

LET'S COMPUTE!

99p

The great
**TURTLE
CONTEST**
hots up!



Compute a
gingerbread
ROM – Yum!



For **ALL** users
of the Electron,
BBC Micro, PC,
Amiga, C64/128,
Archimedes,
Amstrad CPC,
Atari ST and
Spectrum

No 2 September 1990
A Database Publication



HOW A COMPUTER WORKS
Start exploring... on the centre pages

AL'S GORITHMS

THIS IS IT! The page where the budding artists among you can pick up a tip or two. And a chance for you to show what **YOU** can do yourself.

Making a lot with a few lines of code is what it's all about. Alan McLachlan gives you some more bright ideas.



GOING ROUND IN CIRCLES...

Less talk this month, but lots for you to do. The rules are simple: Using circles – and only circles – create something more interesting than just pretty repetitive patterns.

With the programs on this page you can create two unusual pictures of a clown and a worm. They're unusual because they are both built up out of circles filled with colour.

You'll find them very easy to change as you wish. You can wipe out parts of the picture by over-painting them.

You only need type in the core of the program once – use the one suitable for your own micro. Then save it on disc or tape and add one of the sets of DATA lines – or try your own.

Let's look at how the picture is drawn. The first DATA line – line 330 – just tells your micro how many lines follow.

Each of the following lines contains four figures needed to



draw a circle: its x and y positions, its radius and the colour you want to make it.

Each of these is read by the computer. The four figures are then used to draw a circle.

If you are going to use this program to draw your own picture it's best to plan it in advance.

Start by drawing it on paper – working out the position of each circle and the order it is to be drawn. You'll find the result can be quite spectacular and well worth the effort.

And as with last month's routine, if you think you can do something much better using just circles – just send your masterpiece to Let's Compute! and let us all see it.

● Next month we'll see what we can do with ellipses.

Electron/BBC Micro/Archimedes core

```
10 REM Circles
20 REM By Alan McLachlan
30 REM (c) Let's Compute!
40 MODE2
50 VDU5
60 VDU23;8202;0;0;0;
70 PROCINIT
80 READcircles
90 FOR P = 1 TO circles
```

```
100 READ x%,y%,r%,c%
110 PROCcircle(x%,y%,r%,c%)
120 NEXT
130 PROCcircle(x%,y%,r%,c%)
140 REPEAT UNTIL FALSE
150 DEFPROCcircle(x%,y%,r%,c%)
160 LOCAL IX
170 GCOLOR,c%
180 VDU29,x%;y%;
```

```
190 MOVE0,0
200 FOR IX=0 TO 30
210 MOVE 0,0
220 PLOT85,r%*c(IX),r%*s(IX)
230 NEXT
240 VDU29,0;0;
250 ENDPROC
260 DEFPROCINIT
270 DIM s(30),c(30)
```

```
280 FOR loop%=0 TO 30
290 s(loop%)=SIN(loop%*2.2*PI/30)
300 c(loop%)=COS(loop%*2.2*PI/30)
310 NEXT loop%
320 ENDPROC
```


Typing tip

If you get an Out of Data in line 100 error it does not always mean you have made a mistake in that line. It could be that you've not typed in one of the DATA lines correctly. Or you may have forgotten to enter one of the lines.

So check them all carefully. The program works fine providing you key it in exactly as it is printed.

Data for worm

330 DATA 38	530 DATA 710,254,134,0
340 DATA 200,600,200,2	540 DATA 710,254,130,2
350 DATA 170,570,220,0	550 DATA 620,280,154,0
360 DATA 600,600,200,2	560 DATA 620,280,150,2
370 DATA 630,570,220,0	570 DATA 550,350,184,0
380 DATA 160,800,20,2	580 DATA 550,350,180,2
390 DATA 680,800,20,2	590 DATA 450,500,224,0
400 DATA 1200,300,20,2	600 DATA 450,500,220,2
410 DATA 1180,280,34,0	610 DATA 550,500,30,0
420 DATA 1180,280,30,2	620 DATA 550,490,30,2
430 DATA 1150,260,50,0	630 DATA 440,450,130,0
440 DATA 1150,260,46,2	640 DATA 430,490,110,2
450 DATA 1090,240,62,0	650 DATA 300,600,70,7
460 DATA 1090,240,60,2	660 DATA 450,600,70,7
470 DATA 1000,230,80,0	670 DATA 300,500,80,1
480 DATA 1000,230,76,2	680 DATA 335,535,20,7
490 DATA 910,226,94,0	690 DATA 320,520,25,1
500 DATA 910,226,90,2	700 DATA 320,620,30,8
510 DATA 800,230,114,0	710 DATA 470,620,30,8
520 DATA 800,230,110,2	

Data for clown

330 DATA 54	610 DATA 830,540,120,0
340 DATA 1000,500,100,5	620 DATA 830,540,114,5
350 DATA 1000,500,80,0	630 DATA 650,520,120,0
360 DATA 1000,500,74,5	640 DATA 650,520,114,1
370 DATA 1000,400,30,5	650 DATA 700,580,20,7
380 DATA 300,500,100,5	660 DATA 680,570,30,1
390 DATA 300,500,80,0	670 DATA 650,140,70,3
400 DATA 300,500,74,5	680 DATA 520,180,90,3
410 DATA 300,400,30,5	690 DATA 520,100,70,3
420 DATA 650,550,350,5	700 DATA 780,180,90,3
430 DATA 650,450,200,7	710 DATA 780,100,70,3
440 DATA 650,450,160,0	720 DATA 620,160,14,1
450 DATA 650,450,120,7	730 DATA 680,120,14,1
460 DATA 650,450,80,5	740 DATA 720,200,14,1
470 DATA 790,450,50,7	750 DATA 780,220,14,1
480 DATA 510,450,50,7	760 DATA 840,180,14,1
490 DATA 550,700,100,0	770 DATA 810,120,14,1
500 DATA 750,700,100,0	780 DATA 760,160,14,1
510 DATA 550,650,140,5	790 DATA 760,100,14,1
520 DATA 750,650,140,5	800 DATA 780,60,14,1
530 DATA 550,660,100,7	810 DATA 580,200,14,1
540 DATA 570,660,60,4	820 DATA 520,220,14,1
550 DATA 570,660,30,0	830 DATA 460,180,14,1
560 DATA 750,660,100,7	840 DATA 490,120,14,1
570 DATA 730,660,60,4	850 DATA 550,160,14,1
580 DATA 730,660,30,0	860 DATA 550,100,14,1
590 DATA 470,540,120,0	870 DATA 520,60,14,1
600 DATA 470,540,114,5	

CPC core

```

10 REM CPC Circles
40 MODE 0
50 INK 0,0:INK 1,6:INK 2,18:INK 3,24:INK 4,2:INK 5,16:INK 6,2
0:INK 7,26
60 GOSUB 260
80 READ circles
90 FOR p=1 TO circles
100 READ x%,y%,r%,c%:x%=x%\2:y
%=y%\2:r%=r%\2
110 GOSUB 150
120 NEXT
140 WHILE 1=1:WEND
150 REM circle
170 PLOT -5,-5,c%
180 ORIGIN x%,0.9*y%
190 FOR i%=0 TO 300 STEP 1+12
0\r%
200 FOR j%=-2 TO r%\150 STEP 2
210 MOVE -r%*c(i%),-0.9*r%*s(i
%)+j%
220 DRAW r%*c(i%),0.9*r%*s(i%
+j%
230 NEXT j%
235 NEXT i%
240 ORIGIN 0,0
250 RETURN
260 REM INITIALISE
270 DIM s(310),c(310)
280 FOR loop%=0 TO 310
290 s(loop%)=SIN(loop%*PI/300)
300 c(loop%)=COS(loop%*PI/300)
310 NEXT loop%
320 RETURN

```

ST/Amiga core

```

10 REM ST/Amiga Circles
20 REM written in STOS/AMOS
40 KEY OFF:MODE 0
50 REM in AMOS use CLS instea
d of MODE 0
80 READ num
90 FOR p=1 TO num
100 READ x,y,r,c
110 INK c:CIRCLE x/4,INT(200-(
y/5.15)),r/4
120 NEXT
130 WAIT KEY

```

Spectrum core

```

10 REM Spectrum Circles
70 GO SUB 260
75 RESTORE
80 READ circles
90 FOR p=1 TO circles
100 READ x,y,r,c:LET x=x/6:LET
y=y/6:LET r=r/6
110 GO SUB 150
120 NEXT p
140 GO TO 140
150 REM CIRCLE
170 INK c
180 LET r=r-2:CIRCLE x,y,r+1
190 FOR i=1 TO 601 STEP INT (1
+40/r)
210 PLOT x-r*c(i),y-r*s(i)
220 DRAW 2*r*c(i),2*r*s(i)
230 NEXT i
240 LET r=r+2:CIRCLE x,y,r:IN
VERSE 1:CIRCLE x,y,r:INVERSE 0
250 RETURN
260 REM INITIALISE
270 DIM s(610):DIM c(610)
280 FOR l=1 TO 610
290 LET s(l)=SIN (l*PI/600)
300 LET c(l)=COS (l*PI/600)
310 NEXT l
320 RETURN

```



Will I be
at school
on my next
birthday?

On what day will the
next century start?

When will Christmas Day
next be on a Sunday?

Will Bonfire Night
be at the weekend?

BACK TO THE Future

Stephen Wade
invites you to be a
time traveller – and
unlock the secrets of
the calendar

Want to know which day of the
week a certain date falls on?
It's a real doddle with this
program!

Type in the core, and then the
extra lines for your micro – and
SAVE it.

When the program is RUN,
you are asked to enter the day
number, the month number and
the year of the date you are
interested in. Type these in
using numbers. For example,
the 25th of December 1990
would be entered as:

Day number : 25
Month : 12
Year : 1990

And that's all there is to it. You'll
be told the day of the week on which
that date falls.

If you want to quit the program press
Return each time the computer asks you
a question. If no day, month and year are
entered the program automatically ends.

ST/Amiga using STOS/AMOS

```
40 DIM d$(7)
80 KEY OFF:CLS
100 REPEAT
170 LOCATE 0,0 : CENTRE "DAY OF
THE WEEK"
240 IF d=0 AND m=0 AND y=0 THE
N CLS : END
250 IF m<3 THEN m=m+12 : DEC y
260 a# = d + 2 * m + INT(0.6 * (m + 1)) + y +
INT(y / 4) - INT(y / 100) + INT(y / 400) + 2
270 a# = INT(((a# / 7) - INT(a# / 7)) * 7
)+ 1.5)
290 PRINT d$(a#)
310 REPEAT
320 UNTIL INKEY$ <> ""
330 UNTIL 0
```

Commodore 64/128

```
40 DIM D$(7)
150 PRINT CHR$(147);
170 PRINT "DAY OF THE WEEK"
180 PRINT:PRINT
240 IF D=0 AND M=0 AND Y=0 THE
N PRINT CHR$(147);:END
250 IF M<3 THEN M=M+12:Y=Y-1
260 A=D+2*M+INT(.6*(M+1))+Y+(Y
DIV 4)-(Y DIV 100)+(Y DIV 400)+
2
270 A=INT((A/7-(A DIV 7))*7+1.
5)
290 PRINT D$(A)
310 GET AS:IF AS="" THEN 310
320 GOTO 150
```

Amstrad CPC

```
40 DIM d$(7)
80 MODE 1
90 PRINT CHR$(2)
100 WINDOW 5,36,5,21
110 INK 0,24
120 INK 1,2
130 PAPER 1
140 PEN 0
160 WINDOW 6,35,6,20
170 PRINT TAB(8) "DAY OF THE W
EEK"
180 WINDOW 6,35,8,20
190 WHILE 1=1
200 CLS
205 PRINT:PRINT
240 IF d=0 AND m=0 AND y=0 THE
N INK 0,1:INK 1,24:WINDOW 1,40,1
,25:PEN 1:PAPER 0:CLS:END
250 IF m<3 THEN m=m+12:y=y-1
260 a=d+2*m+INT(0.6*(m+1))+y+(
y\4)-(y\100)+(y\400)+2
270 a=INT((a/7-(a\7))*7+1.5)
290 PRINT d$(a)
310 WHILE INKEY$<>"":WEND
320 WHILE INKEY$="":WEND
330 WEND
```

Spectrum

```
40 DIM d$(7,9)
170 PRINT TAB 8;"DAY OF THE WE
EK"
180 PRINT:PRINT
240 IF d=0 AND m=0 AND y=0 THE
N CLS:STOP
250 IF m<3 THEN LET m=m+12:LE
T y=y-1
260 LET a=d+2*m+INT (0.6*(m+1)
)+y+INT (y/4)-INT (y/100)+INT (y
/400)+2
270 LET a=INT ((a/7-INT (a/7))
*7+1.5)
290 PRINT d$(a)
310 IF INKEY$<>"" THEN GO TO 3
10
320 IF INKEY$="" THEN GO TO 32
0
330 GO TO 150
```

THE CORE (ALL micro's)

```
10 REM Day of the week
20 REM by Stephen Wade
30 REM (c) Let's Compute!
50 FOR n=1 TO 7
60 READ d$(n)
70 NEXT n
150 CLS
210 PRINT:INPUT "Day number :
";d
220 PRINT:INPUT "Month : ";m
230 PRINT:INPUT "Year : ";y
280 PRINT:PRINT:PRINT "The day
is ";
300 PRINT:PRINT "Press a key..
."
340 DATA "Saturday","Sunday","
Monday"
350 DATA "Tuesday","Wednesday"
360 DATA "Thursday","Friday"
```

Electron/BBC Micro/Archimedes

```
40 DIM d$(7)
80 MODE 6
90 VDU 23,1,0;0;0;0;
100 VDU 28,4,20,35,4
110 VDU 19,0,3,0,0,0
120 VDU 19,1,4,0,0,0
130 COLOUR 129
140 COLOUR 0
160 VDU 28,5,19,34,5
170 PRINT TAB(8)"DAY OF THE WE
EK"
180 VDU 28,5,19,34,7
190 REPEAT
200 CLS
205 PRINT:PRINT
240 IF d=0 AND m=0 AND y=0 THE
N VDU 20,26:CLS:END
250 IF m<3 THEN m=m+12:y=y-1
260 a=d+2*m+INT(.6*(m+1))+y+(y
DIV 4)-(y DIV 100)+(y DIV 400)+
2
270 a=INT((a/7-(a DIV 7))*7+1.
5)
290 PRINT d$(a)
310 REPEAT
320 UNTIL GET
330 UNTIL 0
```


LET'S COMPUTE!

99p

**Cheats and
passwords
revealed!**



**Compute a
fun brew for
Hallowe^{en}!**



**For ALL users
of the Electron,
BBC Micro, PC,
Amiga, C64/128,
Archimedes,
Amstrad CPC,
Atari ST and
Spectrum**

No 3 October 1990
A Database Publication



9 770960 283003

10



WIN an Amiga computer – and lots more!

Witches' Brew

The party season has arrived. And what better way to make a halloween party – or any other – go with a fizz than to prepare an unusual cocktail of drinks?

Here's a program that gives you a few ideas of what you could mix with what.

But once you've created your tasty concoction, what do you call it?

The program itself comes to your rescue. Along with the list of suggested ingredients you also get a name.

As it's one made up by your computer it may look strange. And you may have trouble saying it. So be warned: With some of them you may tie your tongue in knots trying!

Now type it in – and have fun!



Is YOUR machine here?

```

10 REM The Witches' Brew
20 REM (C) Let's Compute!
30 DEF FNR(n)=RND(n)
40 LET w$="aeiou"
50 LET v$=""
60 LET c$="bcdfghjklmpqrstvwxyzn"
70 DIM d$(3,50):DIM x(3)
80 FOR A=1 TO 3:LET M=0
90 LET M=M+1
100 READ d$(A,M)
110 IF d$(A,M)<>" " THEN GOTO 90
120 LET x(A)=M-1:NEXT A
130 LET DUMMY = RND(-TIME)
140 CLS
150 PRINT "WITCHES' BREW"
160 PRINT
170 PRINT "How to make a ";
180 GOSUB 400
190 PRINT
200 FOR C=1 TO 3
210 GOSUB 270
220 NEXT C
230 PRINT
240 PRINT "Press a key for another dri
nk"
250 LET AX = GET
260 GOTO 140
270 REM SORT OUT WORDS
280 LET LD=0
290 LET T=FNR(3)
300 LET R=x(C)-LD
310 LET D=ABS(FNR(R+2)-FNR(R))+1+LD
320 IF D>x(C) THEN GOTO 390
330 IF C=3 THEN PRINT "and some ";:GOT
O 370
340 PRINT " ";T;" Measure";
350 IF T>1 THEN PRINT "s of ";
360 IF T=1 THEN PRINT " of ";
370 PRINT d$(C,D)
380 LET LD=D:GOTO 290
390 RETURN
400 REM FIND AND PRINT NAME
410 LET R=FNR(21)
420 PRINT CHR$(ASC(MID$(c$,R,1))-32);
430 IF R=12 THEN LET v$="u":PRINT v$;
440 IF R=15 THEN IF FNR(5)=1 THEN PRIN

```

```

T "h";
450 GOSUB 580
460 PRINT v$;
470 IF FNR(15)<2 THEN GOTO 450
480 LET R=FNR(21):PRINT MID$(c$,R,1);
490 IF FNR(15)=1 THEN PRINT MID$(c$,R,
1);
500 IF R=12 THEN LET v$="u":PRINT v$;
510 IF R=1 THEN IF FNR(20)=1 THEN PRIN
T "b";
520 IF R=2 THEN IF FNR(2)=1 THEN PRINT
"k";
530 IF R=15 THEN IF FNR(9)=1 THEN PRIN
T "h";
540 IF R<>12 THEN IF FNR(15)=1 THEN PR
INT "r";
550 IF FNR(5)<4 THEN GOTO 450
560 IF FNR(9)<5 THEN GOSUB 580:PRINT v
$;:IF FNR(15)=1 THEN PRINT "r";
570 PRINT:RETURN
580 REM FIND A VOWEL
590 LET x$=v$
600 LET P=FNR(5)
610 LET v$=MID$(w$,P,1)
620 IF v$=x$ THEN GOTO 600
630 RETURN
640 REM Soft Drinks
650 DATA "Lemon juice","Apple juice","
Pineapple juice","Tomato juice"
660 DATA "Apricot juice","Orange juice
","Mixed fruit juice","Lime juice"
670 DATA "Vimto cordial","Pear juice","
680 REM Fizzy drinks
690 DATA "Coke","Vimto","7 Up","Lemona
de","Bitter Lemon","Tonic water"
700 DATA "Ginger ale","Soda water","
710 REM Floaters
720 DATA "orange peel","ice cubes","cl
oves","lemon peel","lime peel"
730 DATA "cherries","crushed ice","app
le slices","raisins","dried apricots"
740 DATA "dried peaches","fresh pineap
ple cubes","orange slices"
750 DATA ""

```

How to make a Zuno

2 Measures of Apricot juice
 3 Measures of Pear juice
 2 Measures of Bitter Lemon
 1 Measure of Soda Water
 add some lime peel
 add some dried peaches
 add some fresh pineapple cubes

BBCMicro/Archimedes/Electron

The listing works as shown

Amstrad CPC

Change the following lines:

```

30 DEF FNR(n)=INT(RND*n+1)
130 RANDOMIZE TIME
250 WHILE INKEY$="" :WEND

```

C64/128

Enter the program in capitals.

Change the following lines:

```

30 DEF FNR(N)=INT(RND(0)*N+1)
140 PRINT CHR$(147)
250 GET AS:IFAS="" THEN GOTO 250
420 PRINT MID$(CS,R,1)

```

PC (GW-Basic)

Change the following lines:

```

30 DEF FNR(n)=INT(RND*n+1)
130 RANDOMIZE TIMER
250 WHILE INKEY$="" :WEND

```

Spectrum

Use 48k mode.

Change the following lines:

```

30 DEF FNR(n)=INT(RND*n+1)
70 DIM d$(3,50,22): DIM x(3): DIM s$(22)
110 IF d$(A,M)<>" "s$ THEN GOTO 90
130 RANDOMIZE
250 PAUSE 0
310 LET D=ABS (FN r(R+2)-FN r(R))+1+LD
420 PRINT CHR$ (CODE (c$(R))-32);
490 IF FN r(15)=1 THEN PRINT c$(R);
610 LET v$=w$(P)

```

ST (Stos)/Amiga (Amos)

Change the following lines:

```

30 MODE0:HIDE
130 LET a=RND(TIMER)
250 WAIT KEY
290 LET t=RND(2)+1
310 LET D=ABS(RND(R+2)-RND(R))+1+LD
600 LET p=RND(4)+1

```

AMOS: Use CLS instead of MODE0

THE PROGRAM DOCTOR



There's always a better way of doing things, and here's another program just waiting to be operated on. Our resident micro medic diagnoses several shortfalls.

Remember all those games programs where, having lost all your lives all too quickly, the micro offers you a chance to play the game again? Well, that's what the program at the top of the page does. It prints out a simple message until you press N to finish.

Recognise the standard question:

Do you want to finish?(Y/N)

Actually all that happens is that line 40 tests whether Y was pressed. If it wasn't it RUNs the program again. If you pressed any other key - including a small y - the program ends.

But we can do a much better job. The second program shows how.

Here are improvements that have been made:

- You should always test for lower case letters as well as capitals - y as well as Y. That's what the OR is doing in the new Line 40.

- It's also polite to let the user know that the program has ended. So Line 50 is there to round things off.

Those are the programming changes. But you might notice something else - put in to make the program easier to read.

It now starts with two lines of REMs. These don't affect the way it runs, but they act as a useful signpost. They

This is the original program:

```
10 PRINT "This is all the program does."
20 PRINT "Do you want to finish?(Y/N)"
30 k$=GET$
40 IF k$<>"Y" THEN RUN
```

How the Doc altered it:

```
1 REM ### Program Doctor ###
2 REM ### Version Two ###
10 PRINT "This is all the program does."
20 PRINT "Do you want to finish?(Y/N)"
30 k$=GET$
40 IF k$<>"Y" OR k$<>"y" THEN RUN
50 PRINT "Goodbye then"
```

IS YOUR MACHINE HERE?

Electron/BBC/Archimedes

The programs work as shown

CPC/PC (GW-Basic)

Replace k\$=GET\$ with:

`k$="":WHILE k$="":k$=INKEY$:WEND`

Commodore 64/128

Enter the program in capitals

Replace k\$=GET\$ with:

`GET k$:IF k$="" THEN GOTO 30`

Spectrum

Select 48k mode

Replace k\$=GET\$ with:

`k$=INKEY$:IF k$="" THEN GOTO 30`

Amiga and Atari ST

Use Amos or Stos

Replace k\$=GET\$ with k\$=INPUT(1)

tell you things like which version of the program it is and so on. Still, things could be even better. At the moment, any key except Y or y will repeat the program, despite the fact we've asked for a yes or no answer (Y/N).

There is one way round this. Just add another IF ... THEN. Put in the following line and you've got a much more friendly program:

```
45 IF k$<>"N" OR k$<>"n" THEN GOTO 30
```

All right, the main part of the program does nothing except print a bit of text. But the point is that **WHATEVER** someone types it is dealt with properly now.

The program caters for every input so there can be no freak happenings when someone presses X, say. And that's what makes a more polished job.

These routines will work - with the small changes listed - on most home micros. But there are more elegant ways of doing the same thing.

The snag is that you need to use different commands for different micros.

For example, on the BBC Micro you can use the command INSTR as part of your check for which letter was pressed.

On Page 9 of the August issue of *Let's Compute!* the green part of the listing shows this.

If you can write a super sophisticated bit of program to check for a Y/N reply on your micro, let us see it.

We'll pin the best to the Noticeboard and send the author a baseball cap.

Do you think YOU'VE written a program that can't be improved? Then send it along to the Program Doctor. He LOVES a challenge!

**Think you know a lot
about pop music?
Here's a program
to test you out.**

TINA'S

If you fancy being a quizmaster - now's your chance!

Clever Tina Turtle has devised a Pop Quiz you can play on your computer. And she's made it easy for you to change it into any type of quiz you like.

After you've typed in the program and RUN it you'll be asked 10 questions. Get one right and you move on to the next.

If you get one wrong, you'll be given the right answer before you move on to the next question. Note that as the data is all in capitals you must answer in capitals.

Once you've answered all 10 questions you're given your score and a message.

At the moment there are two messages - printed by Lines 190 and 200. You can easily change them or - by inserting more similar lines - add messages of your own.

If you want to add more questions change Line 330 to say how many there are in all. Then add more lines similar to those already at 350 to 440 at the end of the program.

Start numbering the extra DATA statements at Line 450.

Obviously, if you wish you can change all the data at the end of the program and put anything you want there. Geography, history, trivial pursuit - any type of question will do. Don't forget that if you change the type of quiz you'll also have to alter lines 100 and 120. This simple quiz has many uses. If you come up with any novel questions and answers let us see them. There will be a prize for any DATA lines we pin to the noticeboard.



```

10 REM Pop quiz
20 REM (c)Let's Compute!
30 READ M
40 DIM SS(M):DIM S(M)
50 LET P=0
60 FOR C=1 TO M
70 READ SS(C),GS(C)
80 NEXT
90 CLS
100 PRINT:PRINT "Let's Compute! Pop Q
uiz"
110 FOR C=1 TO M
120 PRINT:PRINT "Which group or artis
t sang ":PRINT SS(C);"?
130 INPUT AS
140 PRINT
150 IF AS=GS(C) THEN GOSUB 230
160 IF AS<>GS(C) THEN GOSUB 270
170 NEXT
180 PRINT:PRINT:PRINT "You got ";P;"
out of ";M
190 IF P>0.75*M THEN PRINT "That's no
t bad"
200 IF P<=0.75*M THEN PRINT "You coul
d do better"
210 PRINT
220 END
230 REM RIGHT ANSWER SUBROUTINE
240 PRINT "Well done, you're right"
250 LET P=P+1
260 RETURN
270 REM WRONG ANSWER SUBROUTINE
280 PRINT "Hard Luck! You're wrong"
290 PRINT "The right answer is: ";GS(
C)
300 RETURN
310 REM DATA LINES
320 REM NEXT IS NUMBER OF QUESTIONS
330 DATA 10
340 REM NEXT ARE SONGS, GROUPS
350 DATA "ITSY BITSY TEENY WEENY", "BO
MBALURINA"
360 DATA "TURTLE POWER", "PARTNERS IN
KRYME"
370 DATA "LISTEN TO YOUR HEART", "ROXE
TTE"
380 DATA "NAKED IN THE RAIN", "BLUE PE
ARL"
390 DATA "TONIGHT", "NEW KIDS ON THE B
LOCK"
400 DATA "YOU CAN'T TOUCH THIS", "MC H
AMMER"
410 DATA "WHERE ARE YOU BABY?", "BETTY
BOO"
420 DATA "ENGLISHMAN IN NEW YORK", "ST
ING"
430 DATA "PRAYING FOR TIME", "GEORGE M
ICHAEL"
440 DATA "THIEVES IN THE TEMPLE", "PRI
NCE"

```

IS YOUR MACHINE HERE?

**Electron/BBC/Archimedes/
ST (Stos)/Amiga (Amos)/
PC (GW-Basic)**

The program will work as shown.

Commodore 64/128

*Use capital letters throughout.
Change line 90 to:*

90 PRINT CHR\$(147)

Spectrum

*Use 48k mode.
Change line 40 to:*

40 DIM SS(M+1,32):DIM GS(M,32):
DIM AS(32)

Insert a new line:

135 IF LEN(AS)<10 THEN LET AS=AS+" ":GO
TO 135



Leave £10 in a piggy bank and a year later you'll still have £10. But put it in a bank account and in the same time it could have grown by an extra 70p.



Pennies from Heaven

This program teaches your computer how to work out your bank interest.

And shows you how that nestegg will grow and grow.

How much you can make depends on two things: The present interest rate - which can change at any time. And how long you leave your money in the bank.

To find out how much interest you're earning just answer the simple questions your computer will ask you. Have your bank book in front of you. You'll need to tell the computer about each time you put money in or took some out.

First you'll be asked to give the date you did this. Type it in as three numbers, separated by commas.

So, November 5 this year would be entered as:

5,11,90

Then you need to say how much you put in or took out. If you took money out, put a minus (-) in front of the number you type in.

Next, tell the program what the interest rate was at the time. You're then asked if you want to enter any more. You can reply Y or N.

If you answer Y you'll get the same questions again. Like this, you can make as many entries as are in your bank book. If you answer N you'll be asked for today's date.

Once you've typed this in, you'll be told what interest you're now due.

```

10 REM Interest
20 REM by Ken Hughes
30 REM (c)Let's Compute!
40 DIM D(100)
50 DIM M(100)
60 DIM Y(100)
70 DIM C(100)
80 DIM R(100)
90 CLS
100 LET N=0
110 LET N=N+1
120 INPUT "What is the entry date (d,m,y)";D(N),M(N),Y(N)
130 INPUT "How much";C(N)
140 INPUT "What interest rate";R(N)
150 LET R(N)=R(N)/100/365
160 INPUT "Do you want to enter more (Y/N)";AS
170 IF AS="y" OR AS="Y" THEN GOTO 110
180 IF AS<>"n" AND AS<>"N" THEN GOTO 1
60
190 IF Y(1)<100 THEN Y(1)=Y(1)+1900
200 LET N=N+1
210 INPUT "Enter today's date: d,m,y ";D(N),M(N),Y(N)
220 LET TBAL=0
230 FOR M = 2 TO N
240 IF Y(M)<100 THEN Y(M)=Y(M)+1900
250 LET T1=D(M-1)+(M(M-1)*30.5)+(Y(M-1)*12*30.5)
260 LET T2=D(M)+(M(M)*30.5)+(Y(M)*12*30.5)
270 TEMPDATE=T2-T1
280 LET TBAL=TBAL+C(M-1)
290 LET TBAL=TBAL+(TBAL*R(M-1)*TEMPDATE)
300 NEXT M
310 LET TBAL=INT(TBAL*100)
320 LET TBAL=TBAL/100
330 PRINT:PRINT "You now have: ";TBAL

```

Averaging it out

To keep the listing short we've told your computer each month has 30.5 days rather than 28, 29, 30 or 31. This means that the results are not quite accurate - but not that far out.

IS YOUR COMPUTER HERE

C64/128: Change line 90 to:
90 PRINT CHR\$(147)

ST (Stos) Amiga (Amos): Put # after all the variable names that don't end in \$. So D(100) becomes D\$(100), TBAL becomes TBAL# and so on.

THE LISTING WORKS AS IT IS ON OTHER MICROS

LET'S

Secret message maker!

**START TO
BUILD YOUR
OWN DATABASE**

**For ALL users
of the Electron,
BBC Micro, PC,
Amiga, C64/128,
Archimedes,
Amstrad CPC,
Atari ST and
Spectrum**

No 4 November 1990
A Database Publication



9 770960 283003

Do-it-yourself Bonfire Hangman

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

BUILD YOUR OWN

BU

*Who needs a database?
We all do! Over the
next few months
Let's Compute! is
going to show you
how easy it is to
build your own.*

ON THE CARDS

We are surrounded by databases! You may not realise it but we use them all the time.

Have you ever had to use the phone book to find out the number of a friend? This is a kind of database – it's a list of names, addresses and numbers.

You have probably seen teletext – Ceefax and Oracle – on your TV. Hundreds of pages can be called up by pressing a few buttons.

There is a wide variety of information on teletext. It ranges from jokes to the latest news. But it's just another kind of database.

What would YOU use a database for? You can keep a record of your favourite football team's results. Or a list of the names of the songs on your cassettes.

If you collect anything – stamps, toy cars, records, cards, stickers and so on – you need to keep track of what you've got. The best way to do this is with a database.

It's easy to create your own database on pieces of paper. Just collect the information and write it down. But it's much more fun to use your computer.

Paper is easily lost. And it can take quite a long time to search for the item you need.

But you can do it in a flash if you have a database

in your computer. And that's what we're going to make over the next few months.

It will be like a card index file. You'll have about 51 "cards" and you can write down anything you want on them.

Then, if you want, you'll be able to sort them into alphabetical order. This will make it really easy to find any card you want.

You'll be able to store them on tape or disc so they can be used again and again. And you'll be able to print them out. So you can always have an up to date written copy of your cards.

How do you use your database? You start by choosing what you want to do from a list on your screen. This is called the menu. It will look like the panel above.

So let's begin building our database. And the first thing we need to create is the menu. That is what this month's program does.

Type it in and then SAVE it on tape or disc. Then you can RUN it and have a look at the things you'll be able to do with your computerised database.

Try picking one of the choices. All you need to do is type in the number of the one you want.

At the moment nothing useful will happen. In fact you'll just get an error message telling you that lines are missing!

DATABASE BUILD YOUR OWN DATABASE BUILD YOUR OWN DATABASE

```

10 REM Card Index Database
20 REM By Roland Waddilove
30 DIM C$(255)
40 FOR X=1 TO 255
50 LET C$(X)="*"
60 NEXT X
70 REM ===== MAIN MENU =====
80 CLS
90 PRINT " CARD INDEX DATABASE"
100 PRINT "-----"
110 PRINT
120 PRINT "1. Load a box of cards"
130 PRINT "2. Save a box of cards"
140 PRINT "3. Read/search for a card"
150 PRINT "4. Write on a new card"
160 PRINT "5. Throw away old cards"
170 PRINT "6. Change something on a card"

```

```

180 PRINT "7. Print the cards"
190 PRINT "8. Sort the cards into order"
200 PRINT "9. Stop the program"
210 PRINT
220 LET N=0
230 FOR X=1 TO 251 STEP 5
240 IF C$(X)="*" THEN LET N=N+1
250 NEXT X
260 PRINT "Cards free=";N;" / Cards used=";51-N
270 PRINT
280 PRINT "What do you want to do?"
290 INPUT "Please type in a number (1-9)";N
300 CLS
310 IF N=9 THEN END
320 ON N GOSUB 1000,2000,3000,4000,5000,6000
,7000,8000
330 GOTO 70

```

CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1-9)

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes/ CPC/PC (GW-Basic)

The program works as shown

Amiga (Amos)

Change Line 80 to:

80 MODE 0:KEY OFF:HIDE

Atari ST (Stos)

Change Line 80 to:

80 CLS:HIDE

Spectrum

Select 48k mode

Change or add the following lines:

```

30 DIM C$(255,32)
35 DIM B$(32):LET B$="*"
240 IF C$(X)=B$ THEN LET N=N+1
310 IF N=9 THEN STOP
320 GOSUB N*1000

```

Commodore 64/128

Enter the program in capitals

Change CLS in lines 80 and 300 to:
PRINT CHR\$(147);

How it works

First, the program tells the computer to save room in its memory for the cards.

It's called DIMension-ing the array C\$. In this will be held all the information on the 51 cards.

It's dimensioned to 255 so that each card can use five of the memory areas reserved by the DIM.

Next, the menu is printed on the screen. An INPUT command is then used to find out which choice you've asked for.

The number that you type in is held in N and the command on Line 260 will make the program jump to the right subroutine.

All we've got to do now is put the subroutines into the program. We'll start doing that next month.

NEXT MONTH

We'll start adding the missing lines. And then you'll be able to begin writing your cards



Remember the 5th of November

An old favourite gets a new look in this unique version of Hangman

As bonfire night approaches, here's a game to get you in the mood.

It's our special version of Hangman. On your screen is a picture of a bonfire. And the words you have to guess all have a firework theme.

Just type in the listing. Then **SAVE** it and **RUN** it.

The dashes that appear on the screen show you the length of the word you have to guess. Press any letter and see what happens.

If it's in the word it will appear in the correct place in the row of dashes.

But if you're wrong? The bonfire will start building up from the bottom of the screen.

Try to guess all the letters in the word before the picture is complete.

The program has been written so that you can easily change it. Look at Lines 550 to 700.

They draw the picture out of letters – just like they are in our Alphabet Artist contest. You can put any picture you want in the program. It must

be 16 lines high and no more than 25 letters long.

If your computer has special features you can use them. For instance, C64 owners could use the graphic characters. BBC Micro owners could design a colourful Mode 7 picture.

If you can define other special characters on your computer you can draw the picture from your own shapes.

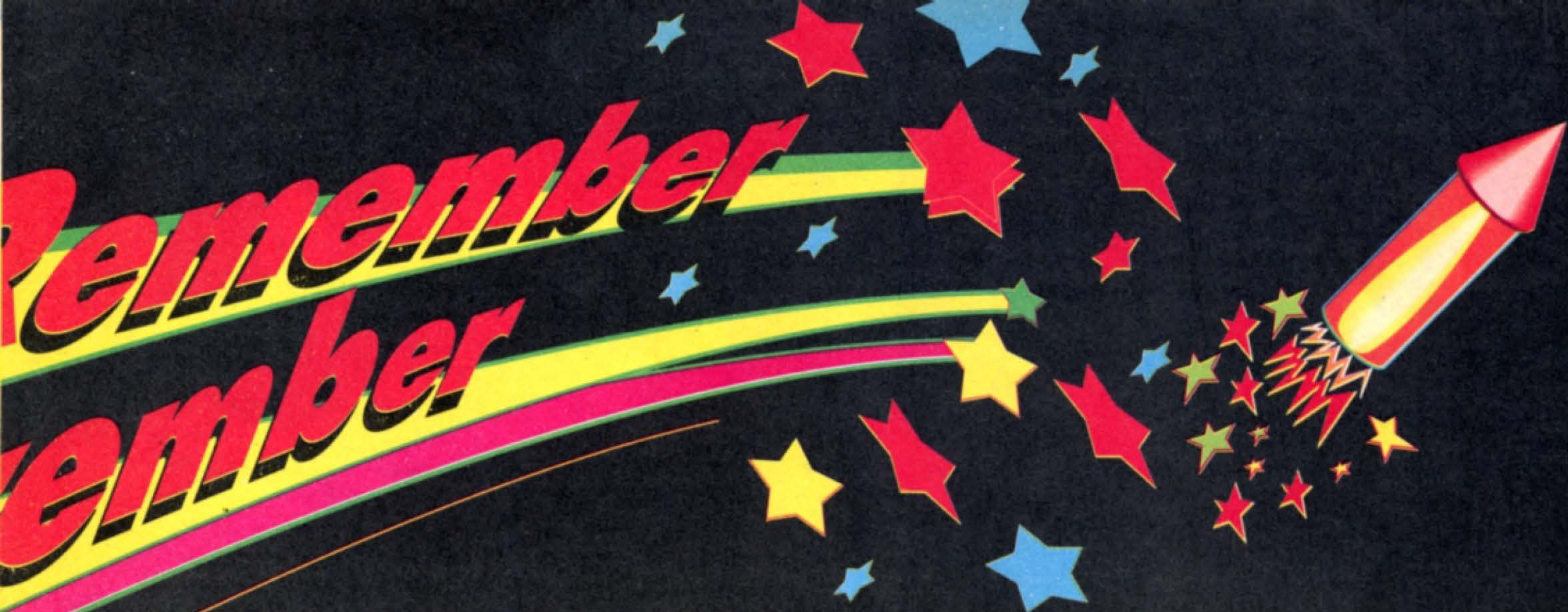
As well as putting your own pictures in the game, you can also use your own words. Put them in **DATA** statements like Lines 720 to 740.

