN line%=?lines:?lines=23
570 PROC_screen_window:CLS
580 PROC_title
590 PROC_text_instructions
600 IF loaded THEN PROC_message(STRING$(10," ")+"PRESS ANY KEY"):X=GET:?lines=line%:ENDPROC
610 IF texts<2 THEN PROC_message(STRING$(10," ")+"PRESS ANY KEY"):Y=GET:X=1:GOTO660
620 REPEAT
630 PROC_message("Choose a text (1-"+STR$(texts)+")?")
640 X=GET-48
650 UNTIL X>0 AND X<texts+1
660 PROC_message("      INITIALISING - PLEASE WAIT")
670 FOR I%=0 TO 851 STEP 4:array!I%=&A0A0A0A0:NEXT:?lines=0
680 RESTORE 3640:REPEAT:READ A$:UNTIL VAL(A$)=X
690 I%=0
700 REPEAT
710 READ A$:IF VAL(A$)=-99 THEN 790
720 J%=0
730 REPEAT
740 chr=ASC(MID$(A$,J%+1,1)):IF chr<65 OR (chr>90 AND chr<97) THEN chr=chr OR &80
750 array?(J%+I%*37)=chr
760 J%=J%+1
770 UNTIL J%=37 OR MID$(A$,J%+1,1)="
780 I%=I%+1
790 UNTIL I%=23 OR VAL(A$)=-99
800 IF I%<23 THEN ?lines=I% ELSE ?lines=23
810 PROC_screen_window:CLS
820 ENDPROC
830 REM=====
840 DATA 1,7,7,4,1,5,5,6,1,9,8,4,7,3,2,5,8,3,2,3,2,8,4,9,6,9
850 REM=====
860 DEF PROC_write_text
870 PROC_title
880 PROC_create_instructions
890 PROC_message("Create a new text file ?")
900 IF FN_yn="N" THEN ENDPROC
910 REPEAT
920 PROC_message(STRING$(12," ")+"PLEASE WAIT")
930 FOR I%=0 TO 851 STEP 4:array!I%=&A0A0A0A0:NEXT:?lines=23
940 PROC_message("")
950 PROC_title

```

```

960 REPEAT
970 CLS
980 PROC_screen_window
990 PROC_input_text
1000 PROC_message("Finished ?")
1010 UNTIL FN_yn="Y"
1020 PROC_save_text
1030 ?lines=23
1040 PROC_screen_window:CLS
1050 PROC_message("Another one ?")
1060 UNTIL FN_yn="N"
1070 ENDPROC
1080 REM=====
1090 DEF PROC_title
1100 PROC_screen(CHR$135)
1110 FOR I%=1 TO 2
1120 PRINTTAB(11,I%)CHR$129;CHR$141;"DISCOVERY"
1130 NEXT
1140 ENDPROC
1150 REM=====
1160 DEF PROC_input_text
1170 IF finished THEN ENDPROC
1180 PRINTTAB(0,0);
1190 REPEAT
1200 AX=135:screen_chr=(USR(&FFF4) AND &FFFF) DIV &100
1210 XX%=POS:YY%=VPOS:ch%=array?(XX%-1+YY%*37)
1220 X=GET
1230 IF (screen_chr=32 OR array?(XX%+YY%*37)<161) AND (X>31 AND X<127) THEN PRINT CHR%X; ELSE IF X<127 THEN SOUND1,-8,200,2:PRINTTAB(POS+1,VPOS);
1240 IF X=127 AND ch%<161 AND POS<>0 THEN PRINTCHR%X; ELSE IF X=127 AND POS=0 AND VPOS<>0 AND array?(XX%-2+YY%*37)<161 THEN PRINTCHR%X; ELSE IF X=127 THEN SOUND1,-8,200,2
1250 IF X=136 AND POS=0 AND VPOS<>0 THEN PRINTTAB(36,VPOS-1); ELSE IF X=136 THEN PRINTTAB(POS-1,VPOS);
1260 IF X=137 AND POS=36 THEN PRINTTAB(0,VPOS+1); ELSE IF X=137 THEN PRINTTAB(POS+1,VPOS);
1270 IF X=138 AND VPOS=(?lines)-1 THEN PRINTTAB(POS,0); ELSE IF X=138 THEN PRINTTAB(POS,VPOS+1);
1280 IF X=139 AND VPOS=0 THEN PRINTTAB(POS,(?lines)-1); ELSE IF X=139 THEN PRINTTAB(POS,VPOS-1);
1290 IF VPOS>=(?lines)-1 AND POS>35 THENPRINTTAB(0,0);
1300 UNTIL X=13
1310 ENDPROC
1320 REM=====
1330 DEF PROC_save_text
1340 line%=0
1350 FOR YY%=0 TO 22
1360 VDU26:PRINTTAB(0,YY%+1)CHR$131;

```



```

:PROC_screen_window
1370 FOR XX%=0 TO 36
1380 VDU31,XX%,YY%
1390 AX=135
1400 chr=(USR(&FFF4) AND &FFFF) DIV
&100
1410 IF chr<>32 THEN line%=YY%+1
1420 IF chr<65 OR (chr>90 AND chr<97
) OR chr>122 THEN chr=chr OR &80
1430 array?(XX%+YY%*37)=chr
1440 NEXT
1450 NEXT
1460 ?lines=line%
1470 PROC_message("Save text ?")
1480 IF FN_yn="N" THEN ENDPROC
1490 REPEAT

1500 PROC_message("Enter file name :
")
1510 INPUT "name$:name$=LEFT$(name
$,7)
1520 UNTIL LEN(name$)>0
1530 PROC_oscli("SAVE ""+name$+""
"+STR$(array)+" "+STR$(853))
1540 ENDPROC
1550 REM=====
1560 DEF PROC_load_text
1570 PROC_screen_window:CLS
1580 PROC_title
1590 PROC_load_instructions

1600 loaded=FALSE
1610 PROC_message("Load text ?")
1620 IF FN_yn="N" THEN ENDPROC
1630 loaded=TRUE
1640 REPEAT
1650 PROC_message("Enter file name :
")
1660 INPUT "name$:name$=LEFT$(name
$,7)
1670 UNTIL LEN(name$)>0
1680 PROC_oscli("LOAD "+name$+" "+ST
R$(array))
1690 ENDPROC
1700 REM=====
1710 DEF FN_yn
1720 REPEAT
1730 X=GET AND &DF
1740 UNTIL X=78 OR X=89
1750 CLS
1760 =CHR$(X)
1770 REM=====
1780 DEF PROC_score
1790 PROC_screen(CHR$135)

1800 right=0:wrong=FALSE:gap%=(23-?l
ines)/DIV2
1810 FOR YY%=0 TO ?lines-1
1820 VDU26:PRINTTAB(0,gap%+YY%+1)CHR
$131;:PROC_screen_window
1830 FOR XX%=0 TO 36
1840 IF (YY%=?lines-1 AND XX%=36) TH
EN 1960

```

```

1850 VDU31,XX%,YY%
1860 chr=array?(XX%+YY%*37):IF chr>1
60 THEN 1950
1870 AX=135
1880 input_chr=(USR(&FFF4) AND &FFFF
) DIV &100
1890 IF input_chr=32 THEN chr=32:60T
01950

1900 IF (input_chr OR &20)<>(chr OR
&20) THEN chr=32:score%=score%-5:wron
g=TRUE:60T01950
1910 score%=score%+score?((chr OR &2
0)-96)
1920 stake%=stake%+1:right=right+1:r
ightchr=chr
1930 chr=chr OR &80
1940 array?(XX%+YY%*37)=chr
1950 VDU (chr AND &7F)
1960 NEXT
1970 NEXT
1980 IF right=1 AND NOT wrong THEN P
ROC_message("Free ride ?"):IF FN_yn="
Y" THEN PROC_free_ride
1990 PROC_score_window:PRINTTAB(0,0)
CHR$133;"SCORE :";score%; TAB(20,0);"
STAKE :";stake%;

2000 PROC_screen_window
2010 I%=-1:REPEAT:I%=I%+1:UNTIL arra
y?I%<160 OR I%=850
2020 IF I%=850 finished=TRUE ELSE fi
nished=FALSE
2030 ENDPROC
2040 REM=====
2050 DEF PROC_buy_or_predict
2060 IF finished THEN ENDPROC
2070 IF stake%<2 THEN PROC_message("
Insufficient money! You must predict"
):PROC_screen_window:ENDPROC
2080 PROC_message(" Buy or Predi
ct (B/P) ?")
2090 REPEAT:X=GET AND &DF:UNTIL X=66
OR X=80
2100 IF X=80 THEN CLS:PROC_screen_wi
ndow:ENDPROC
2110 IF NOT initials AND stake%>19 T
HEN PROC_initials:IF initials THEN 20
60
2120 PROC_message("Enter letter you
wish to buy :")
2130 REPEAT:X=GET AND &DF:UNTIL X>64
AND X<91
2140 price=2*(10-(score?(X-64)))
2150 IF stake%-price<0 THEN PROC_mes
sage("You can't afford that ! It cost
s ""+STR$(price)):VDU7:TIME=0:REPEAT
UNTIL TIME>300:60T0 2080
2160 stake%=stake%-price
2170 PROC_message(STRING$(12," ")+"P
LEASE WAIT")
2180 FOR I%=0 TO 850

```

```

2190 chr=array?I%:IF (chr AND &DF)=X
THEN array?I%=chr OR &80
2200 NEXT
2210 PROC_score
2220 GOTO 2060
2230 REM=====
2240 DEF PROC_initials
2250 PROC_message("Buy initial lette
rs (Y/N) ?")
2260 IF FN_yn="N" THEN ENDPROC
2270 stake%=stake%-20
2280 PROC_message(STRING$(12," ")+"P
LEASE WAIT")
2290 FOR I%=0 TO 22
2300 FOR J%=0 TO 36
2310 chr=array?(J%+I%*37):IF chr>159
THEN 2350
2320 IF J%=0 THEN array?(J%+I%*37)=c
hr OR &80
2330 prechr=array?(J%-1+I%*37)
2340 IF (prechr>159 AND prechr<193)
OR (prechr>218 AND prechr<225) OR pre
chr>250 THEN array?(J%+I%*37)=chr OR
&80
2350 NEXT
2360 NEXT
2370 initials=TRUE
2380 PROC_score
2390 ENDPROC
2400 REM=====
2410 DEF PROC_screen(C$)
2420 VDU26:gap%=(23-?lines)/DIV2
2430 FOR I%=1+gap% TO gap%+?lines
2440 PRINT TAB(0,I%)C$;CHR$157;CHR$1
32
2450 NEXT
2460 PRINTTAB(0,24)CHR$130;
2470 PROC_screen_window
2480 ENDPROC
2490 REM=====
2500 DEF PROC_screen_window
2510 gap%=(23-?lines)/DIV2
2520 VDU28,3,gap%+?lines,39,gap%+1
2530 ENDPROC
2540 REM=====
2550 DEF PROC_score_window
2560 VDU28,1,0,39,0,12
2570 ENDPROC
2580 REM=====
2590 DEF PROC_message(M$)
2600 VDU28,1,24,39,24,12
2610 PRINTM$;
2620 ENDPROC
2630 REM=====
2640 DEF PROC_oscli(osi$)
2650 $cmd=osi$
2660 X%=$cmd MOD 256:Y%=$cmd DIV 256
2670 CALL &FFF7
2680 ENDPROC

```


From Page 53

```

2690 REM=====
2700 DEF PROC_free_ride
2710 score%=score%-score?((rightchr
OR &20)-96)
2720 stake%=stake%-1:gap%=(23-?lines
)DIV2
2730 PROC_screen(CHR$135)
2740 FOR YYZ=0 TO ?lines-1
2750 VDU26:PRINTTAB(0,gap%+YYZ+1)CHR
$131;:PROC_screen_window
2760 FOR XXZ=0 TO 36
2770 IF (YYZ=?lines-1 AND XXZ=36) TH
EN 2820
2780 VDU31,XXZ,YYZ
2790 C=array?(XXZ+YYZ*37)
2800 IF (C OR &20)=(rightchr OR &20)
THEN array?(XXZ+YYZ*37)=C OR &80
2810 IF array?(XXZ+YYZ*37)>160 THEN
VDU C ELSE VDU 32
2820 NEXT
2830 NEXT
2840 ENDPROC
2850 REM=====
2860 DEF PROC_well_done
2870 ?lines=23
2880 PROC_screen_window:CLS
2890 PROC_title
2900 PRINTTAB(13,14)"WELL DONE"
2910 ENDPROC
2920 REM=====
2930 DEF PROC_create_instructions
2940 PRINT'"SPC8"TO CREATE A TEXT FI
LE"
2950 PRINTSPCB"-----
"
2960 PRINT"Just enter the text using
upper or"
2970 PRINT"lower case and any punctu
ation as"
2980 PRINT"required. The cursor will
pass to"
2990 PRINT"the next line when you re
ach the end"
3000 PRINT"of a line. The cursor key
s may be"
3010 PRINT"used to position the curs
or and any"
3020 PRINT"alterations may be made b
y just"
3030 PRINT"overtyping the previous t
ext."
3040 PRINT'"When text entry is compl
ete, press"
3050 PRINT"the RETURN key. You will
then be"
3060 PRINT"offered the chance to sav
e the text."
3070 ENDPROC

```

```

3080 REM=====
3090 DEF PROC_load_instructions
3100 PRINT'"SPC6"LOAD TEXT INSTRUCTI
ONS"
3110 PRINTSPC6"-----
-"
3120 PRINT'"Any text file, previous
ly created"
3130 PRINT"and saved with this progr
am, may be"
3140 PRINT"loaded."
3150 ENDPROC
3160 REM=====
3170 DEF PROC_text_instructions
3180 PRINTSPC9"GAME INSTRUCTIONS"
3190 PRINTSPC9"-----"
3200 PRINT"A piece of text is hidden
and all"
3210 PRINT"you can see is the punctu
ation. You"
3220 PRINT"must discover what the te
xt is by"
3230 PRINT"buying or predicting lett
ers."
3240 PRINT"If you have more than '2
stake money"
3250 PRINT"you may buy a letter. Occ
urrences"
3260 PRINT"of that letter will be re
vealed and"
3270 PRINT"this will, of course, lea
ve you less"
3280 PRINT"to score. You may also bu
y all the"
3290 PRINT"initial letters. You can
predict 1"
3300 PRINT"or more letters but you w
ill score"
3310 PRINT"-5 for each wrong guess.
To enter"
3320 PRINT"a guess just use the curs
or keys or"
3330 PRINT"space bar to position the
cursor"
3340 PRINT"and type it in. Upper or
lower case"
3350 PRINT"may be entered and will b
e scored"
3360 PRINT"when the RETURN key is pr
essed."
3370 PROC_message(STRING$(10,"")+P
RESS ANY KEY")
3380 X=GET
3390 PROC_screen_window:CLS
3400 PROC_title
3410 PRINTSPC9"GAME INSTRUCTIONS"
3420 PRINTSPC9"-----"
3430 PRINT"If you enter only one let
ter and it"
3440 PRINT"is correct, you will be o

```