You can have as many words as the memory of your computer will allow. Just make sure that you leave the line that says **DATA "EOF"** at the very end of the program.

At the moment it's Line 750. You can give it a higher number if you want.

So type it in, play it, change it and play it again.

You'll have hours of fun with this one!



```

10 REM BONFIRE HANGMAN
20 REM (c) Let's Compute!
30 LET X=RND(-TIME)
40 DIM W$(500)
50 LET c=0:LET ES="EOF"
60 LET c=c+1
70 READ W$(c)
80 IF W$(c)<>ES THEN GOTO 60
90 LET numwr=c-1
100 LET X$=W$(RND(numwr))
110 LET c$=X$
120 LET wr=0
130 LET lgth=LEN(X$)
140 LET g=1
150 LET gap=(20-lgth)/2
160 CLS
170 LET X=26:LET Y=1:GOSUB 500:PRINT "
TRIED"
180 LET X=gap:LET Y=17:GOSUB 500:PRINT
STRING$(lgth,"-")
190 LET ok=0
200 LET X=3:LET Y=19:GOSUB 500:PRINT "
This is move ";g
210 PRINT:PRINT "  PRESS A LETTER";
220 GOSUB 510
230 LET g=g+1
240 FOR c=1 TO lgth
250 IF L$<>MID$(X$,c,1) THEN GOTO 290
260 LET X$=LEFT$(X$,c-1)+" "+MID$(X$,c
+1)

```

```

270 LET X=gap+c-1:LET Y=17:GOSUB 500:P
RINT L$
280 LET ok=-1
290 NEXT c
300 IF ok=0 THEN LET X=28:LET Y=3+2*wr
g:GOSUB 500:PRINT L$
310 IF X$=STRING$(lgth," ") THEN GOTO 3
40
320 IF ok=-1 THEN GOTO 190
330 LET wr=wr+1:GOTO 370
340 CLS
350 LET X=5:LET Y=17:GOSUB 500:PRINT "
WELL DONE"
360 GOTO 450
370 LET X=0:LET Y=16-2*wr:GOSUB 500
380 IF wr=8 THEN GOTO 410
390 ON wr GOSUB 690,670,650,630,610,5
90,570
400 GOTO 190
410 GOSUB 550
420 LET X=1:LET Y=17:GOSUB 500:PRINT "
HARD LUCK "
430 PRINT:PRINT "  THE WORD WAS "
440 PRINT SPC((18-LEN(c$))/2)c$;SPC((1
8-LEN(c$))/2)
450 PRINT "  ANOTHER GO? ";
460 GOSUB 510
470 IF L$="Y" THEN GOTO 100
480 IF L$<>"N" THEN GOTO 460
490 CLS:END

```

```

500 PRINTTAB(X,Y);:RETURN
510 LET L$=GET$
520 IF L$>"Z" THEN LET L$=CHR$(ASC(L$)
-32)
530 RETURN
540 REM Picture by Colin Steele
550 PRINT "  0"
560 PRINT " /@_ *"
570 PRINT "* I"
580 PRINT " L *"
590 PRINT " ***"
600 PRINT "***** *"
610 PRINT " # #* *"
620 PRINT "##### *"
630 PRINT "### # ** *"
640 PRINT " ## # *      00"
650 PRINT "#### ####      <>"
660 PRINT " # #####* **      <<<####"
670 PRINT "### ### ## * *      < ####"
680 PRINT " # ### # # **      ####"
690 PRINT "##### ## #* *      II"
700 PRINT " # ## ## ## **      II"
710 RETURN
720 DATA "BONFIRE","FIREWORK","SPARKLER
","TREACLE","APPLES","ROCKET"
730 DATA "CHESTNUT","GUY","GUNPOWDER",
"LOT","CANDLE","MATCHES"
740 DATA "BANGERS","FUN","DISPLAY"
750 DATA "EOF"

```

IS THIS IS YOUR COMPUTER?

BBC Micro/Electron/ Archimedes

The listing works as shown

Amstrad CPC

Change the following lines:

```

30 RANDOMIZE TIME
100 LET X$=W$(INT(RND*numwr+1))
500 LOCATE X+1,Y+1: RETURN
510 LET L$="":WHILE L$="":LET L$=INKEY
$:WEND

```

ST(stos)/Amiga(AMOS)

Change the following lines:

```

30 KEY OFF : MODE 0 : LET x=RND(TIMER
)
100 LET X$=W$(RND(numwr-1)+1)
440 PRINT SPACES((18-LEN(c$))/2)c$;SPA
CES((18-LEN(c$))/2)
500 LOCATE x,y:RETURN
510 LET L$=INPUT$(1)

```

Amiga: Use CLS instead of MODE 0

C64/128

Enter the program in capitals.
Change or add the following lines:

```

25 FOR N=1 TO 20:DASH$=DASH$+"-":SKIP
$=SKIP$+" ":NEXT
100 LET X$=W$(INT(RND(0)*NUMWRD+1))
160 PRINT CHR$(147)
180 PRINT LEFT$(DASH$,LGTH)
310 PRINT LEFT$(SKIP$,LGTH)
340 PRINT CHR$(147)
440 PRINT LEFT$(SKIP$, (18-LEN(c$))/2);
c$;LEFT$(SKIP$, (18-LEN(c$))/2)
490 PRINT CHR$(147):END
500 POKE 211,X:POKE214,Y:SYS58732:RETU
RN
510 GET L$:IF L$="" THEN GOTO 510

```

PC (GW-Basic)

Change the following lines:

```

30 RANDOMIZE TIMER
100 LET X$=W$(INT(RND*numwr+1))
500 LOCATE X+1,Y+1: RETURN
510 LET L$="":WHILE L$="":LET L$=INKEY
$:WEND

```

Spectrum

Use 48k mode.
Change or add the following lines:

```

30 RANDOMIZE
40 DIM W$(500,16): DIM W(500): DIM E
$(16): DIM S$(16)
45 LET S$=" ": LET B$="-----"
-"
70 READ X$: LET W$(c)=X$: LET W(c)=LE
N (X$)
100 LET c= INT (RND*numwr+1): LET X$=
W$(c)
130 LET lgth=W(c)
180 LET X=gap:LET Y=17:GOSUB 500:PRINT
B$(1 TO lgth)
250 IF L$<>X$(c) THEN GOTO 290
260 LET X$=X$(1 TO c-1)+" "+X$(c+1 TO
)
310 IF X$=S$ THEN GOTO 340
390 LET LINE=710-20*wr: GO SUB LINE
440 PRINT S$(1 to (18-lgth)/2);c$
490 CLS:STOP
500 PRINT AT Y,X;:RETURN
510 LET L$=INKEY$: IF L$="" THEN GOTO
510
520 IF L$>"Z" THEN LET L$=CHR$(CODE(L$)
)-32)

```


CRACK THE CODE

Have you ever wanted to send someone a secret message? Alan McLachlan has just the program for you. It will mix up your words so that only people who know the code can read them.



Secret agents use a computer program to send top secret messages to each other. Now *Let's Compute!* readers can do the same.

Using our program, your messages are turned into coded text. The way they are coded depends on a secret number that YOU decide.

Without knowing the secret number it is very difficult for anyone else to read your message. Don't forget it. And only let authorised people know it!

But a program that codes messages is no use unless you have a program to decode them as well. So this program does both.

Type it in, SAVE it and RUN it. The first question you'll be asked is whether you want to code or decode a message. Answer C or D.

Then you'll be asked for your secret code number. This can't be any old number. The panel opposite tell you how to make one up.

If the number you enter is not valid the

computer will inform you. It will also give you some suitable suggestions. And make sure you remember the number – you won't be able to decode your message without it!

Next, just type in your message. It can be up to 255 letters long on most computers (80 on a C64). After you press Return there will be a slight pause and then you will see your words converted into secret code!

Here's what a coded message looks like:

```
iw$XLvxiigkri$e(wex$exw$piri$wxJ$$empi)
```

Can you crack it? Any coding system can be broken. But you'll discover that the *Let's Compute!* code is very tricky to break.

If you do discover what the message says, let us know. We'll send a baseball cap to the person who sends us the first correct translation!

THINK OF A NUMBER

The way a message is encoded depends on your secret number. When you make one up there are three rules you must follow:

- It must be six figures long.
- The first five figures must be all the numbers 1 to 5, but they can be in any order.
- The last figure has to be a number between 0 and 4.

Here are some examples of numbers you can use:

132542543210
543212

HOW THE COMPUTER CODES THE MESSAGE

The computer takes your message and mixes it up. The way it does this depends on the secret number you typed in.

Here's how the the message *Hi there Pete* is coded using the number 23154

- The message is split into groups of five letters. So our message splits like this:

Hi th ere P ete

Note that the space counts as a letter. The number of letters in the last group will sometimes be less than five. That is what has happened here.

- Next, the groups of letters are mixed up. The first five numbers of the secret number are used to decide the new order.

So, using the pattern 23152, The first group – *Hi th* – becomes *i Hht*.

The last group is not mixed up unless it is exactly five characters long.

- The letters are then all changed to letters higher up the alphabet. The last letter of your secret code lets the computer know how much to shift them.

The last number of our example code is 2. So A becomes C, B becomes D and so on. Space and letters at the end of the alphabet become characters such as brackets and commas.

With the shift of two places, our message (*i Hht*) is changed to *k"Jjv*.

Here's how the word *COMPUTER* is coded using several secret numbers:

Code number	Coded word
123450	COMPUTER
123451	DPNQVUFS
543210	UPMOCTER
543212	WROQEVGT

Now for an example you can try for yourself. Use the code 341253 and key this in:

There's a message on your answering machine

The result – in code – is:

huWkh#d*v#vvphd#rjhqrx#juqv#dzlqhujdf#pklqh

```

10 REM Crack the code!
20 REM (c) Let's Compute!
30 CLS
40 DIM P(6)
50 PRINT "The Let's Compute! Secret C
oder"
60 PRINT:PRINT:PRINT
70 GOSUB 200
80 PRINT:PRINT
90 PRINT "Code or Decode (C/D)":GOSUB
670
100 IF AS="C" OR AS="c" THEN GOSUB 410
:GOTO 130
110 IF AS="D" OR AS="d" THEN GOSUB 490
:GOTO 130
120 GOTO 90
130 PRINT:PRINT "Do you want to use th
e program again?":GOSUB 670
140 IF AS="N" OR AS="n" THEN CLS:END
150 IF AS<>"Y" AND AS<>"y" THEN GOTO 1
30
160 CLS:PRINT "Do you want to use the
same code number?":GOSUB 670
170 IF AS="Y" OR AS="y" THEN GOTO 80
180 IF AS="N" OR AS="n" THEN GOTO 70
190 GOTO 160
200 REM SPECIAL NUMBER
210 INPUT "What is your code number";N
$
220 IF LEN (NS)<>6 THEN LET FLAG=1:GOTO
280
230 LET FLAG=0:LET MS=LEFT$(NS,5)
240 FOR I = 49 TO 53
250 IF INSTR(NS,CHR$(I))=0 THEN LET FL
AG=1
260 NEXT I
270 IF RIGHT$(NS,1)<"0" OR RIGHT$(NS,1
)>"4" THEN GOSUB 310:GOTO 210
280 IF FLAG=1 THEN GOSUB 310:GOTO 210
290 RETURN
300 REM WRONG NUMBER
310 CLS:PRINT "Code number wrong (see
Let's Compute!)":PRINT
320 RETURN
330 REM MESSAGE
340 CLS:PRINT "Type in your message"
350 LET MS="":LET L=0
360 GOSUB 670:IF AS=CHR$(13) THEN GOTO
390

```

```

370 LET MS=MS+AS:LET L=L+1:IF AS=CHR$(
127) AND L>1 THEN LET L=L-2:LET MS=LEFT$
(MS,L)
380 LET X=0:LET Y=4:GOSUB 700:PRINT MS
;" ":GOTO 360
390 IF MS="" THEN GOTO 340
400 RETURN
410 REM CODE
420 FOR I=1 TO 6:LET P(I)=VAL(MID$(NS,
I,1)):NEXT I
430 GOSUB 330:LET MS=LEFT$(MS,L)
440 CLS:PRINT "Your message is being c
oded"
450 LET SS=MS:GOSUB 570:LET CS=RS
460 CLS:PRINT "Your message:";PRINT:PR
INT MS
470 PRINT:PRINT"has been coded to:":PR
INT:PRINT CS
480 RETURN
490 REM DECODE
500 FOR I=1 TO 5:LET P(VAL(MID$(NS,I,1
)))=I:NEXT I:LET P(6)=-VAL(MID$(NS,6,1))
510 GOSUB 330:LET CS=LEFT$(MS,L)
520 CLS:PRINT "Your message is being
decoded."
530 LET SS=CS:GOSUB 570:LET MS=RS
540 CLS:PRINT "Your coded message:":PR
INT:PRINT CS
550 PRINT:PRINT"has been decoded to:":
PRINT:PRINT MS
560 RETURN
570 REM DO CODE/DECODE
580 LET RS="":IF L<5 THEN GOTO 640
590 FOR I = 0 TO INT((L-5)/5)
600 FOR J=1 TO 5
610 LET RS=RS+CHR$(ASC(MID$(SS,I*5+P(J
),1))+P(6))
620 NEXT J
630 NEXT I
640 LET I=5*INT(L/5)
650 IF I<L THEN LET I=I+1:LET RS=RS+CH
R$(ASC(MID$(SS,I,1))+P(6)):GOTO 650
660 RETURN
670 REM GET
680 AS=GET$
690 RETURN
700 REM TAB(X,Y)
710 PRINT TAB(X,Y);:RETURN

```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The programs work as shown

Amiga and Atari ST

Remove line 370

Change the following lines:

```

10 MODE 0:KEYOFF:HIDE
680 AS=INPUT$(1)
710 LOCATE X,Y:RETURN

```

NOTE Delete won't work as you type your message in, so take care.

Amiga use CLS instead of MODE 0

Commodore 64/128

Enter the program in capitals.

Replace every CLS with PRINT CHR\$(147);

Change or add the following lines:

```

250 LET F=0: FOR J=1 TO 5:IF MID$(NS,J
,1)=CHR$(I) THEN LET F=1
255 NEXT J: IF F=0 THEN LET FLAG=1
370 LET MS=MS+AS:LET L=L+1
375 IF AS=CHR$(20) AND L>1 THEN LET L=
L-2:LET MS=LEFT$(MS,L)
680 GET AS:IF AS="" THEN GOTO 680
710 POKE 211,X:POKE 214,Y:SYS58732:RET
URN

```

Spectrum

Select 48k mode

Change or add the following lines:

```

140 IF AS="N" OR AS="n" THEN CLS:STOP
230 LET FLAG=0:LET MS=NS ( TO 5)
250 LET FOUND=0: FOR J=1 TO 5:IF MS(J)
=CHR$(I) THEN LET FOUND=1
255 NEXT J: IF FOUND=0 THEN LET FLAG=1
270 IF NS(6)<"0" OR NS(6)>"4" THEN LET
FLAG=1
370 LET MS=MS+AS: LET L=L+1: IF AS=CHR
$(12) AND L>1 THEN LET L=L-2:LET MS=MS T
O L)
420 FOR I=1 TO 6: LET P(I)=VAL(NS(I)):
NEXT I
430 GOSUB 330:LET MS=MS( TO L)
500 FOR I=1 TO 5:LET P(VAL (NS(I)))=I:
NEXT I:LET P(6)=-VAL (NS(6))
510 GOSUB 330:LET CS=MS( TO L)
610 LET RS=RS+CHR$(CODE (SS(I*5+P(J))
)+P(6))
650 IF I<L THEN LET I=I+1:LET RS=RS+CH
R$(CODE (SS(I))+P(6)):GO TO 650
675 IF INKEY$<>" " THEN GO TO 675
680 LET AS=INKEY$: IF AS="" THEN GO TO
680
710 PRINT AT Y,X;: RETURN

```

Amstrad CPC/PC (GW-Basic)

Change the following lines:

```

680 LET AS="":WHILE AS="":LET AS=INKEY
$:WEND
710 LOCATE X+1,Y+1:RETURN

```


SHOO



Discover how easy it is to make your computer move things round the screen. Then type in and play this fast-action game

CONTROLLING YOUR SPEED

Line 10 lets you decide how fast you want the objects to move. You can start with a slow game and make it faster as you get better.

Increase the number in *D=10* to slow the game down.

Have you ever wondered how the objects are moved around the screen in games? The answer is that they never move at all!

Put simply, they are wiped out from one position then redrawn nearby. It happens so fast that your eye is tricked into thinking they're moving.

How realistic the movement is depends on two things: How far the object is shifted. And how quickly it can be drawn.

Imagine you want to move a star - * - around the screen. Start by printing it.

Next you print a space right over. Then you reprint the star nearby. That's all that Lines 50 and 60 in the program do.

The effect is a falling star. We can make a game from this simple start. Let's add an arrow which can from right to left.

Line 130 draws an arrow, together with the space that wipes out the one already there. It starts moving when you press the spacebar.

So, with just a few lines of program, we've got a game. The object is to hit the star with the arrow.

Type the listing in, SAVE it and RUN it. See if you can hit ALL the stars that fall!

TING STARS



```

10 LET D=10
20 CLS
30 LET X=31:LET Y=10:GOSUB 220:PRIN
T"<"
40 LET I=1
50 LET X=3:LET Y=1-1:GOSUB 220:PRIN
T" "
60 LET X=3:LET Y=1:GOSUB 220:PRINT"
*"
70 FOR J=1 TO D:NEXT J
80 LET I=I+1
90 GOSUB 240
100 IF I<20 AND K$<>" " THEN GOTO 50
110 IF I=20 THEN LET X=6:LET Y=21:GO
SUB 220:PRINT "TOO SLOW":GOTO 180
120 FOR K=30 TO 3 STEP -1
130 LET X=K:LET Y=10:GOSUB 220:PRINT
"<"
140 FOR J=1 TO D:NEXT J
150 NEXT K
160 IF I=11 THEN LET X=6:LET Y=21:GO
SUB 220:PRINT "WELL DONE"
170 IF I<>11 THEN LET X=5:LET Y=21:G
OSUB 220:PRINT "YOU MISSED"
180 LET X=5:LET Y=22:GOSUB 220:PRINT
"ANOTHER GO?"
190 GOSUB 240
200 IF K$="Y" THEN RUN
210 IF K$<>"N" THEN GOTO 190
220 PRINTTAB(X,Y);
230 RETURN
240 LET K$=INKEY$(0)
250 RETURN

```

MORE IDEAS

This fun game is designed to show you how easy it is to move things on a computer screen.

Use it as a base to build a bigger and better game. Here are just a few ideas for features you could add:

- Put in a timer so you can see how many you can hit in a minute.
- Make the speed of the star slightly different each time.
- Make the arrow start from a different height each time.
- Use your own designs of characters to make it a more realistic game.
- Have some stars going up as well as down.
- Put messages like *open your eyes* in place of the ones we have used. And you could make it print different messages. These would depend on how near the arrow was to the star.

IS THIS YOUR COMPUTER?

BBC Micro/Electron/Archimedes

The listing works as shown

C64/128

Enter the program in capitals.
Change or add the following lines:

```

20 PRINT CHR$(147)
130 POKE 211,X:POKE 212,Y:SYS 58732
220 GET K$

```

ST(Stos)/Amiga(AMOS)

Change the following lines:

```

20 KEY OFF : MODE 0
130 LOCATE X,Y
240 LET K$=INPUT(1)

```

Amiga: Use CLS instead of MODE 0

Amstrad CPC

Change the following lines:

```

130 LOCATE X+1,Y+1
240 LET K$=INKEY$

```

PC (GW-Basic)

Change the following lines:

```

130 LOCATE X+1,Y+1
240 LET K$=INKEY$

```

Spectrum

Use 48k mode.

Change or add the following lines:

```

130 PRINT AT Y,X;
240 LET K$=INKEY$

```


A large, stylized logo featuring the number "99" in a bold, black, sans-serif font, with a small "¢" symbol to its right. The entire logo is enclosed within a thick, black, circular border that has a slightly irregular, hand-drawn appearance. The background of the entire advertisement is a solid, vibrant orange.

to
in
h

A black silhouette of the Eiffel Tower is centered on a red background. To the left of the tower, the words "to", "in", and "h" are stacked vertically in a bold, yellow, sans-serif font.

Start collecting



badges
See Page 4

No 5 December 1990
A Database Publication



WIN A YEAR'S SUPPLY OF FREE GAMES!

PLUS LOTS OF OTHER
CONTEST PRIZES!



**"IT'S
AS EASY
AS UN,
DEUX,
TROIS!"**

**Check
how well
you can
count in
French.
It's the
fun way
to learn**

Have you ever stared at lists of foreign words you need to learn – and not know where to start? Help is now

at hand with the *Let's Compute!* French number test.

- Type in the listing, **SAVE** it and **RUN** it.
- You will first be asked for the highest number you want testing on? The biggest number you can answer is 69.
- Then you're asked what the lowest number is. This can be any figure from one up to your highest.
- Now the test begins. A number will be shown. Just type in the French words for it. If you're right a tick will be shown next to your answer. Wrong answers get a cross.
- After 15 questions you're given your score. You can press the spacebar for another test or press **C** to look at the correct answers.
- It's always a good idea to check your answers. You can see where you went wrong. And try not to make the same mistake again.
- So start with the numbers from 1 to 5. Even if you don't know any and get 0 out of 15 in your first test, you'll soon learn them. And once you know the first five, try a few more.
- So start now. Before long you'll know all the French numbers up to 69.

Take care!

Make sure you enter the **DATA** lines exactly as they are printed. If you make a mistake you could be learning the wrong spelling of a French word.

Note that, as we've used lower case letters you should answer in lower case. If, instead, you enter the listing in capitals, answer in capitals.

Un

What is the highest number you want?75
69 is the biggest

What is the highest number you want?45

What is the lowest number you want?56
Don't be silly!

What is the highest number you want?45

What is the lowest number you want?20

Choosing the size of your test

Trois

Press Space for
another go.
Press C for
correct answers.

quarante cinq	x
vingt-trois	/
trente-sept	/
vingt-neuf	/
vingt-huite	x
quarante-quatre	/
trente-trois	/
quarante-quatre	/
trente-et-un	x
trente-trois	/
vingt-cinq	/
quarante-quatre	/
vingt	/
trente-sept	/
vingt et un	/

You got 12 out of 15.

Checking your score

```
10 REM FRENCH NUMBERS
20 REM (c) Let's Compute!
30 DIM N(100):DIM WS(100)
40 PRINT:LET X=RND(-TIME)
50 LET S=0:LET SS=" "
60 FOR I=1 TO 19:LET SS=SS+" ":NEXT I
70 FOR I=1 TO 19:READ WS(I):NEXT I
80 FOR I=20 TO 60 STEP 10
90 READ AS:LET WS(I)=AS
100 LET WS(I+1)=AS+" et un"
110 FOR J=2 TO 9
120 LET WS(I+J)=AS+"-"+WS(J)
130 NEXT J:NEXT I
140 CLS
150 PRINT:INPUT "What is the highest n
umber you want";H
160 IF H>69 THEN PRINT "69 is the bigg
est":GOTO 150
170 PRINT:INPUT "What is the lowest nu
mber you want";L
180 IF L=0 OR L>=H THEN PRINT "Don't b
e silly!":GOTO 150
190 CLS
200 FOR I=1 TO 15
210 LET N(I)=RND(H-L+1)+L-1
```

```
220 NEXT I
230 FOR Q=1 TO 15
240 LET QU=N(Q)
250 LET X=0:LET Y=19:LET TS="What is F
rench for "+STR$(QU):GOSUB 430
260 INPUT RS
270 LET X=0:LET Y=19:LET TS=SS+SS:GOSU
B 430
280 IF RS=WS(QU) THEN LET S=S+1:LET MS
="/"
290 IF RS<>WS(QU) THEN LET MS="x"
300 LET X=20:LET Y=Q+2:LET TS=RS:GOSUB
430
310 LET X=39:LET Y=Q+2:LET TS=MS:GOSUB
430
320 NEXT Q
330 LET X=0:LET Y=19:LET TS="You got "
+STR$(S)+" out of 15.":GOSUB 430
340 LET X=0:LET Y=8:LET TS="Press Spac
e for":GOSUB 430
350 PRINT:PRINT "another go."
360 PRINT "Press C for"
370 PRINT "correct answers."
380 GOSUB 510
390 IF AS="C" OR AS="c" THEN GOSUB 450
```

quarante-deux	/
vingt-cinq	/
trente-trois	/
trente-huit	/
quarante-quatre	/
quarante-et-un	x
trente-neuf	/
quatreante-trois	x
trente-trois	/
vingt-quatre	/
quatrenhte-cinq	x
vingt	/
quateante-cinq	x

What is French for 34?

Deux

Answering the questions

43	quarante-trois	quatreante-trois	x
33	trente-trois	trente-trois	/
24	vingt-quatre	vingt-quatre	/
45	quarante-cinq	quatrenhte-cinq	x
20	vingt	vingt	/
45	quarante-cinq	quateante-cinq	x
33	trente-trois	trente-trois	/
44	quarante-quatre	quarante-quatre	/
45	quarante-cinq	quarante-cinq	/
35	trente-cinq	trente-cinq	/
24	vingt-quatre	vingt-quatre	/
20	vingt	vingt	/
43	quarante-trois	quarante-trois	/
44	quarante-quatre	quarante-quatre	/
34	trente-quatre	trente-trois	x

Press Space for another go.

Quatre

Looking at the right answers

(Note: Spectrum display is slightly different)

```
400 IF AS<>" " THEN GOTO 380
410 RUN
420 REM ** PRINT SUBROUTINE **
430 PRINTTAB(X,Y)TS;
440 RETURN
450 REM ** PRINT CORRECT ANSWERS **
460 LET X=0:LET Y=8:LET TS=SS:GOSUB 43
0
470 PRINT:PRINT SS:PRINT SS:PRINT SS:L
ET X=0:LET Y=3:LET TS="":GOSUB 430
480 FOR Q=1 TO 15:PRINT STR$(N(Q));" "
;WS(N(Q)):NEXT Q
490 PRINT:PRINT "Press Space for anoth
er go.":GOSUB 510
500 RETURN
510 LET AS=GET$
520 RETURN
530 DATA "un","deux","trois","quatre",
"cing","six","sept","huit","neuf","dix"
540 DATA "onze","douze","treize","quat
orze","quinze","seize","dix-sept"
550 DATA "dix-huit","dix-neuf","vingt"
,"trente","quarante","cinquante"
560 DATA "soixante"
```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown

Commodore 64/128

Enter the program in capitals.

Change these lines:

```
140 PRINT CHR$(147);
190 PRINT CHR$(147);
210 LET N(I)=INT(RND(0)*(H-L+1)+L)
310 LET X=38:LET Y=Q+2:LET TS=MS:GOSUB
430
430 POKE 211,X:POKE 214,Y:SYS 58732:PR
INT TS;
510 GET AS:IF AS="" THEN 510
```

Spectrum

Select 48k mode. Change these lines:

```
30 DIM N(100): DIM WS(100,16): DIM RS
(16)
40 PRINT : RANDOMIZE
60 FOR I=1 TO 11:LET SS=SS+" ":NEXT I
210 LET N(I)=INT(RND*(H-L+1)+L)
300 LET X=14:LET Y=Q+2:LET TS=RS:GOSUB
430
310 LET X=31:LET Y=Q+2:LET TS=MS:GOSUB
430
430 PRINT AT Y,X;TS;
510 LET AS=INKEY$:IF AS="" THEN GOTO 5
10
```

PC (GW-Basic)/Amstrad CPC

Change these lines:

```
40 PRINT:RANDOMIZE TIMER
210 LET N(I)=INT(RND*(H-L+1)+L)
430 LOCATE Y+1,X+1:PRINT TS;
510 LET AS=INKEY$:WHILE AS="":L
ET AS=INKEY$:WEND
```

CPC: Use TIME instead of TIMER in Line 40.

Use X+1,Y+1 instead of Y+1,X+1 in Line 430

Amiga and Atari ST

Use Amos or Stos. Change these lines:

```
40 PRINT:LET X=RND(TIMER)
210 LET N(I)=RND(H-L)+L
430 LOCATE X,Y : PRINT TS;
510 AS=INPUT$(1)
```


Electron/BBC Micro/Archimedes core

```

10 REM AL'S GORITHMS ... ELLIPSES
20 REM (c) LET'S COMPUTE!
30 MODE 1
40 GCOLOR,129
50 CLS
60 PRINT:PRINT "JUST A MO, CALCULATIN
6....."
70 REM LOOK-UP TABLE
80 DIM T(72,2)
90 FOR I=0 TO 72
100 LET T(I,1)=COS (RAD (I*5))
110 LET T(I,2)=SIN (RAD (I*5))

```

```

120 PRINT TAB(5,5) I
130 NEXT I
140 CLG
150 FOR P=0 TO 27
160 READ M1,M2,IC,XC,YC,CL
170 GOSUB 200: REM DRAW ELLIPSE
180 NEXT P
190 GOTO 190
200 REM ELLIPSE ROUTINE
220 GCOL 0,CL
230 MOVE XC,YC
240 DRAW XC+M1*T(IC/5,1),YC+M1*T(IC/5,

```

```

2)
250 FOR A=0 TO 360 STEP 15
260 MOVE XC,YC
270 LET X=M1*T(A/5,1)
280 LET Y=M2*T(A/5,2)
290 LET XT=X*T(IC/5,1)-Y*T(IC/5,2)+XC
300 LET YT=X*T(IC/5,2)+Y*T(IC/5,1)+YC
310 PLOT 85,XT,YT
320 NEXT A
330 RETURN

```

PROGRAMMING NOTE!

This program uses SIN and COS in a lot of calculations.

To save your computer working them out each time an ellipse is drawn, they are all calculated at once in Lines 70 to 130.

The results are saved in all the

memory locations called T. Then the answer can be looked up whenever it is needed.

This is a common method of speeding up programs.

It's called using a look-up table.

Spectrum core

```

10 REM AL'S GORITHMS ... ELLIPSES
20 REM (c) LET'S COMPUTE!
30 PAPER 7
40 INK 0
50 CLS
60 PRINT:PRINT "JUST A MO, CALCULATIN
6....."
70 REM LOOK-UP TABLE
80 DIM T(750,2)
90 FOR I=1 TO 750
100 LET T(I,1)=COS (I*PI/360)
110 LET T(I,2)=SIN (I*PI/360)
120 PRINT AT 1,0; I
130 NEXT I
135 FOR I=0 TO 6 STEP 2: POKE USR "A"+
I,BIN 01010101: POKE USR "A"+I+1,BIN 101
01010: NEXT I
140 FOR I=0 TO 21: PRINT AT I,0;"
";NEXT I

```

```

150 FOR P=0 TO 27
160 READ M1,M2,IC,XC,YC,CL
170 GOSUB 200: REM DRAW ELLIPSE
180 NEXT P
190 GOTO 190
200 REM ELLIPSE ROUTINE
210 INVERSE 1:IF CL=0 THEN INVERSE 0
220 LET S=1+500/(M1+M2)
230 LET M1=M1/6: LET M2=M2/6: LETXC=XC
/6: LET YC=YC/6: LET IC=IC*2+1
235 LET XC0=M1*T(IC,1): LET YC0=M1*T(I
C,2)
240 LET XC1=XC+XC0:LET YC1=YC+YC0
250 FOR A=1 TO 721 STEP S
260 PLOT XC1,YC1
270 LET X=M1*T(A,1)
280 LET Y=M2*T(A,2)
290 LET XT=(X*T(IC,1)-Y*T(IC,2))
300 LET YT=(X*T(IC,2)+Y*T(IC,1))
310 DRAW XT-XC0,YT-YC0
315 DRAW -XT-XC0,-YT-YC0
320 NEXT A
330 RETURN

```

NOTE: Line 140 contains 32 graphic characters. To use one, go into graphics mode and type the letter A.

To get into graphics mode press the key labeled GRAPH or press Caps Shift and 9. The cursor will then appear as **G** Press the same key to leave graphics mode.

Until you RUN the program the character will show as A. Once it has been run it will look like it is in the listing above.

Amstrad CPC core

```

10 REM AL'S GORITHMS ... ELLIPSES
20 REM (c) LET'S COMPUTE!
30 MODE 1
35 INK 0,0:INK 1,6:INK 2,24:INK 3,26
40 PAPER 1:PEN 0
50 CLS
60 PRINT:PRINT "JUST A MO, CALCULATIN
6....."
70 REM LOOK-UP TABLE
80 DIM T(750,2):DEG
90 FOR I=0 TO 750
100 LET T(I,1)=COS (I/2)
110 LET T(I,2)=SIN (I/2)
120 LOCATE 5,5:PRINT I
130 NEXT I
140 CLS
150 FOR P=0 TO 27
160 READ M1,M2,IC,XC,YC,CL
170 GOSUB 200: REM DRAW ELLIPSE
180 NEXT P

```

```

190 GOTO 190
200 REM ELLIPSE ROUTINE
210 PLOT -5,-5,CL
220 LET S=1+500/(M1+M2)
230 LET XC=XC/2:LET YC=YC*5/12:LET IC=
IC*2
235 LET XC0=M1*T(IC,1)/2:LET YC0=M1*T(
IC,2)*5/12
240 LET XC1=XC+XC0:LET YC1=YC+YC0
245 LET XC2=XC-XC0:LET YC2=YC-YC0
250 FOR A=0 TO 360 STEP S
260 MOVE XC,YC
270 LET X=M1*T(A,1)
280 LET Y=M2*T(A,2)
290 LET XT=(X*T(IC,1)-Y*T(IC,2))/2+XC
300 LET YT=(X*T(IC,2)+Y*T(IC,1))*5/12+
YC
310 DRAW XT,YT:DRAW XC2,YC2
320 NEXT A
330 RETURN

```

ST (Stos)/Amiga (Amos) core

```

10 REM AL'S GORITHMS ... ELLIPSES
20 REM (c) LET'S COMPUTE!
30 KEY OFF:MODE 0:HIDE ON:CURS OFF
40 PALETTE $0,$700,$700,$777
50 REM
60 PRINT:PRINT "JUST A MO, CALCULATIN
6....."
70 REM LOOK-UP TABLE
80 DIM T(72,2)
90 FOR I=0 TO 72
100 LET T(I,1)=COS (RAD (I*5))
110 LET T(I,2)=SIN (RAD (I*5))
120 LOCATE 5,5:PRINT I
130 NEXT I
140 INK 3:BAR 0,0 TO 319,199
150 FOR P=0 TO 27
160 READ M1,M2,IC,XC,YC,CL
170 GOSUB 170: REM DRAW ELLIPSE
180 NEXT P
190 GOTO 160
200 REM ELLIPSE ROUTINE
210 INK CL
220 LET XC=XC/4:LET YC=YC/5:LET M1=M1/

```

```

4:LET M2=M2/5:LET OXT#=XC+M1*T(I/5,1):L
ET OYT#=YC+M1*T(I/5,2)
230 LET XC=320-XC:YC=200-YC
240 LET OXT#=320-OXT#:OYT#=200-OYT#
250 FOR A=0 TO 360 STEP 15
260 LET X#=M1*T(A/5,1)
270 LET Y#=M2*T(A/5,2)
280 LET XT#=X#*T(IC/5,1)-Y#*T(IC/5,2
)+XC
290 LET YT#=X#*T(IC/5,2)+Y#*T(IC/5,1
)+YC
300 POLYGON XC,YC TO OXT#,OYT# TO XT#,
YT#
310 LET OXT#=XT#:OYT#=YT#
320 NEXT A
330 RETURN

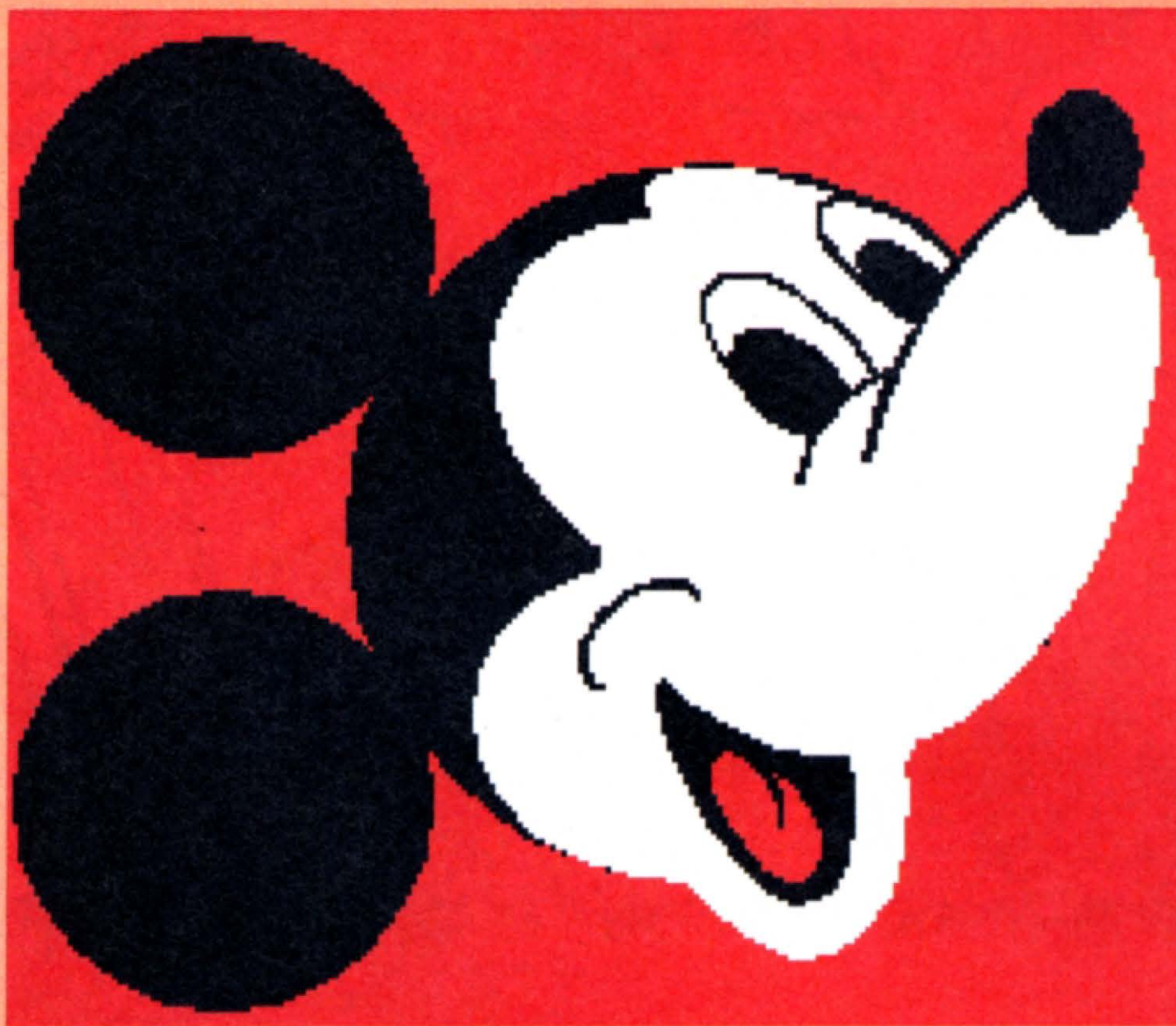
```

Amos users: Replace Line 30 with:

```

30 SCREEN OPEN 0,320,,,200,16,LOWRES:HI
DE:CURS OFF

```

AL'S GORITHMS

THIS IS IT! The page where the budding artists among you can pick up a tip or two. And a chance for you to show what **YOU** can do yourself.