```

ffered a"
3450 PRINT"free ride, ie all similar
letters"
3460 PRINT"will be revealed but you
will score"
3470 PRINT"nothing for the correct g
uess and"
3480 PRINT"will again reduce the max
imum score."
3490 REM
3500 PRINT'"LETTERS","COST"SPC4"SCOR
E"
3510 PRINT"Initials","'20","-"
3520 PRINT"A,E,I","'18","1"
3530 PRINT"O,S,U","'16","2"
3540 PRINT"N,R,T","'14","3"
3550 PRINT"D,L,W","'12","4"
3560 PRINT"F,G,P","'10","5"
3570 PRINT"H,Y","'8","6"
3580 PRINT"B,C,M","'6","7"
3590 PRINT"K,Q,V","'4","8"
3600 PRINT"J,X,Z","'2","9";
3610 ENDPROC
3620 REM=====
3630 REM TENNYSON 'CHARGE OF THE LIG
HT BRIGADE'
3640 DATA"1"
3650 DATA"Half a league, Half a leag
ue,"
3660 DATA"Half a league onward,"
3670 DATA"All in the valley of Death
"
3680 DATA"Rode the six hundred."
3690 DATA"Forward, the Light Brigad
e!"
3700 DATA"Charge for the guns!" he s
aid;"
3710 DATA"Into the valley of Death"
3720 DATA"Rode the six hundred."
3730 DATA"Cannon to right of them,"
3740 DATA"Cannon to left of them,"
3750 DATA"Cannon in front of them"
3760 DATA"Volley'd and thunder'd;"
3770 DATA"Stormed at with shot and s
hell,"
3780 DATA"Boldly they rode and well,
"
3790 DATA"Into the jaws of Death,"
3800 DATA"Into the mouth of Hell"
3810 DATA"Rode the six hundred."
3820 DATA"When can their glory fade?
"
3830 DATA"D the wild charge they mad
e!"
3840 DATA"All the world wonder'd."
3850 DATA"Honour the charge they mad
e!"
3860 DATA"Honour the Light Brigade,"
3870 DATA"Noble six hundred!"
3880 DATA"-99"

```


Educational Software **LCL**

MEGA MATHS



24 PROGRAM & LEVEL STEP-BY-STEP SELF-TUTORIAL
2 REVISION COURSE FOR MATURE BEGINNERS
2 LEVEL STUDENTS & MICRO MATHS USERS

BBC Micro
Tape, 40/80 track Disc,
Econet compatible.

Electron
ZX Spectrum
Commodore 64
unexpanded Vic
ZX81 · Apple · PET
NEC PC8201A
Dragon (book only)
All software for the
above computers.



MEGA MATHS (BBC, Electron). 24 program step-by-step, A-level course of 105 topics for mature beginners, A-level students and Micro Maths users. (2 discs/tapes + 2 books.) £24.

MICRO ENGLISH (BBC, Electron). Complete English Language self-tuition and revision course, to O-Level, of 24 programs. Includes "speak-and-spell" real speech (no extras required). £24.

MICRO MATHS 24 program revision or self-tuition course to O-Level. Includes 59 Topics plus two books. £24.

ANIMATED ARITHMETIC Teaches using moving colour pictures (not words) Ages 3-8. £6.50 (disc £7.50).

BRAINTEASERS Book of IQ tests and puzzle programs. £5.95 "..." excellent" EDUCATIONAL COMPUTING.

*Immediate dispatch Micro Maths, Micro English, Mega Maths and Master Maths. Phone or send orders or requests for free glossy colour catalogue to:

LCL DEPT. E
26 Avondale Ave.
Staines, Middx.
Tel: 0784 58771

24 hrs. Orders

World leaders in educational software
(distributors in 24 countries.)



TRADE DISCOUNTS
NEW HOTLINE SERVICE



AS SEEN ON
TV

MACOB

SYSTEMS

DIRECT TO YOU AT WHOLESALE PRICES

COMPUTERS

BBC B Plus	£489
BBC-B	£329
BBC-BD	£419
DFS (Acorn)	£90
Z80 2nd Processor	£389
6502 2nd Processor	£189
CBM 64	Available for export
Sinclair	

MONITORS

Microvitec 1431	£185
Microvitec 1451	£279
Philips BM7502	£83

DISKS

D/S D/D 80T	£20 (box of 10)
-------------	-----------------

(Fully guaranteed) box of 10

MODEMS

Pace Nightingale	£130
------------------	------

DUAL DRIVES WITH PSU AND MONITOR STAND

MD802D	£320
MD402D	£290

DUAL DRIVES VERTICALLY STACKED WITH CABLES + FORMATTING DISK

MD802C	£255
MD402C	£225

- Price includes VAT at 15% and delivery is FREE in the UK.
- To order just send a cheque or postal order to us and the goods reach you fast.
- Trade enquiries welcome.
- Remember, our name is our guarantee.

PRINTERS

Canon PW1080A	£279
Canon PW1156A	£389
Kaga KP810	£279
Epson RX80F/T	£242
Quendata 20cps	£239
Juki 6100	£332

FIRMWARE

View	£48
Wordwise +	£48
Disc Doctor	£29
Printmaster	£29

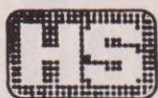
DATA RECORDERS

Omega	£21
-------	-----

MACOB SYSTEMS

8 Lister Gardens, London N18 1HZ. Tel: 01-803 1622

Telex: 8954029 att.MACOB



HIGHLIGHT SOFTWARE

for the

BBC & Electron

Reading Pack 1 (Age 5-8)

SPLASHDOWN & FIREFIGHT

Building 3 letter words sl-, sm-, sn-, sp-, bl- etc.

Reading Pack 2 (Age 6-9)

PYRAMIDS & SPLOOSH

th-, ch-, sh-, wh- oo-, ee-, ea-, oi-, ou-, oa-, ai-

Reading Pack 3 (Age 7-10)

MAGIC E & BREAK IN

"Magic e" spelling rule soft 'c', soft 'g' sounds.

Reading Pack 4 (Age 8-11)

SORT OUT & LETTERBUGS

Alphabetical Sorting Unscrambling hidden words

Latest Release:

"ENTERTAINING ENGLISH" (Age 7-11)

"BUSY": Verbs and adverbs

"TIMESLIP": Past and present tenses of verbs.

Price per Pack

BBC 40 Track Disc: £11.95 BBC/Electron Cassette: £7.95

Programs can be obtained from selected branches of BOOTS and JOHN MENZIES, from MICROPOWER dealers and from Educational Software specialists including FIVE TO TWELVE, PROTEUS COMPUTING, SOFT CENTRE or by Mail Order.

Send for your FREE catalogue
with full program range NOW!

HIGHLIGHT SOFTWARE
FREEPOST, Barry, CF6 6YZ.
Telephone: (0446) 745482

LEARNING IS FUN WITH HIGHLIGHT!

JP Magnetics

*specialists in the supply of bulk
custom wound audio/computer
cassettes to Trade and Education*

NEW!

**TURBO LOAD FOR
CBM 64 WITH
'ON SCREEN DISPLAY'**

*DUPLICATION FACILITIES FOR MOST
POPULAR HOME COMPUTERS
16 TRACK RECORDING STUDIO*

*Competitive Rates, Contact us for price
list/quotations now.*

JP MAGNETICS LTD

UNIT 4, 7 MARY ST, BRADFORD BD4 8SW
TEL: (0274) 731651

HIGH QUALITY GRAPHICS!

Put a full stop to punctuation problems

THIS program was designed to help children who are already vaguely familiar with the aspects of punctuation to improve their punctuation skills.

It can be used by a wide age range, and because there is a provision to load, save and enter new data, sentences can be created to match the pupils' different abilities.

An unpunctuated sentence is displayed in the centre of the screen. In order to insert the required marks you must move the arrow using the left and right cursor keys.

When it is in the correct position type

By PALI RAO

in the character required from the keyboard. Once you feel that you have completed the task press Return.

The program's correction cursor will then scan your work, pointing out any errors.

Should it encounter one, a brief comment will be made in the lower half of the screen concerning the correct use of the particular punctuation mark required. This process continues until

you have punctuated 10 sentences.

Throughout the exercise the screen continuously displays the score for the last sentence as a percentage, the number of turns you have had, the number of totally correct sentences and the number of mistakes you have made — once again as a percentage.

The program makes use of speech synthesis during the introductory sequence. If no speech chip is present the sound statements from the end of line 350 should be omitted.

PROCEDURES

data	Stores the data for the 50 sentences into the variable <i>sen\$</i> .
instr	Displays the instructions and asks for your name.
init	Initialises variables.
screen	Draws the teletext screen display.
change	Selects a random sentence from those currently in use and strips it of its punctuation marks.
move	Enables you to move the arrow on the screen and type in the desired punctuation.
eval	Evaluates and checks your input. Also updates the screen information.
comment	Prints out the comments if mistakes have been made.
info	Displays information at the end of the test.
header	Prints out the title Punctuation in double height letters.
FNinput	Input routine for your name.
inpdata	The menu.
load] Loads previously stored data.
loadfile	
save] Save the sentences currently stored in memory.
savefile	
enter	Enters new sentences into memory.

MAIN VARIABLES

name\$	Stores the name of the user after being processed by FNinput.
sen\$	Array holding the sentences ready for random selection.
inp\$	String array holding your punctuation marks as entered by the arrow cursor.
SC	Number of correct punctuation symbols entered.
X%	x value for the arrow cursor.
Y%	y value for the arrow cursor.
right%	Number of sentences correctly entered.
score%	Percentage of marks attained for the last sentence displayed.
turn	Number of turns.
nsen\$	Unpunctuated sentence ready to be displayed on the screen.
choice	Random number from which the sentences are chosen.
P	Central position of the sentence.
Pos	Position of the arrow cursor.
mistake%	Number of mistakes made.
sco%	Average score as a percentage.
L	Number of sentences in memory.
mis%	Average number of mistakes as a percentage.
clear%	Loop to clear the input array for the next turn.
tot%	Total number of possible errors that can occur in a sentence.

From Page 57

```

5 REM (C) The Micro User
10 MODE 7:VDU 23,1,0;0;0;0;
15 ON ERROR GOTO 1900
20 DIM sen$(50),inp$(50):L=20:*FX
4,1
30 PROCdata
40 PROCinstr
50 PROCinit
60 PROCscreen
70 FOR turn=1 TO 9
80 PROCchange
90 PROCmove
100 IF I=9 THEN:turn=10:NEXT:PROCin
pdata:GOTO 50
110 PROCeval
120 NEXT
130 PROCinfo
140 GOTO 50
150 DEFPROCdata
160 FOR R=1 TO 20:READ sen$(R):NEXT
170 DATA ""Worship your King!"" sa
id the slave.",""I'd like you to lea
ve,"" she said.",""What?"" David sai
d,startled.",""He looked away.""Not he
re,Captain.""
180 DATA ""A year!"" I cried.""Wer
e you lost?""""I will."" Paul's ja
w thrust forward.",""Why?"" he cried
.""Why?""""It's Pete's."" she repl
ied."
190 DATA ""I hate it!"" cried Jame
s.",""Come aboard,Smith."" I said ag
ain.",""Is that all?"" she asked.",
""Yes,"" I said,""if there is a cave.
""",""Nonsense!"" he cried.",""Well
,well,"" he said,""duty calls.""
200 DATA ""Rumours!"" exclaimed Ro
bin.",""Forward!"" cried the General
.",""We'll see."" he replied.",""Wh
ere are his teeth?"" Jon asked.",""A
ny more?"" asked the man's friend.",
""Why does it end?"" I asked."
210 ENDPROC
220 DEFPROCinstr
230 PROCheader
240 PRINTTAB(0,4);CHR$(131);" You
must try to correctly punctuate ""CH
R$(131);"sentences that appear on the
screen."
250 PRINT:PRINTCHR$(131);" To do t
his you move the arrow using: ""CHR$(
134);" "L";CHR$(131);"(left" a
nd";CHR$(134);" "I";CHR$(131);"(right
)" ""CHR$(131);"then type in corr
ect character from the""CHR$(131);"ke
yboard.";

```

```

260 PRINTCHR$(131);"Press";CHR$(134
);"RETURN";CHR$(131);"when you finish
."CHR$(131);"Pressing";CHR$(134);"TA
B";CHR$(131);"displays the menu to lo
ad,"CHR$(131);"save or enter data.";
270 PRINT:CHR$(131);"Your attempt w
ill""CHR$(131);"be marked & correcte
d by the computer."
280 PRINTTAB(1,23);CHR$(134);"Press
";CHR$(131);"P";CHR$(134);"to play,
or";CHR$(131);"C";CHR$(134);"to co
ntinue"
290 A=GET:IF A<>80 AND A<>67 THEN 2
90 ELSE IF A=80 THEN:CLS:GOTO 350
300 PROCheader:PRINTCHR$(134);" T
he following information will be "
SPC(7);CHR$(134);"displayed on the s
creen:"
310 PRINTSPC(6);CHR$(131);" The";C
HR$(134);"score";CHR$(131);"for the L
AST""SPC(3);CHR$(131);"question, give
n as a percentage."
320 PRINTSPC(5);CHR$(134);"Turn -"
CHR$(131);"The number of turns";SPC(
17);CHR$(131);"that you have had."
330 PRINTSPC(3);CHR$(131);"The num
ber of sentences totally""SPC(14);CHR
$(134);"correct."
340 PRINTCHR$(131);"The number of"
CHR$(134);"mistakes";CHR$(131);"that
have been""CHR$(131);" made, given a
s an average percentage."PROCspace(4
,22)
350 PROCheader:PRINTTAB(7,10);CHR$(
131);"Please enter your";CHR$(134);"n
ame:"SOUND -1,241,0,0:SOUND -1,192,0
,0:SOUND -1,290,0,0:SOUND -1,223,0,0
360 name$=FNinput
370 ENDPROC
380 DEFPROCinit
390 mone%=0:mtwo%=0:SC=0:MC%=0:Y%=1
2:right%=0:score%=0:sco%=0
400 ENDPROC
410 DEFPROCscreen:CLS
420 PRINTTAB(0,0);CHR$(145);STRING$
(13,CHR$(243));CHR$(135);"PUNCTUATION
";CHR$(145);STRING$(13,CHR$(243))TAB(
0,13);CHR$(145);STRING$(39,CHR$(240))
430 FOR col=1 TO 9
440 PRINTTAB(0,col);CHR$(145);CHR$(
181);TAB(38);CHR$(145);CHR$(234)TAB(0
,col+13);CHR$(145);CHR$(181);TAB(38,c
ol+13);CHR$(145);CHR$(234)
450 NEXT
460 PRINTTAB(0,9);CHR$(145);STRING$
(39,CHR$(163));TAB(0,22);CHR$(145);S
tring$(39,CHR$(163))
470 PRINTTAB(5,3);CHR$(135);"Score:

```

```

";CHR$(131);"0%";CHR$(135);SPC(7);"Co
rrect:"CHR$(134);"0"TAB(5,5);CHR$(13
5);" Turn:"CHR$(131);"1";CHR$(135);S
PC(7);"Mistakes:"CHR$(134);"0%"
480 PRINTTAB(3,7);CHR$(135);"Keys:"
CHR$(131);"L";CHR$(134);"left right
";CHR$(131);"I";CHR$(135);" TAB:";CH
R$(130);"menu";TAB(12,1);CHR$(135);"T
est for";CHR$(131);name$
490 ENDPROC
500 DEFPROCchange
510 PRINTTAB(0,10);SPC(39)TAB(0,12)
;SPC(39)
520 nsen$="":choice=RND(L):P=INT((3
9-LEN(sen$(choice)))/2)
530 FOR Y=1 TO LEN(sen$(choice))
540 A=ASC(MID$(sen$(choice),Y,1))
550 IF A=33 OR A=34 OR A=39 OR A=44
OR A=46 OR A=63 THEN EQ=TRUE ELSE EQ
=FALSE
560 IF A>=65 AND A<=90 THEN nsen$=n
sen$+CHR$(A+32) ELSE IF EQ=FALSE THEN
nsen$=nsen$+CHR$(A)
570 IF EQ=TRUE THEN nsen$=nsen$+" "
580 NEXT
590 PRINTTAB(P,10);CHR$(131);nsen$
600 ENDPROC
610 DEFPROCmove
620 XX=18:REPEAT
630 I=INKEY(0):IF I=9 THEN GOTO 700
640 IF I=136 AND XX>(P+1) THEN XX=X
%-1
650 IF I=137 AND XX<INT((P+LEN(nsen
$))) THEN XX=XX+1
660 PRINTTAB(XX,YX);""TAB(XX-1,YX)
";""TAB(XX+1,YX);""
670 IF I>=35 AND I<=38 OR I>=40 AND
I<=43 OR I=45 OR I>=47 AND I<=62 OR
I=64 OR I>=91 AND I<=96 OR I>=123 AND
I<=126 OR I=135 OR I=-1 OR I=127 OR
I=138 OR I=139 THEN 630
680 IF I<>137 AND I<>136 AND I<>13
THEN:PRINTTAB(XX,10);CHR$(I):inp$(XX-
P)=CHR$(I)
690 *FX 15,0
700 UNTIL I=13 OR I=9
710 ENDPROC
720 DEFPROCeval
730 PRINTTAB(XX,YX);""
740 mistake%=0:tot%=0:SC=0:score%=0
750 FOR ch=1 TO LEN(sen$(choice)):P
os=ch+P
760 N$=MID$(sen$(choice),ch,1):IF (
ASC(N$)>=97 AND ASC(N$)<=122) OR ASC(
N$)=32 THEN check=FALSE ELSE check=TR
UE
770 IF inp$(ch)=N$ co%=TRUE ELSE co
%=FALSE

```