Making a lot with a few lines of code is what it's all about. Alan McLachlan gives you some more bright ideas.

SQUASHING THE CIRCLES

Two issues ago we created some great pictures from circles. But squash the circles – some just a little bit and some a lot – and you've got lots more shapes to draw with.

So this month we're going to make even better pictures from ellipses – a fancy name for flattened circles. We'll show you how you can draw an ellipse of any size and shape. And to add even more fun you can also tilt it about its centre.

So the listing is a little more complex than before. It has to work out how much to squash the ellipse and how much to tilt it. But you don't need to know how it works to draw your own pictures. Just change the 27 in Line 150 to the number of ellipses you want to draw.

Then put the numbers that the program needs to draw the ellipses in DATA lines starting at Line 340. Each ellipse needs six numbers. You'll see them in each of the DATA lines opposite. They have to be entered in this order:

Order	What it does
First	Measures the distance across one way
Second	Measures the distance across the other
Third	Gives the angle of tilt (0 to 180)
Fourth	Gives the X location of the centre
Fifth	Gives the Y location of the centre
Sixth	Gives the colour

When you've keyed in the main core of the program you're ready to draw your own pictures.

But before you do so let your computer draw the Micky Mouse shown above.

To do this, enter these DATA lines:

```

340 DATA 150,150,0,300,300,0
350 DATA 150,150,0,300,700,0
360 DATA 260,260,0,650,500,0
370 DATA 160,100,20,750,340,1
380 DATA 160,120,0,640,360,3
390 DATA 160,120,140,640,580,3
400 DATA 120,40,160,720,700,3
410 DATA 160,120,90,780,500,3
420 DATA 80,40,140,710,620,0
430 DATA 74,34,140,710,620,3
440 DATA 70,30,146,788,688,0
450 DATA 64,24,146,788,688,3
460 DATA 50,30,140,710,600,0
470 DATA 40,20,140,792,672,0
480 DATA 180,100,110,680,360,3
490 DATA 140,60,110,680,360,0
500 DATA 140,100,140,700,450,3
510 DATA 60,30,50,760,560,0
520 DATA 60,30,50,764,554,3
530 DATA 200,90,70,860,560,0
540 DATA 200,90,70,866,554,3
550 DATA 80,60,90,760,460,3
560 DATA 50,40,90,920,760,0
570 DATA 40,25,120,700,280,1
580 DATA 30,20,120,680,300,0
590 DATA 36,18,120,676,300,1
600 DATA 50,30,40,600,420,0
610 DATA 44,28,40,604,416,3

```

When you RUN it you'll find that the result is quite impressive.

But can you come up with something better?

The challenge is there!

Send us your masterpiece drawn using just ellipses and let everyone see it!



Learn how to
look after
yourself
with the
Safe Scientist

GOOD HEALTH?

Do you know what's good for you? Then try this fun survey. It gives a score based on your answers to a few questions. And it could open your eyes.

Try it. But be honest - pick the choice which is nearest to the truth.

The program gives a mark of 2 for the healthiest answer and 0 for the least healthy.

After each question you are also given some advice about what might have been the best answer. So you can learn from the program.

But to be a really good scientist you must carry out as big a survey as possible. So why not check the health of your friends?

Keep a record of the results and plot them on some kind of a chart.

You might even be able to say something about the health of your school. Or even the whole population of the country.

If you know something about programming you can easily extend or alter the program. For example, you can have more than 10 different questions.

Whatever you do with the program, remember that there is more to life than your computer screen. Good old fresh air and exercise really are important!

● **Next month you'll need your sunglasses at the ready.**

We'll be investigating the dazzling world of colour!

```
10 REM HEALTHY LIFE STYLE
20 REM By The Safe Scientist
30 DIM S(3)
40 MODE 3:VDU19,0,4;0;
50 FOR N=23 TO 1 STEP-1
60 LET X=32:LET Y=N:GOSUB 400:PRINT "
HEALTHY LIFESTYLE"
70 LET Y=Y+1:GOSUB 400:PRINT "
":REM 16 SPACES
80 LET T=10:GOSUB 420
90 NEXT N
100 LET SC=0
110 READ QS:IF QS<>"end" THEN GOSUB 16
0:GOTO 110
120 GOSUB300
130 PRINT:PRINT"PRESS SPACE TO TRY AGA
IN."
140 GOSUB 440:IF AS<>" " THEN GOTO 140
150 RUN
160 REM QUESTION
170 PRINT:PRINT QS
180 FORN=1 TO 3:READ AS,S(N)
190 PRINT:PRINT N;" ":AS
200 NEXT N
210 PRINT
220 INPUT "Which choice";C:IF C<1 OR C
```

```
>3 THEN GOTO 220
230 LET SC=SC+S(C)
240 PRINT:PRINT"You scored ";S(C);" fo
r that."
250 READ MS:PRINT:PRINT MS
260 PRINT:PRINT "PRESS SPACE"
270 GOSUB 440:IF AS<>" " THEN GOTO 270
280 CLS
290 RETURN
300 REM SCORE
310 CLS
320 PRINT "You have scored ";SC
330 PRINT:PRINT
340 IF SC=20 THEN PRINT "Perfect! What
a healthy little soul you are!":RETURN
350 IF SC>15 THEN PRINT "Pretty good!
You know what's good for you.":RETURN
360 IF SC>10 THEN PRINT "Not bad, but
some aspects could be healthier.":RETURN
370 IF SC>5 THEN PRINT "Do more to mak
e sure you stay healthy.":RETURN
380 IF SC>1 THEN PRINT "Help! You need
it if you value good health.":RETURN
390 PRINT "Check your pulse! With your
lifestyle you're struggling.":RETURN
400 REM TAB(X,Y)
```

```
410 PRINT TAB(X,Y);:RETURN
420 REM DELAY
430 TIME=0:REPEAT UNTIL TIME>T:RETURN
440 REM GET
450 LET AS=GET$:RETURN
460 DATA "It is 7.15 on a Friday morni
ng. Your alarm has just sounded."
470 DATA "Get up, wash and clean your
teeth.",2
480 DATA "Turn the alarm off and go ba
ck to sleep.",0
490 DATA "Get up but be in a really ba
d temper.",1
500 DATA "Keeping clean is important.
So is learning to rely on yourself."
510 DATA "It is now 7.45 and you are t
hinking about breakfast."
520 DATA "Skip breakfast because you h
aven't time.",0
530 DATA "Tuck into bacon and eggs wit
h fried bread.",1
540 DATA "Have a bowl of cereal and fr
uit juice.",2
550 DATA "It is most important to star
t the day with a well balanced meal."
560 DATA "Twenty to nine and you are n
```




IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown.

Commodore 64/128

Enter the program in capitals.

Change the following lines:

```
40 PRINT CHR$(147);
60 LET X=10:LET Y=N:GOSUB 400:PRINT "
HEALTHY LIFESTYLE"
280 PRINT CHR$(147);
310 PRINT CHR$(147);
410 POKE 211,X:POKE 214,Y:SYS 58732:RE
TURN
430 LET T=TIME+T*3
435 IF T>TIME THEN 435
436 RETURN
450 LET GET AS:IF AS="" THEN 450
455 RETURN
```

Spectrum

Change or add the following lines:

```
40 CLS
50 FOR N=20 TO 1 STEP-1
60 LET X=8:LET Y=N:GOSUB 400:PRINT "H
EALTHY LIFESTYLE"
410 PRINT AT Y,X;:RETURN
425 LET T=PEEK(23672)+T/2:IF T>255 THE
N LET T=T-256
430 IF PEEK(23672)<>T THEN GOTO 430
435 RETURN
450 LET AS=INKEY$:IF AS="" THEN GO TO
450
455 RETURN
```

PC (GW-Basic)

Change the following lines:

```
40 CLS
410 LOCATE Y,X:RETURN
430 LET T=TIMER+T/8:WHILE T>TIMER:WEND
:RETURN
450 LET AS=INKEY$:WHILE AS="":LET AS=I
NKEY$:WEND:RETURN
```

Atari ST (Stos)

Change the following lines:

```
40 MODE 1:KEY OFF:CURS OFF:HIDE
50 FOR N=21 TO 1 STEP-1
410 LOCATE X+1,Y+1:RETURN
430 LET T=TIMER+T*3:WHILE T>TIMER:WEND
:RETURN
450 LET AS=INPUT$(1):RETURN
```

Amiga (Amos)

Change the following lines:

```
40 CLS
50 FOR N=21 TO 1 STEP-1
410 LOCATE X+1,Y+1:RETURN
430 LET T=TIMER+T*3:WHILE T>TIMER:WEND
:RETURN
450 LET AS=INPUT$(1):RETURN
```

Amstrad CPC

Change the following lines:

```
40 MODE 2
410 LOCATE X+1,Y+1:RETURN
430 LET T=TIME+T*3:WHILE T>TIME:WEND:R
ETURN
450 LET AS=INKEY$:WHILE AS="":LET AS=I
NKEY$:WEND:RETURN
```

Health in the National Curriculum
You need know something about diet and a healthy life style.

You are also expected to know that some things, like tobacco, really are bad for you.

ow at school."

570 DATA "Tuck into a Mars bar you bou
ght on the way to school.",1

580 DATA "Nip round the back of the bi
ke shed for a quick fag.",0

590 DATA "Play a game of tennis with y
our mates.",2

600 DATA "It's not a bad idea to get s
ome FRESH air and exercise."

610 DATA "At last! It's morning break
time."

620 DATA "Race to the tuck shop for so
me bags of crisps.",0

630 DATA "Nip into the toilets to brus
h your hair and have a chat.",1

640 DATA "Play a game of tennis with y
our mates.",2

650 DATA "Better to avoid greasy, salt
y tuck. Exercise is much more useful."

660 DATA "It's the dreaded double math
s lesson."

670 DATA "Quietly eat the Polo mints y
ou bought earlier.",1

680 DATA "Just get on quietly with the
problems you've been set.",2

690 DATA "Claim to feel ill so that yo

u can escape to the medical room.",0

700 DATA "You'll feel much better if y
ou stick to your set tasks."

710 DATA "Hurrah! It's lunchtime."

720 DATA "Eat the balanced 'meal of th
e day'.",2

730 DATA "Tuck into hot dog and chips.
",1

740 DATA "Skip dinner so you can keep
the money you were given.",0

750 DATA "Once again, the well balance
d meal is best."

760 DATA "It's games this afternoon."

770 DATA "Take a full and active part
in the game.",2

780 DATA "Forge a note from your mum t
o say you are ill.",1

790 DATA "Skip off and hide in the toi
lets.",0

800 DATA "You may or may not enjoy it
but the exercise is good for you."

810 DATA "School has ended. You are no
w back at home thinking about tea."

820 DATA "Scoff a couple of chocolate
bars and a bag of crisps.",1

830 DATA "Eat a tuna fish salad.",2

840 DATA "Don't bother with tea becaus
e you're watching Neighbours.",0

850 DATA "You need that well balanced
meal again."

860 DATA "After tea. What a range of a
ctivities are possible."

870 DATA "Go to the park to play with
your mates.",2

880 DATA "Flop in front of the telly f
or the evening.",0

890 DATA "Sit in your room doing your
homework.",1

900 DATA "Get the fresh air and exerci
se but make sure homework is done."

910 DATA "Your parents tell you it is
time for bed."

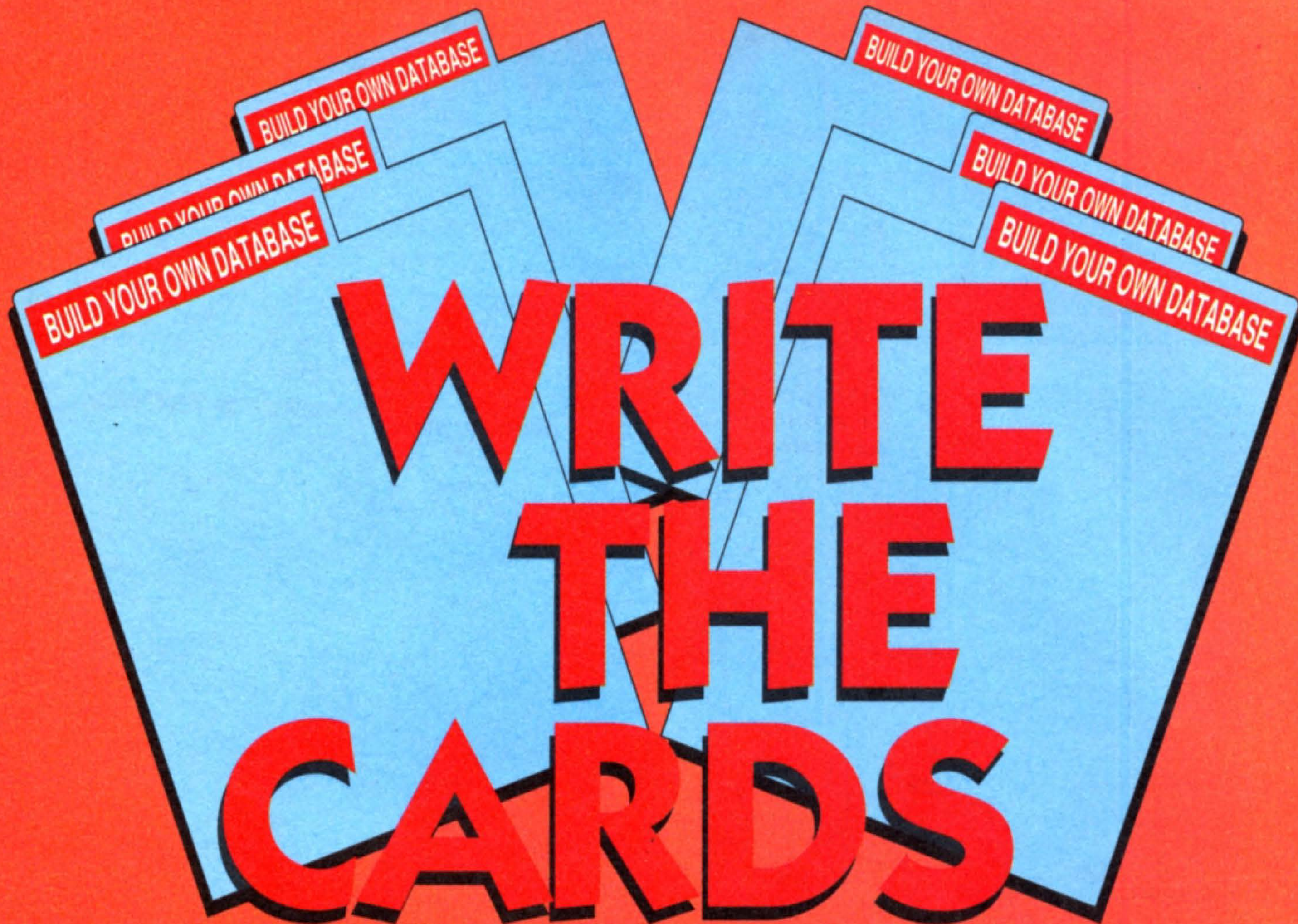
920 DATA "Settle down because you know
your bed time is sensible.",2

930 DATA "Pretend to settle down but g
et up later to see the horror movie.",0

940 DATA "Ask Dad about dinosaurs' int
estines to hang out bed time.",1

950 DATA "You really need your sleep."

960 DATA "end"



Last month we started creating the Let's Compute! Card Index Database. Now Roland Waddilove shows what to do next

What is a database?

A database is a collection of facts and figures. We are surrounded by them.

A bus or train timetable is a database. So is the list of programmes in Radio Times.

We often use a database to look up facts and figures. But it takes time to find out information when its only written on paper.

How long would it take you to go through the phone book and find the number of Mr Smith who lives at 57 High Street, Manchester? You would have to spend ages wading through pages of Smiths looking for the right address.

But ring directory enquiries and you'll be told the correct number in a flash.

How do they do it?

The answer is that everyone's name, address and telephone number is stored on a computer.

The operator types in a name, presses a few more keys and, hey presto, the number is displayed on the screen.

Last month we started building our database by creating the main menu screen. This month we'll add more lines to the program so you'll be able to enter information on to the blank cards.

The section of code you need is in Lines 4000 to 4100. It is a subroutine called Write on Card.

Line 4000 is a REM to tell us what the following lines do. The next line sets a counter, called X, which points to the first card.

Line 4020 searches for the first card that hasn't been written on. It contains an asterisk character, which is what the program is looking for.

There can be up to five lines of text on each card. So a FOR ... NEXT loop (Lines 4050 to 4090) is used to put the lines into the array C\$. This is where the card information is held. If you want to leave a blank line just press Return without typing anything else.

But what if you make a mistake while entering information? Or if you change you mind after typing something in and want to throw away a card? A second subroutine starting at Line 5000 will sort things out.

Again, a REM at the start tells us what the subroutine does. Then Lines 5010 and 5020 search for cards that have information written on them. That means any card that hasn't got an asterisk.

When a card is found it is printed on the screen. You are then asked whether this is the card you want to throw away.

All you need to do is type Y or N. Your answer is read by the INPUT instruction in Line 5080.

Type in these two subroutines and try selecting Choices 4 and 5 from your menu. You can now enter some information and try throwing away cards or enter some new information.

But lets face it: A database isn't much use if you have to enter the information each time you want to use it!

Next month you'll find out how to save your cards on to tape or disc.

CARD INDEX DATABASE

1. Load a box of cards
 2. Save a box of cards
 3. Read/search for a card
 4. Write on a new card
 5. Throw away old cards
 6. Change something on a card
 7. Print the cards
 8. Sort the cards into order
 9. Stop the program
- Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1-9)


```

4000 REM ===== Write on Card =====
4010 LET X=1
4020 IF C$(X)<>"*" AND X<255 THEN X=X+
5 : GOTO 4020
4030 PRINT "Enter 5 lines of text on t
his card..."
4040 PRINT
4050 FOR Y=0 TO 4
4060 PRINT "Line";Y+1;
4070 INPUT T$
4080 LET C$(X+Y)=T$
4090 NEXT Y
4100 RETURN
5000 REM == Throw away old cards ==
5010 LET X=1
5020 IF C$(X)="*" THEN GOTO 5100
5030 CLS
5040 FOR Y=0 TO 4
5050 PRINT C$(X+Y)
5060 NEXT Y
5070 PRINT
5080 INPUT "Throw away this card (Y or
N)";K$
5090 IF K$="y" OR K$="Y" THEN LET C$(X
)="*"
5100 LET X=X+5
5110 IF X<255 THEN GOTO 5020
5120 RETURN

```

TYPING TIP!

When you type things on your cards avoid commas (,) and quotes ("). On some computers these could make the program mess up your data.

IS THIS YOUR COMPUTER?

**Electron/BBC/Archimedes/PC(GW BASIC)
Amstrad CPC/ Atari ST(stos)/Amiga(AMOS)**

The program will work as shown

Commodore 64/128

Change or add the following lines:

```

4050 FOR Y=0 TO 4:TS=""
4075 IF TS="" THEN 4090
5030 PRINT CHR$(147);

```

Spectrum

Change or add the following lines:

```

4020 IF C$(X)<>B$ AND X<255 THEN LET X=X
+5 : GOTO 4020
4075 PRINT " ";T$
5020 IF C$(X)=B$ THEN GOTO 5100

```

NEXT MONTH

You'll find out how to save your cards on to tape or disc

Fine fonts Fun fonts Frightening fonts FORMAL FONTS

See us on Stand
J5
at the
Computer Shopper
Show

We have over 200 high quality fonts for use with Archimedes and 13000 DTP programs. Contact us for a free 24pp font catalogue.

As well as fonts, we supply everything you need for DTP at very competitive prices: computers, monitors, printers, software and more. We also give discounts to educational establishments.

Whatever you need for DTP, contact The Electronic Font Foundry first.

The Electronic Font Foundry, 18 Brockenhurst Rd., Ascot, SL5 9DL.
Telephone or Fax (24hrs) 0344 872923

MASTER COMPACT CARTRIDGE ADAPTOR

Connects MASTER and ELECTRON cartridges (including our ROMPlus-144) to the COMPACT - Price £25. SAE for details

INCREASE ELECTRON PERFORMANCE AND BBC COMPATIBILITY WITH OUR RANGE OF ADD-ONS

E2P SECOND PROCESSOR

Gives the Electron equivalent performance to a BBC with a Second Processor. Adds a complete processor with its own 64k memory bank to run all programs while the Electron handles screen, keyboard, discs, etc.

Gives 30k memory for BASIC programs and 44k memory for HIBASIC programs. (Electron HIBASIC available separately) Gives up to 300% speed increase in programs.

Fits into a Plus One/ROMbox Plus slot.

Operating system available on ROM or 3.5" ADFS disc or Tape (state requirement)
Price £79 + £5 for case.

MODE 7 Mk2 UNIT

Gives exactly the same Mode 7 display screen as the BBC. Allows BBC Mode 7 software to run on the Electron: Magazine programs, Adventure games, Educational, Wordwise, Prestel, Teletext. Includes Prestel software - Prestel requires a modem and our RS423 adaptor. You will need a ROM socket to hold the Mode 7 ROM, eg. in our RS423 adaptor. For Mode 7 programs with HIMEM above &4000 you will need a Shadow RAM board. Fits inside the Electron, alongside the Shadow RAM Board.

Price £49 for you to fit, or £59 fitted.

Also available as a kit - board and ROM for £25 - you buy the components.

RS423 SERIAL CARTRIDGE

Connects the Electron to serial RS423/RS232 devices, eg. printers, modems. Allows the transfer of programs, data, etc. from one computer to another. Has the same socket, so uses the same leads as the BBC. Includes a socket for a 32k RAM or 16k ROM, eg. the Mode 7 ROM. Fits into a Plus One/ROMbox Plus slot.

Price £29.95 + £5 for a case + £20 for a 32k RAM.

ROMPLUS - 144 ROM CARTRIDGE

Sideways ROM cartridge with Printer Buffer and RAM Filling System. Holds 8 ROMs or 7 ROMs and a 32k RAM. Fits into a Plus One/ROMbox Plus slot.

Price £39 + £20 for a 32k RAM

All prices inclusive of post, etc. - Please allow 28 days delivery.

WE ALSO REPAIR ELECTRONS

JAF SYSTEMS

9, Lon-y-Garwa, Caerphilly
Mid-Glamorgan CF8 1NL. Tel: 0222 887203



THE BRAIN GAME

How good is your memory? Here's a game that will put it to the test!

In the popular game Simon you have to remember the sequence of four flashing lights. This unique *Let's Compute!* version uses the alphabet in place of the lights. But the idea is the same.

When you RUN the program the alphabet is shown on your computer screen. Press the spacebar to start. Arrows will point to a letter and then disappear.

All you have to do is remember which letter the arrows pointed to. Then press the same letter on your keyboard. You must use capital letters. If you get it right it will appear again, followed by a second letter.

You now have two keys to press. If you get them right another will be added and you have another go. So the pattern of letters keeps increasing one by one.

If you press the wrong letter your computer will show you the order in which you should have pressed the letters.

Now type the listing in, RUN it and test your memory!

POSSIBLE PROGRAM IMPROVEMENTS

- The program is set up so your computer can remember a pattern of 100 letters. If you think you can do better, change the DIM S(100) in Line 60 to, say, DIM S(500). The biggest number you're allowed to put there depends on the type of computer you're using.
- Put your own routine in the program to time how long it takes to type the letters. Make the program add some points to the score if you're quick.
- Add colour to brighten up the display.

```

10 REM THE BRAIN GAME
20 REM (c) LET'S COMPUTE!
30 LET AS="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
40 LET SS=" ":LET B=0
50 FOR N = 1 TO 5:LET SS=SS+SS:NEXT
60 DIM S(100)
70 LET W=40:LET X=RND(-TIME)
80 CLS
90 LET Y=3:LET MS="THE LET'S COMPUTE!
BRAIN GAME":GOSUB 340
100 LET Y=10:LET MS=AS:GOSUB 340
110 LET M=1:LET N=1
120 GOSUB 730
130 GOSUB 690
140 LET X=26:GOSUB 320:LET S(M)=X
150 LET Y=15:LET MS="WATCH!":GOSUB 340
160 FOR N=1 TO M
170 GOSUB 520
180 LET H=S(N):GOSUB 480
190 LET D=1:GOSUB 440:GOSUB 400
200 NEXT N
210 GOSUB 520
220 LET Y=15:LET MS="NOW IT'S YOUR GO"
:GOSUB 340
230 FOR N=1 TO M
240 LET F=0:GOSUB 460:IF KS<"A" OR KS>
"Z" THEN GOTO 240
250 IF KS<>CHR$(S(N)+64) THEN LET F=1:
LET N=M
260 LET H=ASC(KS)-64:GOSUB 520:GOSUB 4
80:GOSUB 440
270 NEXT N
280 IF F=1 THEN GOSUB 620:GOTO 80
290 GOSUB 560
300 LET M=M+1:GOSUB 520:LET D=2:GOSUB
400:GOTO 140
310 REM ** START OF SUBROUTINES **
320 REM RANDOM
330 LET X=RND(X):RETURN
340 REM PRINT MESSAGE
350 LET X=4:GOSUB 380:PRINT SS
360 LET X=(W-LEN(MS))/2-1
370 GOSUB 380:PRINT MS
380 REM TAB(X,Y)
390 PRINT TAB(X,Y);:RETURN
400 REM DELAY
410 TIME=0
420 IF TIME<D*100 THEN GOTO 420
430 RETURN
440 REM BEEP
450 VDU7:RETURN
460 REM GET A KEY
470 LET KS=GET$:RETURN
480 REM POINTERS
490 LET X=(W-30)/2+H:LET Y=9:GOSUB 380
:PRINT"V"
500 LET Y=11:GOSUB 380:PRINT"^^"
510 RETURN
520 REM CLEAR POINTERS
530 LET Y=9:LET MS=SS:GOSUB 340
540 LET Y=11:GOSUB 340
550 RETURN
560 REM WIN
570 IF (N-1)>B THEN LET B=N-1
580 GOSUB 730
590 SOUND 1,-15,200,30
600 LET Y=15:LET MS="WELL DONE":GOSUB
340:LET D=2:GOSUB 400
610 RETURN
620 REM LOOSE
630 SOUND 1,-15,2,30
640 LET Y=15:LET MS="HARD LUCK":GOSUB
340:LET D=2:GOSUB 400:GOSUB 690
650 LET MS="HERE'S WHAT YOU SHOULD HAV
E DONE":GOSUB 340:LET D=1:GOSUB 400
660 FOR N=1 TO M
670 LET H=S(N):GOSUB 520:GOSUB 480:GOS
UB 440:GOSUB 400:NEXT N
680 RETURN
690 REM PRESS SPACE
700 LET Y=15:LET MS="PRESS SPACE":GOSU
B 340
710 GOSUB 460:IF KS<>" " THEN GOTO 710
720 RETURN
730 REM PRINT SCORE
740 LET Y=6:LET MS="SCORE "+STR$(N-1)+
LEFT$(SS,W/2)+"BEST "+STR$(B):GOSUB 340
750 RETURN

```


IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The programs work as shown

Commodore 64/128

Enter the program in capitals

Change the following lines:

```
25 S=54272:POKE$+24,15:POKE$+6,240
80 PRINT CHR$(147);
330 LET X=INT(RND(0)*X)+1:RETURN
390 POKE 211,X:POKE 214,Y:SYS 58732:RE
TURN
410 TIS="000000"
450 DU=300:FR=32:GOSUB 800:RETURN
470 GET K$:IF K$="" THEN 470
475 RETURN
590 DU=800:FR=64:GOSUB 800
630 DU=800:FR=16:GOSUB 800
800 POKE S+1,FR:POKE S+4,17:FOR T=1 TO
DU:NEXT:POKE S+4,16:RETURN
```

Spectrum

Select 48k mode

```
70 LET W=34: RANDOMIZE
260 LET H=CODE (K$)-64:GO SUB 520:GO S
UB 480:GO SUB 440
330 LET X=INT (RND*X+1): RETURN
350 LET X=4: GO SUB 380: PRINT S$( TO
31)
390 PRINT AT Y,X,: RETURN
410 LET T=PEEK (23672)+D*50: IF T>255
THEN LET T=T-256
420 IF PEEK (23672)<>T THEN GOTO 420
450 BEEP .2,9: RETURN
470 LET K$=INKEY$: IF K$="" THEN GO TO
470
475 RETURN
500 LET Y=11:GOSUB 380:PRINT"T"
530 LET Y=9:LET M$=S$( TO 26): GO SUB
340
590 BEEP 1.5,36
630 BEEP 1.5,-12
740 LET Y=6:LET M$="SCORE "+STR$ (N-1)
+S$( TO W/2)+"BEST "+STR$ (B):GOSUB 340
```

Amstrad CPC

Change the following lines:

```
70 LET W=40:RANDOMIZE TIME
330 LET X=INT(RND*X+1):RETURN
390 LOCATE X+1,Y+1:RETURN
410 LET T=TIME+D*300
420 WHILE T>TIME:WEND
450 SOUND 1,142,20,15:RETURN
470 LET K$=INKEY$:WHILE K$="":LET K$=I
NKEY$:WEND:RETURN
500 LET Y=11:GOSUB 380:PRINT"T"
590 SOUND 1,28,150,15
630 SOUND 1,500,150,15
```

PC (GW-Basic)

Change the following lines:

```
70 LET W=40:RANDOMIZE TIMER
330 LET X=INT(RND*X+1):RETURN
390 LOCATE Y+1,X+1:RETURN
410 LET T=TIMER+D/2
420 WHILE T>TIMER:WEND
450 BEEP:RETURN
470 LET K$=INKEY$:WHILE K$="":LET K$=I
NKEY$:WEND:RETURN
590 SOUND 253,18
630 SOUND 150,8
```

Amiga and Atari ST

Use Amos or Stos

```
70 LET W=40:LET X=RND(TIMER)
80 MODE 1:KEY OFF:CURS OFF:HIDE:CL
ICK OFF
330 LET X=RND(X-1)+1:RETURN
390 LOCATE X,Y:RETURN
410 LET T=TIMER+D*300
420 WHILE T>TIMER:WEND
450 BELL:RETURN
470 LET K$=INPUT$(1):RETURN
590 VOLUME 10:PLAY 2,55,30:VOLUME 0
630 B00M
```

Amos: Leave Line 80 as:

80 CLS

COMPUTER EXPRESS



A500 BATMAN PACK



The Commodore A500 Batman Pack must surely rank as one of the most popular computer packs ever! The pack features the Commodore Amiga 500 computer with mouse controller and TV modulator, plus four top software titles. The software includes: 'Batman The Movie' - rid Gotham City of the cunning Joker, in Ocean's top selling title based on the blockbuster Batman film; 'New Zealand Story' - high quality conversion of the leading arcade game; 'Interceptor' - dogfight with two F-16s in this leading flight simulator; 'Deluxe Paint II' - top quality Amiga graphics package which set the standard for others to follow. Return the coupon for further details.

PACK INCLUDES:

A500 Computer & Mouse	£399.99
A520 TV Modulator	£24.99
Batman The Movie	£24.95
New Zealand Story	£24.95
Interceptor	£24.95
Deluxe Paint II	£49.95

TOTAL RRP: £549.78
Less Pack Saving: £150.78

PACK PRICE: £399.00

£399 INC VAT

AMIGA SOFTWARE

ACCESSORIES

Full range of Amiga accessories available.

CLASS OF THE 90s EDUCATION BUNDLE

Amiga 500, Midi Interface, D Paint II, Superbase, Publisher's Choice, Dr T's Midi Recording Studio, BBC Emulator and BBC Programs, Maxiplan 500, plus lots of accessories.

AMIGA HARDWARE

A1084 colour monitor - stereo sound	259.00
3.5" disk drive external	86.00
A500 memory expansion with clock	80.00
A590 external hard disk	375.99
Philips colour monitor 8833	264.99
Minigen	109.00

PRINTERS

Star LC10 - No1 low cost printer
Star LC10 Colour
Epson LX400
Panasonic 1180



WHEN YOU HAVE BEEN INSPIRED ABOUT BY THE BEST COMING AND HAVE A FRIENDLY COME WITH THE BEST COMPUTER EXPRESS.

24 HOUR SALES LINE

0582 475577

SHOWROOM OPENING HOURS: Monday-Friday 9am-5.30pm Saturday 10am-4pm
HEAD OFFICE: COMPUTER EXPRESS • 15 APEX BUSINESS CENTRE • BOSCOMBE ROAD • DUNSTABLE • BEDS • LU5 4BB • 0582 475577 • FAX 0582 688311

All prices and specifications subject to availability and change without notice. All items guaranteed. All prices correct at time of going to press.

STOP PRESS CHRISTMAS PACK ... 4 arcade games deluxe paint & modulator and mouse ONLY 399.00

LET'S COMPUTE!

99p

No 6 January 1991
A Database Publication

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

Make your own
foot-operated
JOYSTICK
WIN!

A Michaelangelo
souvenir – and
£100s worth
of other
great
prizes



Top Christmas games reviews





Think you're quick on the trigger? This fast-action game will test you to the limit!

NAKES ALIVE!

Here's an action packed game that's full of fun. Type it in and play it just as it is.

But, to add to the fun, we list below loads of suggestions for changes. Some are simple and can be done by anyone. Others will prove a challenge even to more expert programmers.

Once you've typed it in and played it a few times, see how you can improve it. Experimenting and changing a working program is a great way to learn about computing.

In the game a long snake comes out of a hole in your computer screen. But how much of it can come to the surface before it collides with something?

Move it round the screen with care. Don't

let it touch the border or any part of its body that's already come out of the hole. To make it even more difficult, scattered around the screen are obstacles. These must be avoided at all costs.

The idea is to keep going for as long as you can. You'll need to be quick to miss all the things on the screen. And you have to use all your skill to work out the best way to move around.

Your score, and the highest previous one, are shown at the top of your screen. Can you do better?

Type it in, SAVE it and RUN it. Let us know your high score. We'll put the best on the High Score Challenge table.

ADAPTING YOUR GAME

There are lots of ways you can change this program to make your own game. Here are just a few ideas:

- **Change the shapes.**

The letters used to draw the snake, the border and the obstacles are set in Lines 50, 60 and 70. Alter these to any letters you want. If your computer has keys that produce graphic characters you can use these instead.

Do you know how to define characters and use colour on your computer? Then you can use bricks for the border, bombs for the obstacles and a slimy-looking green shape for the snake.

- **A different game each time**

Use your computer's random feature so that some or all of the above are changed each time you play.

- **Use different playing keys**

Lines 190 to 220 sets the keys you use to play the game. Change these as you wish. You could even change these lines so that you can use a joystick or the joyboard described on Pages 10 and 11.

- **Set the speed**

Line 80 contains a number that sets the speed of the game. Make it larger if the game is too fast for you. Reduce it for a bigger challenge.

- **Modify the display**

Line 30 contains numbers that define the size of the playing areas. Reduce the numbers and you'll get a smaller area.

This makes it more difficult to get a high score!

- **Add noise**

Put some sound instructions in the program.

- **Alter the number of obstacles**

Line 40 lets you have about 40 obstacles. You can have more or less. Note that we say about 40 because an obstacle may be drawn on top of one that is already there. Each time this happens one less will be displayed.

- **A better crash!**

Lines 680 to the end of the program are the crash routine. Put anything you want there: flashing screens, explosions and so on. Remember to put the word RETURN on the last line.