```

780 IF check AND co% THEN SC=SC+1
790 IF check THEN tot%=tot%+1
800 PRINTTAB(Pos,Y%-2);N$:IF Pos<=3
5 THEN PRINTTAB(Pos+1,Y%);" ^"
810 IF check AND NOT co% THEN:PRINT
TAB(Pos,Y%);"^^":SOUND 2,-15,67,1:mist
ake%=mistake%+1:PROCcomment
820 IF ch+Pos<=35 THEN PRINTTAB(Pos
+1,Y%);" ^"
830 NEXT
840 score%=(SC/tot%)*100:sco%=sco%+
score%
850 mone%=(mistake%/tot%)*100:mtwo%
=mtwo%+mone%:mis%=(mtwo%/(turn*100))*
100
860 IF score%=100 THEN:right%=right
%+1:PRINTTAB(3,17);CHR$(136);CHR$(131
);"Well done, you made no mistakes";C
HR$(137):SOUND 2,-15,88,4:SOUND 2,-15
,98,2:SOUND 2,-15,100,5:I=INKEY(300):
PRINTTAB(2,17);SPC(34)
870 PRINTTAB(12,3);CHR$(131);score%
;"% ";CHR$(135);TAB(12,5);CHR$(131);
turn+1;CHR$(135);TAB(31,3);CHR$(134);
right%;CHR$(135);TAB(31,5);CHR$(134);
mis%;"% "
880 FOR clear=1 TO 40:inp$(clear)="
":NEXT
890 ENDPROC
900 DEFPROCcomment
910 comment%=ASC(N$):PRINTTAB(Pos+1
,Y%);" ":VDU 28,2,21,37,15
920 com=INSTR(".,'?!",N$)
930 IF comment%>65 AND comment%<=9
0 THEN com=7
935 IF com=0 THEN GOTO 950
940 ON com GOTO 960,970,980,990,100
0,1010,1020
950 PROCspace(4,5):CLS:VDU 28,0,24,
39,0:ENDPROC
960 PRINTCHR$(135);" The";CHR$(131
);"full stop";CHR$(135);"should be us
ed at ";CHR$(135);"the end of comp
lete sentences,or ";CHR$(135);"afte
r initials or abbreviations.":GOTO 95
0
970 PRINTCHR$(135);" The";CHR$(131
);"apostrophe";CHR$(135);"should be us
ed to ";CHR$(135);"mark the omission
of a letter in a ";CHR$(135);"word,
or as the sign of the case of";CHR$(1
35);"possession e.g.""Jo's book"".":G
OTO 950
980 PRINTCHR$(135);" The";CHR$(131
);"comma";CHR$(135);"is used to replac
e 'and'";CHR$(135);"or as a pause,or
to mark off,or to ";CHR$(135);"separ
ate parts of a sentence,or ";CHR$(1

```

```

35);"after addressing a person or thi
ng.":GOTO 950
990 PRINTCHR$(135);" The";CHR$(131
);"inverted commas";CHR$(135);"are use
d to ";CHR$(135);"indicate directly
reported speech.":GOTO 950
1000 PRINTCHR$(135);" The";CHR$(131
);"question mark";CHR$(135);"is used a
fter ";CHR$(135);"all direct questi
ons.":GOTO 950
1010 PRINTCHR$(135);" The";CHR$(131
);"exclamation mark";CHR$(135);"is pla
ced ";CHR$(135);"after a sentence
expressing strong ";CHR$(135);"emotio
n or implying loud outcry.":GOTO 950
1020 PRINTCHR$(135);" The";CHR$(131
);"capital letter";CHR$(135);"is used
at the ";CHR$(135);"start of a new se
ntence or as the ";CHR$(135);"first
letter of a proper noun.":GOTO 950
1030 DEFPROCspace(X,Y)
1040 *FX 15,0
1050 PRINTTAB(X,Y);CHR$(134);"Press"
;CHR$(131);"space bar";CHR$(134);"to
continue"
1060 REPEAT A=GET:UNTIL A=32
1070 ENDPROC
1080 DEFPROCinfo
1090 finsco%=(sco%/1000)*100
1100 PRINTTAB(9,15);CHR$(135);"Out o
f";CHR$(134);"10";CHR$(135);"question
s,"TAB(9,16);CHR$(135);" you got";CHR
$(134);right%;CHR$(135);"correct."
1110 PRINTTAB(6,17);CHR$(131);"Your
average score was";CHR$(134);finsco%;
"%";TAB(4,18);CHR$(129);mis%;"%";CHR$(
135);"of entries were mistakes"
1120 PROCspace(6,20)
1130 ENDPROC
1140 DEFPROCheader
1150 CLS:FOR head=1 TO 2:PRINTTAB(0,
head);CHR$(141);".....";CHR$(1
29+head);"PUNCTUATION";CHR$(135);"...
.....":NEXT
1160 ENDPROC
1170 DEFFNinput
1180 PRINTTAB(8,15);SPC(6);TAB(2,15);
SPC(40);TAB(10,17);SPC(40);TAB(16,12)
;".":DEL=127:RET=13:HI=90:LO=6
5:Co=0:name$="":REPEAT
1185 REPEAT
1190 *FX 15,1
1200 Ok=FALSE:En%=GET$:En=ASC(En$)
1220 IF (En)=LO AND En=(HI) OR En=32
THEN Ok=TRUE
1230 UNTIL Ok OR En=DEL OR En=RET
1231 IF Co=7 THEN IF En<>DEL AND En<
>RET THEN GOTO 1185

```

```

1240 IF En=DEL AND Co>0 THEN:Co=Co-1
:VDU 127:name$=LEFT$(name$,LEN(name$)
-1):GOTO 1185
1241 IF En=DEL THEN 1185
1245 Co=Co+1
1250 name$=name$+En$:IF En<>RET THEN
PRINTTAB(15+Co,12);En$;
1270 UNTIL En=RET
1280 IF Co=1 THEN:PRINTTAB(2,15);CHR
$(129);"YOU HAVE NOT TYPED IN YOUR NA
ME":TAB(10,17);CHR$(129);"PLEASE TRY A
GAIN":FOR Pause=1 TO 4000:NEXT:GOTO 1
180
1290 =name$
1300 DEFPROCinpdata
1310 PROCheader:PRINTTAB(15,5);CHR$(
130);"-MENU-"
1320PRINTTAB(1,7);CHR$(131);"Please
type in your choice, ";name$
1330 PRINTTAB(11,13);CHR$(129);"A:";
CHR$(131);"LOAD new data":TAB(11,15);C
HR$(129);"B:"CHR$(131);"SAVE data":TAB
(11,17);CHR$(129);"C:"CHR$(131);"ENTE
R new data":TAB(11,19);CHR$(129);"D:";
CHR$(131);"CONTINUE"
1340 PRINTTAB(18,17);I$=GET$:I$=CHR
$(ASC(I$) AND &DF):C=INSTR("ABCD",I$)
:IF C<>0 THEN PRINTTAB(18,17);I$ ELSE
GOTO 1340
1350 IF C=1 THEN PROCload:GOTO 1310
1360 IF C=2 THEN PROCsave:GOTO 1310
1370 IF C=3 THEN PROCcenter:GOTO 1310
1380 ENDPROC
1390 IF I$<>CHR$(13) AND I$<>CHR$(12
7) THEN S=S+1
1400 DEFPROCload
1410 PROCheader:PRINTTAB(14,10);"LOA
DING DATA"
1420 PROCloadfile
1430 PROCspace(5,23)
1440 ENDPROC
1450 DEFPROCsave
1460 CLS:PROCheader:PRINTTAB(14,10);
"SAVING DATA":TAB(11,16);PROCsavefil
e:PROCspace(5,23)
1470 ENDPROC
1480 DEFPROCcenter
1490 PROCheader:PRINTTAB(13,6);"ENTE
RING DATA":TAB(3,8);CHR$(130);"Please
input the sentences up to"
1500 PRINTCHR$(130);" THIRTY FIVE C
HARACTERS IN LENGTH, with"CHR$(130);
" all the necessary punctuation, an
d ""CHR$(130);CHR$(136);" press RE
TURN";CHR$(137);"after each entry."
1510 PRINTTAB(6,17);CHR$(131);"How m

```


From Page 59

```
any sentences do you"CHR$(131);SPC(8
);"wish to enter (10-50) ? ";C=1
1520 I=GET:IF C=1 AND I<49 OR C=1 AN
D I>53 OR C=2 AND I<48 OR C=2 AND I>5
7 THEN:IN=1:SOUND 2,-15,67,4:GOTO 152
0
1530 IF C=1 THEN L=(I-48)*10 ELSE L=
L+(I-48)
1540 IF L<>50 THEN PRINTCHR$(I); ELS
E PRINT"50"
1550 IF C=1 AND L<>50 THEN C=2:GOTO
1520
1560 PROCspace(5,23)
1570 U=8:FOR R=1 TO L:IF R=1 OR R/14
=INT(R/14) PROCheader:PRINTTAB(0,3);C
HR$(129);"SENTENCE NUMBER"TAB(3,4);CH
R$(131);STRING$(35,"."):U=8
1580 S=0:PRINTTAB(0,U);CHR$(129);R;C
HR$(135);
1590 I$=GET$:IF S=0 AND I$=CHR$(127)
THEN GOTO 1590
1595 IF S=35 AND I$<>CHR$(13) AND I$
<>CHR$(127) THEN GOTO 1590
1600 IF I$=CHR$(127) THEN S=S-1:S$=L
EFT$(S$,LEN(S$)-1):PRINTTAB(4,U);S$+"
";GOTO 1590
1605 IF I$="0" AND I$<" THEN GOTO
1590
1610 IF I$<>CHR$(13) THEN S=S+1:S$=S
$+I$:PRINTTAB(4,U);S$
1640 IF I$=CHR$(13) AND S>2 THEN 167
0
1660 GOTO 1590
1670 sen$(R)=S$:S$="":U=U+1:NEXT R:PR
OCspace(5,23)
1680 ENDPROC
1690 DEFPROCloadfile.
1695 PROCfilename
1700 F=OPENUP(file$):INPUT#F,L
1710 FOR R=1 TO L:INPUT#F,D$:sen$(R)
=D$:NEXT
```



```
1720 CLOSE#F:PRINTTAB(9,18);CHR$(129
);"THE DATA HAS LOADED"
1730 ENDPROC
1740 DEFPROCsavefile
1745 PROCfilename
1750 F=OPENOUT(file$):PRINT#F,L
1760 FOR R=1 TO L:PRINT#F,sen$(R):NE
XT
1770 CLOSE#F:PRINTTAB(10,18);CHR$(12
9);"THE DATA HAS SAVED"
1780 ENDPROC
1790 DEFPROCfilename
1800 PRINTTAB(2,12)"Enter filename :
";
1810 INPUTfile$
1820 ENDPROC
1900 IF ERR=222 THEN PRINTTAB(2,18)"
File not found":PRINTTAB(2,20)"Press
a key to continue":REPEAT UNTIL GET:P
ROCinpdata:GOTO 50
1910 IF ERR=204 THEN PRINTTAB(2,18)"
Bad filename":PRINTTAB(2,20)"Press a
key to continue":REPEAT UNTIL GET:PRO
Cinpdata:GOTO 50
1920 MODE 7:REPORT:PRINT" at line ";
ERL
1930 *FX4
```

SALE

BBC/ELECTRON PROFESSIONAL SOFTWARE

"Our educational software is used in thousands of schools and homes throughout Great Britain.

EDUCATIONAL 1

BBC/ELECTRON

Tape £8.00 Disc £10.00

Hours of fun and learning for children aged five to nine years. Animated graphics will encourage children to enjoy counting, maths, spelling and telling the time. The tape includes six programs: MATH 1, MATH 2, CUBECOUNT, SHAPES, SPELL and CLOCK.

... 'An excellent mixture of games'... Personal Software - Autumn 1983.

Tape £6.95 Disc £8.95

EDUCATIONAL 2

BBC/ELECTRON

Tape £8.00 Disc £10.00

Although similar to Educational 1 this tape is more advanced and aimed at seven to twelve year olds. The tape includes MATH 1, MATH 2, AREA, MEMORY, CUBECOUNT and SPELL.

Tape £6.95 Disc £8.95

FUN WITH NUMBERS

BBC/ELECTRON

Tape £8.00 Disc £10.00

These programs will teach and test basic counting, addition and subtraction skills for four to seven year olds. The tape includes COUNTING, ADDING, SUBTRACTION and an arcade type game called ROCKET MATHS which will exercise addition and subtraction. With sound and visual effects.

"These are excellent programs which teachers on the project have no hesitation in recommending to other teachers."... Computers in Classroom Project.

Tape £6.95 Disc £8.95

FUN WITH WORDS

BBC/ELECTRON

Tape £8.00 Disc £10.00

Start your fun with alphabet puzzle, continue your play with VOWELS, learn the difference between THERE and THEIR, have games with SUFFIXES and reward yourself with a game of HANGMAN.

... 'Very good indeed'... A&B Computing - Jan/Feb 1984

JIGSAW AND

SLIDING PUZZLES by P. Warner BBC/ELECTRON

Tape £7.00 Disc £9.00

There are two jigsaw and four sliding puzzles on a 3 x 3 and 4 x 4 grid. Each program starts off at an easy level to ensure initial success but gradually becomes harder. It helps children to develop spatial imagination and in solving problems. The tape includes: OBLONG, JIGSAW, HOUSE, NUMBERS, CLOWN and LETTERS.

★SPECIAL OFFER★

Buy three titles and deduct £3.00

Add 50p p&p per order. Please state BBC or ELECTRON or 40 or 80 track for discs.

Golem Ltd, Dept E, 77 Qualitas, Bracknell, Berks RG12 4QG. Tel: 0344 50720

Made in England

World Patents Pending

S. Lomo

SLOMO SPEED CONTROLLER

A UNIQUE COMPUTER TEACHING AID FOR BBC, CBM64 AND SPECTRUM

SLOW MOTION - enables the teacher to set the speed of the program to individual pupils ability. Infinitely variable speed control for games and teaching programs without the need for special software. Simply press for 'on', adjust speed, press for 'off'.

FREEZE FRAME - allows programs to be interrupted and halted at any point for discussion purposes or special emphasis. Press to 'Freeze', release to return to previously set speed.



WITH CABLE & CONNECTOR FOR FITTING DIRECTLY TO THE COMPUTER
£14.95 incl. VAT

PARTICULARLY SUITABLE FOR YOUNGER CHILDREN OR THOSE WITH SPECIAL NEEDS

ASK FOR MORE DETAILS OF SLOMO AND OUR FULL RANGE OF PERIPHERALS INCLUDING SPECTRUM JOYSTICK INTERFACES WITH INTEGRAL SLOW MOTION

We accept official orders from UK Government and educational establishments, otherwise cheque or P.O. for £14.95 to the address shown below - PLEASE STATE FOR WHICH COMPUTER REQUIRED.

NIDD VALLEY MICRO PRODUCTS LTD

Dept MUES 985 FREEPOST, KNARESBOROUGH, N.YORKS. HG5 8YZ
Telephone: Harrogate 0423-864488

A GREAT PACKAGE DEAL!

For serious users of the BBC Computer.
A professional trio which can be integrated to produce an information management system of exceptional power.

1 SCRIBE

THE PROFESSIONAL WORD PROCESSOR designed for the serious user ■ All operations fully prompted ■ No special knowledge of the computer system necessary ■ Document size NOT limited by computer memory ■ Automatic disc buffering ensures text is moved between disc and computer memory without user intervention ■ Up to 255 pages in a single document ■ 80 column display ■ on screen underline and right justify ■ See it as it will be printed. **SCRIBE** comes in a chip with 5 minute fitting instructions, utilities disc and manual. **DISC MACHINES ONLY.**
£59.95 + 60p p&p



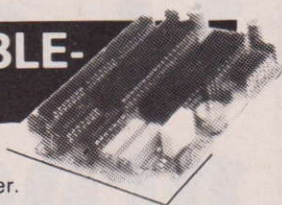
2 DATABASE

A superb information management system with an incredible operating speed. 96 fields per record ■ One record 2 Kbytes max ■ One field 900 characters max ■ 4000 records per database ■ 16 level conditional search ■ Find any record in 2 secs ■ Wild card search ■ Record match ■ 8 automatic sub indexes ■ Total flexibility of output via report writer with auto write back to any field ■ Maths pack ■ Semi programming language. Integrates with Scribe to give conditional search with MAIL MERGE and high powered report formatting. Database is in a chip. **DISC MACHINES ONLY**
£49.00 + 60p p&p.
Scribe & Database in one chip
£95.00 + 60p p&p.



3 DOUBLE-DOS

Double density disc interface for the BBC computer.



A high capacity database or word processor is enhanced even further with this state of the art disc interface. The many features include ■ almost 800 Kb of CONTINUOUS file space on a standard 80 track disc drive ■ Automatically read standard Acorn files ■ Compensate for discs formatted in single density ■ Provide up to 156 files IN ONE DIRECTORY ■ Automatically read, write and format double sided drives to appear as one disc surface ■ Allow maximum use of **MERTEC** Scribe and Database. Fitting to your computer is made exceptionally safe and reliable by providing connection via a flexible ribbon cable. Full instructions provided.
£59.95 + 90p p&p

Ask about our
"Typing Checker"
for Scribe, Wordwise
& View

NOW AVAILABLE IN ONE GREAT PACKAGE
£95^{inc} VAT
DATA SCRIBE
IN A SINGLE CHIP!!
DATA SCRIBE & DISK INTERFACE
£139.95^{inc} VAT



Purchasers of MCP software are provided with a free update service and an unmatched level of technical support. (Updates subject to postal charge.)

Please supply the following products/information (Delete as applicable).

QTY ☐ _____

QTY ☐ _____

QTY ☐ _____

In enclose £ _____

Inc P&P My Credit Card No. is _____

Visa ☐

Access ☐

Tick which

Name _____

Address _____



To: **MERLIN COMPUTER PRODUCTS,**
33/36 Singleton Street, Swansea SA1 3QN. Telephone: 0792 467980



MERLIN COMPUTER PRODUCTS

IMPORTANT NOTE:

Please check with us if you decide to use any other disc interface than Acorn or Double-Dos with Scribe or Database.

Introducing you to a very powerful programming technique

List processing using Logo

IF words are the building bricks in Logo, then lists are the buildings. In Logo a word is simply any collection of alphanumeric characters terminated by a space, which means that in Logo spaces are significant.

How the system responds to any word it encounters depends on whether or not it is preceded by a syntactic marker – Logo's equivalent of a punctuation mark.

Words which are not preceded by anything are seen by the Logo interpreter to be commands to be executed. They may be either primitives, that is the commands which the system begins with, or they may be procedures, which are lists of instructions attached to a chosen name which you have taught the system.

The point to notice here is that the system gives equal weight to primitives and procedures when it comes to interpretation. This makes Logo an extensible language.

If the word has a colon : placed immediately in front of it without any intervening space, it looks for a value attached to that name. In other words it tries to evaluate it.

If, finally, there is a quote mark " immediately in front, then the system treats this as it literally is and does nothing to it. The most convenient environment in which to work on Logo lists is the textscreen.

When you begin with Acornsoft Logo the default – the value which is always assumed – is the graphics screen. To get the textscreen, type TS and, of course, press Return. With Logotron this is not necessary since the

By
DEREK RADBURN

default is the textscreen.

Let's try some simple exercises with words. Try typing:

```
PR "GOLLY_GOSH
```

Then type:

```
MAKE "GOLLY_GOSH "SURPRISE
```

then:

```
PR :GOLLY_GOSH
```

Next try:

```
MAKE "SURPRISE "WOW!
```

We'll try a new command. See if you can work out what's happening:

```
PR THING "GOLLY_GOSH
```

followed by:

```
PR THING :GOLLY_GOSH
```

and to make sure, try:

```
PR :SURPRISE
```

If you're still not sure, well you asked for it. Try:

```
PR THING THING "GOLLY_GOSH
```

Life's full of surprises, isn't it? Now try:

```
PR WORD "GOLLY_GOSH "SURPRISE
```

and afterwards, try:

```
PR (WORD "GOLLY_GOSH "SURPRISE  
:SURPRISE)
```

Now if you've really got a sharp mind

you'll have twigged what's happening. But if you're like me, in five years time... Essentially, the WORD operator is glue. It sticks words together and makes them one.

Normally it requires two inputs, as in the way we used it in the first example. But in the second example we gave it warning to expect more than two by enclosing it, with words it had to work on inside parentheses.

Notice also that the last word was not literal like the first two, but was the value attached to a name. This means you may use suitable variables as well as literal words.

The list is the Logo structure which makes it possible for Logo to do things. Lists are words or other lists enclosed in square brackets []. Let's start to use lists:

```
PR [THIS IS A LIST]  
MAKE "TRIAL [THIS IS A LIST]  
PR :TRIAL
```

Try:

```
SHOW :TRIAL
```

Now let's introduce some glue that works on anything, in the form of the sentence operator:

```
MAKE "RUN [SO IS THIS:]
```

Now try:

```
PR SE :TRIAL :RUN
```

Notice the difference between WORD and SE. At the moment we have two lists to manipulate. However, like WORD, SE can be a greedy procedure – that is if enclosed in parentheses it can take more than two inputs.

Here's another thing about SE which distinguishes it from WORD. Try:

```
PR SE :TRIAL "SNIFFER
```

We can take lists and words to pieces as well as put them together. Try these:

```
PR FIRST "HELLO  
PR LAST "HELLO  
PR BF "HELLO  
PR WORD BF "HELLO BL "HELLO
```

Do you see what is happening? It is usual to use the shortened two letter forms of BUTFIRST and BUTLAST – indeed, in Logotron you have to.

You write a Logo procedure by typing TO followed by a space and the name you have chosen for the

procedure. Here is a simple example to try:

```
TO BORE
REPEAT 5 [PR "HELLO]
END
```

The reason for the name should soon become apparent. One feature which Logo procedures have is their ability to take local variables as parameters which are declared on the title line of the procedure. Enter the editor with your procedure BORE by typing:

```
ED "BORE
```

Once you are in there, use the cursor keys (top right of keyboard) and the Delete key to get your procedure looking like this:

```
TO BOREMORE :NAME
REPEAT [PR SE "HELLO, :NAME]
END
```

The appearance of :NAME on the top line is like a place holder, it tells the procedure that it must be given some more information when it is called if it is to work. If, after exiting the editor, you just type BOREMORE, you will get:

```
BOREMORE NEEDS MORE INPUTS
```

Just a few moments ago I referred to

local variables, I would like to expand a little – so move over a bit.

Local variables are local to the procedure in which they occur. They are not known to the system as whole or any other procedures.

The variables which we created earlier by using the MAKE command are what are termed global variables. They are known to the whole system and are available to all procedures.

In truly structured programming, which is possible in Logo, it is virtually possible to write programs which use only local variables in the form of parameters.

So, to call BOREMORE correctly we could type in, say BOREMORE "ALAN. The value ALAN would then be passed to :NAME.

I'll end by introducing a procedure which uses much of what we have just been dealing with, but in a way which requires some understanding – recursion.

This is a very powerful programming technique, which is not exclusive to Logo. It just happens that it is extremely useful in list processing.

Possibly one way of seeing recursion is to think in terms of swimming – particularly if you feel all at sea! To

swim move arms and legs then swim. Swimming is a type of recursive activity, part of its definition is an invoking of itself.

Consider, a good intellectual word:

```
TO REVERSE :OBJECT
IF EMPTYQ :OBJECT [OP "]
OP WORD LAST :OBJECT REVERSE BL
:OBJECT
END
```

(Note in Logotron – EMPTYQ would be EMPTY?)

The first thing to explain is OP, which is short for OUTPUT (it has to be OP in Logotron). The OUTPUT command causes whatever follows to be output to the calling level, either another procedure or else TOPLEVEL.

TOPLEVEL is the interactive command level we have been working at most of the time so far – that is you type something and the computer responds.

In this procedure OP is unable to output and terminate the running procedure because when it is called part of what it needs to output has to be provided by a recursive call to the running procedure itself.

Only when the word has become empty does the recursive spiral for OP cease, and then it is able to spiral back up again.

The easiest way to see this is to run the procedure, but to type TRACE 1 first (Logotron users should just type TRACE). Try this:

```
PR REVERSE "ACORN
```

'Now if you have TRACE on you will see each input as the procedure winds downwards into its recursion. This should then be followed by a building up of its output once it has reached the empty state with its input.

Try it out and see it work with other inputs. You might like to try:

```
PR REVERSE REVERSE "ACORN
```

If you have the TRACE on, you will see what happens. To remove the TRACE type TRACE 0 (or in Logotron just type TRACE – it's a toggle).

The purpose of this article was to give some idea of what list processing is. It was an attempt to de-mystify, while at the same time not removing the self-learning element that is fundamental to the philosophy of Logo.

I hope that you have acquired a taste to explore.



RECURSION: If you're all at sea, think in terms of swimming...

BIZZELL'S BLOCKBUSTER BARGAINS



**HIGH SPEED, HIGH QUALITY, FOUR COLOUR, FLATBED.
PRINTER/PLOTTER Only £399**

- Prints horizontally or vertically in text.
- Prints four directions and 255 sizes in graphics.
- Plotting area 298 x 216 mm.
- Plotting speed 100mm/sec.
- Step size 0.1mm.
- Self test and auto functions.
- Red, Green, Blue and Black.
- Parallel and RS232C interfaces.
- Full guarantee.
- BASIC ASCII command codes and parameters.
- Full manual (cassette and commands for BBC Micro).

Ideal for graphs, drawings, symbols, axes, geometric patterns, charts, diagrams, circuits, computer art, flowcharts, 3-D, contours etc.

RD DIGITAL TRACER

Produce illustrations on your computer simply and easily using the tracer to copy existing material. Simply run the tracer head over anything you want to produce. Complete with full instructions, manual and software. BBC £69.95



MICAD

Computer Aided Design program for the BBC B creates sophisticated drawings quickly and easily using lines, arcs and circles hatching and text of any size. Edit and store your drawing and output them to your plotter. Two dimensional and three dimensional shapes can be created. £34.44

All prices include VAT and delivery. UK mainland. Write with order and cheque (stating computer) to:

**BIZZELL COMPUTERS
WALNUT TREE HOUSE,
FORNCET, ST. PETER,
NORWICH NR16 1HR.**

PERSONAL CALLERS WELCOME (10.00-5.00) Phone for appointment (095389-592)

M-J SOFTWARE

EDUCATIONAL PROGRAMS

We have a wide range of Software on both Cassette & Educational Disk.

Produced by leading Software Houses to meet the needs of Children and Students of all ages.

Send today for your copy of our catalogue.

We have programs for the

**BBC Micro, Electron, Spectrum, Dragon,
Commodore 64, Amstrad and Vic20.**

Please send a copy of your catalogue.

We have a Computer

Name

Address

Town Postcode

Post to:

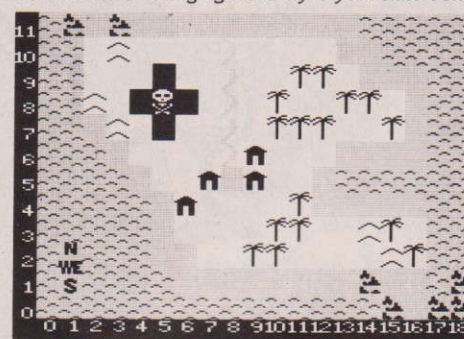
MU1

**MICRO JENN SOFTWARE LTD
81 Squirrels Heath Road, Harold Wood, Essex RM3 0LR**



HUNT THE TREASURE

Maths with a difference - bringing creativity to your classroom computer.



A do-it-yourself games program based on using co-ordinates to make treasure hunts. It encourages logical thinking, design and language skills and is sufficiently flexible to present a challenge to children aged 8 to 13.

BBC "B" and ELECTRON

Disc (specify 40/80) £11.50
and user guide inclusive

Tape £9.50

Teachers' notes and planning sheets 0.50p

Other titles available on request.

**Hilditch Software, 4 Church Road, Felixstowe, Suffolk IP11 9NF.
Tel: Felixstowe (0394) 271747**

MicroLink

*Your personal passport to the
world of communications with*

TELECOM GOLD

Telecom Gold is a trademark of British Telecommunications plc

What it offers the BBC Micro user...

Give your micro mainframe power

With MicroLink your micro becomes a terminal linked directly to the Telecom Gold mainframe computer, and able to tap its tremendous power and versatility. Right away you'll be able to use giant number-crunching programs that can only run on a mainframe. You can set up your own computerised filing systems, store and update statistics and other information, cross-reference material between files, selectively extract the information you want, perform massive calculations and design reports to display information from any of the files and in any format you choose.

The biggest bulletin board of them all

The number of bulletin boards is growing rapidly. New ones are springing up in all parts of Britain and all over the world, with people of like minds chatting to each other on all manner of subjects. The only snag is that the vast majority are single-user boards – which means lots of other people are also trying to make contact and all too often all you get is the engaged tone. But with the MicroLink bulletin board there is no limit to the number of people using it at the same time. And no limit to the number of categories that can be displayed on the board.

We're only a local phone call away

More than 96 per cent of MicroLink subscribers can connect to our mainframe computer in London by making a local phone call. This is possible because they use British Telecom's PSS system, which has access points all over Britain. A local phone call is all you need, too, for access to the international Dialcom system through MicroLink.

Telemessages – at a third of the cost

The modern equivalent of the telegram is the telemessage. Send it before 10pm and delivery is guaranteed by first post the following day (except Sunday). The service was intended for people phoning their message to the operator, and it costs £3.50 for 50 words. But you can now use it via MicroLink for only £1.25 for up to 350 words!

Send and receive telex messages

With MicroLink you can turn your micro into a telex machine, and can send and receive telex messages of any length. You will be able to

communicate directly to 96,000 telex subscribers in the UK, 1½ million worldwide – and even with ships at sea via the telex satellite network. Business people can now send and receive telexes after office hours, from home or when travelling. You can key in a telex during the day and instruct MicroLink not to transmit it until after 8pm – and save 10 per cent off the cost!

The mailbox that is always open

MicroLink is in operation 24 hours a day, every day. That means you can access your mailbox whenever you want, and from wherever you are ... home, office, airport – even a hotel bedroom or golf club! No-one needs to know where you are when you send your message.

What does it all cost?

Considering all the services you have on tap, MicroLink is remarkably inexpensive. You pay a once-only registration fee of £5, and then a standing charge of just £3 a month. On-line costs are 3.5p a minute (between 7pm and 8am) or 10.5p a minute during office hours. There is an additional 2p a minute PSS charge if you are calling from outside the 01- London call area. Charges for telex, tele-messages and storage of files are given on the next page.

How much it costs to use MicroLink

Initial registration fee: £5.

Standing charge: £3 per calendar month or part.

Connect charge: 3.5p per minute or part – cheap rate; 10.5p per minute or part – standard rate.

Applicable for duration of connection to the Service. Minimum charge: 1 minute.

Cheap rate is from 7pm to 8am, Monday to Friday, all day Saturday and Sunday and public holidays; Standard rate is from 8am to 7pm, Monday to Friday, excluding public holidays.

Filing charge: 20p per unit of 2,048 characters per month.

Applicable for storage of information, such as telex, short codes and mail files. The number of units used is an average calculated by reference to a daily sample.

Information Databases: Various charges.

Any charges that may be applicable are shown to you before you obtain access to the database.

MicroLink PSS service: 2p per minute or part (300 baud); 2.5p per minute or part (1200/75 baud).

Only applies to users outside the 01-London call area.

Telex registration: £10.

Outgoing telex: 5.5p per 100 characters (UK); 11p per 100 (Europe); 16.5p per 100 (N. America); £1.15 per 400 (Rest of world); £2.75 per 400 (Ships at sea).

Deferred messages sent on the night service are subject to a 10 per cent discount.

Incoming telex: 50p for each correctly addressed telex delivered to your mailbox. Obtaining a mailbox reference from the sender incurs a further charge of 50p.

It is not possible to deliver a telex without a mailbox reference. If a telex is received without a mailbox reference the sender will be advised of non-delivery and asked to provide a mailbox address.

Each user validated for telex and using the facility will incur a charge of 6 storage units a month. Further storage charges could be incurred depending on the amount of telex storage and the use made of short code and message file facilities.

Telemessages: £1.25 for up to 350 words.

Radiopaging: No charge.

If you have a BT Radiopager you can be paged automatically whenever a message is waiting in your mailbox.

International Mail: For the first 2,048 characters – 20p to Germany and Denmark; 30p to USA, Australia, Canada, Singapore, Hong Kong and Israel. For additional 1,024 characters – 10p; 15p.

These charges relate to the transmission of information by the Dialcom service to other Dialcom services outside the UK and the Isle of Man. Multiple copies to addresses on the same system host incur only one transmission charge.

Billing and Payment: All charges quoted are exclusive of VAT. Currently all bills are rendered monthly.

Software over the telephone

MicroLink is setting up a central store of software programs which you'll be able to download directly into your micro. The range will include games, utilities, educational and business programs, and will cover all the most popular makes of micros.

Talk to the world – by satellite

MicroLink is part of the international Dialcom network. In the USA, Australia and a growing number of other countries there are many thousands of users with electronic mailboxes just like yours. You can contact them just as easily as you do users in Britain – the only difference is that the messages from your keyboard go speeding around the world via satellite.

What you need to access MicroLink

You must have three things in order to use MicroLink: a computer (it can be any make of micro, hand-held device or even an electronic typewriter provided it has communications facilities), a modem (it can be a simple Prestel type using 1200/75 baud, or a more sophisticated one operating at 300/300 or 1200/1200 baud), and appropriate communications software.

MicroLink

in association with

TELECOM GOLD *

Application Form

I/We hereby apply to join MicroLink

(✓) ☐ I enclose my cheque for £5 payable to Database Publications as registration fee to MicroLink.

(✓) ☐ I also wish to use Telex. I authorise you to charge an additional £10 to my initial bill for validation.

☐ I confirm that I am over 18 years of age.

Signature _____

Date _____

I intend to use the following computer _____

FOR OFFICE USE ONLY:

Mailbox assigned _____

Start date _____

Password _____

SEND TO:
MicroLink
Database Publications
Europa House
68 Chester Road
Hazel Grove
Stockport SK7 5NY.

MU ED

Name

Position

Company

Address

Postcode Daytime telephone

Commencement of Service

Please indicate month of commencement
Allow 10 days for validation of mailbox

19

Payment

Whilst Database Publications Ltd is the supplier of all the services to you, the commission and billing thereof will be handled by Telecom Gold as agents for Database Publications Ltd.

Date of first payment to be on 15th of month following commencement.

Please complete billing authorisation form A, B or C below:

A. Direct Debiting Mandate (Enter full postal address of Bank Branch)

To

I/We authorise you until further notice in writing to charge to my/our account with you on or immediately after 15th day of each month unspecified amounts which may be debited thereto at the instance of British Telecommunications plc – TELECOM GOLD by Direct Debit. Bills are issued 10 days before debit is processed.

Name of Account to be debited

Account Number

B. Please debit my/our Access/Visa account number

I/We authorise you until further notice in writing to charge to my/our account with you on or immediately after 15th day of each month unspecified amounts which may be debited thereto at the instance of British Telecommunications plc – TELECOM GOLD. Bills are issued 10 days before charge is applied to your account.

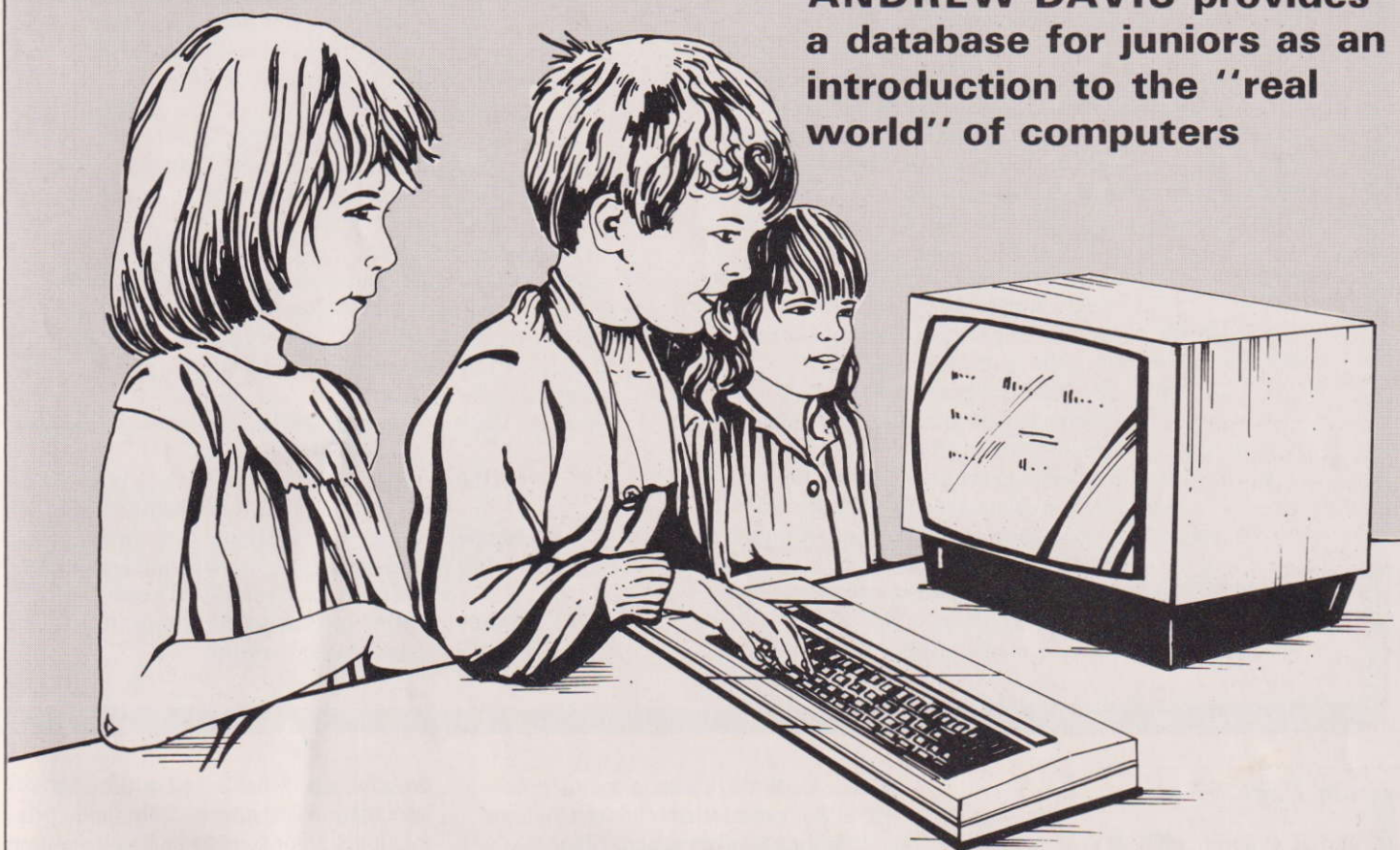
C. Please invoice the company/authority.

(✓) ☐ If you select this option, which is ONLY AVAILABLE to government establishments and public limited companies, you will be sent an authorisation form for completion which will require an official order number to accept unspecified amounts.

* Telecom Gold is a trademark of British Telecommunications plc.

A base for your data

ANDREW DAVIS provides a database for juniors as an introduction to the "real world" of computers



COMPUTERS have proved to be exciting resources in the primary classroom, helping children to learn in new ways the sorts of things they have always learned. But many teachers know that we need not stop there.

In the real world, computers have many important functions which bear little relation to the learning package software many of us use in school.

One important function to which computers in their present stage of development are well adapted is that of storing large amounts of information, and allowing the user to access it in a variety of ways which are convenient to him.

Filing and databases are commonplace in the office and in industry.

It seems a good idea then if we can introduce children as soon as possible to these "real world" computer uses.

Often there turn out to be important "education" aspects as well. This is very obviously true of databases.

The Chiltern M.E.P. group produced a well-known and widely-respected data-base program called Quest.

It comes in three versions. One works on the BBC Micro, the second on

cassette-based 380Z and 480Z Research Machines, and the third on 480Z Research Machines having 256k of memory.

The program is a powerful package, allowing quite complex questioning of

Whole classes can work together on building up files on animals, trees, friends and so on

stores of information held by the computer.

Even the closest friends of this program, however, would probably agree that some of the language the user has to employ is somewhat stilted and tortuous.

Because of this there seemed to be a need for a much simpler and more user-friendly program which young children could use, which would lead

them later into Quest or similar programs.

The program offered here, Junior Data Base, attempts to meet this need.

It doesn't do a fraction of what Quest does – and it is not supposed to.

It does provide a format which is easy to use, with a simple data matching facility which young children fairly readily come to understand.

The program has been tried in school with some success and a lot of care has been taken with the "user interface":

- Any text entries are turned into upper-case letters. This is not ideal for young children, admittedly, but we wanted to avoid matching failures occasioned by the inconsistent use by children of capitals and lower case.
- Extra spaces are stripped off.
- Irrelevant areas of the keyboard are disabled, hence making the program very difficult to crash.

Whole classes can work together in building up files of information on animals, trees, weather, friends, buildings, transport and so on.

Large files consume a lot of time and labour if it is just one person putting it

MAIN PROGRAM

10 PROCc	Turns off the cursor; the cursor keys are also disabled here.	
20 PROCe	Deals with errors, and returns the user to the main menu.	
30	<i>FS(100,9)</i> is declared here – it reserves space for all the details of the records and fields. <i>sort\$(19,2)</i> is used in PROCask, the procedure which allows the user to interrogate the data base. The array holds various bits of information to enable those records which match the questioner's specifications to be filtered out from the rest.	
50 PROCintro	Calls the introductory menu.	
80	The main menu. PROCdh is used widely. It prints in double height to a specified colour.	
130	If questions are to be asked, the array <i>sort\$</i> is cleared of any garbage, since it is used to store information during the attempt to match.	
160 PROCn	Is the main procedure called when you start to build up a new file. PROCi (defined at line 200) is an all-purpose	
		input routine for numbers. It outputs <i>a\$</i> , which is VAled and the numerical result put into <i>i%</i> for the number of records to be made up, and into <i>f%</i> for the number of fields under which the records are to be described.
		170 Invites the user to supply the descriptions of the fields.
		180 Systematically invites the user to supply information for each record under all the fields.
	280 PROCt	Time delay procedure.
	290 PROCT	This is an all-purpose text input routine, used throughout the program. <i>Q\$</i> contains the prompt, and <i>L%</i> the permitted length in characters of the input.
		330 Changes any lower case input into upper case.
		350 Prevents accidental typing in of extra spaces in the middle of input.
	370,380	Looks at the input, contained in <i>a\$</i> , and strips off any unwanted spaces on the right hand side of the word.
	390,400	Strips off unwanted spaces on the left hand side of the word.

From Page 67

in, but if children take it in turns for 15-20 minutes each, you will have masses of information after a day or so.

Incorrect entries can always be reviewed later.

Children can then be set information-retrieval tasks . . . "Are there any animals which live in the sea *and* suckle their young? "How many easy books on space travel are there in our book corner?" and so on.

Graphs and bar charts can be made using information gathered from questioning the computer.

Very valuable discussion can be had by groups of children about the kinds of fields to store records under, and then about interesting problem cases of classification.

When starting out, children will certainly need help before they can understand the notions of record, field and file.

It is suggested that these "technical" words may as well be used from the start with the children – there are no obvious "ordinary language" synonyms.

The program can be used with either disc or tape, but disc is recommended.

In addition to the keys specified at various points in the program, the Escape key has the function of taking

the user back to the main menu.

It is suggested that children put in small amounts of data at a time.

It's a good idea to save your data onto cassette or disc quite often. In this way you may hope to avoid wasting work through some accident.

To understand what a record is, think of a library record card. Each book has its own card.

You could classify the book by subject, author, publisher, and so on. These classifications would correspond to the fields in this database program.

Here is an example of a record.

NAME ANDREW DAVIS ADDRESS CAMBRIDGE HOBBY FISHING

It is described in terms of just three fields – name, address and hobby.

This program allows you to have up to ten fields.

A maximum of 100 records is theoretically possible, but if records are long, don't try to go up to the maximum!

It is impossible to say in advance exactly how many records you can store. It depends on how many fields you have, how long your field descriptions are, and how much information you put in under each field.

As a very rough guide, using long

descriptions of fields and putting lots of information in under each field, you might get 36 or so records if you wanted 10 fields.

This is not too much of a limitation, since it probably means 72 records with five fields, and even more if you try to keep the descriptions of the fields shorter, and/or put less information in.

If by some unlucky chance you do run out of space, you will get a message saying No Room. Anything you have just typed in will be lost, and you will have to run the program again.

The implication of this is that you should add small amounts of information at a time when your data-base is beginning to get large.

Each time you add such an amount, the updated file should be saved.

The description of a field can be up to 20 characters long. Remember that characters include spaces.

The information stored for each record under a particular field can be up to 40 characters long.

If you exceed these limits, the computer simply repeats the prompt for you to type in the information.

Files can be updated at any time. They might be newly created, or they might have been stored on disc or tape on a previous occasion and now loaded in for revision.

430 PROC R Allows user to look through all the records entered in so far.

460 PROC dh Procedure for producing double-height characters.

470 PROC sp Procedure for halting program at prompt "Press Space to go on".

480 PROC ask The routine for questioning the data base.

510 Permits lower or upper case Y/N response, and highlights the fields selected for inquiry in flashing green.

550 *sort\$(T%,1)* contains the number of the fields selected for inquiry. When used in *F\$(0,VAL(sort\$(T%,1)))* the description of the field is called up from the main array. *sort\$((T%,0))* is used to hold the specific bits of information the questioner is trying to get the data base to match.

560,580 Loops which attempt the matching.

600,610 Records the number of matching records found, and prints them out.

630 PROC file Saves files created with the program to disc or tape.

640 PROC load Loads files previously created with the program from disc or tape for viewing

or updating.

660 PROC Y A subsidiary menu called up when you are altering or adding to files.

690,740 The routine for changing information in a record.

760 PROC error An attempt to catch one or two of the more likely classroom errors.

830 PROC add For adding to records and fields.

850 The number of fields (*f%*) is set to be the old number of fields plus the extra fields now required. *mf%* contains the extra number of fields requested.

860 In this loop, the user enters in the descriptions of the new fields, if any.

870 If there are to be new fields, PROCZ (defined at line 900) takes the user through the records to put in information on them under each new field.

880 If there are to be new records, this loop invites the user to supply information for them under all the fields.

930 PROC X A procedure for changing the descriptions of fields.

960 FN Rand Function for supplying colour numbers which change appropriately. Used in the printing of the double-height text.

The word used for a field description may be altered, or new fields added or new records added.

The option "Ask me about a file" allows children to experience one simple aspect of searching a data base.

The computer will only come up with records which match all the specifications given. Thus, on the whole, the more specifications you give, the less records will match them.

The program comes in two sections. The first section Listing I consists of the instructions, whilst the second Listing II

makes up the main program. When using Junior Data Base in your classroom, you may prefer to leave out the first section altogether.

Ideally, however, the instructions should be retained on the same disc or tape for convenient reference, and it is important that lines 110 and 420 contain the filename under which you have saved the main program. We have used the filename DATABASE.

I make no comment on the programming of the first section, since it merely involves printing text.

VARIABLES

f% Number of fields.
i% Number of records.
mf% Extra fields requested.
mi% Extra records requested.
F\$(n,n) String array storing fields and information.
sort\$(n,n) String array storing information about records which match questions.

Listing I

```
5 REM (C) THE Micro User
10 MODE7:VDU23;8202;0;0;0;
12 PROCTITLE
14 END
20 DEFPROCTITLE
30 PRINT CHR$130;STRING$(38,"#")
40 PRINT CHR$133;CHR$141;"Junior Database":PRINT CHR$133;CHR$141;"Junior Database"
50 PRINT CHR$130;STRING$(38,"#")
60 PRINT CHR$134;CHR$141;"By Andrew Davis, Marion Abiks":PRINT CHR$134;CHR$141;"By Andrew Davis, Marion Abik
```

```
5"
70 PRINT CHR$134;CHR$141;"and Mike Aylen":PRINT CHR$134;CHR$141;"and Mike Aylen"
80 PRINT "Type 'I' for INSTRUCTIONS, or 'press RETURN for the program"
90 REPEAT:AX=GET:UNTILAX=73ORAX=13
100 IFAX=13AND PAGE=&E00 PRINT "Press PLAY on your tape recorder"to load the main program."
110 IFAX=13CHAIN"DATABASE":ENDPROC
120 CLS
```

```
130 PRINTCHR$130;CHR$141"These instructions are for teachers.":PRINTCHR$130;CHR$141"These instructions are for teachers."
140 PRINT"The program allows small data bases ""to be built up, and f or the user to ""ask simple questions of them."
150 PRINT""Use either with disk or tape.""Disk is recommended."
160 PROCs:PRINT""Pressing ESCAPE a
```


From Page 69

always takes you to the "main MENU."
170 PRINT "It is suggested that children put in "small amounts of data at a time." :PROCs

180 PRINT "It's also a good idea to save your data "on to cassette or disk quite often." "In this way, you may hope to avoid "wasting work through some accident."

190 PROCs

200 PRINT "To understand what a RECORD is, think "of a library record card." "Each book has its own card." "You could classify the book by"

210 PRINT "SUBJECT, AUTHOR, PUBLISHER, ETC." "These classifications would "correspond to the FIELDS" "in this Data base program." :PROCs

220 PRINT "Here is an example of a Record." :PRINTCHR\$130"RECORD 3":PRINTCHR\$131;"NAME";" ";CHR\$133;"ANDREW DAVIS":PRINTCHR\$131;"ADDRESS";" ";CHR\$133;"CAMBRIDGE":PRINTCHR\$131;"HOBBY";" ";CHR\$133;"FISHING"

230 PRINT "It is described in terms of just three "fields, i.e. NAME, ADDRESS, and "HOBBY. This program allows you to "have up to ten fields."

240 PROCs

250 PRINT "A maximum of 100 records is "theoretically possible but if "records are long, don't try "to go up to the maximum!"

260 PRINT "It is impossible to say

in advance" "exactly how many records you can "store. It depends on how many fields "you have, how long your field names are" "and how much information you put in" "under each field." :PROCs

270 PRINT "As a VERY ROUGH guide, using long "descriptions of fields and putting "lots of information under each "field, you might only get 36 or so "records if you wanted ten fields."

280 PRINT "This is not much of a limit, since it "means 72 records with five fields" "and more if you try to keep the "description of the field shorter, "and/or put less information in." :PROCs

290 PRINT "If you do by some unlucky chance run "out of space, you will get a "message saying NO ROOM. Anything you "have just typed in will be lost, and "you will have to RUN the program "again."

300 PRINT "The implication of this is that you "should add small amounts at a time "when your data base is getting large." "Each time you add such an amount, "the updated file should be SAVED." :PROCs

310 PRINT "The description of a field can be up "to 20 characters long. (A character is "a letter, number, SPACE, or indeed "anything you can put on the screen "with a single key press.)"

315 PRINT "The information stored for each record "under a particular

field can be up "to 40 characters long."

320 PRINT "If you exceed these limits, the "computer simply repeats the prompt "for you to type in the information." :PROCs

350 PRINT "Files can be updated at any time. They "might be newly created, or they "might have been stored on disk or "tape on a previous occasion and now "loaded in for revision."

360 PRINT "The word used for a field may be "altered, or new fields added, or "new records added."

370 PROCs:PRINT "The option 'Ask me about a file' allows "children to experience one simple "aspect of searching a data base." "The computer will only come up with "records which match ALL the"

380 PRINT "specifications given. Thus on the whole "the more specifications you give, "the less records will match them." :PROCs

390 PRINT "You have to give files to be saved a "name. This can be anything you like "but not longer than 7 letters."

400 PRINT "Do you want these instructions again ?" :Y/N":REPEAT:AX=GET:UNTILAX=89ORAX=78:IFAX=89GOTO120

410 IF PAGE=6E00 PRINT "Press PLAY on your tape recorder, and "the main program will now load."

420 CHAIN "DATABASE":END

430 DEFPROC:PRINT "PRESS SPACE":REPEAT:AX=GET:UNTILAX=32:CLS:ENDPROC

Listing II