- **High Scores**

Give the game a high score table.

- **A maze game**

Make the obstacles form a maze so you can guide the snake to freedom.

THE KEYS TO USE:

Up K
Down M
Left Z
Right ... X
Pause ... Any other key
NOTE: USE CAPITAL LETTERS

IS THIS YOUR COMPUTER?

ATARI (Stos)/Amiga (Amos)

Change the following lines:

```
30 LET W=40:LET D=20:REM SCREEN SIZE
80 LET T=200:REM SET SPEED
110 MODE 0:KEY OFF:HIDE ON:CURS OFF
310 LET KS=INKEYS
360 PRINT LOCATE X-1,Y-1:AS;
390 LET X=RND(W-2)+1:LET Y=RND(D-8)+6
```

Commodore 64/128

Change the following lines:

```
30 LET W=38:LET D=25:REM SCREEN SIZE
110 PRINT CHR$(147);
310 GET KS
360 POKE 211,X-1:POKE 214,Y-1:SYS
58732:PRINT AS;
390 LET X=INT(RND(0)*(W-2))+2:LET
Y=INT(RND(0)*(D-8))+7
```

Spectrum

Change the following lines:

```
30 LET W=32:LET D=22:REM SCREEN SIZE
110 CLS
310 LET KS=INKEYS
360 PRINT AT Y-1,X-1:AS;
390 LET X=INT (RND*(W-2))+2:LET Y=INT
(RND*(D-8))+7
```

PC (GW Basic)

Change the following lines:

```
30 LET W=40:LET D=24:REM SCREEN SIZE
110 SCREEN 1:CLS
310 LET KS=INKEYS
360 LOCATE Y,X:PRINT AS;
390 LET X=INT(RND*(W-2))+2:LET
Y=INT(RND*(D-8))+7
```

Amstrad CPC

Change the following lines:

```
110 MODE 1
310 LET KS=INKEYS
360 LOCATE X,Y:PRINT AS;
390 LET X=INT(RND*(W-2))+2:LET
Y=INT(RND*(D-8))+7
```

```
10 REM THE SNAKE GAME
20 REM (C) LET'S COMPUTE!
30 LET W=40:LET D=32:REM SCREEN SIZE
40 LET N=50:REM NUMBER OF OBSTACLES
50 LET WS="O":REM SHAPE OF OBSTACLES
60 LET YS="*":REM SHAPE OF SNAKE
70 LET BS="#":REM SHAPE OF BORDER
80 LET T=20:REM SET SPEED
90 LET B=0:DIM A(W,D)
100 REM START OF GAME
110 MODE 4
120 LET KS="F":LET S=0
130 GOSUB 470:REM WRITE WORDS AT TOP
140 GOSUB 530:REM DRAW BORDER
150 GOSUB 420:REM DRAW OBSTACLES
160 GOSUB 390:REM SNAKE'S START POS.
170 GOSUB 300:REM CHECK KEYPRESS
180 LET C=0
190 IF KS="Z" THEN LET X=X-1:LET C=1
200 IF KS="X" THEN LET X=X+1:LET C=1
210 IF KS="K" THEN LET Y=Y-1:LET C=1
220 IF KS="M" THEN LET Y=Y+1:LET C=1
230 IF C=1 AND A(X,Y)=1 THEN GOSUB 680
:GOTO 110
240 IF C=1 THEN LET A(X,Y)=1:LET S=S+1
:LET AS=STR$(S)
250 IF C=1 THEN LET X1=X:LET Y1=Y:LET
X=11:LET Y=3:GOSUB 360:LET X=X1:LET Y=Y1
260 IF S>B THEN LET B=S
270 LET AS=YS:GOSUB 360
280 GOTO 170
290 REM GET KEY
300 FOR I=1 TO T:LET RS=KS
310 LET KS=INKEY$(0)
320 IF KS="" THEN LET KS=RS
330 NEXT I
340 RETURN
350 REM TAB
360 PRINT TAB(X-1,Y-1):AS;
370 RETURN
380 REM PRINT AT RANDOM POSITION
390 LET X=RND(W-2)+1:LET Y=RND(D-8)+6
400 RETURN
410 REM OBSTACLES
420 LET AS=WS
430 FOR I=1 TO N
440 GOSUB 390:GOSUB 360:LET A(X,Y)=1
450 NEXT I
460 RETURN
470 REM DRAW TOP OF SCREEN
480 LET X=(W-30)/2+1:LET Y=1:LET AS="T
HE LET'S COMPUTE! SNAKE GAME":GOSUB 360
490 LET X=3:LET Y=3:LET AS="SCORE = ":
GOSUB 360
500 LET X=W-11:LET AS="BEST = "+STR$(B
):GOSUB 360
510 RETURN
520 REM DRAW BORDER
530 LET X=W/2-6:LET Y=6:LET AS="PLEASE
WAIT":GOSUB 360
540 FOR X=2 TO W-1:FOR Y=7 TO D-2
550 LET A(X,Y)=0
560 NEXT Y:NEXT X
570 LET AS=BS
580 FOR X=1 TO W
590 LET Y=6:GOSUB 360:LET A(X,6)=1
600 LET Y=D-1:GOSUB 360:LET A(X,D-1)=1
610 NEXT X
620 FOR Y=7 TO D-2
630 LET X=1:GOSUB 360:LET A(1,Y)=1
640 LET X=W:GOSUB 360:LET A(W,Y)=1
650 NEXT Y
660 RETURN
670 REM CRASH
680 LET X=W/2-4:LET Y=D/2+1
690 LET AS="CRASH":GOSUB 360:GOSUB 300
700 LET AS="":GOSUB 360:GOSUB 300
710 PRINT:PRINT:PRINT "PRESS SPACE!!!"
720 IF KS<>" " THEN GOTO 690
730 RETURN
```





Dear Mr Newsagent

Please deliver *Let's Compute!* to the address below until further notice.

Name _____

Address _____

Note to newsagent:
If you have difficulty
in obtaining
Let's Compute!
please contact the
distributors to the news
trade, COMAG, on 0895
444 055.

"OUTLET"

GRAPHICS INFO AND IDEAS!!
SAM COUPE AND SPECTRUM UTILITIES USEFUL
&
INTERESTING HOMEGROWN SOFTWARE!!

Monthly since 1987 – now on ALL of the popular SPECTRUM systems

SPECIAL OFFER – First time buyers send only £2.50 for the latest issue on...
PLUS 3 DISC, DISCIPLE/PLUS D, OPUS, MICRODRIVE, TAPE AND SAM DISK

CHEZRON SOFTWARE

605 Loughborough Road, Birstall, Leicester LE4 4NJ

The convenient SINGLE SOURCE for EDUCATIONAL SOFTWARE

- * All popular educational programs at publishers prices or below including Mirrorsoft, 4Mation, Sherston, E.S.M., Bourne, Chalksoft, Kosmos, Database Fun School, etc.
- * Wide range of formats: Not only BBC/A3000/ Electron but also Spectrum +2/+3, Amstrad, Nimbus, Commodore 64/Amiga, Atari ST, IBM-PC and others!
- * Vast stocks: 30,000 programs available for immediate delivery.
- * Unconditional guarantee: problems rectified by expert staff – telephone helpline also available.

THE EDUCATIONAL SOFTWARE DIRECTORY

describes and prices hundreds of programs

Write or phone for a free copy

RICKITT EDUCATIONAL MEDIA

FREEPOST • Ilton • Ilminster • Somerset TA19 9HS •
Telephone 0460 57152 • Fax 0460 53176

Please state the age of your children and make of your computer

```
1000 REM ===== Load cards =====
1010 PRINT "Load a box of cards"
1020 PRINT "-----"
1030 PRINT
1040 INPUT "What is its name";FS
1050 IF FS="" THEN RETURN
1060 ZZ=OPENIN FS
1070 FOR X=1 TO 255
1080 INPUT#ZZ,CS(X)
1090 NEXT X
1100 CLOSE #ZZ
1110 RETURN
2000 REM ===== Save cards =====
2010 PRINT "Save a box of cards"
2020 PRINT "-----"
2030 PRINT
2040 INPUT "What is its name";FS
2050 IF FS="" THEN RETURN
2060 ZZ=OPENOUT FS
2070 FOR X=1 TO 255
2080 PRINT#ZZ,CS(X)
2090 NEXT X
2100 CLOSE #ZZ
2110 RETURN
```

**This
program
works on
all home
micros**

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program will work as shown

Atari ST(stos) AMIGA (Amos)

Change the following lines:

```
1060 OPEN IN #1,FS
1080 INPUT#1,CS(X)
1100 CLOSE #1
2060 OPEN OUT #1,FS
2080 PRINT#1,CS(X)
2100 CLOSE #1
```

Commodore 64/128

Change the following lines:

```
1060 OPEN 1,1,0,FS
1080 INPUT#1,CS(X)
1100 CLOSE 1
2060 OPEN 1,1,1,FS
2080 PRINT#1,CS(X)
2100 CLOSE 1
```

Spectrum

Leave out Lines 1070 to 1100
and Lines 2070 to 2100

Change the following lines:

```
1060 LOAD FS DATA CS()
2060 SAVE FS DATA CS()
```

PC(GW-Basic)

Change the following lines:

```
1060 OPEN FS FOR INPUT AS #1
1080 INPUT#1,CS(X)
1100 CLOSE #1
2060 OPEN FS FOR OUTPUT AS #1
2080 PRINT#1,CS(X)
2100 CLOSE #1
```

Amstrad CPC

Change the following lines:

```
1060 OPENIN FS
1080 INPUT #9,CS(X)
1100 CLOSEIN
2060 OPENOUT FS
2080 WRITE #1,CS(X)
2100 CLOSEOUT
```


CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1-9)

STORE THE CARDS

Part 4: Roland Waddilove adds load and save routines to the Let's Compute! card index database

In the last two issues of *Let's Compute!* we've built a simple database that lets us type in facts like phone numbers. The first article showed how the menu was made and last month you were given the part of the program that lets you write on the blank cards.

The problem now is that every time you want to use the database you have to type in the data all over again! It disappears every time you switch your computer off.

So this month's code makes your computer remember the facts and figures on the cards.

You will be able to save them to tape or disc.

You need to add these two routines to the parts we've done already. So start by **LOADing** the program you ended up with last month.

Next you should type in the new lines. Then **SAVE** your program and try **RUNning** it.

You can now try choices 1, 2, 4 and 5 on the menu – see above. What you've got now is a useful way of storing facts. But there's much more to a database than that.

We'll add more next month.

How the Load and Save routines work

You need two separate routines. One to save the cards and one to load them back any time you want to use them.

They are in Lines 1000 to 2110. Both are designed in a similar way. The only difference is that Load Cards reads the information in from tape or disc, and Save Cards writes it out.

Remember, the data is stored in the computer's memory in 255 strings called C\$(1) to C\$(255). On most computers, before you save these strings you need to open a file for output. Take a look at Line 2060.

We then use PRINT# to print the strings to tape or disc – Line 2080. This special form of the PRINT command is different to the one you usually use. Nothing shows on the

computer's display. It sends a message to the tape or disc instead.

When all the strings have been printed using a simple FOR ... NEXT loop the file has to be closed – line 2100.

In the Load Cards routine the data file is opened for input in Line 1060. A FOR ... NEXT loop loads the 255 strings making up the cards using INPUT# in Line 1080.

As you may know, INPUT makes the computer stop and wait for you to type in something. But INPUT# is different. It reads the cassette tape or disc instead of the keyboard – you won't see anything on the screen.

The Spectrum works differently. You just use special forms of LOAD and SAVE. Look at Lines 1060 and 2060.

NEXT MONTH

You'll add a routine to search the cards. Then you'll be able to find out what you want without looking at every card in your database.

Happy New

**Make your own
computer calendar
for 1991 – or any
year you want**

January 1 is the time to hang up your new calendar. So here's a program to let you print out your own.

It works for any year in this century – or the next. If you haven't got a printer, you can still use the electronic calendar and view the dates on your screen.

If you don't like the picture of Mabel – the Let's Compute! cat – you can design your own. The panel below tells you what to do.

Type in the program. **SAVE** it and **RUN** it. The first question you'll be asked is whether you want your calendar printed or just displayed on your screen. Answer **P** for print or **S** for screen.

Next, you're asked for the year. A good reply is 1991.

If you answered **P** to the first question your calendar will then be printed. If you chose a screen version, the cat will be displayed.

Press any key to see the first part of the calendar. There will be a slight pause while your computer works a few things out. Then the first few months will be displayed.

To see more, press a key again. (There will be no messages telling you to press a key. Just take our word for it!)

That's all there is to printing or viewing your own calendar.

If you've got a printer why not print out lots and give them to your friends?

How to change the program

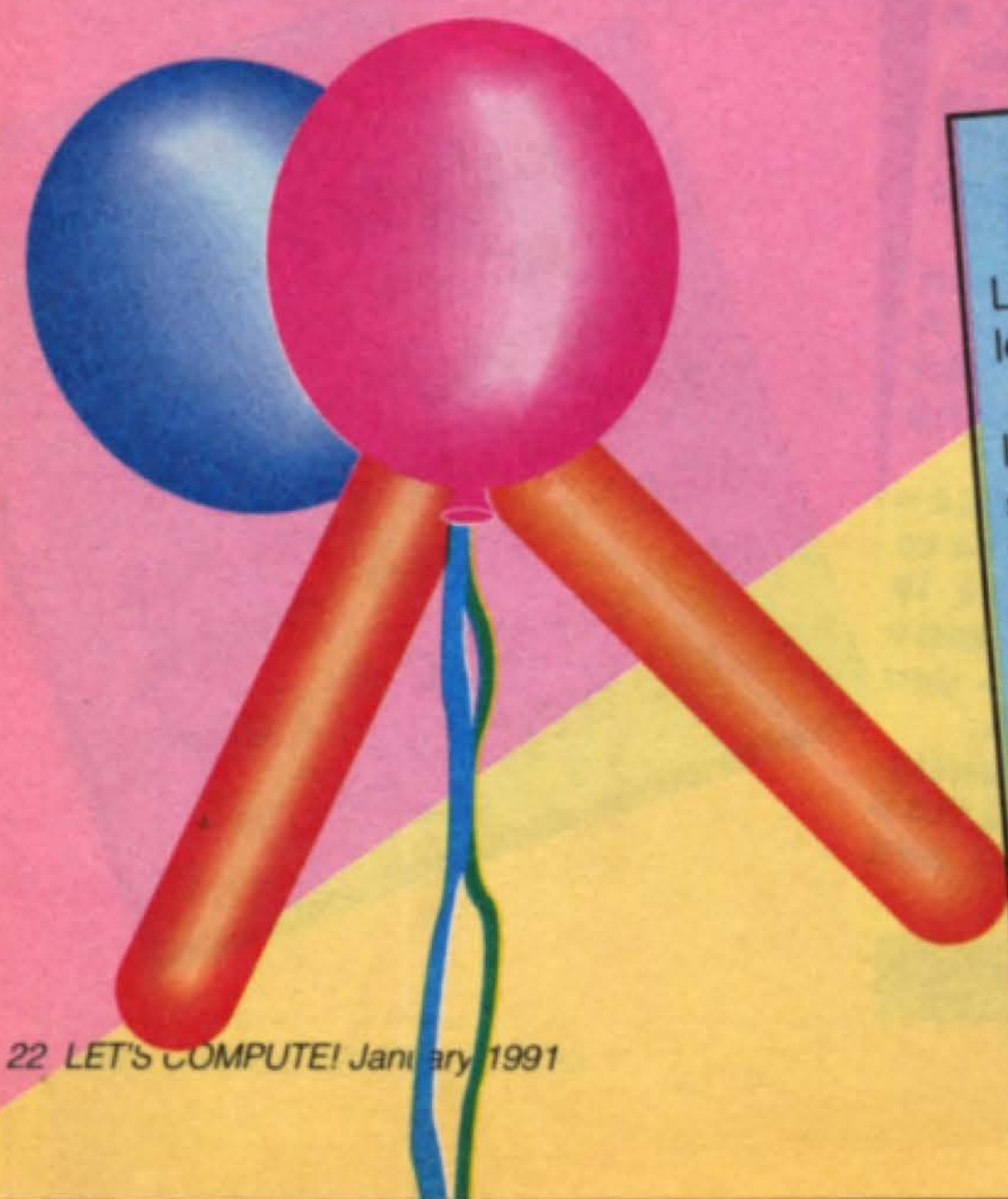
Line 40 sets the width of the printout. At the moment it is set to the number of letters that will fit across the screen of your computer.

If you want to print the calendar – and you have a printer that lets you have more letters on each line – you can change it. When you **RUN** the program, what you see on your screen will be messed up, but the printout will be just what you wanted.

The picture of Mabel is designed to fit on all common printers. It's 32 letters wide so it fits on the smallest computer screen or paper. If it's printed 80 letters wide the whole calendar will be 66 lines long – which is the most common paper size.

Would you like to put your own picture on the calendar? Design it using letters on your keyboard and put them in Lines 770 to 1000. If you use more than 32 letters in any line, change the number 32 in Line 150 to the most you use.

You can make the picture as tall as you want. Just put more **DATA** lines near the end of the program. Make sure that you keep **DATA "end"** as the last line.



1/1
1/1
1/1

Year!

#####

MARCH

SU MO

3 4

11 1

18 1

25 26

TU W

4 5

11 12

8 19

5 26

VE TH

4 5

12 1

19 2

26 27

FR SA

6 7

13 14

20 21

27 28

```

10 REM CALENDAR
20 REM (C) LET'S COMPUTE!
30 MODE 0
40 LET W=80
50 LET SS=" ":FOR I=1 TO 3:LET SS=SS+SS:NEXT I
1:REM THERE ARE THREE SPACES BETWEEN THE QUOTES
60 LET MW=INT (W/24):LET MARGIN=INT ((W-24*MW)/2)
70 DIM MS(12):DIM T(12):DIM WS(6,3)
80 FOR I=1 TO 12
90 READ MS(I),T(I)
100 NEXT I
110 LET DS=" SU MO TU WE TH FR SA "
120 INPUT "Printer or Screen (P/S)"; PS
130 INPUT "What year "; YEAR
140 CLS
150 LET BS="":IF W>33 THEN LET BS=LEFT$(SS,INT ((W-32)/2))
160 LET AS=BS+"The Let's Compute! Calendar "+ST RS(YEAR)
170 GOSUB 570
180 LET AS=" ":GOSUB 570
190 READ AS:IF AS<>"end" THEN LET AS=BS+AS:GOSUB 570:GOTO 190
200 GOSUB 600
210 LET T(2)=28
220 LET Y4=INT (YEAR/4):LET SD=YEAR+Y4-7*INT ((YEAR+Y4)/7)
230 IF YEAR=4*Y4 THEN LET SD=SD-1:LET T(2)=29
240 LET X=(SD-1)*3+1
250 IF SD<=0 THEN LET X=X+21
260 LET MNTH=0
270 REM
280 FOR J=1 TO MW
290 LET R=1
300 LET XS=LEFT$(SS,X)
310 FOR I=1 TO T(MNTH+J)
320 LET T=INT (I/10)
330 LET U=I-10*T+48:LET T=T+48
340 IF T=48 THEN LET T=32
350 LET XS=XS+" "+CHR$(T)+CHR$(U)
360 LET SD=(SD+1)
370 LET X=(X+3)
380 IF X>21 THEN LET X=1:LET WS(R,J)=LEFT$(XS+SS,24):LET R=R+1:LET XS=" "
390 NEXT I
400 LET WS(R,J)=LEFT$(XS+SS,24)
410 LET R=R+1:IF R<7 THEN LET WS(R,J)=SS:GOTO 410
420 NEXT J
430 LET AS=" ":GOSUB 570:LET AS=LEFT$(SS,MARGIN)
440 FOR J=1 TO MW:LET AS=AS+LEFT$(" "+MS(MNTH+J))+SS,24:NEXT J
450 GOSUB 570:LET AS=" ":GOSUB 570:LET AS=LEFT$(SS,MARGIN)
460 FOR J=1 TO MW:LET AS=AS+DS:NEXT J

```

```

470 FOR R=1 TO 6
480 GOSUB 570:LET AS=LEFT$(SS,MARGIN)
490 FOR J=1 TO MW
500 LET AS=AS+WS(R,J)
510 NEXT J
520 NEXT R
530 GOSUB 570
540 GOSUB 600
550 LET MNTH=MNTH+MW:IF MNTH<12 THEN GOTO 270
560 END
570 REM PRINT AS
580 IF PS="P" OR PS="p" THEN VDU2
590 PRINT AS:VDU3:RETURN
600 REM WAIT FOR SPACE
610 IF PS="P" OR PS="p" THEN RETURN
620 AA=GET
630 RETURN
640 REM MONTH DATA
650 DATA "JANUARY",31
660 DATA "FEBRUARY",28
670 DATA "MARCH",31
680 DATA "APRIL",30
690 DATA "MAY",31
700 DATA "JUNE",30
710 DATA "JULY",31
720 DATA "AUGUST",31
730 DATA "SEPTEMBER",30
740 DATA "OCTOBER",31
750 DATA "NOVEMBER",30
760 DATA "DECEMBER",31
770 DATA " /*\/*\"
780 DATA " *****"
790 DATA "\_**0**0**/_ :/: : /-/: -: : : "
800 DATA " ***<*** : : : : : : : "
810 DATA " * **** * : : : : : : : "
820 DATA " /* ** **\"
830 DATA " *****"
840 DATA " **"
850 DATA " ****"
860 DATA " *0000*"
870 DATA " *0000*"
880 DATA " *000000*"
890 DATA " **0000** **"
900 DATA " ***00*** ****"
910 DATA " ***** 000"
920 DATA " **==00==** ****"
930 DATA " **==00==** 000"
940 DATA " **==00==** ****"
950 DATA " **==00==** 000"
960 DATA " **==00==** 000"
970 DATA "****=00==**0*0*"
980 DATA "****=00==**0*0*"
990 DATA " **nnnnn**"
1000 DATA "##### "
1010 DATA "end"

```

IS THIS YOUR COMPUTER?

Spectrum

Change the following lines:

```

30 CLS
40 LET W=32
70 DIM MS(12,22):DIM T(12):DIM WS(6,3,24)
150 LET BS="":IF W>33 THEN LET BS=SS( TO I NT((W-32)/2))
190 READ AS:IF AS( TO 3)<>"end" THEN LET A $=BS+AS:GOSUB 570:GOTO 190
300 LET XS=SS( TO X)
380 IF X>21 THEN LET X=1: LET WS(R,J)=XS: LET R=R+1: LET XS=" "
400 LET WS(R,J)=XS
430 LET AS=" ":GOSUB 570:LET AS=SS( TO MAR GIN)
440 FOR J=1 TO MW:LET AS=AS+" "+MS(MNTH+J ):NEXT J
450 GOSUB 570:LET AS=" ":GOSUB 570:LET AS= SS( TO MARGIN)
480 GOSUB 570:LET AS=SS( TO MARGIN)
560 STOP
580 IF PS="P" OR PS="p" THEN LPRINT AS: R ETURN
590 PRINT AS:RETURN
620 PAUSE 0

```

Amstrad CPC

Change the following lines:

```

30 MODE 2
580 IF PS="P" OR PS="p" THEN PRINT#8,AS:RE TURN
590 PRINT AS:RETURN
620 WHILE INKEYS="":WEND

```

Commodore 64/128

Change or add the following lines:

```

30 PRINT CHR$(147);
40 LET W=40:REM CHANGE TO 80 FOR PRINTER
125 IF PS="P" OR PS="p" THEN OPEN 4,4
140 PRINT CHR$(147);
555 IF PS="P" OR PS="p" THEN PRINT#4:CLOSE#4
580 IF PS="P" OR PS="p" THEN PRINT#4,AS
590 PRINT AS:RETURN
620 GET AA

```

PC(GW BASIC)

Leave out Lines 930 and 940.

Change the following lines:

```

30 SCREEN 1:CLS
580 IF PS="P" OR PS="p" THEN LPRINT AS
590 PRINT AS:RETURN
620 WHILE INKEYS="":WEND

```

Atari ST (Stos)/Amiga (Amos)

Change the following lines:

```

30 MODE 1:KEY OFF:CURS OFF:HIDE
580 IF PS="P" OR PS="p" THEN LPRINT AS
590 PRINT AS:RETURN
620 WHILE INKEYS="":WEND

```

For Amiga use:

```

30 MODE 1:CURS OFF:HIDE

```


Pendown is already a favourite writing tool in schools across the country. Now it has been completely redesigned to take full advantage of all the facilities that make the Archimedes (A3000) so popular. It features a powerful dictionary, thesaurus and spell checker, along with a friendly user-interface that can be reconfigured to make as much sense to a six-year-old as it does to an adult. Special fonts, simplified layouts and on-screen help make it perfect for the very young, whilst powerful word-processing features mean it is equally at home with students working on 'A' Level projects.

Longman Logotron has a wealth of experience in producing educational software, and also produces the National Curriculum standard Logotron Logo, Numerator – a revolutionary maths package, and Hyperbook software and texts, all of which are available for the Acorn Archimedes. For a brochure and full details of all Longman Logotron's educational software, write – stating whether your interest is primary or secondary education – to the address below.

Archimedes PenDown



Longman Logotron
Dales Brewery
Gwydir Street
Cambridge CB1 2LJ
Tel: 0223 323656
Fax: 0223 460208



LONGMAN
LOGOTRON

make
Words
make
Sense

```
10 REM Distance calculator
20 REM by Stephen Wade
30 REM (c) Let's Compute!
40 CLS
50 PRINT "CYCLIST'S CALCULATOR"
60 PRINT "=====
70 PRINT:PRINT
80 INPUT "Diameter of wheel (inches) ";W
90 INPUT "Teeth on cog behind pedals ";F
100 INPUT "Teeth on back wheel cog ";B
110 INPUT "Number of pedal turns ";P
120 INPUT "How long (minutes) ";T
130 LET T=T/60
140 IF T=0 THEN LET T=0.0001
150 PRINT:PRINT
160 REM Circumference of wheel
170 LET C=W*3.14159
180 REM Gear ratio
190 LET R=F/B
200 REM Distance (inches)
210 LET Z=C*R
220 LET D=P*Z
230 REM Distance and speed
(km)
240 LET K=D/39600:LET KS=K/T
250 REM Distance and speed (miles)
260 LET M=D/63360:LET MS=M/T
270 PRINT "You travelled"
280 PRINT K;" km (";M;" miles)"
290 IF T=0.0001 THEN GOTO 330
300 PRINT
310 PRINT "Your average speed was"
320 PRINT KS;" km/h (";MS;" mph)"
330 PRINT
340 INPUT "Another? (Y/N)";K$
350 IF K$="Y" THEN RUN
360 IF K$<>"N" THEN GOTO 340
370 PRINT:PRINT
```

This
program
works on
all home
micros

IS THIS YOUR COMPUTER?

**BBC/Electron/Archimedes/
PC (GW-Basic)/Amstrad CPC/
Spectrum**

The listing works as shown.

Commodore 64/128

*Enter the program in capitals.
Change the following line:*

40 PRINT CHR\$(147);

ST (Stos)/Amiga (Amos)

*Put a # after all the variable names except K\$.
For example, change Line 170 to:*

170 LET RH=F#/#



On Yer Bike!

Ever wondered
how far you've travelled on a bike
journey? Or how fast were you going?
Stephen Wade has the answers.

Cars have speedometers and mileometers. But not many bikes do. So how do you work out how far you've travelled and at what speed?

Use your computer! All you need is the *Let's Compute! Cyclist's Calculator*. It works for any bike that doesn't have Sturmey Archer gears.

Type in the program and SAVE it. When you RUN it you'll be asked for some numbers. Have the following answers ready and type them in when you're asked:

- *The diameter of your wheel.* This is the distance from one edge to the opposite one. Measure it in a straight line across the centre.
- *The number of teeth on the cog behind the pedals.* If you've several cogs there, note each so you can use the program for any gear. If you're lucky the numbers may be in your cycle owner's manual. It's worth checking there first.
- *The number of teeth on the back wheel cog.* Count these, like you did for the front.

You will also be asked two questions about the journey you are checking.

- *How many pedal movements did you make?* Once up and and down with one leg counts as one movement.
- *The time taken.* To work out your speed, your computer needs to know how many minutes you took.

Sometimes you may not be interested in the speed. You might just want to use the program to work out the distance between two places. In that case, enter zero when your computer asks for the time taken.

Once your computer knows the facts it will quickly tell you how far you travelled in both kilometres and miles. If it also knows the time taken it will also tell you your speed.

Experiment with the program. Try out different gears and see what happens to your speed.

But be careful. If you try any speed trials on your bike make sure you do it in safety.

Fine fonts Fun fonts Frightening fonts Formal fonts

"They all look good on screen,
and are properly hinted."

- New Computer Express,
Issue 107.

We have over 200 high quality fonts available for use
with Archimedes and A3000 DTP programs. Contact us
for a free font catalogue.

As well as fonts, we supply everything you need for DTP
at very competitive prices, computers, monitors, printers,
software and more. We also give discounts to educational
establishments.

Whatever your DTP requirements contact The Electronic
Font Foundry first.

The Electronic Font Foundry, 18 Brockenhurst Rd., Ascot SL5 9DL
Telephone or Fax (24hrs) 0344 872923

Mijas Software

Small C System £69 (inclusive of VAT p&p)

Use the Mijas Small C System on the BBC B or MASTER series computers for games,
educational and industrial software. Small C programs run up to 12 times faster than in
BASIC. Your code is highly portable and can be recompiled to run on the ARCHIMEDES
or R140 using the Acorn ANSI C compiler (not supplied) and on many other machines.
You can also use the power of the ARCHIMEDES for the rapid development and test of
BBC B or MASTER programs. Manuals and post sales support included.

This Small C System produces stand-alone programs with up to 40K of code in ROM
and/or MAIN memory. Libraries, extendable by the user, provide memory allocation,
multi-mode graphics, file handling, i/o formatting, string handling, and system calls. Full
assembler interface.

The system includes the V3.0 Small C compiler & Libraries*, Optimiser, Assembler,
Linker, Source-Level Debug, and SHELL. SOURCE CODE is supplied for the compiler
and all libraries. Using the MAKE facility and editable makefiles, C code is automatically
compiled to assembler source, assembled and then linked with the minimum necessary
library code.

Available for the MASTER series Computers, BBC B+ or B with sideways ram, and
ARCHIMEDES. Software supplied on 80T double sided 5.25" or 3.5" ADFS or DFS
disk. The ADFS disk contains the SHELL source code. The system is also available for
other hardware configurations, including systems for the Mitsubishi MELPS processors,
please write for details.

Laser Typesetting Program £23 inclusive

Low cost DTP for any BBC B, MASTER or ARCHIMEDES. Requires an HP Deskjet or
Laserjet printer or emulation. Use for letters, booklets, forms and manuals with rules and
shading, full multi-font justification, in single or multi-column. Fast printing using the
printer's internal fonts. ADFS or DFS disk with rom image, £23 inclusive of VAT, p&p.
Eprom (avoids the need for sideways ram on BBC B) £7 extra.

Please state your computer system when ordering from:-

**MIJAS SOFTWARE, Winchester Road, Micheldever,
Winchester, Hants SO21 3DG. Tel: 0962 89 352.**

*Official orders, ACCESS and VISA welcome

*Includes original Small C code supplied at the cost of distribution

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes
The program works as shown

Amstrad CPC

Make the same changes as
for the Spectrum to Lines:

290,320,340,420,480,500,610

Change the following lines:

```
40 INK 0,0:INK 1,2:INK 2,6:
INK 3,8:INK 4,18:INK 5,20:INK
6,24:INK 7,26:MODE 0:PEN 7
600 WHILE INKEYS="" :WEND
650 LOCATE X+1,Y+1:PRINT AS
```

ATARI (Stos)/ Amiga (Amos)

Make the same changes as
for the Spectrum to Lines:

290,320,340,420,480,500,610

Change the following lines:

```
40 MODE 0:CURS OFF:KEY OFF:
HIDE
45 COLOUR1,$700:COLOUR2,$7:
COLOUR4,$70:COLOUR3,$707:COLOU
R5,$770:COLOUR6,$77:COLOUR7,$7
77
600 WHILE INKEYS="" :WEND
650 LOCATE X,Y:PRINT AS
```

Amos: Leave out KEY OFF
in Line 40

Commodore 64/128

Change the following lines:

```
40 POKE 53280,0:POKE 53281,
0:PRINT CHR$(147);
45 FOR N=1 TO 7:READ A:CS(N
)=CHR$(A):NEXT N
50 LET W=38
60 LET SS=CHR$(18)+" "+C
HR$(146)
290 PRINT CS(C3);
320 PRINT CS(C3+C1);
340 PRINT CS(C3+C2+C1);
360 PRINT CS(C3+C2);
420 PRINT CS(C1);
480 PRINT CS(C1+C2);
500 PRINT CS(C2);
600 GET AS:IF AS="" THEN 600
610 PRINT CHR$(147);
650 POKE 211,X:POKE214,Y:SYS
58732:PRINT AS;
1000 DATA 28,30,158,31,156,15
9,5
```

Spectrum

Change the following lines:

```
40 BORDER 0: PAPER 0: INK 7:
CLS
50 LET W=32
290 PAPER C3
320 PAPER C3+C1
340 PAPER C3+C2+1
360 PAPER C3+C2
420 PAPER C1
480 PAPER C1+C2
500 PAPER C2
570 IF BS(S)<>" " THEN LET
S=S-1:GOTO 570
580 LET AS=BS( TO S-1): LET
BS=BS(S+1 TO )
600 PAUSE 0
610 PAPER 0: CLS
650 PRINT AT Y,X;AS
```

PC (GW Basic)

Change the following lines:

```
40 SCREEN 0:CLS
50 LET W=40
60 LET SS=STRING$(4,219)
160 LET C1=4:GOSUB 410
170 LET C2=2:GOSUB 470
190 LET C1=4:GOSUB 410
200 LET C2=1:GOSUB 470
220 LET C1=2:GOSUB 410
230 LET C2=1:GOSUB 470
260 LET X1=4:LET Y1=4:LET C1
=4:GOSUB 420
270 LET X1=8:LET Y1=8:LET C2
=2:GOSUB 480
280 LET X1=0:LET Y1=8:LET C3
=1
290 COLOR C3+8
320 COLOR C3+C1+8
340 COLOR C3+C2+C1+8
360 COLOR C3+C2+8
400 STOP
420 COLOR C1+8
480 COLOR C1+C2+8
500 COLOR C2+8
600 LET GS=INKEYS:WHILE GS=""
:LET GS=INKEYS:WEND
610 COLOR ,0:CLS
650 LOCATE Y+1,X+1:PRINT AS
```


The Safe Scientist looks at what happens when you...

PAIN WITH LIGHT

Hi there again. This month we're looking at how coloured lights behave.

Imagine some projectors shining coloured squares on to the back of your computer screen. Where the colours mix you get different colours.

Our program shows what happens. The colours from our imaginary projectors are called primary colours.

Where two or more mix they create what are called secondary colours.

It's like mixing paints to make new shades. But note something strange.

The colours you get when you mix lights are different to what you get when you mix paints.

Type the program in, SAVE it and RUN it. After the intro screen you'll see two overlapping areas of primary lights.

Where these cross you'll see the secondary colour. Press the spacebar to see other pairs of colours and their mixes. When the three possible

combinations of two primary lights have been tried all three primary colours are shown at once. Now you can see what happens when all three overlap.

Why should we want to mix coloured lights? Well it's happening all the time inside your TV set. Millions of different colours are being created all the time.

Without colour mixing your favourite TV show would look very dull indeed.

As our program shows, red, green and blue can be mixed by your computer to give other colours. But this is only the start. It's well worth you investigating more about colours yourself.

If you add secondary colours together you get even more. You can also subtract colours.

● Next month get your seat belts ready. We're off to study gravity! And you'll have the chance to touch down on any planet in the solar system!

WHAT GOES ON IN A TV TUBE?

This program gives you an idea how your TV set works. At the back of the tube there are red, green and blue "projectors" - called electron guns.

When these fire at the screen coloured dots light up. The dots are so close that if, say, a red and blue one light up next to each other you see purple.

Every picture you see on your screen is made up from these primary colours. But your TV set can do more with them than our simple program can. It can give the three colours different degrees of brightness.