```
10 MODE7:PROCc:FX4,2
20 VDU15:PROCc
30 DIM sort$(19,2):DIM F$(100,9)
40 VDU12,15:PROCc:FX11,0
50 PROCIntro
70 REPEAT:CLS:PRINT
80 PROCdh("1.I show you the file",
6):PROCdh("2.You make changes in the
file",6):PROCdh("3.You ask me about the
file.",6):PROCdh("4.I save your file",
6):PROCdh("5.Start again",6)
90 PRINT:PROCdh("Type the number".
5):REPEAT:AX=GET:UNTILAX>48ANDAX<54:VDU15
100 IFAX=53PROCIntro
110 IFAX=49PROCc
120 IFAX=50PROCY
130 IFAX=51FOR Y%=0TO19:FOR X%=0TO2:s
```

```
ort$(Y%,X%)="":NEXT:NEXT:PROCask
140 IFAX=52PROCfile
150 UNTILFALSE:END
160 DEFPROCn:CLS:PROCi("How many records
do you want?",100):i%=VAL(a%):PROCi("How
many fields do you want",10):f%=VAL(a%):CLS
170 C%=1:REPEAT:PROCT("What is field "+STR$(C%)+
"?",20):F$(0,C%-1)=a%:C%=C%+1:CLS:UNTILC%>f%
180 C%=1:REPEAT:D%=1:REPEAT:PROCdh("Record "+STR$(C%),2):PROCT("What is the "+F$(0,D%-1)+
"?",40):F$(C%,D%-1)=a%:VDU12,7:D%=D%+1:UNTILD%>f%:C%=C%+1:UNTILC%>i%:CLS
190 ENDPROC
200 DEFPROCi(Z$,N%):REPEAT:PRINT "Z$ "
Type the number":VDU7:a$="":REPE
```

```
AT:REPEAT:AX=GET:UNTIL(AX)>47ANDAX<58)
ORAX=13ORAX=127
210 IFAX<>13ANDAX<>127PRINT:CHR$AX;
220 IFAX=127ANDPOS>0PRINT:CHR$127;:a$=LEFT$(a$,LEN(a$)-1)
230 IFAX<>127ANDAX<>13a$=a$+CHR$AX
240 IFAX=13VDU10,13
250 UNTILLEN(a$)>30ORAX=13
260 IFVAL(a$)>N%PRINTCHR$136;CHR$(128+RND(6));"Too many for me!";VDU7:PROCT(200)
270 UNTILNOT(VAL(a$)>N%)AND LEN(a$)>0:VDU10,13:ENDPROC
280 DEFPROCT(T%):TIME=0:REPEATUNTILTIME>T%:ENDPROC
290 DEFPROCT(Q$,L%):PRINT:REPEAT:a$="":PROCdh(Q$,FNrand):PROCdh("Type it in and press RETURN.",S%):REPEAT:AX=
```



```

GET
300 IFAX<>127PRINT;CHR$A%;
310 IFAX=127ANDPOS>0PRINT;CHR$127;
320 IFAX=127ANDLEN(a$)>0 a$=LEFT$(a$,LEN(a$)-1)
330 AX=FNuc(AX)
340 IFAX<>127ANDAX<>13ANDAX<>32a$=a$+CHR$AX
350 IFAX=32ANDRIGHT$(a$,1)<>" " a$=a$+" "
360 UNTILLEN(a$)=LXORAX=13
370 REPEAT:IFRIGHT$(a$,1)=" " a$=LEFT$(a$,LEN(a$)-1)
380 UNTILRIGHT$(a$,1)<>" "
390 REPEAT:IFLEFT$(a$,1)=" " a$=RIGHT$(a$,LEN(a$)-1)
400 UNTILLEFT$(a$,1)<>" "
410 IFLEN(a$)>LXVDU13,11:PRINT SPC(15)+CHR$13
420 UNTILNOT(LEN(a$))>LXORAX=13:VDU10,13:ENDPROC
430 DEFPROCRC:VDU14,12:PRINT"Press S HIFT to move through the records"
440 CX=1:REPEAT:PRINT"CHR$130;"Record Number "+STR$(CX):DX=1:REPEAT:PRINTCHR$131;F$(0,D%-1);" ";CHR$133;F$(CX,D%-1):DX=DX+1:UNTILDX>f%:PRINT"CHR$134;STRING$(38,"*"):CX=CX+1:UNTILCX>i%
450 VDU15:PROCSP:ENDPROC
460 DEFPROCdh(Z$,CX):PRINT:PRINTCHR$(128+CX);CHR$141;Z$:PRINTCHR$(128+CX);CHR$141;Z$:PRINT:ENDPROC
470 DEFPROCsp:PRINT"Press SPACE to go on":REPEAT:AX=GET:UNTILAX=32:ENDPROC
480 DEFPROCask:VDU12,15:PROCdh("Ask ing questions",6):FORCX=0TOF%-1:PRINT" ";CHR$131;F$(0,CX):NEXT
490 DX=0:P$="":PRINT:FORCX=0TOF%-1:PRINTTAB(0,f%+6)"Do you want to ask a bout "+F$(0,CX);
500 PRINT"Type Y or N ";
510 REPEAT:AX=GET:UNTILAX=89ORAX=121ORAX=78ORAX=110:IFAX=89ORAX=121 DX=0%+1:sort$(DX,1)=STR$CX:PRINT;CHR$129;"Yes":PRINTTAB(0,4+CX);CHR$136;CHR$130;F$(0,CX)
520 IFAX=78ORAX=110PRINT;CHR$130;"N o"
530 PRINTTAB(0,f%+6);SPC(120):VDU11,11,11,7
540 NEXT:IFDX=0ENDPROC
550 CLS:FORTX=1TOOX:PROCT("Which "+F$(0,VAL(sort$(TX,1))),39):sort$(TX,0)=a$:NEXT
560 TAG=0:FORCX=1TOi%:count=0:FORTX=1TOOX:IFsort$(TX,0)=F$(CX,VAL(sort$(TX,1)))count=count+1
570 NEXT:IF count=0% TAG=TAG+1:sort$(TAG,2)=STR$CX
580 NEXT
590 IF TAG=0PRINT"Sorry. I can't help you with your""question.":PROCT(500):ENDPROC
600 CLS:IF TAG=1 THEN PROCdh("I can find "+STR$(TAG)+" matching record",6)ELSE PROCdh("I can find "+STR$(TAG)+" matching records",6)
610 VDU14:FORTX=1TO TAG:PRINT:FORDX=0TOF%-1:PRINT F$(0,D%):V%=VPOS:PRINTTAB(LEN(F$(0,D%))+2,V%-1)F$(VAL(sort$(TX,2)),D%):NEXT:NEXT:VDU15
620 PROCSP:ENDPROC
630 DEFPROCfile:CLS:PRINT":PROCT("Name of file to be saved",8):X=OPENOUT(a$):PRINT#X,i%,f%:FORAX=0TOi%:FORBX=0TOF%-1:PRINT#X,F$(AX,BX):NEXT:NEXT:CLOSE#X
640 DEFPROCload:PROCT("Name of file to be loaded?",8):X=OPENUP(a$):INPUT#X,i%,f%:FORAX=0TOi%:FORBX=0TOF%-1:INPUT#X,F$(AX,BX):NEXT:NEXT:CLOSE#X
650 VDU12,7:PROCdh("Your file is loaded",2):PRINT":PROCSP:ENDPROC
660 DEFPROCv:CLS:PROCdh("1. Add to your file",6):PROCdh("2. Change something in a record",6):PROCdh("3. Change a field name.",6):PROCdh("Type the number",5):REPEAT:AX=GET:UNTILAX>48ANDAX<52
670 IFAX=49PROCadd:ENDPROC
680 IFAX=51PROCx:ENDPROC
690 CLS:REPEAT:PROCI("Which record is to be changed?",i%):CX=VAL(a$):UNTILCX>0
700 CLS:PRINT"CHR$130;"Record Number "+STR$(CX):DX=1:REPEAT:PRINTCHR$131;F$(0,D%-1):V%=VPOS:PRINTTAB(LEN(F$(0,D%-1))+2,V%-1)CHR$133;F$(CX,D%-1):DX=DX+1:UNTILDX>f%
710 DX=1:REPEAT:PROCC1:PROCdh("Change "+F$(0,D%-1)+"?",8%):PRINT"Type Y or N":REPEAT:GX=GET:UNTILGX=89ORGX=78
720 IFGX=89PROCC1:PRINT CHR$136;CHR$129;F$(CX,D%-1):PROCT("What do you want to change this to?",39):F$(CX,D%-1)=a$
730 CLS:PRINT"CHR$130;"Record Number "+STR$(CX):KX=1:REPEAT:PRINTCHR$131;F$(0,KX-1):V%=VPOS:PRINTTAB(LEN(F$(0,KX-1))+2,V%-1)CHR$133;F$(CX,KX-1):KX=KX+1:UNTILKX>f%
740 DX=DX+1:UNTILDX>f%
750 ENDPROC
760 DEFPROCerror:VDU15:IF ERR=222 PRINT"CHR$136;CHR$130;"I can't find the file. Change disks?":PROCT(600)
770 IF ERR=194 PRINT"CHR$136;CHR$130;"You can't load that sort of file here.":PROCT(600)
780 IF ERR=198 PRINT"CHR$136;CHR$130;"Disk full! Change your disk.":PROCT(600)

```

```

t(600)
790 ENDPROC
800 DEFPROCe
810 ONERRORPROCerror:GOTO70
820 ENDPROC
830 DEFPROCadd:CLS:F%=f%:I%=i%
840 PROCI("How many more records do you want"+CHR$13+CHR$10+CHR$10+"stored?",100-i%):i%=VAL(a$)+i%:mi%=VAL(a$):PROCI("How many more fields do you want",10-f%)
850 f%=VAL(a$)+f%:mf%=VAL(a$)
860 IFmf%>0CX=f%+1-mf%:CLS:REPEAT:PROCT("What is field "+STR$(CX)+"?",20):F$(0,CX-1)=a$:CX=CX+1:UNTILCX>f%
870 IFmf%>0PROCCZ
880 IFmi%>0CX=i%+1-mi%:REPEAT:VDU7:DX=1:REPEAT:CLS:PROCdh("Record Number "+STR$(CX),2):PROCT("What "+F$(0,D%-1)+"?",40):F$(CX,D%-1)=a$:DX=DX+1:UNTILDX>f%:CX=CX+1:UNTILCX>i%
890 ENDPROC
900 DEFPROCZ:CX=1:REPEAT:CLS:PRINT"CHR$130;"Record Number "+STR$(CX):DX=1:REPEAT:PRINTCHR$131;F$(0,D%-1);" ";CHR$133;F$(CX,D%-1):DX=DX+1:UNTILDX>(f%-mf%):DX=1+f%-mf%:REPEAT
910 PROCdh("Record Number "+STR$(CX),2):PROCT("What "+F$(0,D%-1)+"?",40):F$(CX,D%-1)=a$:DX=DX+1:UNTILDX>f%:CX=CX+1:UNTILCX>(i%-mi%)
920 ENDPROC
930 DEFPROCX:CLS:DX=0:REPEAT:PRINT;CHR$131;CHR$(65+DX);". ";F$(0,D%):DX=DX+1:UNTIL DX=f%:PRINT"Which one do I change? Type its letter.":REPEAT:AX=GET:UNTILAX>64ANDAX<(65+f%)OR(AX>96ANDAX<123):AX=FNuc(AX):PROCdh(CHR$AX,5)
940 DX=AX-65:PROCT("Type what you want instead.",20):F$(0,DX)=a$:ENDPROC
950 DEFPROCc:VDU23;8202;0;0;0;0:ENDPROC
960 DEF FNrand:S%=S%+1:IFS%>6 S%=1
970 =S%
980 DEFPROCc:FORXX=0TO100:FORY%=0TO9:F$(XX,Y%)="":NEXT:NEXT:f%=0:i%=0:ENDPROC
990 DEF FNuc(AX):IFAX>96ANDAX<123AX=AX-32
1000 =AX
1010 DEFPROCc1:VDU31,0,13:FORDX=1TO2:PRINT;SPC200;NEXT:VDU31,0,13:ENDPROC
2000 DEFPROCintro:VDU12,10,10:PROCdh("1. Start a new file.",6):PROCdh("2. Load a file",6):PRINT"Press 1 or 2.":REPEAT:AX=GET:UNTILAX=49ORAX=50:IF AX=49 PRINT"Please wait.":PROCc:PROCCn
2010 IFAX=50PROCload
2030 ENDPROC

```


You've read the magazine - now run the programs!

We've managed to squeeze all the programs from this Micro User Education Special into a software package that's packed with educational value.

It's available on tape at £4.95, on 5¼" disc at £6.95 and on 3" disc at £8.95.

At these prices it's an offer no teacher or parent can afford to refuse.

Our first issue was a sell-out!

The first issue of The Micro User's Education Special was so popular with teachers and parents that it has had to be reprinted.

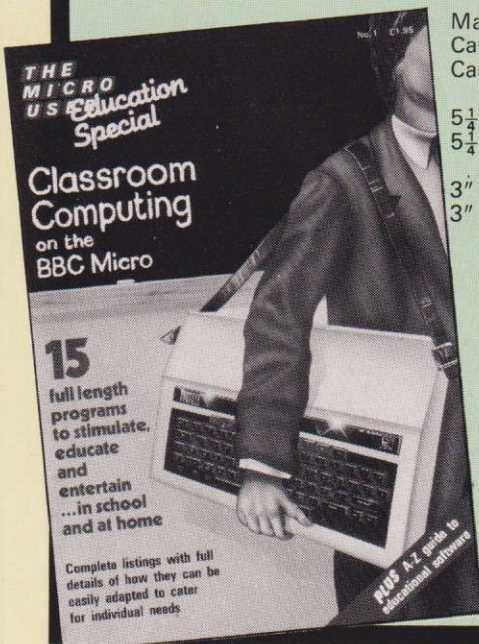
It contains 15 full length programs including graphical, easy to understand instruction in the basics of addition, subtraction and multiplication. There are entertaining table learning and mapping exercises, word and shape recognition for the very young and programs on finding lines of best fit, matrices, and a typing tutor for the not so young.

Plus a variety of word games to stimulate the whole family whatever the age group.

Prices for Education Special No 1 and its companion cassette and disc are:

Magazine	£1.95
Cassette	£5.95
Cassette and magazine	£7.00
5¼" disc	£7.95
5¼" disc and magazine	£9.00
3" disc	£9.95
3" disc and magazine	£11.00

Programs to help children of all ages!



Switch to Logo - the learning language

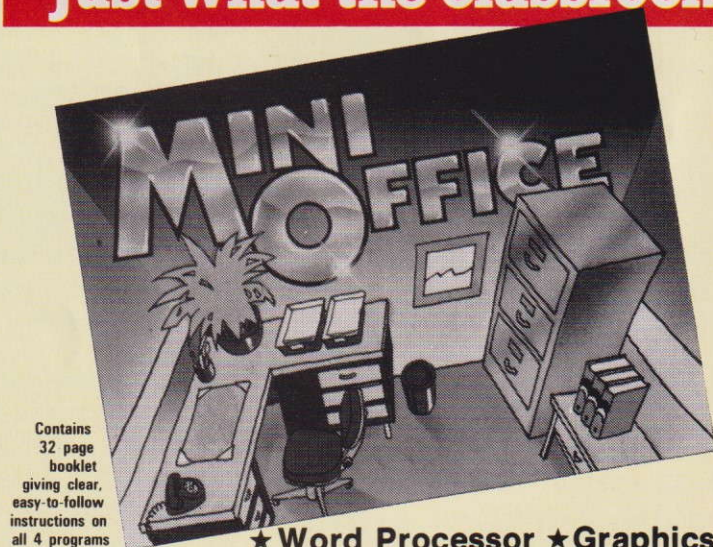
Teachers, parents and enthusiastic pupils all agree - Logo is the ideal language for learning.

Quick to learn, and simple to use, it makes using a microcomputer simplicity itself. Logo's brilliant design stimulates children's creativity and imagination, prompting them to explore the micro's capabilities to the fullest.

And now it's easier than ever to give your micro the power of Logo! After a detailed survey of the various Logos available, we have selected two powerful versions as outstanding: Acornsoft Logo and Logotron Logo. And we can offer these superb implementations to you cheaper than ever before.

So all you have to do - with the help of the articles in this issue - is to select the version that fits your needs. Whichever you choose, you're guaranteed a top-rate Logo - at the lowest possible price.

Just what the classroom's



★ Word Processor ★ Graphics
★ Spreadsheet ★ Database

Word Processor: Mastered in minutes, it's ideal for classroom applications. Whatever their abilities, pupils can produce work of high quality presentation. And being able to correct errors simply and without fuss does wonders for pupil motivation! Its unique double-sized text option in both edit and printer mode make it perfect for younger age groups and the partially sighted.

Database: This will find applications in every subject: wherever you need to store and retrieve information - quickly and easily - this does the job, from library orders to

ORDER FORM

Overseas rate stated as Surface Mail. Air Mail price on application.

All prices include postage, packing and VAT

Valid to November 30, 1985

Please enter number required in box. £ p

EDUCATIONAL SPECIAL No 2

Magazine only	£1.50	2320	<input type="text"/>
Cassette only	£4.95	2321	<input type="text"/>
Cassette + Magazine	£6.00	2322	<input type="text"/>
Disc (40T) only	£6.95	2323	<input type="text"/>
Disc (40T) + magazine	£7.95	2328	<input type="text"/>
Disc (80T) only	£6.95	2324	<input type="text"/>
Disc (80T) + magazine	£7.95	2315	<input type="text"/>
Disc (3") only	£8.95	2325	<input type="text"/>
Disc (3") + magazine	£9.95	2316	<input type="text"/>

TOTAL

EDUCATION SPECIAL No 1

Magazine	£1.95	2210	<input type="text"/>
Cassette	£5.95	2211	<input type="text"/>
Cassette + Magazine	£7.00	2212	<input type="text"/>
Disc (40T) only	£7.95	2213	<input type="text"/>
Disc (40T) + magazine	£9.00	2317	<input type="text"/>
Disc (80T) only	£7.95	2214	<input type="text"/>
Disc (80T) + magazine	£9.00	2318	<input type="text"/>
Disc (3") only	£9.95	2215	<input type="text"/>
Disc (3") + magazine	£11.00	2319	<input type="text"/>

TOTAL

Mini Office

BBC 'B' cassette	£5.95	2231	<input type="text"/>
Electron cassette	£5.95	3062	<input type="text"/>
5 1/4" 40T disc	£7.95	2232	<input type="text"/>
5 1/4" 80T disc	£7.95	2233	<input type="text"/>
3" disc	£9.95	2234	<input type="text"/>

TOTAL

Acornsoft Logo (Rom)

BBC	£59.00	2326	<input type="text"/>
Electron	£52.90	3074	<input type="text"/>

TOTAL

Logotron Logo (Rom)

BBC	£59.95	2327	<input type="text"/>
-----	--------	------	----------------------

TOTAL

Payment: please indicate method(✓)

- ☐ Access/Mastercharge/Eurocard
Card No.
- ☐ Barclaycard/Visa
Card No.
- ☐ Cheque/PO made payable to Database Publications Ltd.

TOTAL

Name

Signed

Address

Send to:

**The Micro User Education Special, FREEPOST,
Europa House, 68 Chester Road, Hazel Grove,
Stockport SK7 5NY.**

(No stamp needed if posted in UK) Please allow 28 days for delivery

**YOU CAN ALSO
ORDER BY PHONE:**

061-480 0171

Don't forget to quote your credit card number and full address

ES2

**Two BBC ROMs
Manuals and
Reference Card**

£59.00

SAVE £10

(also available for the Electron)

ACORN^{SOFT}

2090

A full implementation in
two language ROMs
for the BBC Microcomputer Model B

**16k BBC ROM
183 page Tutorial and
Reference Guide**

£59.95

SAVE £10

The C64 Standard LOGO
for the BBC Model B
Software by SOFT



·LOGO·

A NEW APPROACH TO EDUCATIONAL
AND RECREATIONAL COMPUTING

LOGOTRON

been waiting for!

parish records. Its easy-to-use search and sort facility enables pupils to analyse and classify the information they've collected, as well as test their hypotheses against the database.

Spreadsheet: A package with enormous educational potential for every age group. It allows pupils to set up simple numeric models of the real world, and see the consequences! Not only that, they can experiment by altering their initial assumptions, or seeing how the model 'grows' with time. The subjects could be anything: multiplication tables, seed germination, bus timetables, population models - it takes them all in its stride. And its simplicity and flexibility means that pupils are encouraged to formulate, try out and evaluate their own hypotheses.

Graphics: A superb utility that presents the spreadsheet's data in the form of bar charts, pie charts or graphs, allowing even the less mathematical to gain an intuitive grasp of the spreadsheet's model.

It's ONE simple package that's packed with potential

ONLY £5.95/£7.95

CASSETTE

DISC

For bulk orders telephone Jeanette Diver on 061-429 8008

DAVID EARLE offers a mathematical menu

THIS program is designed to offer practice with simple subtraction sums. The children who use the program call the sums "takeaway sums", hence the title.

There are no written instructions in the program so it is necessary to explain to children that they have to help How-Hi, a Chinese boy, across the stepping stones to meet Lotus Blossom by answering the questions.

Correct answers will help How-Hi across the river, but wrong answers will send him back a stone. If an incorrect answer is given the same question is presented again.

If a second incorrect answer is given a graphical display of the sum is shown before proceeding to the next question.

The keyboard has been disabled apart from the numbers 0 to 9. Auto repeat and the cursor control keys have been disabled. Escape and Break have been trapped.

The version printed is for disc users. It should be transferred to disc as TAKE. When the program is chained it will load, automatically download and run. It is assumed that PAGE will be &1900 or higher for disc use.

When entering the program, line 60 should be altered to read:

```
60 IF PAGE>&1000 THEN 70 ELSE 95
```

and line 95 should be entered to read:

```
95 GOTO 160
```

The program will run, download and set PAGE to &1100. This will allow all necessary disc functions while debugging and preserve the Escape, Break and edit functions. When debugging is complete line 95 should be deleted and line 60 altered to its original form.

The reason for line 130 being as shown instead of the more familiar form used with the tape version is that downloading, Breaking and attempting to re-run causes problems so it is easier to trap Break by reloading the program.

Tape users should omit lines 50 to 80 and enter line 130 as:

```
*KEY10 OLD:MRUNIM
```

If the timing at any part of the program does not suit the user the argument sent to PROCdelay can be altered. The number sent equals the delay in seconds.

To alter the number of times a question is presented in figures before the graphical presentation is displayed change the value compared with T% in line 430.

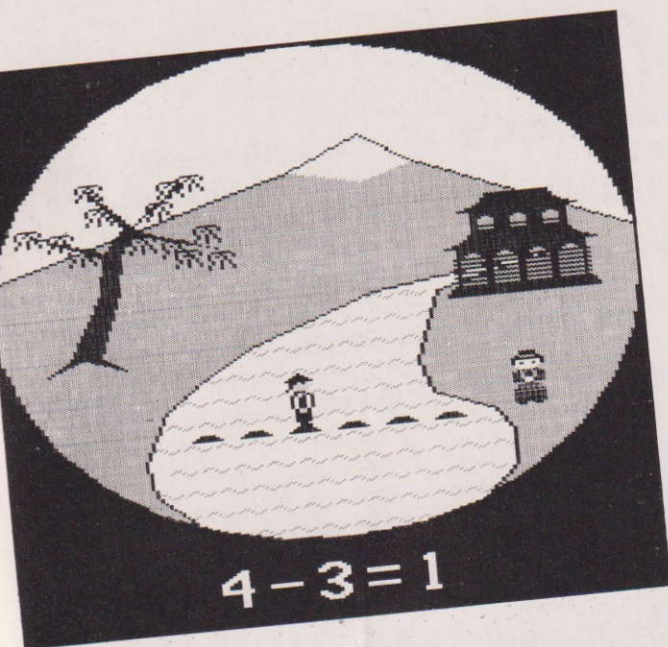
If it is necessary to alter the questions the data statement at line 2010 can be altered. Each group of three figures produces a sum, for example, 9, 8, 1 represents $9 - 8 = 1$.

Any sums can be added provided that no figure in the sum exceeds 9 and that 5, 0, 5 is the last entry as these figures generate a RESTORE in line 280.

PROCEDURES

- colour** Redefinition of several colours used in EOR plotting boy over various background colours.
- define1** Definition of user-defined characters to produce main screen display.
- define2** Re-definition of user-defined characters to produce the boy and the double height characters used in PROCexercise.
- delay** Used to introduce delays into the program at various points to slow the program down to a speed suitable for very young children. The argument passed to the procedure is the length of the delay produced in seconds.
- display** Production of graphic demonstration of a subtraction sum if two incorrect answers are given to the same sum.
- ellipse** Drawing of an ellipse for the outline of the title screen display and the main screen display.
- exercise** Main loop of the program producing the exercise and displays in accordance with questions held in data statements and input from the keyboard. Calls various other procedures as necessary.
- man** Production of the figure of the boy at the horizontal coordinate passed as an argument to the procedure and also deletion of the figure as EOR plotting is used.
- numb** Production of double height figures from user-defined characters at the horizontal coordinate passed as an argument.
- title** Production of title display.
- screen** Production of main screen display for the exercise.





```

10 REM CHINESE TAKEAWAY
20 REM by David Earle
30 REM (C) Micro User 1985
40
50 *KEY9 DX=PAGE-&1100;FORIX=PAGE
TO TOP STEP 4:!(IX-DX)=!IX;NEXT:PAGE=
&1100;MOLD:MRUNIM
60 IFPAGE>&1100 THEN 70 ELSE 100
70 *FX138,0,201
80 END
90
100 *FX11,0
110 *FX18,0
120 *FX4,1
130 *KEY10 PAGE=&1100;MCHAIN"TAKE"
M
140 *FX220,0
150
160 MODE2:VDU23;8202;0;0;0;
170 PROCcolour:PROCdefine1:PROctitl
e:PROCscreen:PROCdefine2:CLEAR
180 RESTORE2010
190
200 REPEAT
210 PROCexercise
220 UNTILFALSE
230
240 DEFPROCexercise
250 Man%=176
260 PROCman(Man%)
270 REPEAT
280 TX=0:READFirst%,Sec%,Ans%:IFFir
st%=5ANDSec%=0THENRESTORE2010
290 GCOL0,7:SOUND1,1,120,10:TX=TX+1
300 PROCnumb(First%,400):PROCnumb(1
1,496):PROCnumb(Sec%,592):PROCnumb(10
,688)

```

VARIABLES

- A%** Horizontal TAB position of display calculated and used in PROCdisplay.
- Ans%** Correct answer to each question read into PROCexercise from data.
- B** Local variable used in PROCellipse.
- C** Local variable used in PROCellipse.
- Chr%** Number of user-defined character calculated and used in PROCnumb.
- D** Delay in seconds used in PROCdelay.
- E** Local variable used in PROCellipse.
- First%** First number in each question read into PROCexercise from data.
- I%** Local variable used in PROCellipse.
- Input%** Corrected Ascii code of input from the keyboard in PROCexercise.
- K%** Local variable used in PROCellipse.
- Leaf** Loop variable used for production of leaves in PROCscreen.
- M%** Local variable used in PROCellipse.
- Man%** Horizontal coordinate of the boy figure in PROCman.
- N** Local variable used in PROCellipse.
- Number%** Number to be displayed in PROCnumb and used to determine which characters to use.
- Pitch%** Pitch of note produced by ENVELOPE and SOUND statements in PROCexercise.
- Press** Dummy variable used to detect a keypress in PROCexercise.
- R** Local variable used in PROCellipse.
- R%** Used to determine the width of the waves in PROCscreen.
- Right%** Horizontal coordinate of number used in PROCnumb.
- S** Local variable used in PROCellipse.
- S%** Horizontal coordinate of the characters equal to Sec% in PROCdisplay.
- Sec%** Second number in each question read into PROCexercise from data.
- Slat%** Loop variable used to produce slats across temple in PROCscreen.
- Stone%** Loop variable used to determine the horizontal coordinate of the stepping stones in PROCscreen.
- T** Local variable used in PROCellipse.
- T%** Counter to limit the number of times a sum is presented in figures before PROCdisplay.
- Wave%** Loop variable used to determine number of ripples in each wave.
- X%** General horizontal coordinate used throughout the program.
- Y%** General vertical coordinate.

From Page 75

```

310 *FX15,1
320 Input%=GET:IF Input%>57OR Input%<
33THEN320
330 IF Input%<48THEN Input%=Input%+16
340 PROCnumb(Input%-48,784)
350 PROCdelay(1):IF Input%-48=Ans%TH
EN360ELSE400
360 PROCman(Man%)
370 Man%=Man%+96
380 SOUND1,2,40,6
390 PROCman(Man%):PROCdelay(1):VDU4
,12,5:GOTO440
400 PROCman(Man%)
410 IF Man%>176 Man%=Man%-96
420 PROCman(Man%)
430 SOUND1,-15,0,10:PROCdelay(3):VD
U4,12,5:IFT%<2 THEN290 ELSE PROCdispl
ay
440 UNTIL Man%>=944:MOVE1000,476:GCO
L0,5:VDU248:FOR Pitch%=20 TO200 STEPS:
SOUND1,1,Pitch%,1:SOUND2,1,Pitch%+5,1
:NEXT
450 VDU4:CLS:COLOUR7
460 *FX15,1
470 PRINT" Press a key":Press=GET:
CLS:VDU5:MOVE1000,476:GCOL0,6:VDU248
480 PROCman(Man%)
490 ENDPROC
500
510 DEFPROCdelay(D)
520 TIME=0:REPEATUNTILTIME=(D*100):
ENDPROC
530
540 DEFPROCman(Man%)
550 MOVEMan%,436
560 VDU18,3,14,251,8,18,3,5,252,10,
8,18,3,12,253,10,8
570 VDU18,3,15,254,8,18,3,5,255:END
PROC
580
590 DEFPROCellipse
600 GCOL0,4:N=2*PI/30:C=COS(N):S=SI
N(N):B=1/SQR(2):E=1/SQR(2)
610 FORIX=1TO31
620 R=B*C-E*S:T=B*S+E*C
630 B=R:E=T:K%=640*B+640:M%=440+E+5
80
640 IFIX=1THENMOVEK%,M%
650 DRAWK%,M%
660 NEXTIX
670 ENDPROC
680
690 DEFPROCscreen
700 PROCellipse:VDU28,3,31,18,28

```

```

710 GCOL0,7:XX=640:FOR Y%=140TO190ST
EP4:PLOT77,XX,Y%:NEXT
720 GCOL0,6:FOR Y%=190TO644STEP4:PL
O77,XX,Y%:NEXT
730 GCOL0,7:FOR Y%=644TO1016STEP4:PL
O77,XX,Y%:NEXT
740 GCOL0,6:MOVE16,644:MOVE750,830:
PLOT85,1264,644
750 GCOL0,7:MOVE600,792:MOVE750,850
:PLOT85,750,764:PLOT85,850,792
760 GCOL0,4:MOVE16,644:DRAW600,792:
DRAW750,850:DRAW850,792:DRAW1264,644
770 MOVE150,497:MOVE200,497:PLOT85,
220,612:PLOT85,250,612:MOVE220,612:MO
VE225,672:PLOT85,250,612:PLOT85,260,6
72:MOVE100,722:PLOT85,225,672:MOVE275
,722:PLOT85,260,672:PLOT85,290,722:MO
VE200,797:PLOT85,275,722:MOVE450,672:
PLOT85,275,712
780 MOVE290,727:DRAW400,797:MOVE200
,497:MOVE250,472:PLOT85,175,497:MOVE1
00,472:PLOT85,150,497
790 VDU5:RESTORE2000:FOR Leaf=0TO12:
READXX,Y%:MOVEXX,Y%:VDU229:NEXT
800 GCOL0,7:MOVE360,190:MOVE940,190
:PLOT85,290,240:PLOT85,1000,240:PLOT8
5,275,300:PLOT85,1000,300:PLOT85,325,
390
810 PLOT85,975,340:PLOT85,400,440:P
LOT85,800,390:PLOT85,500,490:PLOT85,8
75,560:PLOT85,750,570:PLOT85,1125,615
820 MOVE800,390:MOVE830,440:PLOT85,

```



```

850,375
830 GCOL3,8
840 X%=280
850 FOR Y%=624TO172STEP-40
860 IF X%=280THEN X%=X%+32ELSE X%=280
870 MOVEX%,Y%:IFY%>350 RX=0ELSE RX=1
0
880 FOR Wave%=0TORX
890 VDU230
900 NEXT
910 NEXT
920 GCOL0,4

```