This means that, for example, they can mix bright green with dull red. Now that's a really unusual colour!

```
10 REM COLOURS
20 REM BY THE SAFE SCIENTIST
30 REM (c) LET'S COMPUTE!
40 MODE 2
50 LET W=20
60 LET SS="":REM 4 spaces
70 LET AS="Primary Colours":LET Y=1:G
OSUB 640
80 LET BS="Some projector beams shine
onto a screen and overlap.":GOSUB 540
90 LET BS="When colours merge they gi
ve a new colour.":GOSUB 540
100 LET BS="The projectors shine out i
n red, blue or green.":GOSUB 540
110 LET BS="These are called the PRIMA
RY colours.":GOSUB 540
120 LET BS="The new colours are called
SECONDARY colours.":GOSUB 540
130 LET BS="Let's look at the secondar
y colours.":GOSUB 540
140 LET BS="Mixing Two Colours"
150 GOSUB 600
160 LET C1=1:GOSUB 410
170 LET C2=2:GOSUB 470
180 GOSUB 600
190 LET C1=1:GOSUB 410
200 LET C2=4:GOSUB 470
210 GOSUB 600
220 LET C1=2:GOSUB 410
```

```
230 LET C2=4:GOSUB 470
240 LET BS="All three colours."
250 GOSUB 600
260 LET X1=4:LET Y1=4:LET C1=1:GOSUB 4
20
270 LET X1=8:LET Y1=8:LET C2=2:GOSUB 4
80
280 LET X1=0:LET Y1=8:LET C3=4
290 COLOUR 128+C3
300 LET AS=SS+SS:LET IS=8:LET IE=11:LE
T XS=0:GOSUB 440
310 LET AS=SS:LET IS=0:LET IE=7:LET XS
=0:GOSUB 440
320 COLOUR 128+C3+C1
330 LET XS=4:GOSUB 440
340 COLOUR 128+C3+C2+C1
350 LET XS=8:GOSUB 440
360 COLOUR 128+C3+C2
370 LET IS=8:LET IE=11:LET XS=8:GOSUB
440
380 LET BS="That's all folks."
390 GOSUB 600
400 LET BS="":GOSUB 600:RUN
410 LET X1=2:LET Y1=4
420 COLOUR 128+C1
430 LET AS=SS+SS+SS:LET IS=0:LET IE=1
1:LET XS=0
440 LET XS=XS+W/2-10
450 FOR I=IS TO IE:LET X=X1+XS:LET Y=Y
```

```
1+I:GOSUB 650:NEXT I
460 RETURN
470 LET X1=6:LET Y1=8
480 COLOUR 128+C1+C2
490 LET AS=SS+SS:LET IS=0:LET IE=7:LET
XS=0:GOSUB 440
500 COLOUR 128+C2
510 LET AS=SS:LET IS=0:LET IE=7:LET XS
=8:GOSUB 440
520 LET AS=SS+SS+SS:LET IS=8:LET IE=11
:LET XS=0:GOSUB 440
530 RETURN
540 PRINT
550 IF LEN (BS)<=W THEN PRINT BS:RETUR
N
560 LET S=W
570 IF MID$(BS,S,1)<>" " THEN LET S=S
-1:GOTO 570
580 LET AS=LEFT$(BS,S-1):LET BS=MID$(B
S,S+1)
590 PRINT AS:GOTO 550
600 PRINT:PRINT"PRESS SPACE":AA=GET
610 COLOUR 128:CLS
620 LET Y=0
630 LET AS=BS
640 LET X=INT ((W-LEN (AS))/2)
650 PRINT TAB(X,Y);AS
660 RETURN
```


There is more than one way to skin a cat. Our resident micro medic tells of three ways to save you jumping out of loops.

THE PROGRAM DOCTOR

Last month Doc operated on the program shown top right. It tests whether you know the alphabet. If you didn't try it last month, type it in and RUN it now.

It will seem to work as it is. But if you type the alphabet in the wrong order a few times you are likely to get an error message.

That's because each time you press a wrong letter the program jumps out of a loop.

Last month the Doc prescribed a cure to remove this problem. Now he looks at even better methods of doing the same job.

The one you should use depends on the Basic language fitted in your computer. At least one of the ways will work on YOUR machine. The panel on the right tells you which you can use.

First, let's describe the simple game in words. You have to REPEAT typing a key UNTIL you've gone through the whole alphabet OR have made a mistake.

If your Basic understands the commands REPEAT ... UNTIL and OR those words can be put into the program almost exactly as they are spoken.

The You've made a mistake part comes when a flag is set to 1. Remember, we used flags last month. So you can REPEAT the loop UNTIL I>26 OR F=1.

Note that REPEAT ... UNTIL loops don't automatically add one to I so Doc put I=1 near the start of the program and I=I+1 in the loop. All doc's changes are shown on the right.

Some versions of Basic don't understand REPEAT ... UNTIL but have another type of loop. It's called WHILE ... WEND.

Like the repeat type of loop it works as you would say it: "While the 26th letter hasn't been reached and a mistake hasn't been made keep looping". WEND is used to show the end of the loop.

Doc's second set of changes show what you have to do if you want to use this sort of loop.

But some Basics have neither WHILE ... WEND nor REPEAT ... UNTIL - see the panel to check which YOUR machine uses. So Doc's final changes show how you can do REPEAT ... UNTIL without actually having the instructions in your version of Basic.

In place of REPEAT put a REM to show it's the start of the loop. In place of UNTIL something happens you have to go back to the start of the loop if it hasn't happened.

So Doc has now shown four cures for the same problem. And there are plenty of others.

The main point to remember is that you should NEVER jump out of a loop.

This program works on all home micros

This is the original program:

```
10 CLS: REM C64 PRINT CHR$(147);
20 PRINT "TYPE THE ALPHABET NOW!"
30 FOR I = 1 TO 26
40 GOSUB 160
50 PRINT K$;
60 IF K$<>CHR$(I+64) THEN GOTO 100
70 NEXT I
80 PRINT:PRINT:PRINT "WELL DONE"
90 GOTO 110
100 PRINT:PRINT:PRINT "YOU GOT IT
WRONG"
110 PRINT:PRINT "DO YOU WANT ANOTHE
R GO?"
120 GOSUB 160
130 IF K$="Y" THEN GOTO 10
140 PRINT:PRINT "BYE"
150 END: REM ON SPECTRUM USE STOP
160 INPUT K$
170 RETURN
```

Here are the changes Doc made to the original program using REPEAT ... UNTIL:

```
25 LET F=0:LET I=1
30 REPEAT
60 IF K$<>CHR$(I+64) THEN LET F
=1
65 LET I=I+1
70 UNTIL I>26 OR F=1
75 IF F=1 THEN GOTO 100
```

Here are the changes Doc made to the original program using WHILE ... WEND:

```
25 LET F=0:LET I=1
30 WHILE I<=26 AND F=0
60 IF K$<>CHR$(I+64) THEN LET F
=1
65 LET I=I+1
70 WEND
75 IF F=1 THEN GOTO 100
```

Here are the changes Doc made to the original program to make his own REPEAT ... UNTIL:

```
25 LET F=0:LET I=1
30 REM REPEAT
60 IF K$<>CHR$(I+64) THEN L
ET F=1
65 LET I=I+1
70 IF I<=26 AND F<>1 THEN GOTO
30
75 IF F=1 THEN GOTO 100
```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The Basic in these computers understands REPEAT ... UNTIL but not WHILE ... WEND.

CPC/PC(QW-Basic)

The Basic in these computers understands WHILE ... WEND but not REPEAT ... UNTIL.

Amiga and Atari ST

Stos and Amos understand both WHILE ... WEND and REPEAT ... UNTIL.

Spectrum/Commodore 64/128

The Basic in these computers doesn't understand either REPEAT ... UNTIL or WHILE ... WEND.

Do you think YOU have written a program that can't be improved? Then send it along to the Program Doctor. He loves a challenge!

SWEAT

What we call a program is nothing more than a list of instructions to your computer.

These instructions must be in a language that your computer can understand. Nearly all the listings in *Let's Compute!* are in the computer language Basic. This is understood by most home computers.

As a rule, if each line of a program starts with a number it's in Basic. And you must have Basic running in your computer before you start typing in a Basic program.

The BBC Micro, Electron, Spectrum, Amstrad CPC and Commodore 64/128 have Basic running from the moment they're switched on.

If you don't get the message BASIC when you switch on a BBC or Electron, type:

* BASIC

Then press the Return key and you should be in Basic. On the Archimedes and BBC A3000 you usually see the Desktop when you switch on. If you press the F12 key a new line starting with a star will appear at the bottom.

Like on the BBC and Electron, type *BASIC and you're ready to go.

WARNING: The Amiga, Atari ST and PC do not have Basic built in. You have to load it from disc.

You can get several different types of Basic for these computers. The programs in *Let's Compute!*

Keying in a program listing is easy. Or is it? Be warned! There are lots of pitfalls. One simple mistake could stop the whole program from working. But you can make sure YOUR typing is error free by following this simple advice.

are designed to run if you use Stos on the Atari ST or Amos on the Amiga.

Many people already have Stos and Amos. They come from Database Software (0625 859333). If you've got an Amiga, you'll also find that the BBC versions of the programs will work if you use Commodore's BBC emulator.

If you use a PC you will also have to load Basic from disc. Again there are several choices. The PC listings we give are for GW-Basic.

Most of the programs will also run on one of the versions of BBC Basic that you can get for the PC. If you use BBC Basic, choose the BBC listings, not the PC ones.

SIMPLE STEPS TO SUCCESS

- Before you start entering a new program type NEW to empty the computer's memory
- A Basic program line begins with a line number. Don't press Return until you've typed the whole line. (That doesn't mean the line as you see it in print.) Pressing Return is the LAST thing you do before you get to the next line number.
- When you press Return after typing in a line nothing appears to happen. What you've done is store the line in the micro's memory. It will not be activated until you are ready to tell the program to RUN.
- Use the exact figures and letters that you see in the listing. Don't change ANYTHING.
- Capital letters MEAN capital letters (and lower case letters mean lower case letters).
- Take care with keys that can be mistaken for others. Examples are Ø (zero) instead of 0 or 1 instead of l.
- If you make a mistake while you're typing a line you can correct it so long as you haven't pressed Return. Use Delete to move back down the line and rub out the error. Then re-type the part you erased.
- If you notice a mistake AFTER you've pressed Return just retype the whole line – including the line number. It will replace the one you got wrong.
- If you want to clear the screen type CLS and press return. This only wipes the display. Your program remains in the micro's memory. On the C64 use the CLR key instead.
- SAVE your program at frequent intervals while you type it in – don't wait until you've finished.
- To see the program stored in memory, type LIST.
- To make the program in memory work type RUN.

Take care typing!

If you type in a program from *Let's Compute!* just as it is printed it will always work. But if you don't copy it exactly one of two things will happen:

- You may get an error message because your computer doesn't understand something you typed in.
- Something unusual may happen, but not what you expect.

In the case of the first of these you'll get an error message. It usually gives a line number as well.

Check that line first. But your computer isn't always right! Some errors will report the wrong line number. So check other lines too.

An example of the second type of mistake would be if you type 3+1 instead of 3-1. Your program would work but give you the wrong answers.

The motto is simple. If your program doesn't do what you expect – CHECK EVERY LINE!

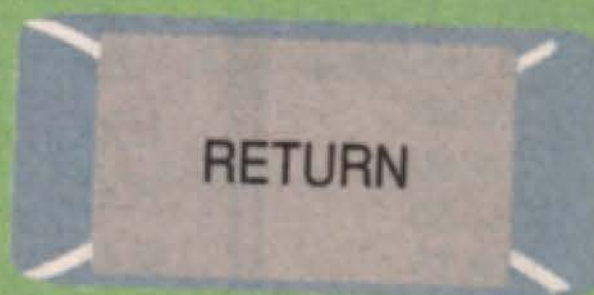
NOT!!



A key difference

Some computers have a key marked RETURN. Others have one labelled ENTER. Some have both.

In *Let's Compute!* when we say Return, you can use either Return or Enter.



SAVING AND LOADING

Make sure you SAVE your program at frequent intervals. Then if you have a power cut or the cat pulls out the plug you won't lose all your work.

First, decide on a short name for your program – say GAME – and then enter:

SAVE "GAME"

Notice that the program's name has quotes before and after it.

What happens next depends on whether you're using tape or disc. But the messages that appear on your

screen are clear and you should have no problems if you follow them.

To put the program back into your micro's memory at the start of a new session, all you have to do is make sure the tape or disc is ready and then enter:

LOAD "GAME"

When it's loaded you can LIST it, RUN it or, if you haven't completed typing it all in, continue adding more lines.

POINTS TO WATCH

The points listed in the main part of this article apply to ALL computers. Here are some more to watch out for on different machines:

Spectrum

On 48k Spectrums you give instructions like PRINT and FOR by pressing one key. If you type all the letters of the word, instead of the single key that's on your keyboard, it will not work even if it looks right.

Amstrad CPC

Spaces are very important on the CPC. Make sure you leave one between the Basic word and the rest of the line.

Joysticks – especially auto-fire ones – can upset how your keyboard works. If the wrong characters appear, unplug your joystick.

PC (GW-Basic), Atari ST (Stos) and Amiga (Amos)

Take care that the line on the screen where you're typing is blank when you start. If there are any characters on it you could be in trouble. When you press Return after typing your new line you might see the message *syntax error* on your screen.

Commodore 64/128

As with the PC, don't type on a line that already has something on it.

After typing a quote mark (") you may get strange characters. If you do, press Return and start again.

Take care when you type + or -. Remember you don't have to press Shift to get these.

If you do use Shift to get + you get something that looks right. But you'll get an error message when you run the program.

LET'S COMPUTE!

99p

Play an
amazing
adventure

LAND YOUR
OWN ROCKET



Write Valentine
verses the easy
way

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

No 7 February 1991
A Database Publication



OVER 100 PRIZES TO WIN!

Introducing...



No.1

Ever wondered how high that big tree is in your garden – or the chimney on your roof? It's easy to find out – using your computer, a piece of string and a few other odds and ends!

Let's make a Clinometer

What you'll need

This issue of *Let's Compute!*

- ☆ A piece of card, about 12cm by 6cm.
- ☆ A tube. The one at the centre of a roll of cling film is ideal. If you can't get one, use the centre of a roll of toilet tissue.
- ☆ Some Sellotape.
- ☆ Some glue for paper.
- ☆ A drawing pin.
- ☆ About 40cm of thin string.
- ☆ A small weight such as an old key or a steel nut.
- ☆ A computer.

How high is your house? It's not very easy to find out using just a tape measure! You COULD use a ladder and long tape. But it's difficult and dangerous. So don't try it!

So how DO you find the height of anything – a house, a tree, a block of flats or even a giraffe?

Surveyors use what's called a **clinometer**. A real one is very expensive.

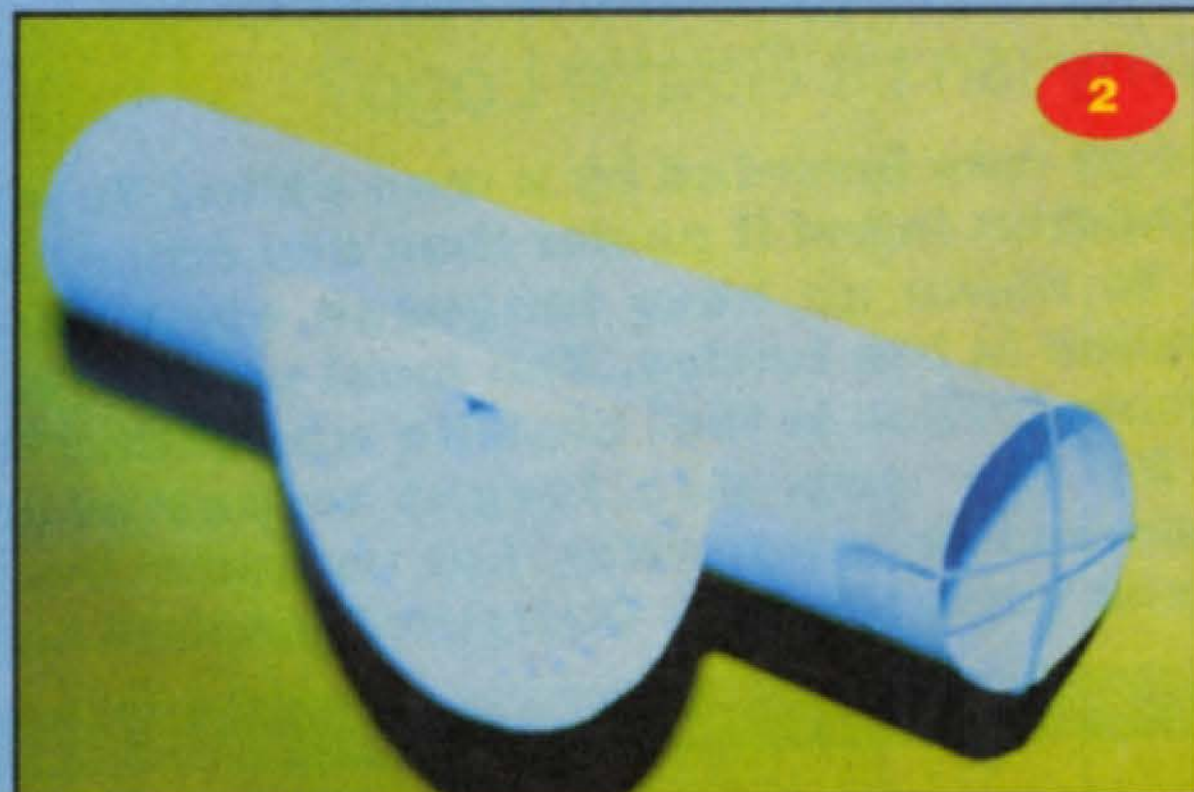
But you can build one from odds and ends that can be found around your home. The step-by-step instructions on this page tell you how to do it.

Clinometers measure angles. Before computers were invented you had to use maths to work out the height from the angle. Now you can use your computer and this *Let's Compute!* program to do it for you.

By making your own clinometer and feeding readings from it into your computer you can find the height of anything you want.

If you discover the height of anything unusual write to *Let's Compute!* and tell us. If you include a photograph of yourself using your clinometer you might even see it pinned to the noticeboard!

Cut out the clinometer scale from Page 8. Then glue it to the card and trim it into the semi-circular shape shown.



Tape the cardboard scale to the tube. Tape two small pieces of string over the end of the tube to form a cross.

HOW TO USE IT

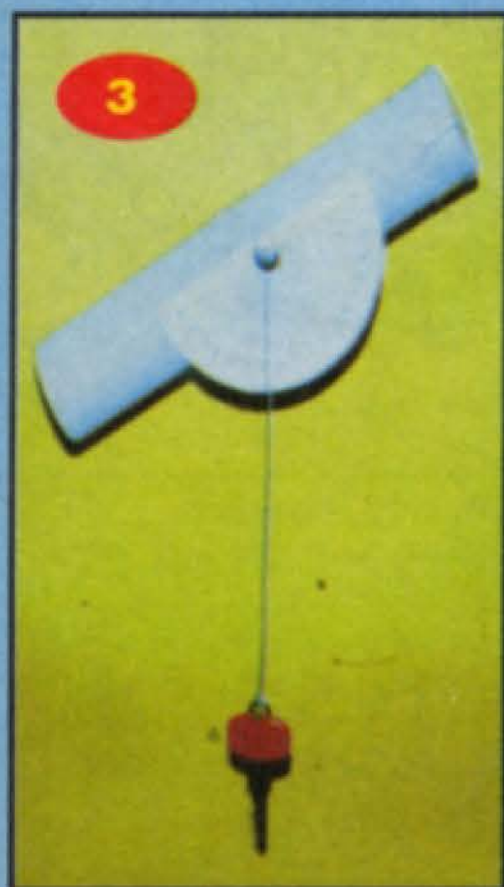
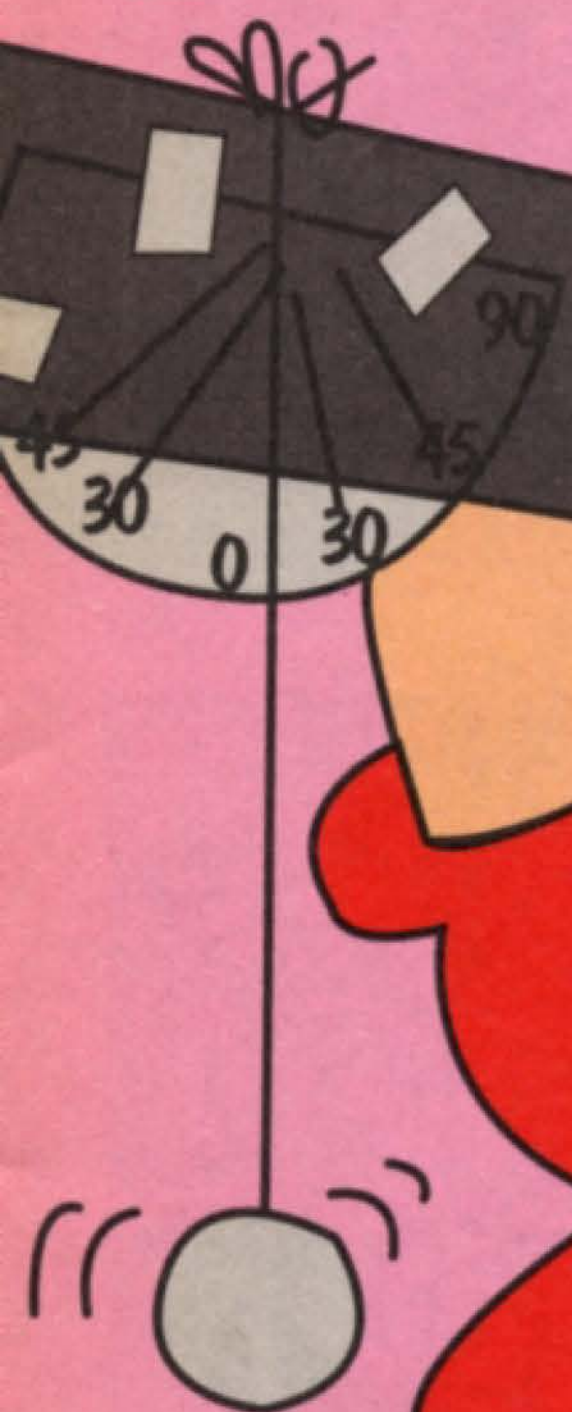
1 Suppose you want to measure the height of a tree. Stand at the foot of the tree and walk away – counting the number of paces.

2 Stop when you can see the top of the tree without stretching your neck. Jot down the number of paces you took.

3 Look at the top of the tree through your clinometer. Make a note of the reading where the string crosses the scale.

4 Take the two readings to your computer and use the program to discover the height of the tree.

If you want to be more accurate, exactly measure the distance you stand away from the tree rather than just count the paces.



Tie a piece of string round the drawing pin. Then push it through the cross on the scale. You may find it best to keep the pin in place with Sellotape. Tie a weight to the other end of the string. Your clinometer is now ready for use.

USING YOUR COMPUTER

Type the program in, SAVE IT and RUN it. You'll be asked five simple questions. Here's how you should answer them:

What units?

You'll usually reply Metres or M. But you can use anything you want such as cm, inches or ft. Make sure you measure all the distances in the units you choose.

How high are your eyes?

Measure from the floor to your eyes and type in the figure.

How big are your paces?

Tell your computer how far you go with each step. If you measured the distance exactly instead of counting paces, answer 1 to this question.

How many paces did you take?

Enter the figure you found when you used your clinometer. If you measured rather than pacing out the distance, give the measurement.

Your computer will then tell you how far away was the object you are measuring. You'll then be asked the final question:

What angle did your clinometer give?

Just enter the reading you got from your clinometer scale.

You'll be told the height of the object you measured.

And that's all there is to it!

You'll then be asked if you want another go. You must answer with a capital Y for yes or N for No.

If you answer Y you won't be asked the first three questions again. The answers are the same every time.

```
10 CLS
20 INPUT "WHAT UNITS";US
30 INPUT "HOW HIGH ARE YOUR EYES";H
40 INPUT "HOW BIG ARE YOUR STEPS";B
50 INPUT "HOW MANY PACES DID YOU TAKE";S
60 LET D=S*B
70 PRINT:PRINT
80 PRINT "THE OBJECT WAS ";D;" ";US;" AW
AY"
90 PRINT:PRINT
100 INPUT "WHAT ANGLE DID YOUR CLINOMETER
GIVE";A
110 LET R=INT(100*(H+D*TAN(3.14159/180*A)
)+.5)/100
120 PRINT "IT IS ";R;" ";US;" HIGH"
130 PRINT:PRINT
140 INPUT "DO YOU WANT ANOTHER GO (Y/N)";
QS
150 IF QS="Y" THEN GOTO 50
160 IF QS<>"N" THEN GOTO 140
```

This program works on all home micros

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes/Spectrum/PC(GW-Basic)

The program works as shown.

Commodore 64/128

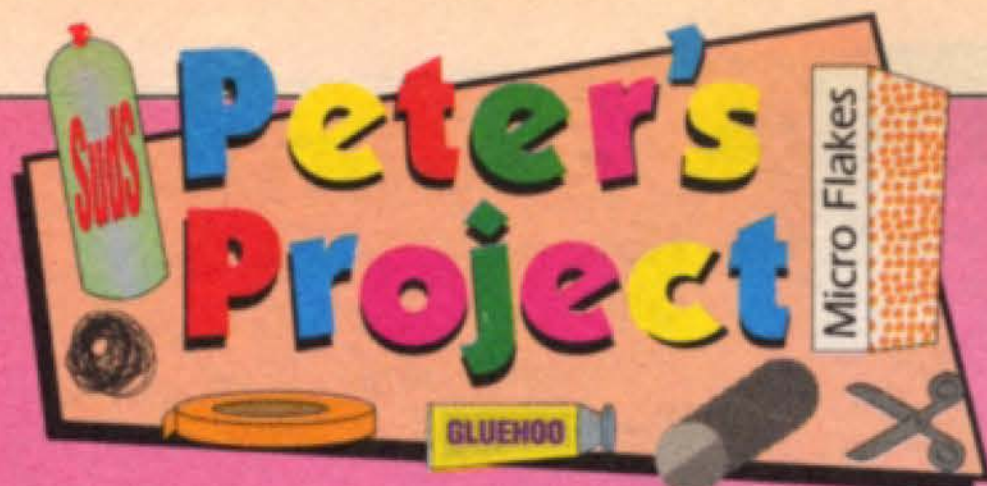
Change Line 10 to:

```
10 PRINT CHR$(147);
```

ST(Stos)

Put # after the variables H, B, D, S, B and R. So, for example, H becomes H#.

NEXT MONTH: Pete will tell you what a Pedometer is, how to make one and how to use it.



THE MATHS!

Using the program on the previous page saves you knowing how to work out distances from angles. But in case you're wondering what happens in Line 110 here's a brief explanation.

In the diagram:

$$\tan A = \frac{H}{D}$$

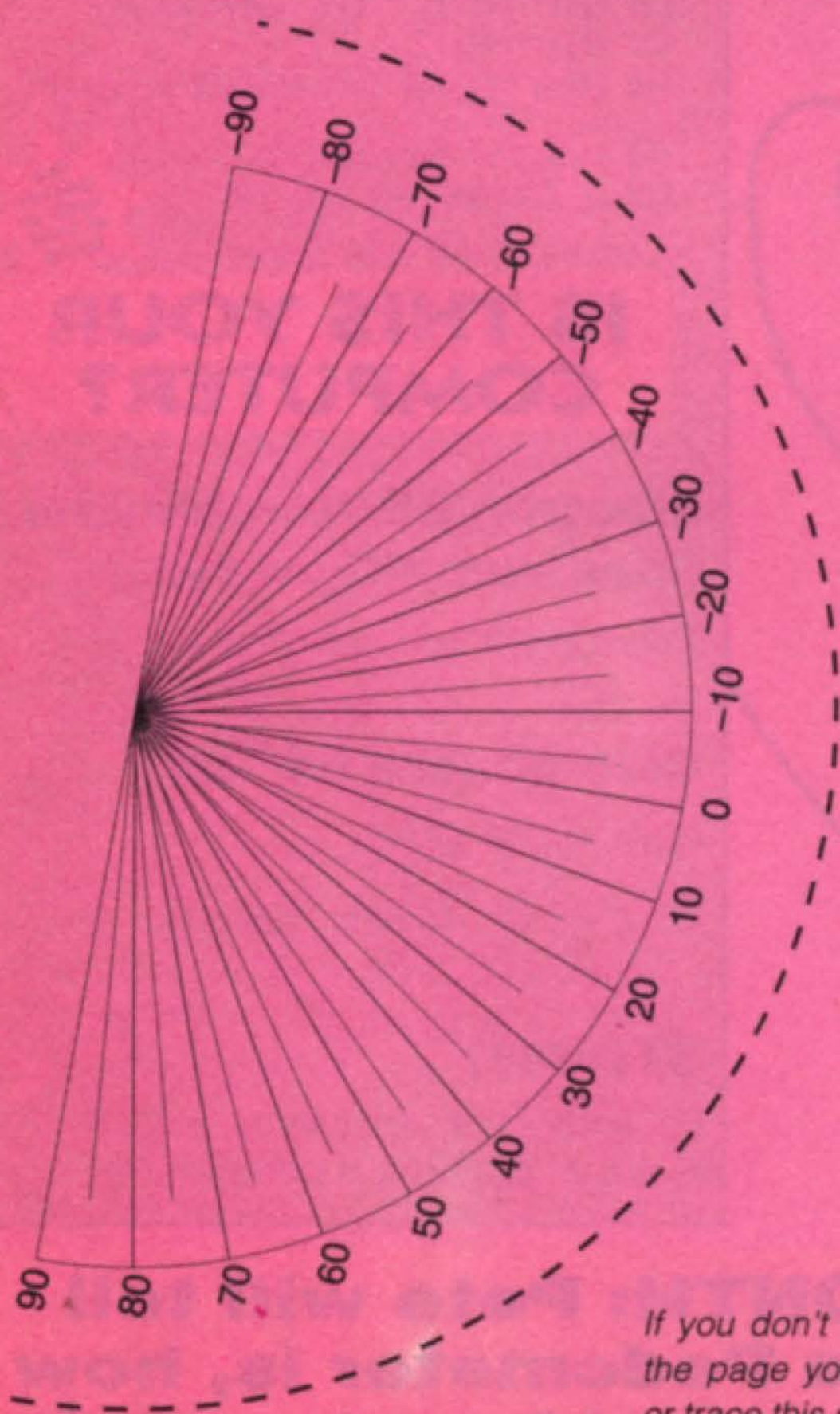
So:

$$H = D \tan A$$

The Height H is worked out from this formula. The height of your eyes is then added because, unless you lay on the ground, that's how far above the ground the clinometer was held.

When you work out angles you probably use Degrees. But most computers use Radians. That's why Line 110 multiplies the angle A by π (3.14159) and divides it by 180.

Final note: The INT and +0.5 are there to round off the numbers. This means that you'll get an answer like 4.57 rather than 4.568237. The Program Doctor explains the method on Page 27.



If you don't want to cut the page you can copy or trace this scale

Fine fonts
Fun fonts
Frightening fonts
Formal fonts

"They all look good on screen, and are properly hinted."

- New Computer Express, Issue 107.

We have over 200 high quality fonts available for use with Archimedes and A3000 DTP programs. Contact us for a free font catalogue.

As well as fonts, we supply everything you need for DTP at very competitive prices, computers, monitors, printers, software and more. We also give discounts to educational establishments.

Whatever your DTP requirements contact The Electronic Font Foundry first.

The Electronic Font Foundry, 18 Brockenhurst Rd., Ascot SL5 9DL
Telephone or Fax (24hrs) 0344 872923



Dear Mr Newsagent

Please deliver *Let's Compute!* to the address below until further notice.

Name _____

Address _____

Note to newsagent:
If you have difficulty in obtaining *Let's Compute!* please contact the distributors to the news trade, COMAG, on 0895 444 055.

THE PROGRAM DOCTOR

Are you troubled by long strings of numbers after a decimal point? Our medic shows how to get rid of them.

The program printed below recently fell on Doc's table. It's a simple VAT calculator. It works out how much things cost once VAT is added to the price.

Type it in and try it. Check that if you enter £100 the price with VAT is £115.

Now try £69. You'll get an answer of £79.35. Nothing wrong with that.

But now try finding out how much something priced at £41. 5 will cost with VAT. The program will tell you it's £47.4375. Now that doesn't look right!

You only want two figures after the decimal point.

How do you lose the others? Try running the program again, but this time with this extra line:

```
45 LET T=INT(T)
```

Again use £41.25 as a test. The computer says the answer is £47. Now ALL the numbers after the decimal point in the answer have gone!

So how do you keep just two of them? Simple! Multiply the answer by 100 before removing everything after the point.

Once they've gone, divide what's left by 100. You'll be left with two figures after the point. Here's what happens to 47.4375:

The original number	47.4375
Multiply by 100.....	4743.75
Remove numbers after the point	4743
Divide by 100	47.43

```
10 LET V=15
20 PRINT
30 INPUT "WHAT IS THE EX. VAT PRICE";P
40 LET T=P*(1+V/100)
50 PRINT
60 PRINT "IT WILL COST ";T;" WITH VAT"
70 GOTO 20
```

ST(Stos)/Amiga(Amos):

Use V#,P# and T# in place of V,P and T.

This
program
works on
all home
micros

Here's a neat way of rounding off money – like £14.56821 – to a much tidier £14.57.

And not only money. Look at Peter's project in this issue. He does exactly the same to round off distances.

You can use it for any numbers: Scores, weights, areas.

It's a useful little routine to put in ANY of your programs.

Try it in the program. The new Line 45 to do this is:

```
45 LET T= INT(T*100)/100
```

But 47.43 is still not quite right! £47.4375 is nearer to £47.44. What you've done is known by programmers as truncating the number.

We really wanted to round it off. One more step is needed.

After multiplying by 100 just add 0.5. Following the steps you get:

The original number.....	47.4375
Multiply by 100.....	4743.75
Add 0.5	4744.25
Remove numbers after the point	4744
Divide by 100	47.44

So type in this new Line 45:

```
45 LET T= INT(T*100+0.5)/100
```

You'll find all your answers now look much better. You'll never get more than two numbers after the decimal point. And what if you want to have three decimal places instead of two?

Just use 1000 in place of 100 in all the above examples. So you can round off to any number of decimal places in a similar way.

Use Doc's method in your own programs to tidy up the printout.

Do you think YOU have written a program that can't be improved? Then send it along to the Program Doctor. He loves a challenge!

MAY THE FORCE BE WITH YOU

This month we are hoping to make a soft landing in the world of forces. The program here is all about making a safe descent in a rocket.

You need to land it on various planets and moons in the solar system. The force of gravity is different on them all.

When you run the program you are first shown the rocket landing on the surface of the Earth. You're told what force was needed to achieve the gentle touchdown.

You must then try to land the same rocket on a different heavenly object. You'll be told the gravity and must type in the force needed.

If you do not use enough force, your spacecraft will hammer into the ground at high speed. On the other hand, if you use too

much force you will run out of fuel before landing. This too will cause a crash.

A successful flight engineer will get the rocket burn correct. A safe, soft landing will result.

If you get it wrong, you'll be told what force you should have used.

The program is easy to alter. You could add more real objects from our solar system.

Or you could let your imagination run riot and invent your own planets.

Do remember that science should be fun! Use the program with a friend. You can often take on new ideas more quickly if you can share your thoughts with someone else.

● Next month we'll look at living things. You be challenged to struggle to maintain the balance in a nature park.

```

10 MODE 0
20 LET W=80:LET H=30:LET Z=9:REM Z IS
THE NUMBER OF PLANETS IN DATA
30 LET X=RND(-TIME)
40 LET RS="  /\  /__\I  IIooII__I//
\\"
50 LET ES=""
60 LET ES=ES+" EARTH":IF LEN(ES)<=W T
HEN GOTO 60
70 LET ES=LEFTS(ES,W)
80 LET D=0
90 LET X=0:LET Y=H-2:GOSUB 580:PRINT
ES
100 LET F=RND(91)+9
110 LET X=0:LET Y=0:GOSUB 580:LET BS=""
To land this rocket on earth a force of "
120 LET BS=BS+STR$(F)+" is needed. Her
e goes!":GOSUB 600
130 GOSUB 410:REM lander routine
140 LET T=TIME:REPEAT UNTIL TIME>T+20
0
150 GOSUB 550:REM clear landed rocket
160 RESTORE
170 LET S=RND(2)
180 FOR I=1 TO S:READ AS,G:NEXT I:LET P
S=""
190 LET PS=PS+" "+AS:IF LEN(PS)<=W THE
N GOTO 190
200 LET PS=LEFTS(PS,W)
210 LET X=0:LET Y=H-2:GOSUB 580:PRINT
PS
220 LET Y=5:GOSUB 580:LET BS="Your ta
sk is to land on "+AS+"."
230 LET BS=BS+" The gravity is "+STR$(
G)+" times that on earth.":GOSUB 600
240 LET Y=10:GOSUB 580:INPUT "Landing
force";L
250 IF L=F*G THEN LET D=0
260 IF L>F*G THEN LET D=-1
270 IF L<F*G THEN LET D=1
280 GOSUB 410
290 LET X=0:LET Y=12:GOSUB 580
300 IF D=0 THEN PRINT "A perfect landi
ng":GOTO 370
310 IF D=-1 THEN LET BS="Too much powe
r. You ran out of fuel.":GOSUB 600
320 IF D=-1 THEN LET BS="Your engines
cut before you landed. You crashed!"
330 IF D=1 THEN LET BS="Not enough for

```


The Safe Scientist goes space-hopping

```

ce. You crashed heavily."
340 GOSUB 600:PRINT:LET B$="The correc
t force was "+STR$(F*G):GOSUB 600
350 LET A$="  \ / '#\< \**#\*":LET X=
W-7:FOR YO=H-9 TO H-5:FOR Y=YO TO YO+3
360 GOSUB 580:PRINT MID$(A$(Y-YO)*4+1
,4):NEXT Y:NEXT YO
370 LET X=W/2-9:LET Y=H-2:GOSUB 580:PR
INT "  PRESS SPACE  ";
380 A=GET
390 CLS:GOTO 80
400 END
410 REM lander routine
420 FOR N=0 TO 31+D
430 LET T=TIME
440 LET YO=N+H-41:GOSUB 490
450 SOUND 1,-15,200-N*5,1
460 REPEAT:UNTIL TIME>T+1
470 NEXT N
480 RETURN
490 REM print rocket
500 IF YO>=0 THEN LET A$=R$:LET SL=6:G
OTO 530
510 IF YO<=5 THEN RETURN
520 LET A$=RIGHT$(R$(YO+6)*4):LET SL=
YO+6:LET YO=0
530 LET X=W-7:FOR L=0 TO SL:LET Y=YO+1
+L:GOSUB 580:PRINT MID$(A$(1+(L*4)),4)
540 NEXT L:RETURN
550 REM clear landed rockets
560 LET X=W-7:FOR Y=H-8 TO H-3:GOSUB 5
80:PRINT "  ":NEXT Y
570 RETURN
580 REM tab
590 PRINT TAB(X,Y);:RETURN
600 REM word wrap
610 IF LEN(B$)<=W-12 THEN PRINT B$:RET
URN
620 LET S=W-11
630 IF MID$(B$,S,1)<>" " THEN LET S=S-
1:GOTO 630
640 LET A$=LEFT$(B$,S-1):LET B$=MID$(B
$,S+1)
650 PRINT A$:GOTO 600
660 DATA "venus",0.8,"mercury",0.4,"th
e moon",0.2,"mars",0.7
670 DATA "jupiter",12,"saturn",10,"ura
nus",5,"neptune",3
680 DATA "pluto",0.5

```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

This program works as shown

PC (Using GW-Basic)

Delete Lines 430 and 460.