```

930 MOVE360,190:DRAW290,240:DRAW275
,300:DRAW325,390:DRAW400,440:DRAW500,
490:DRAW750,570:DRAW1125,615
940 MOVE940,190:DRAW1000,240:DRAW10
00,300:DRAW975,340:DRAW850,380:DRAW83
0,440:DRAW875,560:DRAW1125,615
950 GCOL0,4:MOVE896,656:VDU225:MOVE
1136,656:VDU224:MOVE960,656:MOVE960,6
24:PLOT85,1136,656:PLOT85,1136,624:MO
VE920,624
960 VDU226,226,226,226:MOVE920,592:
VDU227,227,227,227:MOVE956,688:VDU228
,228,228:MOVE928,720:VDU225:MOVE1108,
720:VDU224
970 MOVE992,688:MOVE992,720:PLOT85,
1108,688:PLOT85,1108,720:MOVE916,560:
MOVE900,544:PLOT85,1176,560:PLOT85,11
92,544
980 FOR Slat%=608TO560STEP-8
990 MOVE916,Slat%:DRAW1176,Slat%
1000 NEXT
1010 MOVE1008,436
1020 VDU18,0,0,231,8,18,0,7,232,10,8
,18,0,13,233,8
1030 VDU18,0,7,236,10,8,18,0,13,234,
8,18,0,0,235
1040 GCOL0,4
1050 FOR Stone%=368TO848STEP96
1060 MOVEStone%,352
1070 VDU237
1080 NEXT
1090 ENDPROC
1100
1110 DEFPROCdefine1
1120 ENVELOPE2,1,30,0,-30,6,1,6,13,0
,0,-126,126,0
1130 ENVELOPE1,4,0,0,0,0,0,126,-5,
-3,-1,126,110
1140 VDU23,224,128,128,192,192,224,2
40,252,255
1150 VDU23,225,1,1,3,3,7,15,63,255
1160 VDU23,226,255,255,231,195,129,1
29,129,129
1170 VDU23,227,129,129,129,129,129,1
29,129,129
1180 VDU23,228,255,255,231,195,195,1
95,255,195
1190 VDU23,229,124,146,170,136,84,84
,146,146
1200 VDU23,230,0,0,0,0,0,48,73,6
1210 VDU23,231,60,60,66,255,129,165,
0,0
1220 VDU23,232,0,0,0,0,126,90,126,60
1230 VDU23,233,126,219,219,219,231,2
31,24,126
1240 VDU23,234,126,255,255,255,255,2
55,255,0

```



```

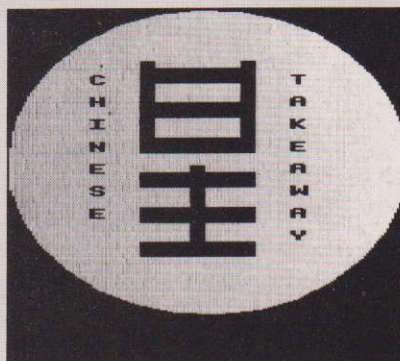
1250 VDU23,235,0,0,0,0,0,0,102
1260 VDU23,236,0,0,0,0,24,24,0,0
1270 VDU23,237,0,0,0,60,126,255,0,0
1280 ENDPROC
1290
1300 DEFPROCdefine2
1310 VDU23,224,60,126,231,195,195,19
5,195,195
1320 VDU23,225,195,195,195,195,195,2
31,126,60
1330 VDU23,226,24,56,56,24,24,24,24,
24
1340 VDU23,227,24,24,24,24,24,24,60,
60
1350 VDU23,228,60,126,231,195,195,3,
7,14
1360 VDU23,229,28,56,112,224,192,192
,255,255
1370 VDU23,230,60,126,231,195,3,7,14
,60
1380 VDU23,231,60,14,7,3,195,231,126
,60
1390 VDU23,232,6,14,30,62,118,230,19
8,198
1400 VDU23,233,255,255,6,6,6,6,6,6
1410 VDU23,234,127,127,96,96,96,96,1
24,126
1420 VDU23,235,15,7,3,3,195,231,126,
60
1430 VDU23,236,14,62,56,112,112,224,
224,252
1440 VDU23,237,254,231,195,195,195,2
31,126,60
1450 VDU23,238,255,255,7,7,14,14,28,
28
1460 VDU23,239,56,56,112,112,224,224
,224,224
1470 VDU23,240,60,126,231,195,195,19
5,231,126
1480 VDU23,241,126,231,195,195,195,2
31,126,60
1490 VDU23,242,60,126,231,195,195,19
5,231,127
1500 VDU23,243,63,7,6,14,28,56,240,2
24
1510 VDU23,244,0,0,0,0,255,255,0,0
1520 VDU23,245,0,0,255,255,0,0,0,0
1530 VDU23,246,0,0,0,0,0,0,255
1540 VDU23,247,255,0,0,0,0,0,0,0
1550 VDU23,248,0,108,254,254,254,124
,56,16
1560 VDU23,249,1,3,15,63,11,15,12,7
1570 VDU23,250,0,128,224,248,160,224
,96,192
1580 VDU23,251,16,56,124,254,104,64,
64,128
1590 VDU23,252,0,0,0,0,20,62,60,60

```

```

1600 VDU23,253,56,100,90,90,90,90,90
,90
1610 VDU23,254,36,36,52,60,60,60,24,
28
1620 VDU23,255,24,24,8,0,0,0,0,0
1630 ENDPROC
1640
1650 DEFPROCcolour
1660 VDU19,10,4,0,0,0,19,11,4,0,0,0,

```



```

19,9,0,0,0,0
1670 VDU19,2,3,0,0,0,19,1,3,0,0,0,19
,14,6,0,0,0,19,5,14,0,0,0
1680 VDU19,15,6,0,0,0,19,8,0,0,0,0,1
9,12,4,0,0,0,19,13,5,0,0,0
1690 ENDPROC
1700
1710 DEFPROCnumb(Number%,Right%)
1720 VDUS
1730 MOVERight%,96
1740 Chr%=224+(Number%*2)
1750 VDUChr%,10,8,Chr%+1
1760 ENDPROC
1770
1780 DEFPROCtitle
1790 PROCellipse
1800 @COL0,6:XX=640
1810 FORYX=140TO1016STEP4
1820 PLOT77,XX,YX
1830 NEXT
1840 @COL0,4:COLOUR4:COLOUR134
1850 FORYX=848TO348STEP-100
1860 MOVE424,YX:PLOT1,0,-40:PLOT81,3
60,40:PLOT81,0,-40
1870 NEXT
1880 FORXX=424TO760STEP320
1890 MOVEXX+40,868:PLOT1,-40,0:PLOT8
1,40,-240:PLOT81,-40,0
1900 NEXT
1910 MOVE628,568
1920 PLOT1,-40,0:PLOT81,40,-240:PLOT
81,-40,0
1930 VDU28,4,31,4,6
1940 PRINT"C H I N E S E "

```

```

1950 VDU28,14,31,14,6
1960 PRINT"T A K E A W A Y"
1970 PROCdelay(3):VDU26:COLOUR128:CL
S
1980 ENDPROC
1990
2000 DATA60,732,190,807,385,807,435,
667,325,762,350,692,165,692,285,707,2
00,762,350,797,410,712,375,672,125,72
2
2010 DATA9,8,1,6,1,5,2,0,2,5,2,3,4,3
,1,7,1,6,3,1,2,8,5,3,1,0,1,9,1,8,6,0,
6,2,2,0,5,4,1,4,2,2,7,2,5,3,3,0,8,4,4
,9,6,3,6,4,2,2,1,1,5,1,4,4,4,0,7,4,3,
3,1,2,8,2,6,9,5,4,6,3,3,2,1,1,5,3,2,4
,0,4,7,3,4,3,1,2,8,1,7,9,3,6,6,2,4,5,
0,5
2020
2030 DEF PROCdisplay
2040 SX=(2*First%)-(2*Sec%):AX=(19-2
*First%)/2:VDU4,26,31,AX,28
2050 COLOUR7
2060 FOR Char=1 TO First%:VDU249,250
:NEXT
2070 PRINTTAB(5,30)STR$(First%)
2080 PROCdelay(2)
2090 COLOUR6
2100 VDU31,SX+AX,28
2110 IF Sec%=0 THEN 2130
2120 FOR Char=1 TO Sec%:VDU249,250:N
EXT
2130 COLOUR7
2140 IF Sec%<>0 THEN 2160
2150 PRINTTAB(7,30)"- 0":GOTO2170
2160 PRINTTAB(7,30)"- ";STR$(Sec%)
2170 PROCdelay(1)
2180 IF Sec%=0 THEN 2230
2190 COLOUR0:VDU31,SX+AX,28,249,250
2200 COLOUR6
2210 VDU31,SX+AX+2,28
2220 FOR Char=1 TO Sec%:VDU249,250:N
EXT
2230 PROCdelay(1)
2240 COLOUR0
2250 VDU31,SX+AX+2,28
2260 FOR Char=1 TO Sec%:VDU249,250:P
ROCdelay(0.5):NEXT
2270 COLOUR7:IF Ans%<>0 THEN 2290
2280 PRINTTAB(10,30)" = 0":GOTO2300
2290 COLOUR7:PRINTTAB(10,30)" = ";ST
R$(Ans%)
2300 PROCdelay(3)
2310 COLOUR0:PRINTTAB(0,28)STRING$(1
9," ")
2320 PRINTTAB(0,30)STRING$(19," ")
2330 VDU28,3,31,18,28
2340 ENDPROC

```


Educational Programs from:

CBA Soft

FLASH-MATCH

- 3 years to adult – a 1 or 2 player game.
- Based on the card game Snap with the computer doing the hard work of judging who wins.
- Play the computer with the 1 player option adjustable response time.
- Encourages shape and colour awareness with colourful entertaining graphics.

PRIME CATCH

- An entertaining sums game – ages 6 years to adult.
- Control fisherman Fred to match a target points catch before the seagull crosses the screen.
- On screen instructions – sound and colourful animated graphics.
- Ideal way to familiarise children with simple sums.

- Both Games **£4.50 each** (inc. P&P)
- BBC (O.S. 1+) ● Electron ● Spectrum 48k

**CBA Soft. PO Box 58, Barnet, Herts.
01-637 1913**

ADVERTISERS INDEX

4mation	38
Beebug	11
Bizzell Computers	64
Brain Train Club	56
C&C Co.	44
CBA Software	78
Care Electronics	19
Communtel	56
Cumana	26, 29
Datapen Micro	44
Golem Ltd	60
T. Hall	33
Highlight Software	55
Hilditch Software	64
Integrex Ltd	79
J.P. Magnetics	55
Kosmos Software	13
L.C.L.	55
M.J. Software	64
Macob Systems	55
Mertec	61
Microvitec	2
Miracle Technology	49
Mirrorsoft	23
Mushroom Computers	64
Nidd Valley	60
Northern Computers	78
Resource Facilities	13
Silent Computers	56
Viglen	34, 35
Willow Software	44

AMSTRAD EDUCATION SCHEME

(FOR BONA FIDE EDUCATION AND TRAINING ESTABLISHMENTS ONLY)

- £200 educational software and utility pack on disc supplied FREE with each disc drive system.
- Substantial discounts and software for schools. ● 1 year FREE hardware service contract.
- Business and training software for higher education. ● Amstrad, BBCBASIC(Z80) language pack.
- Details of the Amstrad network and hard disc system now available.

CPC464 Colour Computer

64K RAM, 32K ROM
standard high speed Basic,
27 colours, 80 column text,
3 channel sound, printer interface,
built-in data recorder.
System supplied with £100
software pack on cassette, also
with green or colour monitor.

**SYSTEM
PRICES
START
AT**

£189



CPC664 Colour Computer

As the CPC464, but built-in
unit is a 180K disc drive.
Includes CP/M 2.2, Dr LOGO
and AMSDOS. System supplied with
£200 educational software and utility
pack on disc, also with green
or colour monitor.

**SYSTEM
PRICES
START
AT**

£279

THE AMSTRAD CPC 664 COMPUTER SYSTEM

FURTHER INFORMATION FOR SCHOOLS, COLLEGES, POLYTECHNICS, UNIVERSITIES AND
TRAINING SCHEMES CONTACT THE AMSTRAD EDUCATIONAL DISTRIBUTORS:

NORTHERN COMPUTERS LTD.

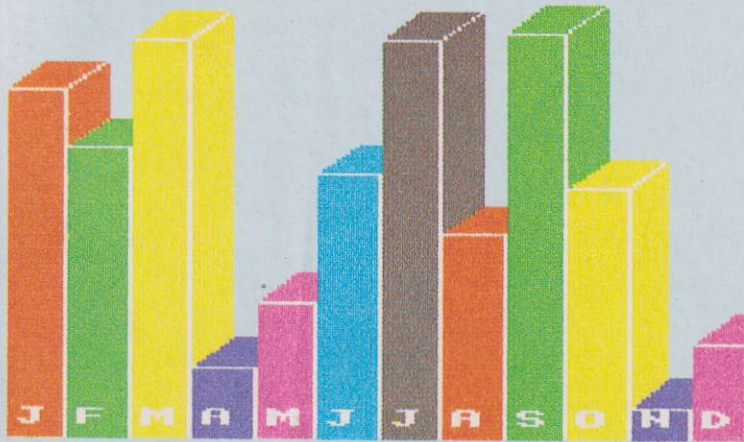
Churchfield Road, Frodsham, Cheshire WA6 6RD. Tel: (0928) 35700 (10 lines)

COLOURJET

36 COLOUR INK JET PRINTER

Prints all BBC Micro screen Modes inc. Mode 7 ... in full colour

NOW 132 CHARACTERS/LINE COMPRESSED



£569 suggested retail price excl VAT

Features: videotext printing
Prints overhead transparencies

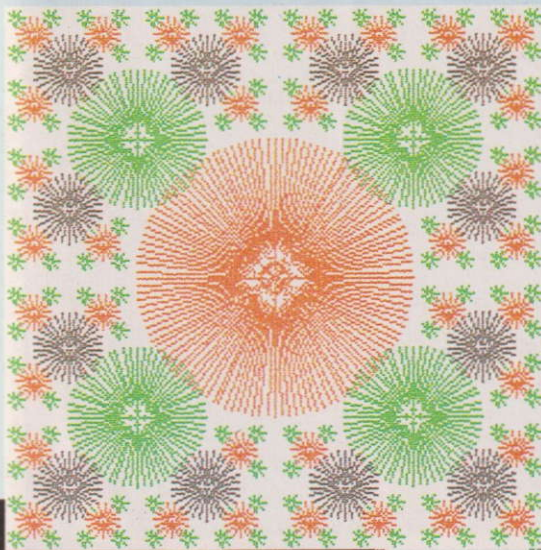
Options available:

- Buffered RS232 interface.
- Viewdata & RS232 interface.
- Apple II Interface.
- IBM PC dump.

Specifications:

640 dots/line & 1280 dots/line Hires mode.
84 dots/inch both axes.
Viewdata Printing (BBC MODE 7).
Friction feed rollpaper with single sheetfeed A4.
40 cps in full colour.
DOUBLE STRIKE MODE.
Ink cartridges, 4 million character life.

Virtually
silent



INTEGREX LTD.

Church Gresley, Burton-on-Trent, Staffs DE11 9PT
Tel: 0283 215432 Telex: 377106

Dealers, OEM & Educational enquiries welcome

If you think the front looks good—you should see the back.

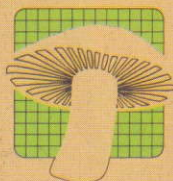


Once again Mushroom has answered a special need of all BBC computer users with their new WORKSTATION — no more untidy wiring, no more pulled plugs, transforms your system into a handsome console unit.

Features include:—

- * ADVANCED CABLE MANAGEMENT
- * TWIN 400K MITSUBISHI SLIM LINE DRIVES SWITCHABLE FROM 40 TO 80 TRACK WITH STATUS INDICATOR LED'S
- * ON BOARD REGULATED 4 AMP POWER SUPPLY WITH OVERLOAD PROTECTION
- * MAINS INPUT FILTER FOR CLEAN POWER SUPPLY
- * FILTERED POWER TO THE 3 TAKE-OFF SOCKETS AT REAR FOR MICRO, MONITOR AND PRINTER

The WORKSTATION is manufactured to the highest safety specifications from lightweight aluminium attractively finished in oatmeal with orange trim. When not in use the computer is stored underneath, affording useful protection from accidental knocks or spillage.



Mushroom

COMPUTERS LIMITED

Another Mushroom product from Broadway Electronics.
Patents pending.



The WORKSTATION, complete with twin disc drives, is suitable for all Areas of use — Domestic, Schools, Local Authorities, Commercial and Industrial.

This is a must for every user at only £399.00 Exc. VAT.

Those people already owning disc drives can have them fitted into the WORKSTATION. Prices on request.

Available direct from:
Broadway Electronics, 55 Tavistock Street, Bedford, MK40 2RF
Telephone: (0234) 58303. Telex: 826112 MUSCOM G, and selected computer dealers nationwide.