Change or add the following lines:

```

10 CLS
20 LET W=80:LET H=22:LET Z=9
30 RANDOMIZE TIMER
100 LET F=INT(RND*10+1)*10
140 LET T=TIMER:WHILE T+1>TIMER:WEND
170 LET S=INT(RND*Z+1)
380 WHILE INKEY$="":WEND
450 SOUND 500-N*10,3
590 LOCATE Y+1,X+1:RETURN

```

ST (Using Stos)

Delete Lines 430 and 460.

Change or add the following lines:

```

10 MODE 1:KEY OFF:HIDE ON
20 LET W=80:LET H=22:LET Z=9
30 LET X=RND(TIMER)
100 LET F=RND(90)+10
140 LET T=TIMER:WHILE T+20>TIMER:WEND
170 LET S=RND(Z-1)+1
380 WHILE INKEY$="":WEND
450 VOLUME 15:PLAY 1,96-(N*2),2:VOLUME
0
590 LOCATE X+1,Y+1:RETURN

```

Commodore 64/128

Use capital letters, delete Line 450.

Change or add the following lines:

```

10 PRINT CHR$(147);
20 LET W=38:LET H=25:LET Z=9
30 LET X=RND(-TI)
100 LET F=INT(RND(0)*10+1)*10
140 LET TI="000000"
145 IF TI<100 THEN GOTO 145
170 LET S=INT(RND(0)*Z+1)
190 PRINT CHR$(147);:GOTO 80
380 GET A$:IF A$="" THEN GOTO 380
430 LET TI$="000000"
460 IF YO <=0 THEN IF TI<50 THEN GOTO 460
590 POKE 211,X:POKE 214,Y:SYS 58732:RE
TURN

```

Amiga (Amiga Basic)

Delete Lines 430 and 460.

Change or add the following lines:

```

10 CLS
20 LET W=80:LET H=22:LET Z=9
30 RANDOMIZE TIMER
100 LET F=INT(RND*10+1)*10
140 LET T=TIMER:WHILE T+2>TIMER:WEND
170 LET S=INT(RND*Z+1)
380 WHILE INKEY$="":WEND
450 SOUND 4000-(N*100),1,100,0
590 LOCATE Y+1,X+1:RETURN

```

Amstrad CPC

Change or add the following lines:

```

10 MODE 2
20 LET W=80:LET H=25:LET Z=9
30 RANDOMIZE TIME
100 LET F=INT(RND*10+1)*10
140 LET T=TIME:WHILE T+1>TIME:WEND
170 LET S=INT(RND*Z+1)
380 WHILE INKEY$="":WEND
430 LET T=TIME
450 SOUND 1,33+N*24,20,15
460 WHILE T+60>TIME:WEND
590 LOCATE X+1,Y+1:RETURN

```

Spectrum

Delete Lines 430 and 460.

Use J in place of YO in all cases.

Change or add the following lines:

```

10 CLS
20 LET W=32:LET H=22:LET Z=9
30 RANDOMIZE
70 LET ES=ES ( TO W)
100 LET F=INT(RND*10+1)*10
140 PAUSE 100
170 LET S=INT(RND*Z+1)
200 LET PS=PS ( TO W)
360 GOSUB 580:PRINT A$((Y-J)*4+1 TO 4): NE
XT Y:NEXT J
380 PAUSE 0
400 STOP
450 BEEP 0.2,-N
520 LET A$=R$((YO+6)*4 TO ):LET SL=J+6:LE
T J=0
530 LET X=W-7:FOR L=0 TO SL:LET Y=J+L:GOS
UB 580:PRINT A$(1+(L*4) TO 4):NEXT L
590 PRINT AT Y,X;:RETURN
630 IF B$(S)<>" " THEN LET S=S-1:GOTO 630
640 LET A$=B$(S):LET B$=B$(S+1 TO)

```

This program works on all home micros

The convenient SINGLE SOURCE for EDUCATIONAL SOFTWARE

- * All popular educational programs at publishers prices or below including Mirrorsoft, 4Mation, Sherston, E.S.M., Bourne, Chalksoft, Kosmos, Database Fun School, etc.
- * Wide range of formats: Not only BBC/A3000/ Electron but also Spectrum +2/+3, Amstrad, Nimbus, Commodore 64/Amiga, Atari ST, IBM-PC and others!
- * Vast stocks: 30,000 programs available for immediate delivery.
- * Unconditional guarantee: problems rectified by expert staff - telephone helpline also available.

THE EDUCATIONAL SOFTWARE DIRECTORY
describes and prices hundreds of programs

Write or phone for a free copy

RICKITT EDUCATIONAL MEDIA

FREEPOST • Ilton • Ilminster • Somerset TA19 9HS •

Telephone 0460 57152 • Fax 0460 53176

Please state the age of your children and make of your computer

"OUTLET"

GRAPHICS INFO AND IDEAS!!
SAM COUPE AND SPECTRUM UTILITIES USEFUL
&
INTERESTING HOMEGROWN SOFTWARE!!

Monthly since 1987 - now on ALL of the popular SPECTRUM systems

SPECIAL OFFER - First time buyers send only £2.50 for the latest issue on...
PLUS 3 DISC, DISCIPLE/PLUS D, OPUS, MICRODRIVE, TAPE AND SAM DISK

CHEZRON SOFTWARE

605 Loughborough Road, Birstall, Leicester LE4 4NJ

Mijas Software

*The New Issue of
'A Book on C'
by Berry and
Meekings
£11 inclusive*

Small C System £69 (inclusive of VAT p&p)

Use the Mijas Small C System on the BBC B or MASTER series computers for games, educational and industrial software. Small C programs run up to 12 times faster than in BASIC. Your code is highly portable and can be recompiled to run on the ARCHIMEDES or R140 using the Acorn ANSI C compiler (not supplied) and on many other machines. You can also use the power of the ARCHIMEDES for the rapid development and test of BBC B or MASTER programs. Manuals and post sales support included.

This Small C System produces stand-alone programs with up to 40K of code in ROM and/or MAIN memory. Libraries, extendable by the user, provide memory allocation, multi-mode graphics, file handling, i/o formatting, string handling, and system calls. Full assembler interface.

The system includes the V3.0 Small C compiler & Libraries*, Optimiser, Assembler, Linker, Source-Level Debug, and SHELL. SOURCE CODE is supplied for the compiler and all libraries. Using the MAKE facility and editable makefiles, C code is automatically compiled to assembler source, assembled and then linked with the minimum necessary library code.

Available for the MASTER series Computers, BBC B+ or B with sideways ram, and ARCHIMEDES. Software supplied on 80T double sided 5.25" or 3.5" ADFS or DFS disk. The ADFS disk contains the SHELL source code. The system is also available for other hardware configurations, including systems for the Mitsubishi MELPS processors, please write for details.

Laser Typesetting Program £23 inclusive

Low cost DTP for any BBC B, MASTER or ARCHIMEDES. Requires an HP Deskjet or Laserjet printer or emulation. Use for letters, booklets, forms and manuals with rules and shading, full multi-font justification, in single or multi-column. Fast printing using the printer's internal fonts. ADFS or DFS disk with rom image, £23 inclusive of VAT, p&p. Eprom (avoids the need for sideways ram on BBC B) £7 extra.

Please state your computer system when ordering from:-

**MIJAS SOFTWARE, Winchester Road, Micheldever,
Winchester, Hants SO21 3DG. Tel: 0962 89 352.**

Official orders, ACCESS and VISA welcome

*Includes original Small C code supplied at the cost of distribution

USING YOUR OWN WORDS

The words are in the program in lines following a REM that explains what they are. Like this:

```
5000 REM NEXT VERB
5010 DATA "TAKE","SEE","SEND","BRING"
5020 DATA "X"
```

You can put as many words as you like in the lines like Line 5010 in the above example. If you want to put more than will fit on one line put another data line between that one and the next. For example:

```
5015 DATA "GIVE","SMELL","SHOUT","PUSH"
```

Make sure you leave all the lines that say:

```
DATA "X"
```

There's only one point to watch. The words in Line 590 rhyme with those in Line 930. Lines 720 and 1050 also rhyme.

Look at them. See how the first word in Line 590 rhymes with the first in Line 930, the second rhymes with the second and so on.

Again, you can use your own words. Use as many or as few as you want. But make sure corresponding ones rhyme.

IS THIS YOUR COMPUTER?

Spectrum

Change or add the following Lines:

```
10 RANDOMIZE
20 DEF FNR(N)=INT(RND*N+1)
320 LET B$="A":IF A$(LEN(A$))="S" THEN
LET B$="THE"
410 IF A$(1)<>" " THEN PRINT " "
440 PAUSE 0
```

Amiga (Amiga Basic)

Change or add the following Lines:

```
10 RANDOMIZE TIMER
20 DEF FNR(N)=INT(RND*N+1)
440 IF INKEY$="" THEN GOTO 440
```

ST (Stos)

Change or add the following Lines:

```
10 LET X=RND(TIMER)
20 DEF FNR(N)=RND(N-1)+1
440 IF INKEY$="" THEN GOTO 440
```

C64/128 Use capital letters

Change or add the following Lines:

```
20 DEF FNR(N)=INT(RND(0)*N+1)
440 GET A$:IF A$="" THEN GOTO 440
```

PC (GW Basic)

Change or add the following Lines:

```
10 RANDOMIZE TIMER
20 DEF FNR(N)=INT(RND*N+1)
440 IF INKEY$="" THEN GOTO 440
```

Amstrad CPC

Change or add the following Lines:

```
10 RANDOMIZE TIME
20 DEF FNR(N)=INT(RND*N+1)
440 IF INKEY$="" THEN GOTO 440
```


**“ Your
computer’s
a poet
– and
doesn’t
know it! ”**

Valentine verses

Are you ever stuck for words? It can be quite a problem when you want to send a message to your Valentine.

But not any more! Here’s a program that will write poetry for you.

Type it in, SAVE it and RUN it. You’ll be surprised at the rhyming poems that your computer can come out with.

You can teach it even more. Just alter the DATA lines in the program. The panel at the bottom of the page tells you how to do it.

You never need to be stuck for words again. We’ve set the program up for Valentine’s Day. But with a change of words it can provide you with poems for just about any occasion.

Easter, birthdays, weddings and Christmas – they are all just as easy.

If your computer writes any great poems let us see them!

**Say something
sweet using
Mike Goldberg’s
poetry maker**

```
10 LET X=RND(-TIME)
20 DEF FNR(N)=RND(N)
30 REM line 1
40 FOR I = 1 TO 2
50 PRINT "I ";
60 RESTORE:GOSUB 470
70 PRINT AS;" YOUR ";
80 GOSUB 470
90 PRINT AS;
100 IF I=1 THEN PRINT ". ";
110 IF I=2 THEN LET R1=R:PRINT
120 NEXT I
130 REM line 2
140 LET RN=FNR(2):IF RN=1 THEN PRINT "
I'D ";
150 IF RN=2 THEN PRINT "I ";
160 GOSUB 470
170 PRINT AS;" TO ";
180 GOSUB 470
190 PRINT AS;" YOU ";
200 GOSUB 470
210 LET R2=R:PRINT AS
220 REM line 3
230 GOSUB 470
240 PRINT AS;" ";
250 GOSUB 470
260 PRINT AS;" ";
270 GOSUB 470
280 PRINT AS;" ";
290 GOSUB 470
300 PRINT AS;" ";
310 READ W:LET R=R1:GOSUB 480
320 LET BS="A":IF RIGHTS(AS,1)="S" THE
N LET BS="THE"
330 PRINT BS;" ";AS
340 REM line 4
350 GOSUB 470
360 PRINT AS;" ";
370 LET B=FNR(3):IF B=1 THEN PRINT "I"
;
380 IF B=2 THEN PRINT "YOU";
390 IF B=3 THEN PRINT "WE";
```

```
400 GOSUB 470
410 IF LEFT$(AS,1)<>"I" THEN PRINT " "
;
420 PRINT AS;" ";
430 READ W:LET R=R2:GOSUB 480:PRINT AS
440 Z=GET
450 PRINT:GOTO 30
460 REM select word
470 READ W:LET R=FNR(W)
480 FOR N=1 TO W+1:READ W$:IF N=R THEN
LET AS=W$
490 NEXT N
500 IF W$<>"X" THEN PRINT:PRINT "ERROR
IN DATA STATEMENT":STOP
510 RETURN
520 REM first line
530 REM verb
540 DATA 7
550 DATA "LIKE","LOVE","ENJOY","COVET"
,"LONG FOR","DESIRE","RELISH"
560 DATA "X"
570 REM noun rhyme1
580 DATA 7
590 DATA "FACE","TEETH","EYES","LIPS",
"HAIR","SMILE","HANDS"
600 DATA "X"
610 REM second line
620 REM first verb
630 DATA 4
640 DATA "HOPE","LOVE","LIKE","WANT"
650 DATA "X"
660 REM second verb
670 DATA 4
680 DATA "TAKE","SEE","SEND","BRING"
690 DATA "X"
700 REM rhyme2
710 DATA 7
720 DATA "DANCING","EATING","WALKING",
"FLYING","WISHING","SKATING","SAILING"
730 DATA "X"
740 REM third line
750 REM conjunction
```

```
760 DATA 3
770 DATA "BUT","AND","SO"
780 DATA "X"
790 REM prep
800 DATA 5
810 DATA "WHEN","AS","THEN","SHOULD",
IF"
820 DATA "X"
830 REM verb
840 DATA 6
850 DATA "SIT","KISS","EMBRACE","LINGE
R","WANDER","WALK"
860 DATA "X"
870 REM location
880 DATA 6
890 DATA "UPON","BEHIND","ABOVE","BENE
ATH","AROUND","BEYOND"
900 DATA "X"
910 REM matching noun rhyme 1
920 DATA 7
930 DATA "CASE","HEATH","RISE","DIPS",
"CHAIR","STILE","LANDS"
940 DATA "X"
950 REM fourth line
960 DATA 6
970 DATA "IT'S THEN","I KNOW","PERHAPS
","WITH LUCK","OH DARLING","MY SWEET"
980 DATA "X"
990 REM next bit
1000 DATA 5
1010 DATA "'LL END UP","CAN'T STOP","SH
ALL BE","'LL JOIN IN","'LL FEEL LIKE"
1020 DATA "X"
1030 REM MATCHING RHYME 2
1040 DATA 7
1050 DATA "ROMANCING","MEETING","TALKIN
G","SIGHING","KISSING","DATING","WAILING"
"
1060 DATA "X"
```


Part 5

SEARCH THE CARDS

Roland Waddilove adds a powerful search routine to the Let's Compute! database

If you've been typing in the *Let's Compute!* database for the last four months you now have a program that lets you type in and store cards.

But how can you quickly search for those important facts you've saved?

No matter how big or small a database is – or how much information is held in it – it is useless if we can't get the facts from it.

For example, suppose you use your database to keep a list of all the LP records you have bought. It can also hold the names of the tracks that are on them.

One day you might want to know which albums you have by New Kids On The Block. Or which one contains your favourite track.

It would be nice to be able to type in the name of the track and make the computer tell you the name of the album.

That's better than digging out every record and reading through all the writing on the back of each sleeve!

You also want a faster way than looking at every card in a card index.

HOW? IT'S QUITE EASY!

All we need to do is to add a search subroutine to our *Let's Compute!* database.

The option is already there on the menu. If you have tried to choose it the program will have stopped with an error report. That's because the subroutine hasn't yet been typed into the program.

Now it's time to do it. The extra lines are shown on the right.

Once you've added this new section, using it is a piece of cake.

First, make sure you have some cards in your database. Either type them in or

load some that you saved before.

Now choose the search option from the menu. You'll be asked to type in a word to search for.

But don't press the Return or Enter key without typing anything. If you do every card in your database will be displayed in turn.

This is one way of seeing the cards. But if you have a lot it might take you a long time to find the one you want!

To use the search facility type in one or two words that you know are on the card you want to find.

Make sure you use capitals or lower case letters exactly as you did when you wrote the card. Now press Return.

The first card that contains the words you entered will be printed on the screen.

You can search for more cards by pressing N for next card. Or press M to go back to the main menu screen.

More lines to add to your program. They'll let you sort your cards into alphabetical order.

CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1-9)

```
3000 REM ===== Search for card =====
3010 PRINT "Search for a card"
3020 PRINT "-----"
3030 PRINT
3040 PRINT "Type in a word to search for,
or"
3050 PRINT "press Return to show all the
cards."
3060 PRINT
3070 INPUT "Search for";SS
3080 X=1
3090 IF CS(X)="*" THEN 3230
3100 F=0
3110 FOR Y=0 TO 4
3120 IF INSTR(CS(X+Y),SS)>0 THEN F=1
3130 NEXT Y
3140 IF F=0 THEN GOTO 3230
3150 CLS
3160 PRINT
3170 FOR Y=0 TO 4
3180 PRINT CS(X+Y)
3190 NEXT Y
3200 PRINT
3210 INPUT "Next card or Menu (N or M)";KS
3220 IF KS="M" OR KS="m" THEN X=250
3230 X=X+5
3240 IF X<255 THEN GOTO 3090
3250 RETURN
```

This program works on all home micros

IS THIS YOUR COMPUTER?

**Electron/Archimedes/
BBC/Amstrad/CPC/
ST(stos)/Amiga(amos)/
PC (GW-Basic)**

The listing works as shown

C64/128

Change or add the following lines:

```
3120 FOR Z=1 TO LEN(CS(X+Y))
3122 IF MID$(CS(X+Y),Z,LEN(SS))=SS THEN
N F=1
3130 NEXT Z
3150 PRINT CHR$(147);
```

Spectrum

Change or add the following lines:

```
3090 IF CS(X) =BS THEN GO TO 3230
3095 IF SS="" THEN GO TO 3150
3105 IF LEN (SS)>32 THEN LET SS=SS( TO
32)
3115 FOR I=1 TO 33-LEN (SS)
3120 IF SS=CS(X+Y,I TO I+LEN (SS)-1) T
HEN LET F=1
3125 NEXT I
```


Pendown is already a favourite writing tool in schools across the country. Now it has been completely redesigned to take full advantage of all the facilities that make the Archimedes (A3000) so popular. It features a powerful dictionary, thesaurus and spell checker, along with a friendly user-interface that can be reconfigured to make as much sense to a six-year-old as it does to an adult. Special fonts, simplified layouts and on-screen help make it perfect for the very young, whilst powerful word-processing features mean it is equally at home with students working on 'A' Level projects.

Longman Logotron has a wealth of experience in producing educational software, and also produces the National Curriculum standard Logotron Logo, Numerator – a revolutionary maths package, and Hyperbook software and texts, all of which are available for the Acorn Archimedes. For a brochure and full details of all Longman Logotron's educational software, write – stating whether your interest is primary or secondary education – to the address below.

Archimedes PenDown



Longman Logotron
Dales Brewery
Gwydir Street
Cambridge CB1 2LJ
Tel: 0223 323656
Fax: 0223 460208



LONGMAN
LOGOTRON

make
Words
make
Sense

```

10 PRINT:INPUT"HOW MANY";M
20 LET S=0
30 DIM PS(M):DIM G(M)
40 GOSUB 90:PRINT:PRINT "PRESS 1, 2 OR 3":GOSUB
160
50 IF K<1 OR K>3 THEN GOTO 40
60 ON K GOSUB 200,360,450
70 GOTO 40
80 REM MENU
90 CLS
100 PRINT:PRINT"1 .. TYPE IN ANOTHER"
110 PRINT:PRINT"2 .. SEE THE ONES THERE"
120 PRINT:PRINT"3 .. FIND THE WINNER"
130 RETURN
140 REM WAIT TILL KEY PRESSED
150 PRINT:PRINT"PRESS ANY KEY"
160 REPEAT:LET KS=INKEY$(0):UNTIL KS<>""
170 LET K=ASC(KS)-48
180 RETURN
190 REM ENTER A NAME
200 IF S=M THEN PRINT:PRINT "NO MORE
ROOM":RETURN
210 PRINT:INPUT "NAME";N$
220 PRINT:INPUT "GUESS";G
230 LET F=0:FOR I=1 TO S
240 IF G=G(I) THEN PRINT:PRINT "SOMEONE HAS
ALREADY GOT THAT":LET F=1
250 NEXT I
260 IF F=1 THEN GOTO 220
270 LET S=S+1:LET C=1
280 IF S=1 THEN LET PS(1)=N$:LET G(1)=G:RETURN
290 IF G<G(C) THEN LET C=C+1:GOTO 290
300 FOR I=S TO C+1 STEP -1
310 LET PS(I)=PS(I-1):LET G(I)=G(I-1)
320 NEXT I
330 LET PS(C)=N$:LET G(C)=G
340 RETURN
350 REM PRINT THE NAMES
360 PRINT:IF S=0 THEN PRINT "NO NAMES
ENTERED":GOTO 410
370 FOR I=1 TO S
380 PRINT I;" ";PS(I);" ";G(I)
390 IF ABS(I/5-INT(I/5))<0.01 THEN GOSUB
150:PRINT
400 NEXT I
410 PRINT:PRINT"ROOM FOR ";M-S;" MORE"
420 GOSUB 150
430 RETURN
440 REM FIND WINNER
450 LET W=75
460 LET D=10000:LET Y=0:PRINT
470 FOR I=1 TO S
480 IF ABS(G(I)-W)=0 THEN LET Y=I
490 IF ABS(G(I)-W)<D THEN LET X=I:LET
D=ABS(G(I)-W)
500 NEXT I
510 IF Y=0 THEN PRINT "THE WINNER IS
";PS(X):GOSUB 150:RETURN
520 PRINT "IT'S A DRAW BETWEEN ";PS(X);" AND
";PS(Y)
530 GOSUB 150:RETURN

```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown

CPC/PC/ST(Stos)/ Amiga(Amiga Basic)

Change the following line:

```
160 LET KS=INKEY$:WHILE KS="" :LET KS=INKEY$:WEND
```

Spectrum

Change or add the following lines:

```

30 DIM PS(M,16):DIM G(M):DIM NS(16)
60 IF K=1 THEN GOSUB 200
61 IF K=2 THEN GOSUB 360
62 IF K=3 THEN GOSUB 450
160 LET KS=INKEY$:WHILE KS="" :LET KS=INKEY$:WEND
170 LET K=CODE(KS)-48

```

Commodore 64/128

Change or add the following lines:

```

90 PRINT CHR$(147);
160 GET KS:IF KS="" THEN GOTO 160

```


LET'S MAKE MONEY

Part 3

GUESS THE WEIGHT OF THE CAKE

Use your computer to help with traditional fund-raising events

Guess the weight of the cake! How many sweets in the jar? What does this man weigh? How many coins in the pile?

These, and lots of other guessing games, are used to make money at school fairs. They are all organised in a similar way.

Let's look at guessing the weight of a cake. The idea is simple.

Someone bakes a cake and weighs it exactly. At the fair people pay, say, 10p to say how much they think it weighs.

At the end of the fair, the closest guess wins the cake.

So where does a computer come into this?

It can keep a record of all the names and guesses. The program here shows how it can be done.

Type it in, SAVE it and RUN it.

First you'll be asked *How Many?*

It wants you to say how many people you are going to let have a go. The figure you can enter depends on what type of computer you're using.

Most will work with numbers even as big as 1000.

You don't need to answer the question exactly – any number that's more than the

most you expect will do.

To see how the program works type in 100. Next, a menu with three choices will be shown. You're ready to start entering names and guesses.

Choose *TYPE IN ANOTHER* from the list. You'll then be asked two questions. Answer the first with a friend's name and the other with an imaginary guess. Use a number between 50 and 100 for your test.

Do the same several times until you've got a few names and guesses in your computer. Then try *SEE THE ONES THERE* from the menu. This will show you the list so far.

Finally, try *FIND THE WINNER* to see who is winning. You can then go back and use the first choice and put in a few more names and guesses. See if the winner is still the same person.

Why use a computer?

With this program you can see who is winning at any time. So if someone has a go and finds they aren't in the lead they may want to have another try.

Someone who is winning may come back later to see if they are still winning. If they aren't, they may have another go.

All of which brings in more money!

- 1 .. TYPE IN ANOTHER
- 2 .. SEE THE ONES THERE
- 3 .. FIND THE WINNER

The menu

SET THE WEIGHT

The actual weight of the cake is set in line 440:

```
440 LET W=75
```

Change the 75 to the weight of YOUR cake.

Take care that anyone who is going to enter the contest doesn't see the listing once it's changed!

WHAT ELSE YOU CAN DO

You can easily change the program to make it do different things. For example, you can make it print a ticket for everyone as they make their guess.

You can make it print messages depending how good the guess is. If someone got a "miles away" message they may want to try for a "very warm".

The program is a frame you can build on as you like. Put bright graphics around the menu. Alter some of the features.

The idea is to end up with your own unique fund-raiser.

LET'S COMPUTE!

99p

**BE
BOSS
OF A
NATURE PARK**



**FUND RAISING
CAN BE FUN**

**WRITE
YOUR
OWN
ADVENTURE!**



For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

No 8 March 1991
A Database Publication



LOADSA CHANCES TO WIN PRIZES!



We're off on the road to

PART 1

Why not write your own adventure game? This new series will show how easy it is.

Adventure!



Writing an adventure is fun. And not only for you. It can give your friends lots of pleasure when they play it later.

And there's another good thing about writing adventures. You don't have to write a completely new program each time you want to create a new game.

Over the next few months we're going to show you how to write a program that will work for lots of different games.

It's called an *Adventure Manager*.



We'll be building it up section by section. In programming terms each of these is called a *routine*.

There will be lines in the program that will make the game jump to these routines when they are needed. These are known as *calls* to the routine.

That's enough of the jargon for now. Let's start building the game!

When you play an adventure your computer needs to understand the words the players type in. Each game will use different words. The program must be able to deal with all of them.



So the first thing we have to do is give our computer a list of words we want it to understand. Type in Lines 4999 to 5500.

This is a simple word list. It includes direction words like

NORTH and **SOUTH** - all your adventures will need these.

The **DATA** at line 5500 looks different. It's called a *rogue value* and it marks the end of the list.

The program will be written so that your computer knows it has reached the end of the list when it comes across a star (*). That way you can add more words without having to tell the program each time you do so.



This is an example of an important rule of programming: The more work you do when you start, the less you'll have to do later.

Now type in Lines 999 to 2000. These set up lots of things your program will need later. It's called an *initialisation routine*.

Next you need the line that makes the program jump to the routine - the *call*. That's line 10. Type it in now.

Before you **RUN** the program you should also type in Line 990. That stops the program when all the calls to the routines have been carried out.

Now you're ready to **SAVE** and **RUN** the program. Try it.

Nothing will seem to happen yet. What you have done is give your computer a list of words, each with its own space number.

You'll notice that in the **DATA** we've only used the first four letters of each word. This is usually

enough for the computer to recognise what the word is. But make sure you don't confuse your machine by using similar words like **LIST** and **LISTEN**!

All the words are directions. Look at Line 5010. You can see that **EAST** and **E** have both been given the same number - 2. This means the computer will treat each of them as the same word.

So it doesn't matter whether a player types in **EAST** or **E**. The computer will look up the figure 2 in its memory and take him in the same direction.



You play an adventure game by typing in instructions or directions. We'll now add some lines that print *What next?* This invites the player to tell the computer what he wants to do.

Type Lines 2009 to 2499. You also need the call to them, so add Line 20.

You'll want the program to show you the word numbers it's found, so add Line 100. *We'll take it out when we add the routines that use descriptions instead of the room numbers.*

SAVE and **RUN** the program again. It will now sit and wait for you to type something in. Try typing anything.



If you type a word that's not in the data two zeros will be printed. If you type something it understands the word or words will be changed into numbers. Try typing **North E**. See what happens?

All this may seem a complicated way of recognising a few words. You may think you could write a five line

What happens when you type something in?

When you answer the *What next?* question your reply is stored in **Y\$**. When you've **RUN** the program try entering: **PRINT Y\$**

You'll find your answer stored there. Now get your computer to check if any of the words are in your list.

Line 30 calls Lines 2500 to 2699 and they do this job.

The **REMs** in some of the lines are there to help you

understand what is going on. But don't worry if you don't quite get the hang of how it works. It will all come clear to you shortly.

REMark

REMs are put in programs to make them more understandable to humans. Your computer ignores them completely!

In this program some of the REMs are at the end of lines. Like this:

```
10 GOSUB 1000:REM SET UP START
```

You don't need to type in :REM or the following words.

Other REMs are alone on a line. For example:

```
999 REM SORT THINGS OUT BEFORE YOU START
```

These lines can be missed out entirely. But it's best not to. They could help you find things later.

program to do the same job.

But our program is very friendly. Any of the following will produce the same result:

```
N
NORTH
GO NORTH
RUN NORTH
PLEASE MAY I GO TO THE NORTH
```

Now you have a program that will look for ANY word you put in the data list. Try adding some of your own.

Just add a line between 5050 and 5500. Such as:

```
5060 DATA 10,HELLO
```

Look what happens when you run the program and type HELLO FROG. You will get:

```
10 0
```

That's because Hello has been given the number 10 and FROG is not in the list. It gets a 0.

Your program can now recognise direction words. But it's not an adventure yet. Your players will need somewhere to go - and exciting things to do.

Next month we'll add the lines that put places in your program. Then you'll be well on the way to seeing your own adventure come to life.

A CHALLENGE

So far your players will only be able to type what they want in capital letters.

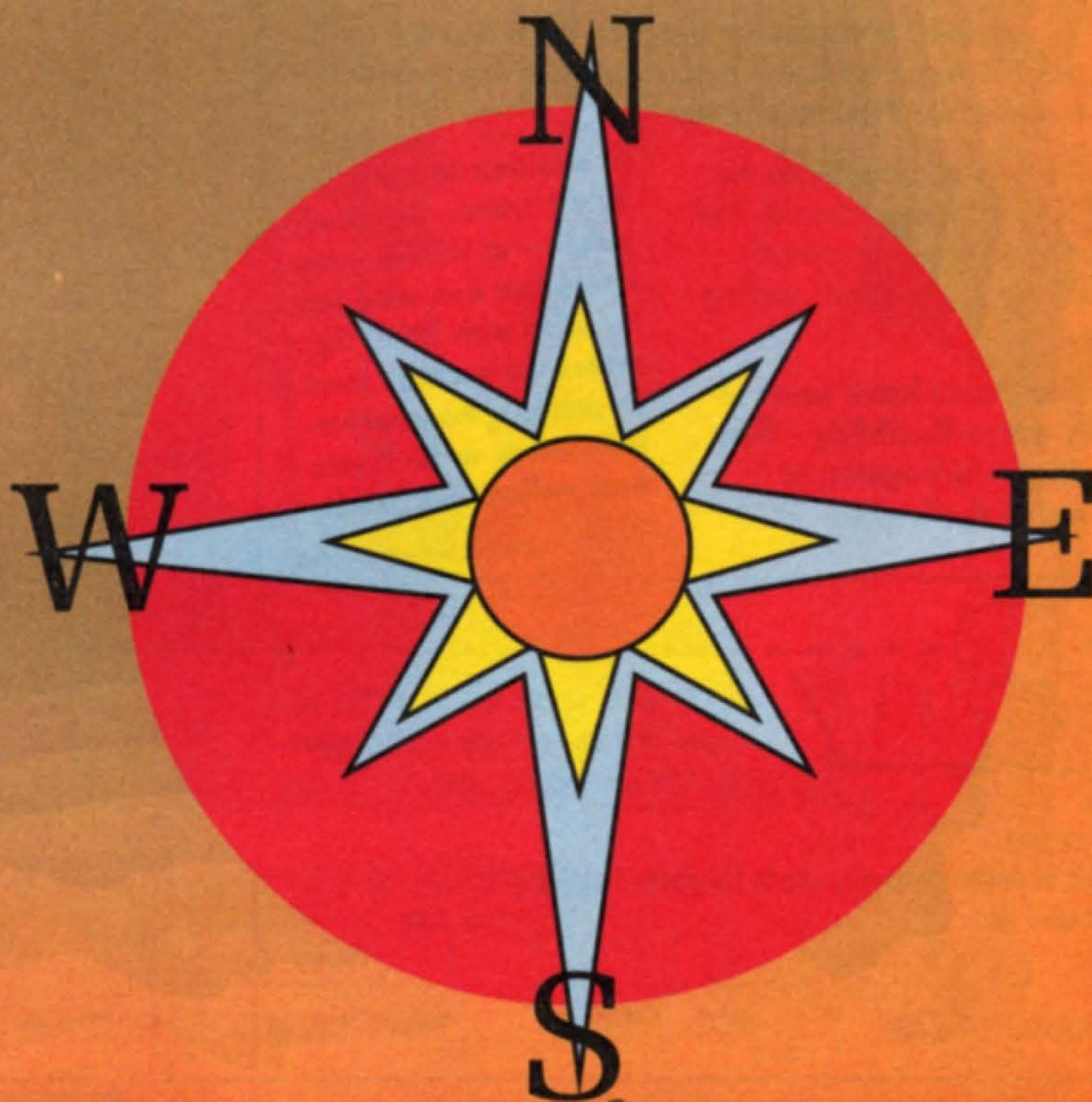
So while you wait for next month's *Let's Compute!* here's a challenge. See if you can make the program work for small letters as well.

IS THIS YOUR COMPUTER?

This program works on a BBC, Archimedes, Electron, CPC, ST(Stos), Amiga and PC(GW-Basic). It will not work on a C64/128 or Spectrum.

```
10 GOSUB 1000:REM SET UP START
20 GOSUB 2010:REM DISPLAY
30 GOSUB 2510:REM GET WORD NUMBERS
100 PRINT"THE WORD NUMBERS WERE ";W(1);
" ";W(2)
990 STOP
998 REM
999 REM SORT THINGS OUT BEFORE YOU STA
RT
1000 LET W=0:REM THIS WILL BE THE NUMB
ER OF WORDS
1005 DIM W(2):REM WORD NUMBERS FOUND G
O HERE
1010 RESTORE 5000
1020 READ X,X$:REM THESE VARIABLES ARE
JUST FOR COUNTING
1030 IF X>0 THEN LET W=W+1:GOTO 1020:R
EM NOT THE END OF THE LIST?
1040 DIM WRDNUM(W):DIM WRD$(W):REM SET
UP ARRAYS TO PUT WORDS IN
1045 RESTORE 5000:REM GO BACK TO THE S
TART OF THE LIST
1050 FOR X=1 TO W:READ WRDNUM(X),WRD$(
X):NEXT X
2000 RETURN:REM-FINISHED SORTING THING
S OUT
2005 REM
2009 REM DISPLAY THINGS
2010 REM BITS GO IN HERE LATER
2200 PRINT "What next ";:INPUT Y$:LET
Y$=Y$+" "
2499 RETURN
2500 REM GET WORD NUMBERS
2510 LET W(1)=0:LET W(2)=0:LET P=0:LET
WHICH=1:REM THE WORD NUMBERS GO IN HERE
, P FOR POINTER, WHICH IS FIRST OR SECON
```

```
D WORD FOUND
2520 LET P=P+1:IF P>LEN(Y$) THEN RETUR
N:REM HAVE WE FINISHED?
2530 IF MID$(Y$,P,1)=" " THEN GOTO 252
0:REM FIND THE NEXT WORD
2540 LET W$="":REM NO WORD YET
2550 IF MID$(Y$,P,1)=" " THEN GOTO 260
0:REM END OF THE WORD?
2560 LET W$=W$+MID$(Y$,P,1):IF LEN(W$)
=4 THEN GOTO 2600:REM 4 LETTERS YET?
2570 LET P=P+1:GOTO 2550:REM GET THE N
EXT LETTER
2599 REM NOW SEE IF THE WORD MATCHES
2600 FOR X = 1 TO W
2610 IF WRD$(X)=W$ THEN W(WHICH)=WRDNU
M(X):LET WHICH=WHICH+1:REM FOUND MATCH -
SAVE IT
2620 NEXT X
2630 IF WHICH=3 THEN RETURN:REM STOP I
F YOU'VE GOT 2 WORDS
2640 IF MID$(Y$,P,1)<>" " THEN LET P=P
+1:GOTO 2640:REM IGNORE THE REST OF THAT
WORD
2650 GOTO 2520:REM GET THE NEXT WORD
2699 RETURN
4900 REM
4999 REM START DATA HERE ----- WORDS
FIRST
5000 DATA 1,"NORT",1,"N"
5010 DATA 2,"EAST",2,"E"
5020 DATA 3,"SOUT",3,"S"
5030 DATA 4,"WEST",4,"W"
5040 DATA 5,"UP",5,"U"
5050 DATA 6,"DOWN",6,"D"
5500 DATA 0,"X"
```





No.2

What you'll need

- ★ A pram wheel.
- ★ A piece of wood (about 4cm by 4cm by 1m)
- ★ A screw to fix the wheel to the wood.
- ★ A piece of stiff card about 4cm by 10cm.
- ★ Two drawing pins
- ★ A computer

HOW TO USE IT

1. Place the pedometer at the point where you want to start measuring.
2. Walk to the point where you want to measure up to, pushing the pedometer in front of you. Count the clicks as you go.
3. Feed the number of clicks into your computer to find the distance.

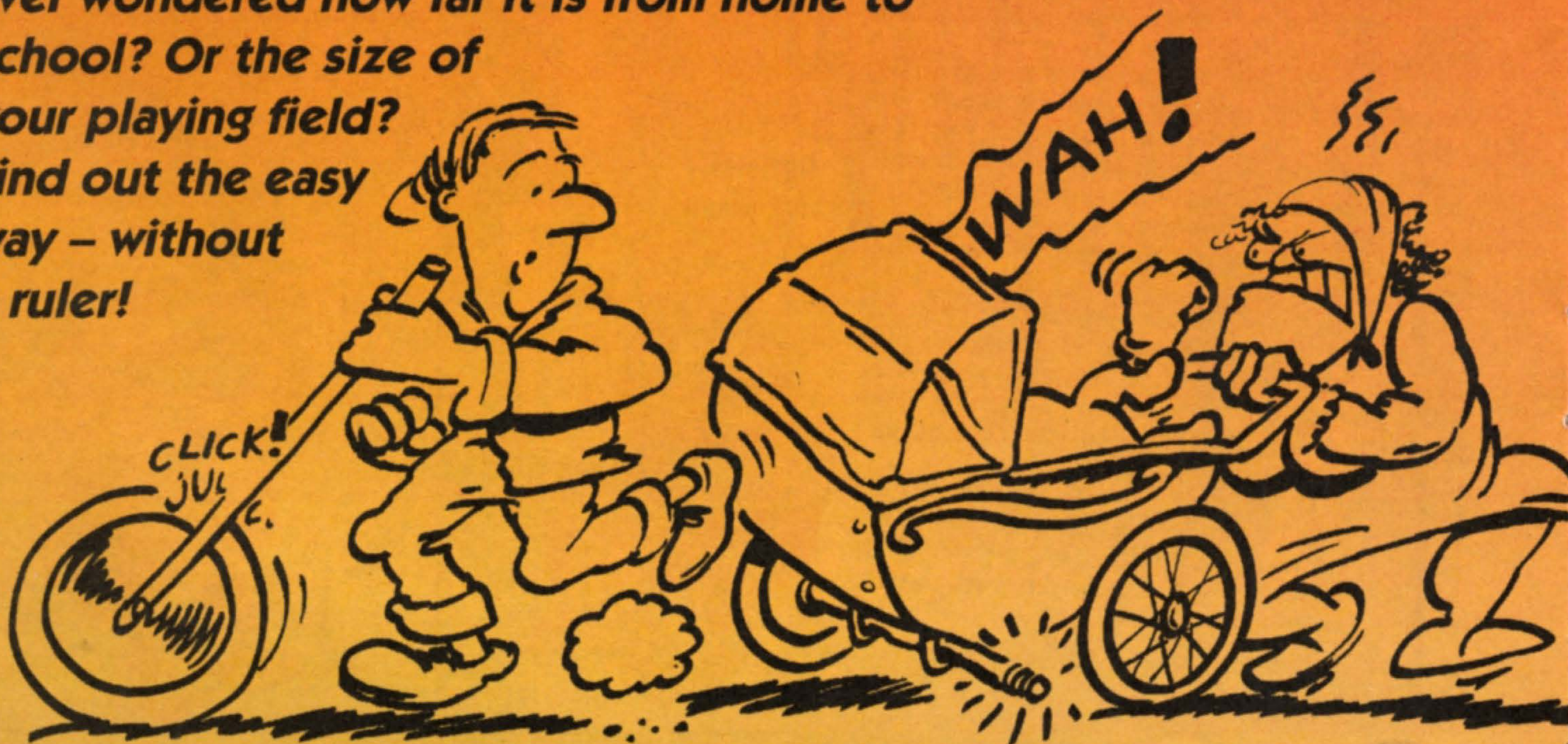
So you want to measure a field but you haven't a ruler long enough! Not to worry. All you need is your computer - and the *Let's Compute* pedometer.

What's a pedometer? Just a wheel on the end of a stick. What you want to find out is the distance the wheel has rolled. A real one uses things like gears and complicated electronics. These work out the distance automatically.

On your simple one, you'll hear a click each time the wheel goes round. All you have to do is count the number of clicks. Then give that figure to your computer and it will do the rest.

Let's make a PEDOMETER

Ever wondered how far it is from home to school? Or the size of your playing field? Find out the easy way - without a ruler!



AT YOUR COMPUTER

Type the program in, SAVE it and RUN it. You'll be asked two simple questions. Here's how you should answer them:

How big is the wheel?

Measure the distance across the wheel (the diameter). Type it into your computer. Make sure you use the same units as you want to measure the big distance in.

For example, suppose you want to measure a field in metres. If your wheel measures 28.4cm your answer to the computer's question would be .284.

How many clicks?

Just key in the number of clicks your pedometer made.

You'll then be told how far you pushed your pedometer. If you measured the wheel in metres the distance is in metres. If you used feet, it's in feet.

And that's all there is to it.

You'll be asked if you want another go. You must answer with a capital Y for yes or N for no.

If you answer Y you won't be asked the first question again. The size of the wheel is the same all the time.

HOW TO MAKE IT

There are lots of ways you can make a pedometer. It depends on what odds and ends you have.

Here's one suggestion for how to make one. But you can always use your imagination to design your own.

Use a screw to fix a pram wheel to a piece of wood. All you then need is something to make a click each time it goes round.

One way to do this is to fold a small piece of card and fix it to the tyre. If it's a solid tyre you can use drawing pins to do it. But that isn't such a good idea if the tyre is full of air!

Remember, all you have to make sure is that every time the wheel goes round the card touches the wood. That makes the click.

That's one way to make a pedometer. Now make your own and start measuring.

If you use it to measure anything unusual write to *Let's Compute!* and tell us. If you include a photograph of yourself using your pedometer you might even see it pinned to the noticeboard.

HOW THE PROGRAM WORKS

Line 40 works out the distance round the wheel (the circumference). It uses the well known formula:

$$\text{CIRCUMFERENCE} = \pi \times \text{DIAMETER}$$

Line 50 multiplies this by the number of clicks to give the total distance.

Line 60 converts this to the nearest whole number. The method used was described by the Program Doctor last month.

```
10 CLS
20 INPUT "HOW BIG IS YOUR WHEEL";D
30 INPUT "HOW MANY CLICKS";N
40 LET C=3.14*D
50 LET T=C*N
60 LET T=INT(T+0.5)
70 PRINT:PRINT
80 PRINT "IT MEASURED ";T
90 PRINT:PRINT
100 INPUT "ANOTHER GO (Y/N)";QS
110 IF QS="Y" THEN PRINT:PRINT:GOTO 30
120 IF QS<>"N" THEN GOTO 100
```

IS THIS YOUR COMPUTER?

Electron/BBC/Amiga/Archimedes/Spectrum/PC(GW-Basic)

The program works as shown.

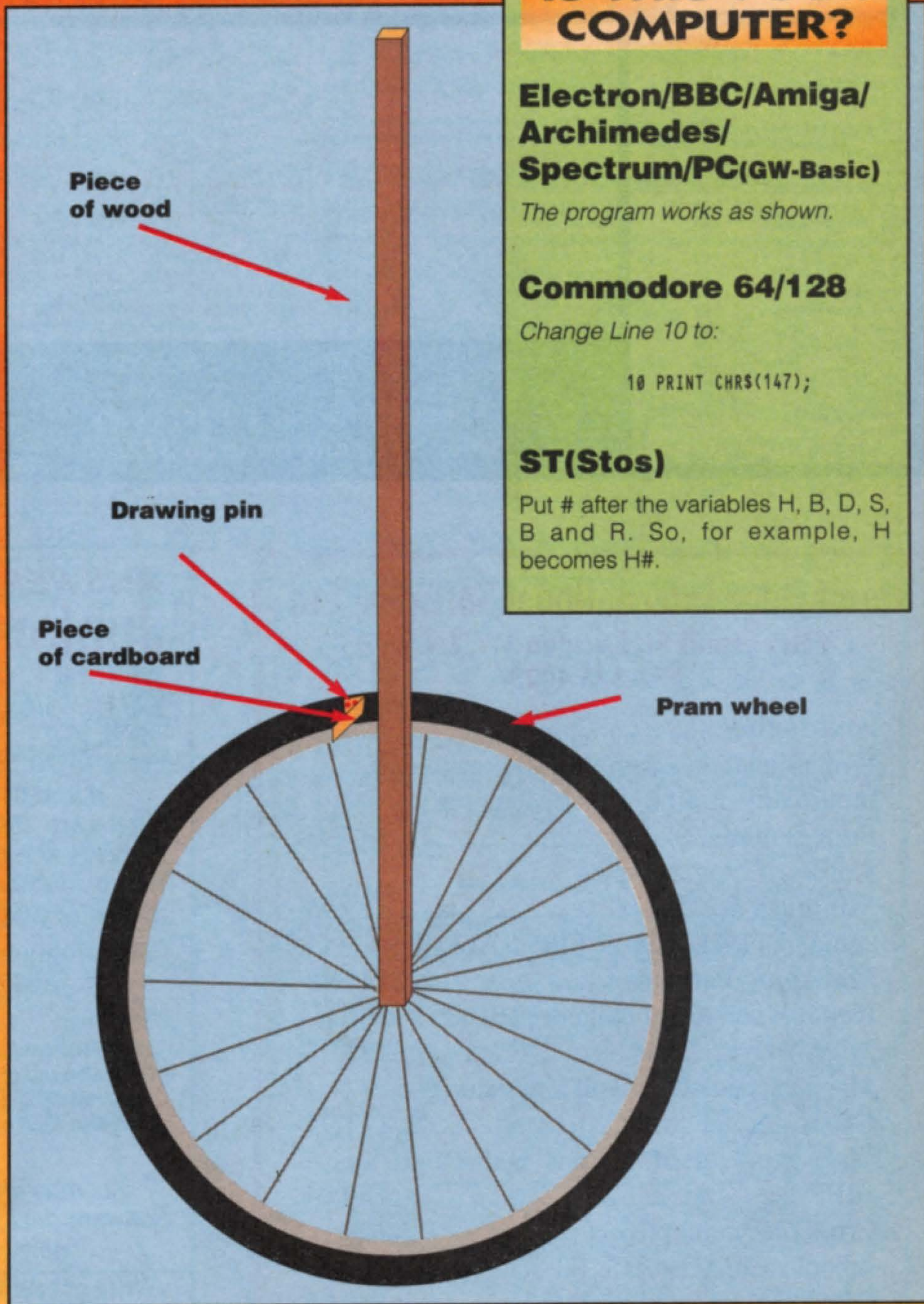
Commodore 64/128

Change Line 10 to:

```
10 PRINT CHR$(147);
```

ST(Stos)

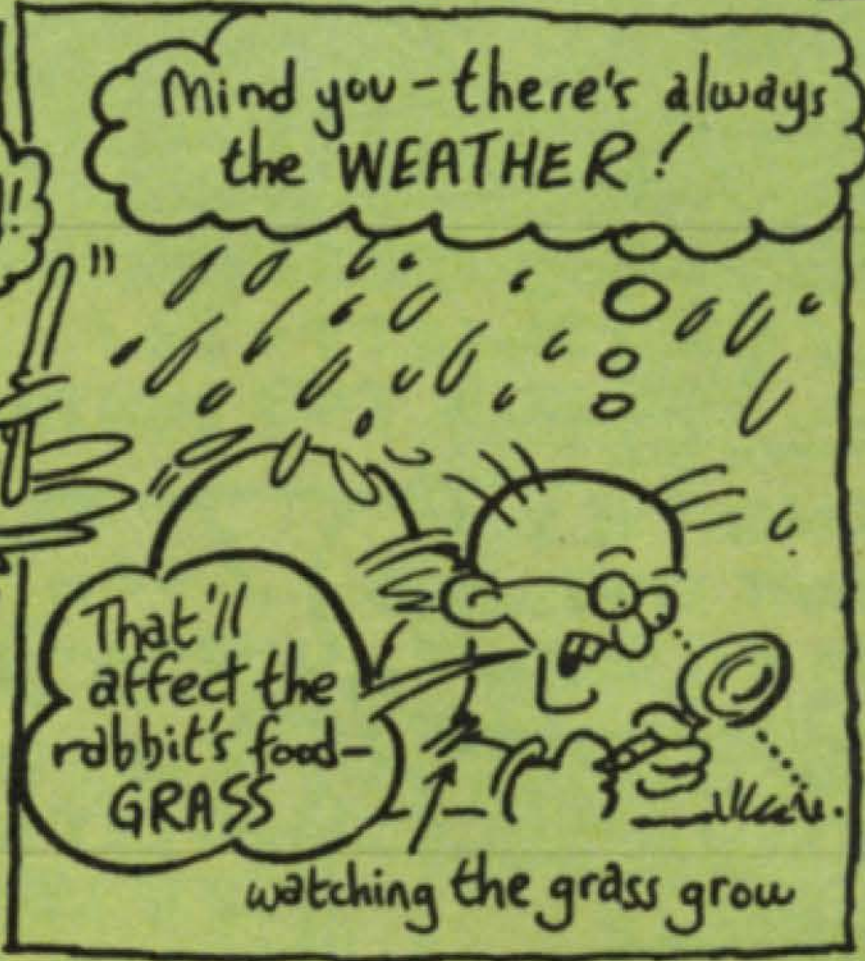
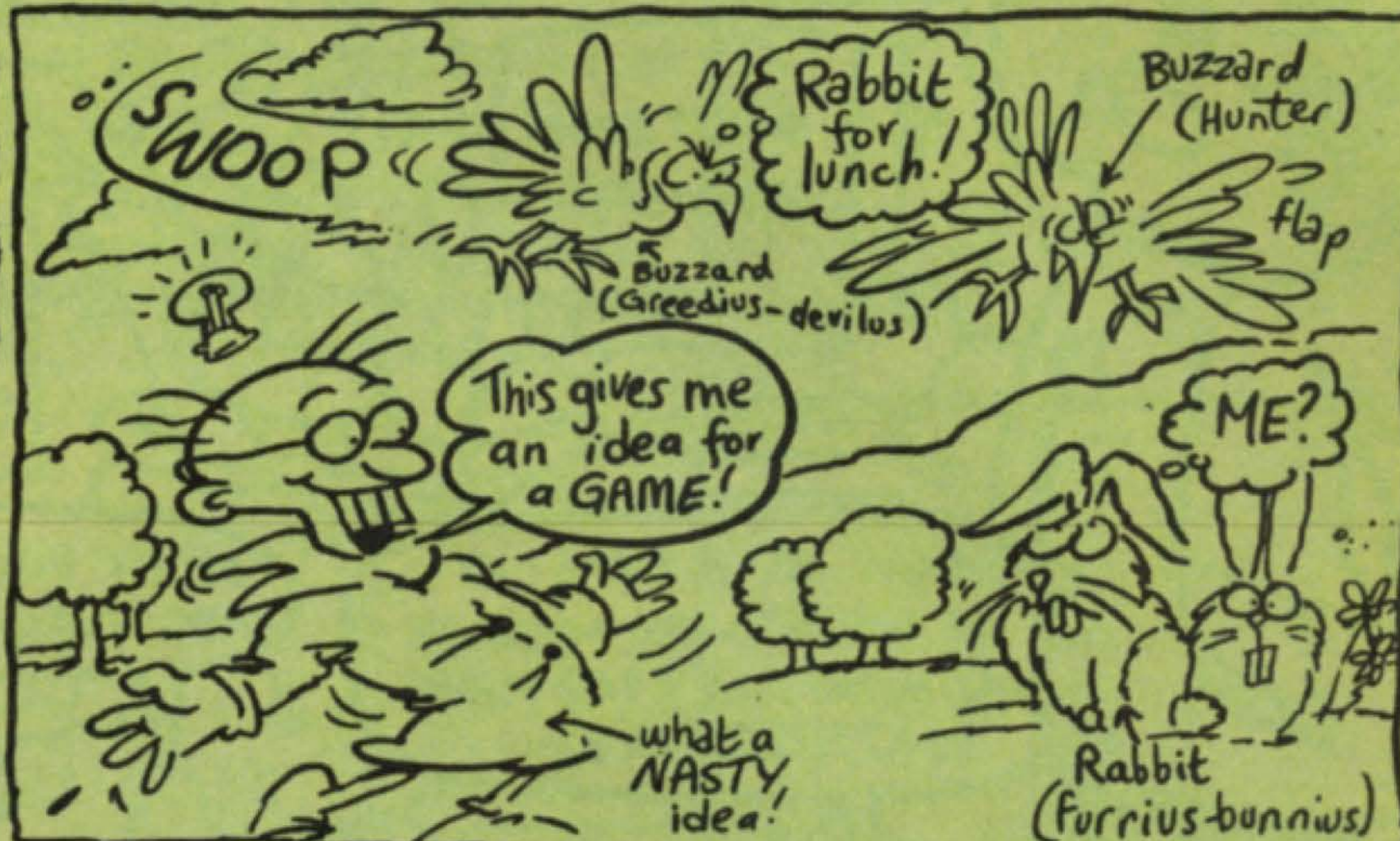
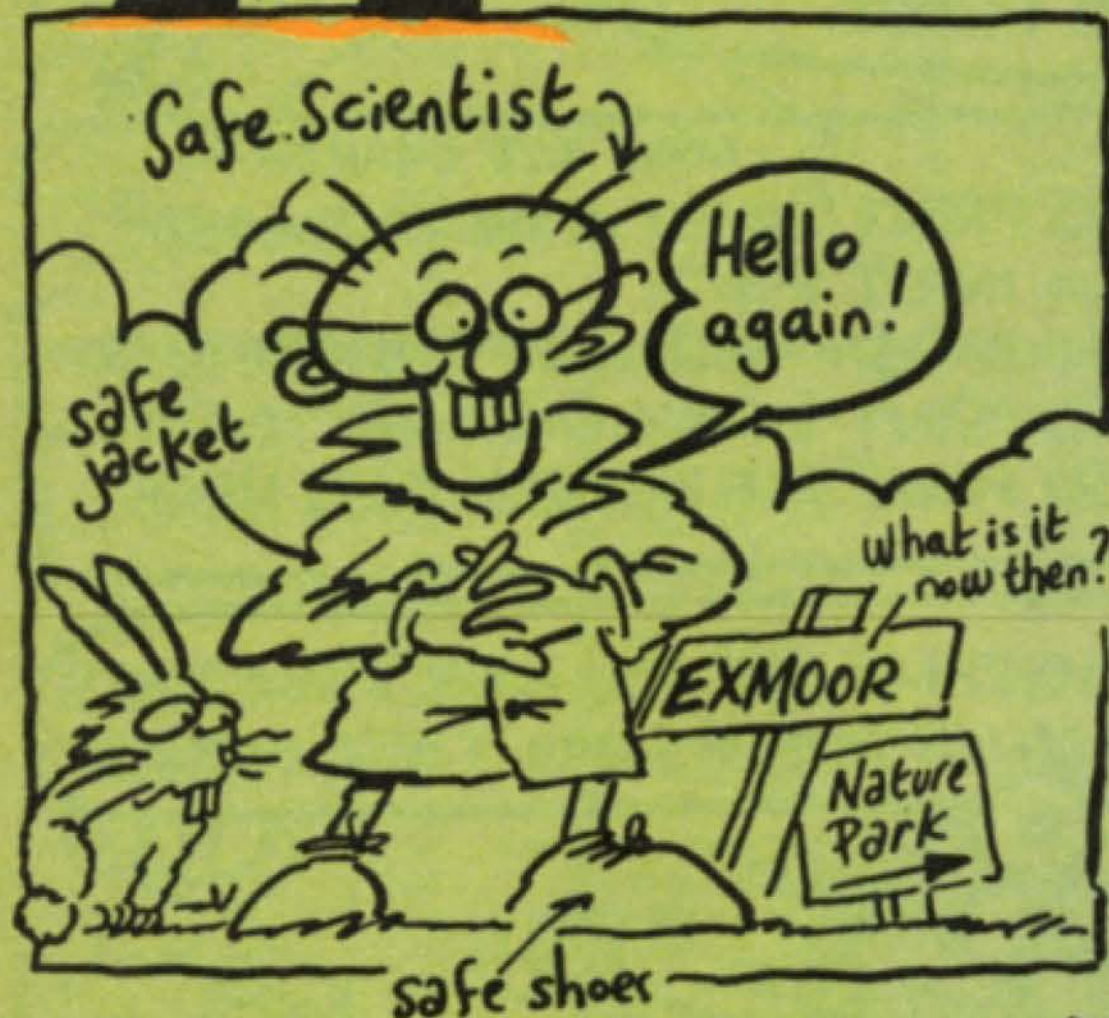
Put # after the variables H, B, D, S, B and R. So, for example, H becomes H#.



NEXT MONTH: Check your reflexes with Peter's super reaction tester



The Safe



Scientist

Nature IN THE Balance



The Safe Scientist looks at animal problems in the park

Here's a game that's based on a small park in Exmoor. It contains a few pairs of buzzard along with a supply of their favourite food – rabbits.

You play the part of the warden. You must try to maintain a balance between the hunters and their prey.

A lot of the problem is out of your control. For example, the growth of the rabbit's food – grass – depends on the weather.

You run the park for four years. Each season you can decide whether to cull rabbits or introduce some more.

This may seem harsh. But if there are too many of our furry friends they will eat all the grass. Then they'll starve! That means the buzzard population will also suffer. In any case, selling the rabbits can help to pay your wages.

The Nature Park Problem makes use of what's called a mathematical model. All the possible events are simplified to basic equations in your computer.

For example, in your model each pair of buzzards will always produce three young birds. Their survival depends on the weather and how much food is available.

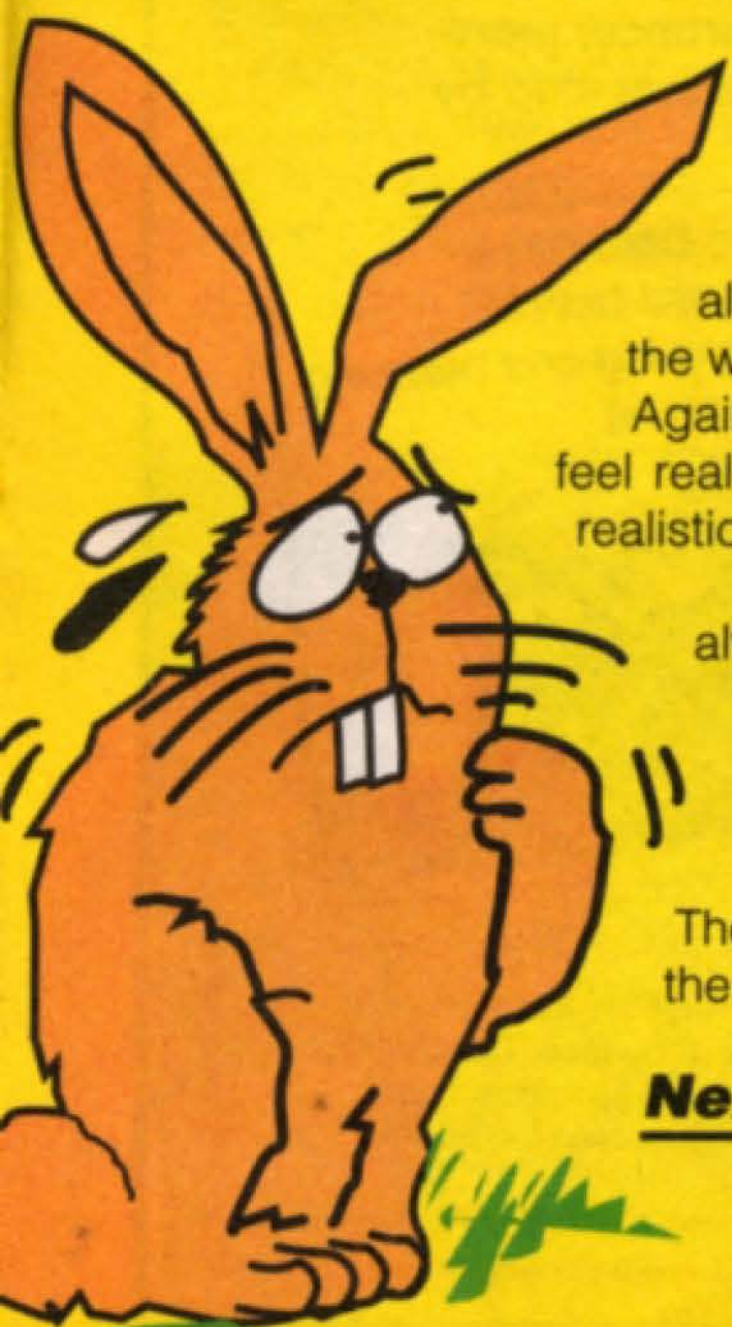
Again, your computer works this out using a simple sum. If you feel really brainy you could refine these sums and produce a more realistic picture of life in the park.

As ever, the way you use the program is up to you. But, as always in science, the idea is to investigate. Questions like:

What happens if you do nothing? Can you keep up a healthy buzzard population if you do intervene every season?

You could keep your own records of weather conditions. Then use them to try to work out how to manage the rabbits for the best effect.

**Next month we'll be putting the brakes on.
Can you stop a car before it crashes?**



NATIONAL CURRICULUM AREAS COVERED

What's this program got to do with science? And is it any help with the National Curriculum?

Well, one whole attainment target is about *Human Influences on the Earth*.

Also, the *Variety of Life* topic wants you to understand predator/prey relationships. You also need to know about pyramids of numbers and biomass.

The program covers all these areas. It also encourages you to investigate. And that's something every good scientist has to do!

**TURN OVER THE PAGE
FOR THE PROGRAM**

The Safe Scientist

```

10 REM NATURE PARK CHALLENGE
20 DIM S$(4):DIM G$(5)
30 DEF FNR(R)=RND(R)
40 CLS:FOR X=1 TO 4:READ S$(X):NEXT X
:FOR X=1 TO 5:READ G$(X):NEXT X
50 LET B=3:LET R=50:LET G=3
60 FOR Y=1991 TO 1994
70 FOR S=1 TO 4
80 NR=0
90 LET SS=S$(S)
100 LET W=FNR(6):LET G=W-S
110 IF G<1 THEN LET G=1
120 RESTORE
130 LET GS=G$(G)
140 GOSUB 410
150 IF S=3 THEN LET R=R/(6-W/3)
160 IF FNR(5)=1 THEN LET R=R+10
170 IF R<1 THEN LET R=1
180 IF S=3 THEN LET B=B+NB/2
190 IF S=3 THEN LET B=B-20*B/R
200 IF B<0 THEN LET B=0
210 IF FNR(5)=1 THEN LET B=B+1
220 IF S=4 THEN LET R=R/(6-W/2)
230 GOSUB 320
240 NEXT S
250 NEXT Y
260 CLS
270 PRINT "You have managed the nature
park for four years."
280 PRINT:PRINT "You now have a buzzard
population of ";INT(B);" pairs."
290 STOP

```

```

300 DATA "SPRING","SUMMER","AUTUMN","W
INTER"
310 DATA "VERY BAD","POOR","FAIR","GOO
D","FIRST RATE"
320 CLS
330 PRINT "How many rabbits will you cu
ll";
340 INPUT C
350 LET R0=(R*2)-C:LET R=R0/2
360 PRINT "How many new rabbits will yo
u release";
370 INPUT C
380 LET R=R+C/2
390 CLS
400 RETURN
410 PRINT "The year is ";Y;". It is ";
SS
420 PRINT:PRINT "There are ";INT(B);" p
airs of buzzard"
430 PRINT:PRINT "There are ";INT(R);" p
airs of rabbit"
440 PRINT:PRINT "The grass is in a ";GS
;" condition"
450 PRINT:IF S<>2 THEN GOTO 480
460 LET NB=B*3:LET NR=R*7:LET R=R+NR/2
:PRINT
470 PRINT "There are ";INT(NB);" new bu
zzards and ";INT(NR);" new rabbits."
480 PRINT "PRESS SPACE"
490 IF INKEY$(0)<>" " THEN GOTO 490
500 CLS
510 RETURN

```

IS THIS YOUR COMPUTER?

BBC/Archimedes/Electron

The program works as shown

Spectrum/CPC/ST(Stos)/ Amiga(Amiga Basic)/ PC(GW-Basic)

Change these lines:

```

30 DEF FNR(R)=INT(RND*R+1)
490 IF INKEY$<>" " THEN GOTO 490

```

C64/128

Use capital letters throughout
Replace:

CLS

with:

PRINT CHR\$(147);

in the listing, change these lines:

```

30 DEF FNR(R)=INT(RND(0)*R+1)
490 GET AS:IF AS<>" " THEN GOTO 490

```



rite On!

The word processor
that grows up
with your children

A revolutionary new word processor from SPA – **in full colour!** You can choose exactly how you want WriteOn to appear. Beginners can use a very simple screen, with large letters and an uncluttered appearance; more experienced users can have access to WriteOn's advanced features step by step. And you can include PaintSpa or NewSPAper pictures in your writing.

- Full What You See Is What You Get word processing
- Simple, uncluttered screen
- Choice of large, easy-to-read typefaces
- Easy PaintSpa-like interface

- Type in colour or black and white
- Supports many colour and black and white printers
- Include PaintSpa or NewSPAper pictures
- 'Minipage' window on the text

More advanced users can use more versatile features – but WriteOn is designed so that younger children don't need to be worried by features like these. You can easily tailor WriteOn suit anyone's needs.

SPA

Nimbus or IBM £44.85 (state which)
(IBM requires EGA or VGA graphics)
**PO Box 59, Leamington Spa,
Warwickshire CV31 3QA**
Telephone: (0926) 422959



There has been a need for some time for a modern, mainstream primary word processor which is able to take children from simple beginnings and introduce them gradually to a wider range of word processing features. WriteOn! fills the gap perfectly.

T.E.S.

LET'S MAKE MONEY

Part 4

Ready, steady - GO!

Raise funds with the Let's Compute! car racing game

Bingo, raffles, wheel of fortune and lots of other fund raising events have one thing in common. They all involve an element of chance.

The idea behind them all is simple. You collect money from lots of people, keep some and split the remainder between a few winners.

Here's a short computer program that can give a new look to that sort of event. It's a car racing game!

Type in the program and try it. You'll be asked how many cars you want. Answer

5 as a test. Then press the spacebar and you'll see the cars racing. At the end you'll be told which car won. Press the spacebar and off they go again.

So how do you use the program to make money?

All you have to do is charge people to guess which car will win. Once a few people have paid run the program to find the winner.

Then keep, say, a quarter of the money you took. Split the remainder between the ones who guessed correctly.

WRITE YOUR OWN PROGRAM

This program is intended as a frame on which you can base your own version. It can easily be changed. For example, you could improve the graphics and add a finish line. Or use sound to add more atmosphere to the race. To get you started, here are some changes that you can easily try. First, you can have more cars racing.

At the moment the program is set up for eight. That means there's room to fit the cars on all computer screens. But you can usually fit more. Just change the 8 in Line 10 to fit as many cars on your own screen as you can.

You can, in fact, easily fit more cars on your screen by moving them closer together. Remove the *2 in Line 80 to do that.

If you only want a few cars racing change the *2 to *3 to move them further apart. Now, with more space between the cars there's room to fit lane markings as well!

If you make any changes to the program we'd love to see them.

```
10 INPUT "HOW MANY CARS";N:IF N>8 THE
N GOTO 10
20 DIM C$(N):DIM P(N)
30 FOR L = 1 TO N:LET C$(L)=" "+STR$(
L)+">":LET P(L)=2:NEXT L:LET R=1
40 PRINT:PRINT"PRESS SPACE TO START T
HE NEXT RACE"
50 REPEAT UNTIL GET=32
60 CLS
70 FOR L=1 TO N
80 PRINTTAB(P(L),L*2);C$(L)
90 NEXT L
100 IF P(R)>30 THEN GOTO 140
110 LET R=RND(N)
120 LET P(R)=P(R)+1
130 GOTO 70
140 PRINT:PRINT "NUMBER ";R;" WON"
150 GOTO 30
```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown

Commodore 64/128

Change these lines:

```
50 GET AS:IF AS <> " " THEN GOTO 50
60 PRINT CHR$(147);
80 POKE 211,P(L):POKE 214,L*2:SYS 587
32:PRINT C$(L)
110 LET R=INT(RND(0)*N)+1
```

Amstrad CPC

Change these lines:

```
50 IF INKEY$<>" " THEN GOTO 50
80 LOCATE P(L),L*2:PRINT C$(L)
110 LET R=INT(RND*N)+1
```

Amiga/PC(GW Basic)

Change these lines:

```
50 IF INKEY$<>" " THEN GOTO 50
80 LOCATE L*2,P(L):PRINT C$(L)
110 LET R=INT(RND*N)+1
```

Spectrum

Change these lines:

```
20 DIM C$(N,5):DIM P(N)
50 IF INKEY$<>" " THEN GOTO 50
80 PRINT AT L*2,P(L);C$(L)
110 LET R=INT(RND*N)+1
```

ST (Stos)

Change these lines:

```
50 IF INKEY$<>" " THEN GOTO 50
80 LOCATE P(L),L*2:PRINT C$(L)
110 LET R=INT(RND*N)+1
```


Part 6

CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1- 9)

SORT THE CARDS

Find the Facts:
Roland Waddilove adds
a search routine to the
Let's Compute! database

DON'T BE IMPATIENT!

"Your database program doesn't work", wrote one very annoyed Let's Compute! reader.

"I typed it in and tried to sort the cards into order and the program stopped with an error message".

Don't panic! Some features of the program won't work yet. So far we haven't typed them all in.

We are building up our database month by month and looking at how it works at every stage. Remember, Rome wasn't built in a day!

In this month's article you'll find the missing subroutine to sort the cards into alphabetical order.

You can already use the database that we've been creating over the last five months.

So far you can type information in, save it, load it and search for any facts you want.

But an important feature of any good database is still missing. Very often you will want to sort the cards into order.

This month we'll add the lines to do that.

Start by loading the database program that you've already built up. Then enter the new lines listed below. SAVE your new, improved database.

When you've done that it's time to

check whether the program works. Run it program and choose *write a card* from the menu.

As a test, type in a set of five dummy cards. Just use a few letter Gs on to the first card, Bs on the second, then Zs, As and Fs.

These are just letters chosen at random. Use others if you want.

Now select the sort option on the menu. Look at the cards afterwards and you should see they're in alphabetical order. If they're not, carefully check your typing.

Let's see how the new subroutine works.

The information you type on to the cards is stored in an array called *C\$()*. Each card has five lines of text so the first one is *C\$(1)*, *C\$(2)*, *C\$(3)*, *C\$(4)* and *C\$(5)*.

The second card is five lines after the first card and so starts at *C\$(6)*. Card three is at *C\$(11)* and so on.

The card sorting program

```
8000 REM ===== Sort cards =====
8010 CLS
8020 PRINT "Sorting the cards..."
8030 FOR X=6 TO 246 STEP 5
8040 FOR Y=251 TO X STEP -5
8050 IF C$(Y)>C$(Y-5) THEN GOTO 8110
8060 TS=C$(Y+0):C$(Y+0)=C$(Y-5):C$(Y-
5)=TS
8070 TS=C$(Y+1):C$(Y+1)=C$(Y-4):C$(Y-
4)=TS
8080 TS=C$(Y+2):C$(Y+2)=C$(Y-3):C$(Y-
3)=TS
8090 TS=C$(Y+3):C$(Y+3)=C$(Y-2):C$(Y-
2)=TS
8100 TS=C$(Y+4):C$(Y+4)=C$(Y-1):C$(Y-
1)=TS
8110 NEXT Y
8120 NEXT X
8130 RETURN
```

**This
program
works on
all home
micros**

IS THIS YOUR COMPUTER?

**BBC/Electron/Archimedes/
Spectrum/Amiga/ST(Stos)/
CPC/PC (GW-Basic)**

The program works as shown .

C64/128

Change the following line:

8010 PRINT CHR\$(147);

NEXT MONTH
**You'll add a new
routine to print
the cards. Then
if you have a
printer, you'll be
able to produce
printed cards.**

LET'S COMPUTE!

99p

SHOOT'EM!
Fast-action
fun for two

WIN



DICK TRACY!

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

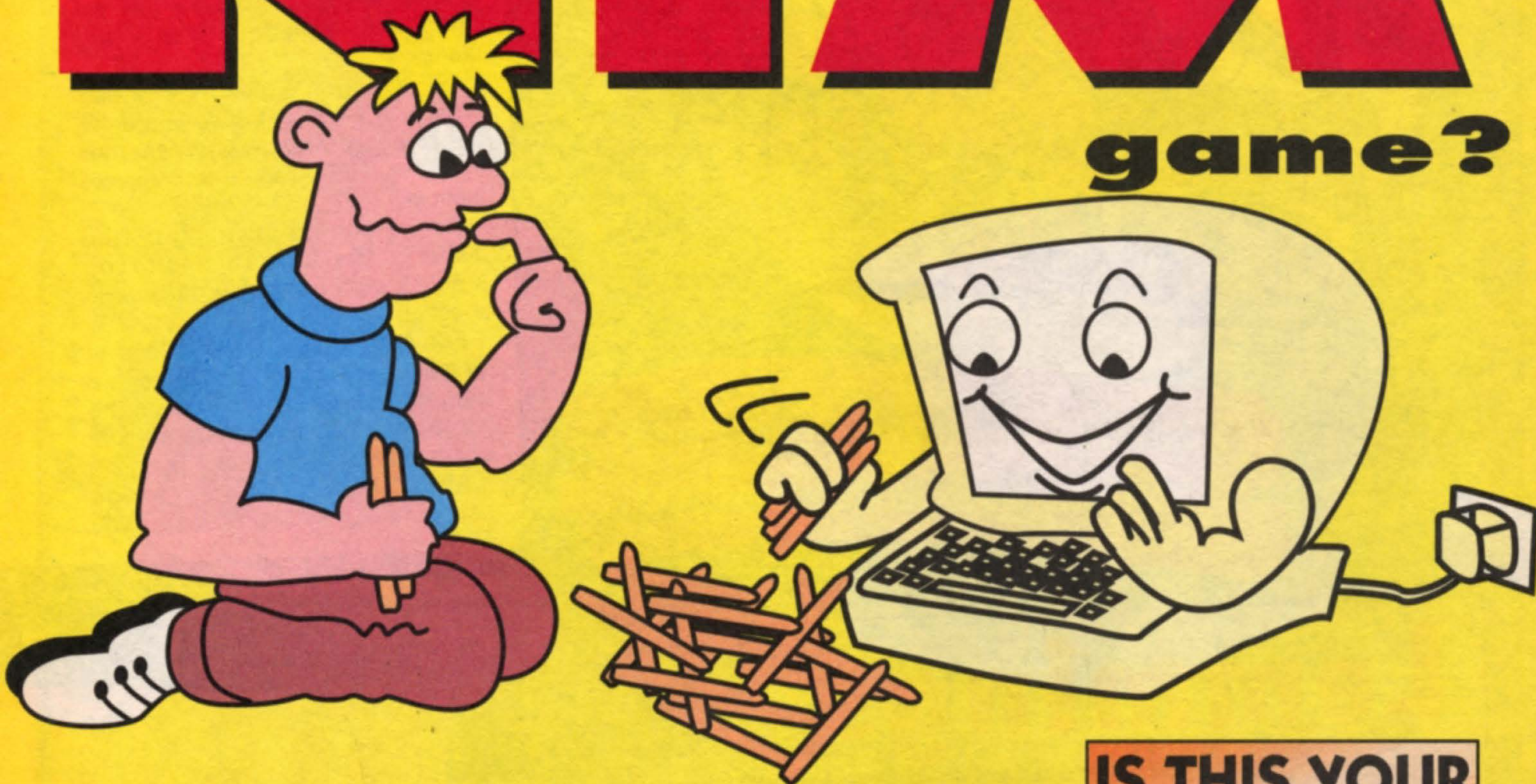
No 9 April 1991
A Database Publication



PARENTS & TEACHERS!
Turn to Page 6 for ways of developing
the National Curriculum content
of Let's Compute!

Can you beat your computer at the

NIM game?



Nim is an ancient two-player game – and one of the first ever to be played against a computer. Type in the short program below and **YOUR** computer becomes an ace Nim player.

The rules are simple: First you need a pile of small sticks. It doesn't matter how many. Any number from 10 to 1,000 is OK.

The two players then take it in turn to pick up sticks. Up to half the pile can be taken each turn.

The loser is the one who picks up the last stick. So, to win, you must force your opponent to take it.

Now let's try the same thing on your computer, using our program. You start with between 100 and 1,000 sticks. You will find your computer is an expert: It will always win if it goes first.

To give you a sporting chance, it lets you have the first move 8 times out of 10. Can you work out the way to win every time you start?

Once you've sussed it out you should win every time you start it's your turn to start.

```
10 CLS
20 N=RND(901)+99
30 PRINT "There are ";N;" in the pile"
40 IF N>900 THEN GOTO 90
50 PRINT:PRINT "How many will you take (upto half)";:INPUT M
60 IF M>N/2 OR M<1 THEN PRINT "That is more than half":GOTO 50
70 LET N=N-M
80 PRINT:PRINT "You have left ";N
90 LET X=1
100 LET X=X*2
110 IF X<=N THEN GOTO 100
120 LET X=X/2-1
130 IF 2*X<N THEN LET X=N-1
140 PRINT:PRINT "I take ";N-X;" leaving ";X
150 LET N=X
160 IF N>1 THEN GOTO 50
170 IF N=1 THEN PRINT:PRINT "I win!!":GOTO 190
180 PRINT:PRINT "I lose"
190 PRINT:INPUT "Play again (Y/N)?";AS
191 IF AS="Y" OR AS="y" THEN GOTO 10
192 STOP
```

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown.

Amiga (Amiga Basic)/CPC/
PC (GW-Basic)/Amstrad/
AtariST(Stos)/Spectrum

Change these lines:

```
20 LET N=INT(RND*901)+100
```

Commodore 64/128

Use capital letters.

Change these lines:

```
10 PRINT CHR$(147)
20 LET N=INT(RND(0)*901)+100
```

**This
program
works on
all home
micros**

IS THIS YOUR COMPUTER?

This program works on a BBC, Archimedes, Electron, CPC, ST(Stos), Amiga and PC(GW-Basic). It will not work on a C64/128 or Spectrum.

```

35 IF W(1)>=1 AND W(1)<=10 GOSUB 2710
:GOTO20:REM FOR MORE THAN 10 ROOMS CHANG
E THE 10 IN THIS LINE
990 GOTO 20
1060 ROOM=1:Y$="start the adventure":ME
SS1$="":MESS2$="":MESS3$=""
2010 RESTORE 5520:REM GO TO THE START O
F THE ROOM LIST
2020 FOR C=1 TO ROOM:READ DIR$,DESC$,CO
DESS:NEXT C:REM GET INFO FOR THAT ROOM
2025 CLS:PRINT"You are ";DESC$;".":REM
PRINT WHERE YOU ARE
2030 D$="":IF INSTR(DIR$,"N")>0 D$=D$+"
North "
2031 IF INSTR(DIR$,"E")>0 D$=D$+"East "
2032 IF INSTR(DIR$,"S")>0 D$=D$+"South
"
2033 IF INSTR(DIR$,"W")>0 D$=D$+"West "
2034 IF INSTR(DIR$,"U")>0 D$=D$+"Up "
2035 IF INSTR(DIR$,"D")>0 D$=D$+"Down":
REM D$ CONTAINS THE DIRECTION YOU CAN GO
2190 IF D$>"" THEN PRINT:PRINT"You can
go ";D$
2191 PRINT"*****
*****":REM 39 STARS
2192 PRINT:PRINT "You wanted to ";Y$:PR
INT:REM REMIND THE PLAYER WANT THEY WANT
ED TO DO
2193 PRINT MESS1$
2194 PRINT MESS2$
2195 PRINT MESS3$:PRINT:MESS1$="":MESS2
$="":MESS3$="":REM PRINT ANY MESSAGES AN
D CLEAR THEM
2700 REM TRY TO MOVE TO A NEW ROOM
2710 P=1:REM POINTER IN ROOM CODE
2720 IF VAL(MID$(CODESS,P,2))=W(1) THEN
ROOM=VAL(MID$(CODESS,P+2,2)):RETURN
2730 IF VAL(MID$(CODESS,P,2))=99 THEN M
ESS1$="You can't go that way!":RETURN
2740 P=P+4:GOTO2720
5510 REM ROOMS START HERE
5520 DATA S,in the control room,030399
5530 DATA E,in the weapons room,020399
5540 DATA NSEW,in a corridor,0101020403
05040299
5550 DATA W,in a store room,040399
5560 DATA NS,in a corridor,0103030799
5570 DATA ES,in the recreation room,020
7031099
5580 DATA NSEW,in a corridor,0105020803
11040699
5590 DATA ESW,in the air lock,020903120
40799
5600 DATA W,in the shuttle,040899
5610 DATA NES,in the sleeping quarters,
01060211031399
5620 DATA NSEW,in a corridor,0107021203
14041099
5630 DATA NSW,standing by a crate in th
e hold,01080315041199
5640 DATA NEW,crawling through an air du
ct,01100214042099
5650 DATA NSEW,in a corridor,0111021503
16041399
5660 DATA NW,in the hold,0112041499
5670 DATA N,in the engine room,011499
5680 DATA NEWS,crawling through an air d
uct,011802180319041999
5690 DATA NEWS,crawling through an air d
uct,011702130320041799
5700 DATA NEWS,crawling through an air d
uct,011702200320041799
5710 DATA NEWS,crawling through an air d
uct,011802130319041999
5900 DATA X,X,X

```

*Now your great
adventure really
starts to take
shape as you
design the maze*



Last month we showed you how to start writing your own Adventure Manager.

The program we gave you recognises words the player types in. It lets you put in any words you want your computer to understand.

The first six words we used were direction words. Now that your computer can recognise these it needs places to move to.

First you need to draw a map. Ours is for a spaceship. But you can draw your own for anywhere you want - your house, your school or your town for example.

Try ours first, then have a go with your own.



We'll call each place on the map a room. Even if it's really a cabin, a hold or a corridor. That way we can talk about, for instance, *Room 8*. You can even talk about your garden as being a room on your own map.

Once you've drawn your map you're ready to begin working on your computer.

Start by loading last month's program. You're going to add more lines to it.

Type 100 and press Return. This removes Line 100 - remember it was just there to let you know your program was working.



Now look at the map. This is what your computer needs to know. The instructions for each of the 20 rooms are in Lines 5510 to 5900. Add these to your program.

The panel on the right explains how these DATA lines are made up. Just follow the instructions there to make your own DATA lines.

Now add Line 1060. This tells the computer which room to

Mapp Ad

PART 2

Always draw your map first. Then work out the codes on paper. That way you will find it much easier to sort out any mistakes.

start in. It also gives it the message that needs to be printed at the start.



Lines 2010 and 2020 find the computer's description of the room from the DATA. Then Lines 2025 to 2195 display what's been read. There's also a bit that can send messages to the player. You will need these later.

Type these lines in and run the program. You'll be in the control room with this message showing: *You can go South*

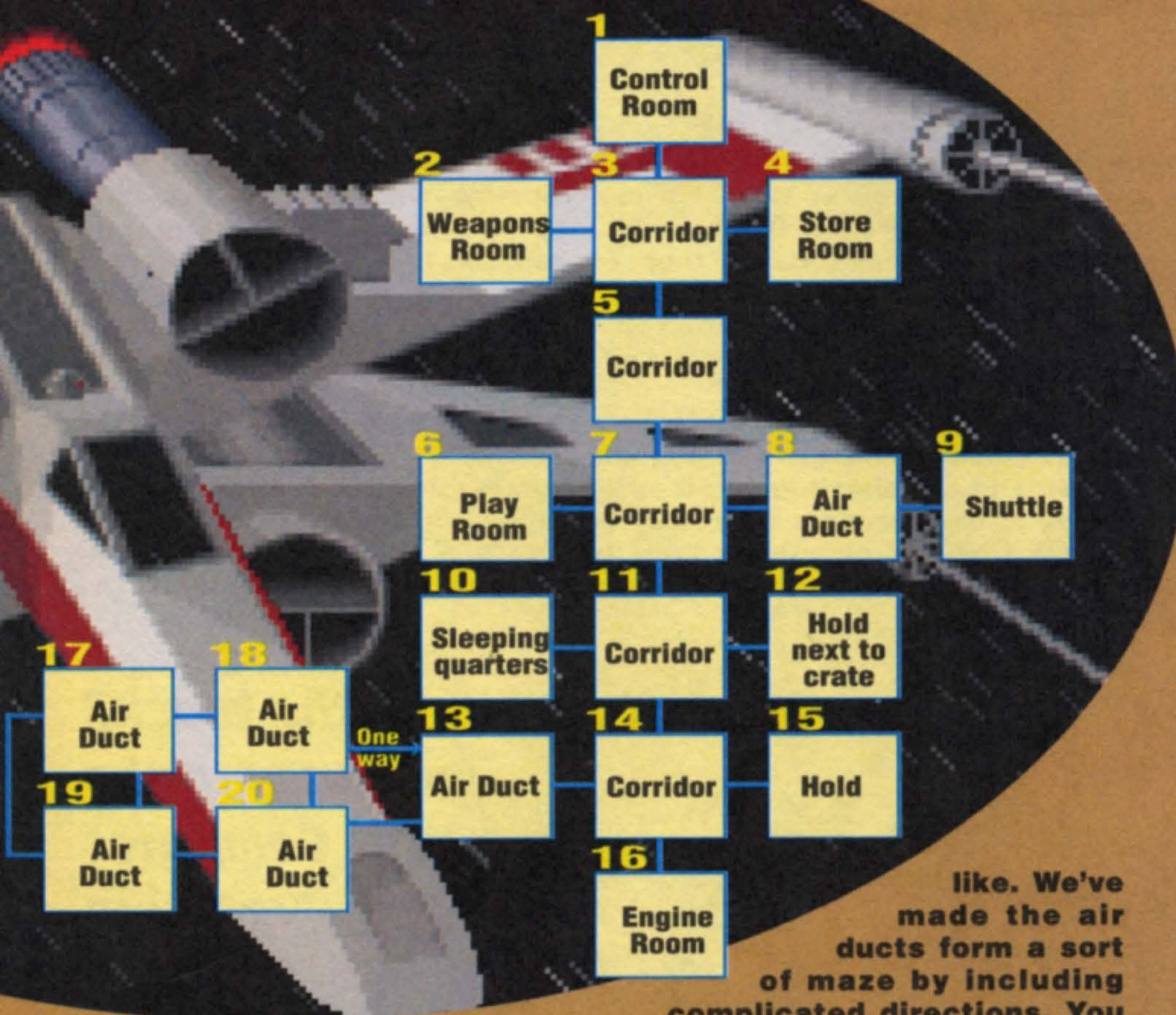


Now type in Lines 2700 to 2740. They check the first word typed in by the player to find which new room to move to.

If your computer doesn't know the word you won't be able to go that way.

Now we must make the computer use this routine. Line 35 checks to see if the first word found is a direction word. If it is

ing the road to venture!



it goes to the movement sub-routine.

Finally, replace the old line 990 with:

```
990 GOTO 20
```

This makes the whole program go in a complete loop so you can keep playing.

Now **SAVE** your complete program and try **RUN**ning it.



You will be able to move around the spaceship in the map. We haven't used any **UP** or **DOWN** to keep things simple. But they are in the program ready for you to use if you write your own version.

The connections between rooms can be as difficult as you

like. We've made the air ducts form a sort of maze by including complicated directions. You can make a few rooms seem like a lot by doing this sort of trick.

You can have one way paths like the one we've put between rooms 18 and 13.

You can have magic tunnels that join one end of your map to the other. You could make our spaceship into a circular space station by adding connections between rooms 1 and 16.

Now try using your own set of rooms. Start with something simple like your house or your school.

Then move on to making up your own adventure landscape.

Next month we'll look at how you can add things like space suits to your adventure.

What's in the DATA lines?

Each room needs three bits of data. As an example, look at Line 5590. This describes room number 8 – the Air duct. The program knows it is room 8 because it is the eighth room DATA line.

The first bit of data, ESW, shows the directions you can move in. The second bit, in the air lock, is the description of the room. Your computer will add *You are* to each description.

The next group of numbers tell the program which words send you where. Word 02 will move us to room 09, 03 to room 12 and 04 to room 07.

Each number must be two figures. So, for example, the number 2 must be written as 02 in the DATA.

The 99 at the end tells the program that's the end of the list.

TRY THIS!

You could make a maze seem to go on for ever by connecting a room to itself. Try changing the codes for room 19 to:

```
01190219031904051906191999
```

Also alter the directions to **NSEWUD** and see what happens when you enter that room.

ANSWER TO LAST MONTH'S CHALLENGE

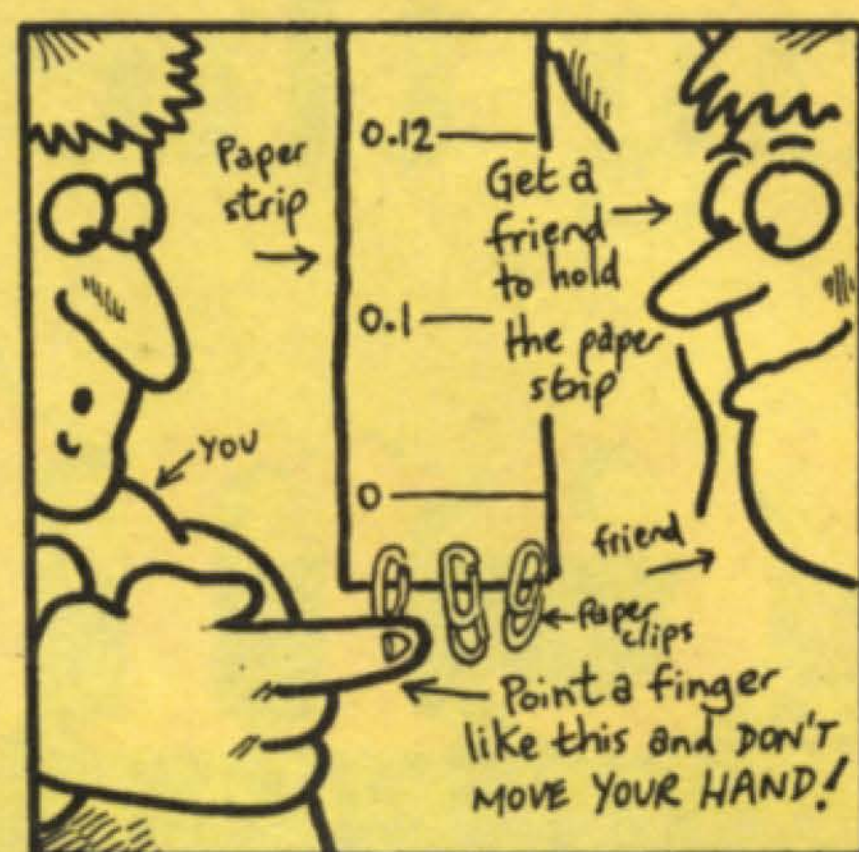
There are many ways to make your computer understand capital or lower case letters. Here's a way that will work on all computers:

```
2201 LET YYS="":FOR ZZ=1 TO LEN(Y$)
2202 LET LLS=MID$(Y$,ZZ,1):LET
LL=ASC(LLS)
2203 IF LL>91 THEN LET LLS=CHR$(LL-32)
2204 LET YYS=YYS+LLS
2205 NEXT ZZ
2206 LET YS=YYS
```




No.3

USING THE PAPER



Q

U

I

C

K

You get two for the price of one with this month's project. There's a paper reaction tester that you can take anywhere - and a special computerised version. Our step-by-step instructions tell you how to make and use them both. Try them out and see how fast **YOUR** reflexes are.

There are lots of experiments you can do with your two testers.

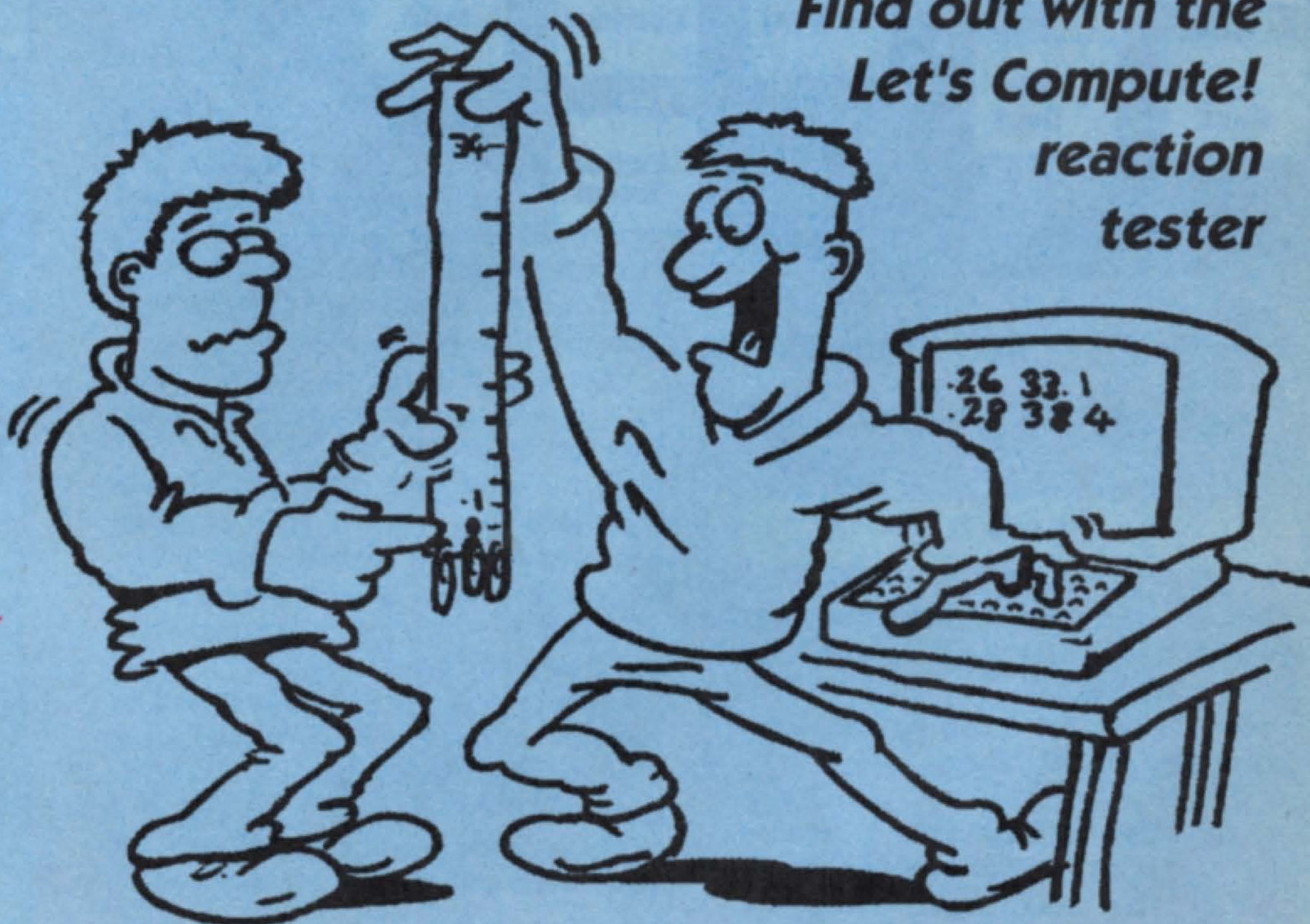
First, try comparing the results of the paper and computer versions. See if they're similar.

You could check which of your friends has the fastest reaction. Is the quickest also the best at zapping aliens?

Draw a graph showing the reaction times of all your friends. Draw another of the speeds achieved by the same number of adults. Who are fastest? Children or adults? We'd love to see your graphs and other results. Send them to *Let's Compute!* and we'll pin some to the Noticeboard.

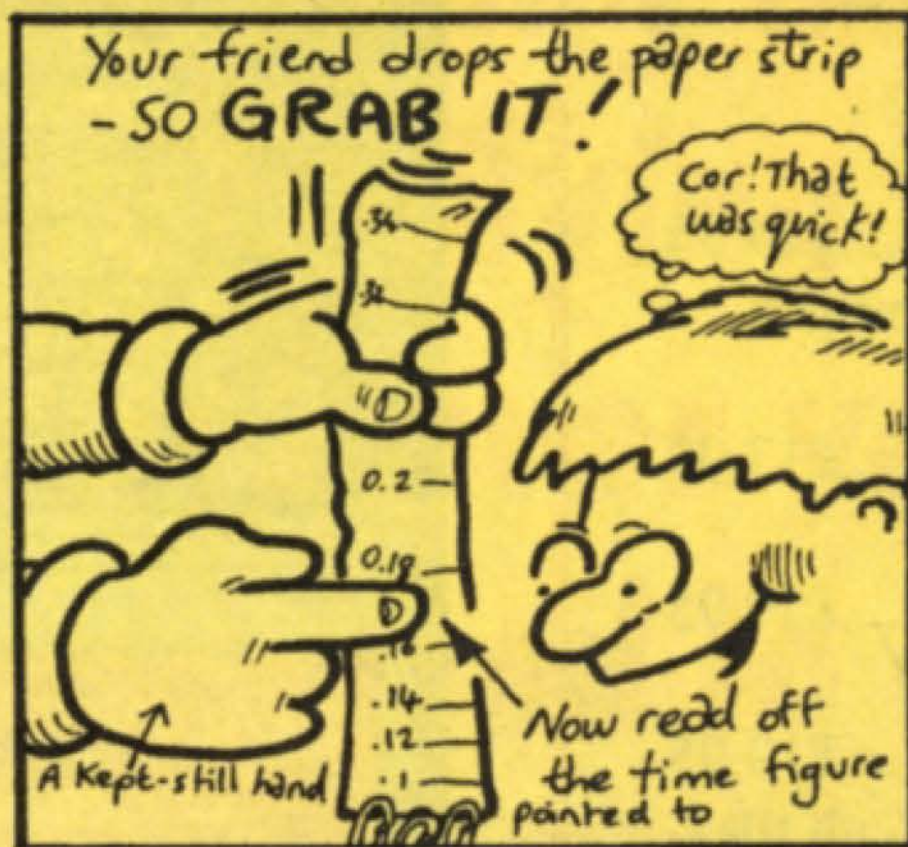
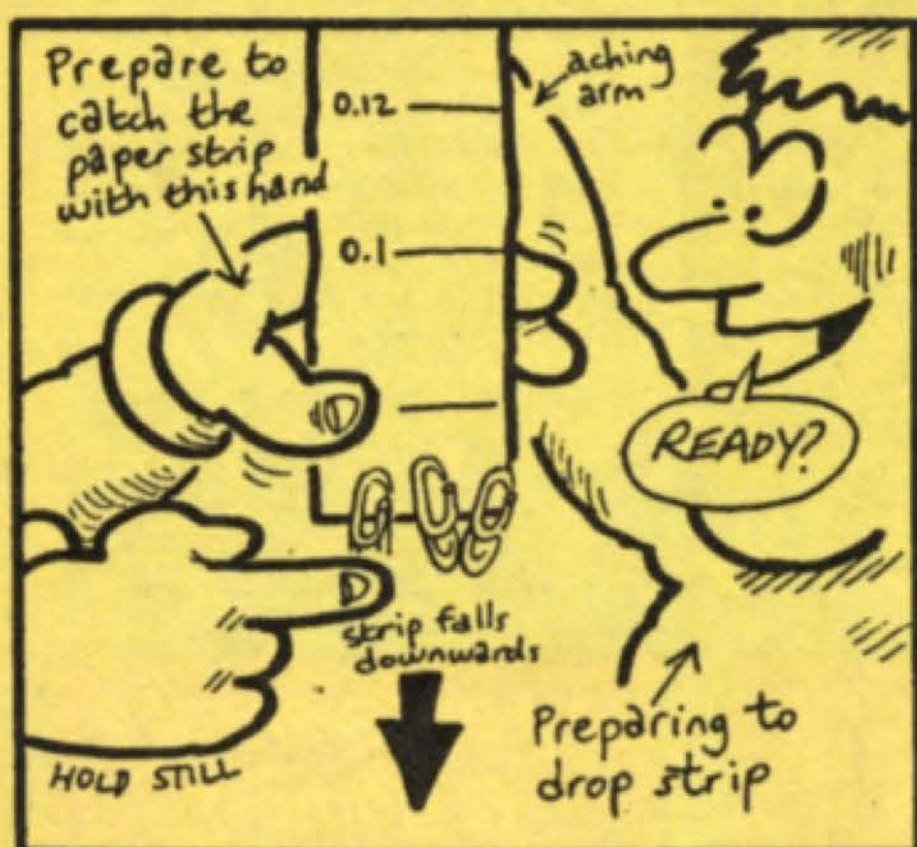
Are you the fastest kid on the block?

**Find out with the
Let's Compute!
reaction
tester**



on the draw!

REACTION TIMER



MAKING THE PAPER REACTION TIMER

Cut the two strips from the right hand edge of this page. Tape them together so the numbers are along one edge. The zero should be at one end and the 0.34 at the other.

Put three paper clips at the end where the zero is. Your reaction timer is now ready to use.

How did we work out how to mark the tester? Simple! We used our computer and this program:

```
10 PRINT " TIME    DISTANCE"
20 FOR T=0.1 TO 0.35 STEP 0.02
30 D=INT(4900*T*T+.5)/10
40 PRINT " ";T;" secs  ";D;" cm"
50 NEXT
```

It works on all computers. Try it.

When you RUN it you get figures like those shown on the right.

The right hand column tells you how far to measure from zero. The left hand one tells you the times you should write

TIME DISTANCE

0.1 secs	4.9 cm
0.12 secs	7.1 cm
0.14 secs	9.6 cm
0.16 secs	12.5 cm
0.18 secs	15.9 cm
0.2 secs	19.6 cm
0.22 secs	23.7 cm
0.24 secs	28.2 cm
0.26 secs	33.1 cm
0.28 secs	38.4 cm
0.3 secs	44.1 cm
0.32 secs	50.2 cm
0.34 secs	56.6 cm

on your tester. If you want to make a longer tester just increase the number 0.35 in Line 20.

But don't make it too big! If you try increasing the time to 0.8 secs you'll find you need a piece of paper about 3 metres long!

USING YOUR COMPUTER

Type in and RUN the program below. You'll first be told to *Press the spacebar then press it again after 10 seconds*. Press it once. Use a watch to time 10 seconds then press it again.

Then watch your screen carefully!

Each time XXX appears press the spacebar – as quickly as you can. You'll be told how quick you were. Make a note of your best time and keep trying to do better.

```
10 PRINT "PETE'S PROJECT REACTION TEST ER"
20 PRINT:PRINT "PRESS THE SPACEBAR."
30 PRINT:PRINT "THEN PRESS IT AGAIN AFTER 10 SECONDS.":PRINT:LET T=0
40 GOSUB 120:LET T=T+0.1
50 GOSUB 120:LET S=T/10
60 PRINT:PRINT "PRESS SPACE EACH TIME YOU SEE XXX":PRINT
70 IF T>99 THEN LET T=T-100:GOTO 70
80 LET R=INT(T/10):LET Z=T-R*10
90 LET R=R*Z*S:FOR Z=1 TO R:NEXT
100 PRINT "XXX":PRINT:LET T=0:GOSUB 120
110 PRINT "REACTION TIME WAS ";INT(T/S*100+.5)/100;" SECS":PRINT:GOTO 70
120 LET AS=INKEY$(0)
130 IF AS<>" " THEN LET T=T+1:GOTO 120
140 RETURN
```

IS YOUR COMPUTER HERE?

Electron/BBC/Archimedes
The program works as shown.

Amiga (Amiga Basic), PC (GW-Basic), Amstrad CPC, Spectrum/Atari ST (Stos)

Change or add these lines:

```
120 IF INKEYS<>" " THEN GOTO 120
125 LET AS=INKEYS
```

Atari ST: Also change S to S# in Lines 50,90 and 110.

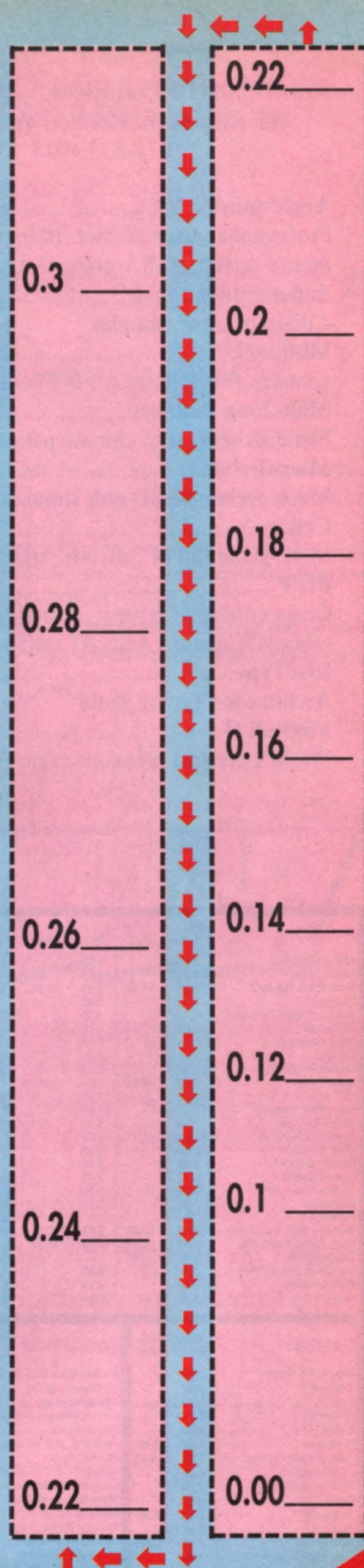
Commodore 64/128

Change this line:

```
120 GET AS
```

ANOTHER USE FOR THIS PROGRAM

Can you estimate, say, one minute without looking at your watch? When XXX appears on your screen don't press the spacebar as quickly as you can. Wait until you think a minute has passed and then press it.



NEXT MONTH: Watch the weather with Peter's rain gauge

LET'S MAKE MONEY

Part 5

STAR GAZING

Here's an electronic game to help boost your funds

Lots of successful fund raising events are based on guessing. People are willing to part with money to have a go – whether it's estimating the number of sweets in a jar or guessing the weight of a cake.

This program lets you use your computer to play a guessing game. The idea is simple.

When you RUN the program lots of stars will appear on your computer screen. A few seconds later they disappear.

People pay you to guess how many there were. Type their estimate into your computer and you'll be told how near they were.

Press the spacebar and another load of stars will be displayed. Someone else can then have a go.

There will be a different number of stars each

time. That means the game doesn't end even if someone guesses the right number.

Exactly how you use the program is up to you. You could, for example, charge 10p a go and refund the money if the guess is 10 or less away from the right answer. Anyone guessing the exact number could win £1.

This short program is intended as a frame for you to build on. You can use it just as it is. But you'll find more people will want to look at your game if you add some extra bits.

There are a few ideas below for changes you could try. Use some of these and add a few of your own. Then take your computer and this program to make money at your next fund-raising event.

CHANGE THE PROGRAM TO SUIT YOUR NEEDS

Here are some ideas for ways you can alter the program:

- You can have more, or less, stars by changing the number 600 in Line 30.
- Change the number 10 in Line 40. If you make it smaller more stars will be shown.
- Use something different to a * in Line 50. If you know how to define characters you could, for instance, make a clown's face.
- Alter the number 5000 in Line 80. The smaller it is, the quicker the stars will be cleared from the screen.
- Put more messages in the program. These should be different depending how near the guess was. For example, try adding this line:

135 IF N>50 AND N<80 THEN PRINT"MILES AWAY: TRY HARDER!!"

```
10 LET C=0
20 CLS
30 FOR I=1 TO 600
40 LET R=RND(10)
50 IF R=1 THEN PRINT"*";:LET C=C+1
60 IF R<>1 THEN PRINT" ";
70 NEXT I
80 FOR I=1 TO 5000:NEXT I
90 CLS
100 PRINT:INPUT"HOW MANY STARS WERE THERE";N
110 LET N=ABS(C-N)
120 IF N=0 THEN PRINT"EXACTLY RIGHT!"
130 IF N<>0 THEN PRINT"THERE WERE ";C;
    ".":PRINT"YOU WERE ";N;" OUT"
140 PRINT"PRESS SPACE FOR ANOTHER GO"
150 REPEAT UNTIL GET=32
160 GOTO 10
```

- Put colour in the program
- Add sound. Maybe a higher note for each star drawn.

IS THIS YOUR COMPUTER?

Electron/BBC/Archimedes

The program works as shown.

Amiga (Amiga Basic), PC (GW-Basic), Amstrad CPC, ST (Stos), Spectrum

Change these lines:

```
40 LET R=INT(RND*10)+1
150 IF INKEY$<>" " THEN GOTO 150
```

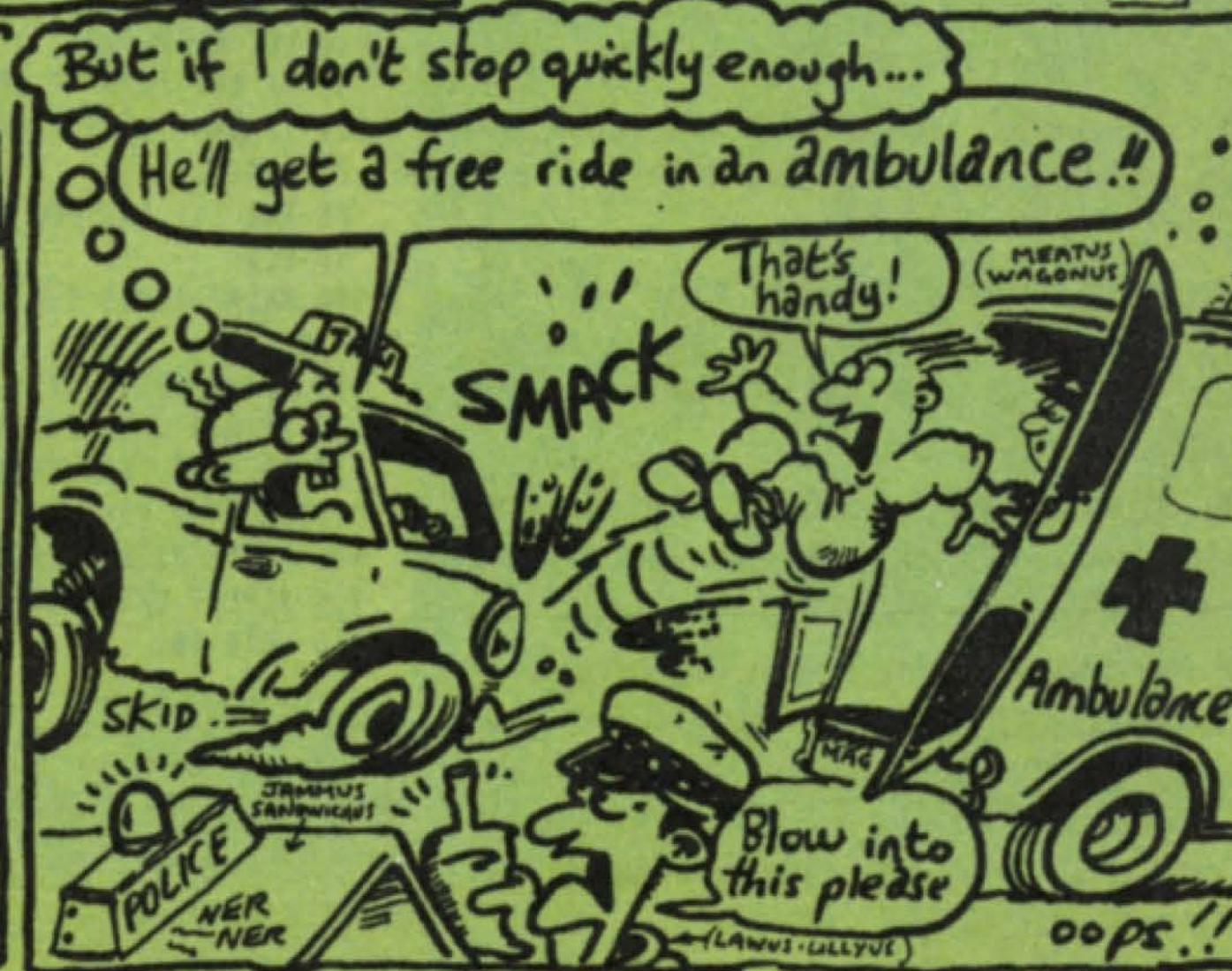
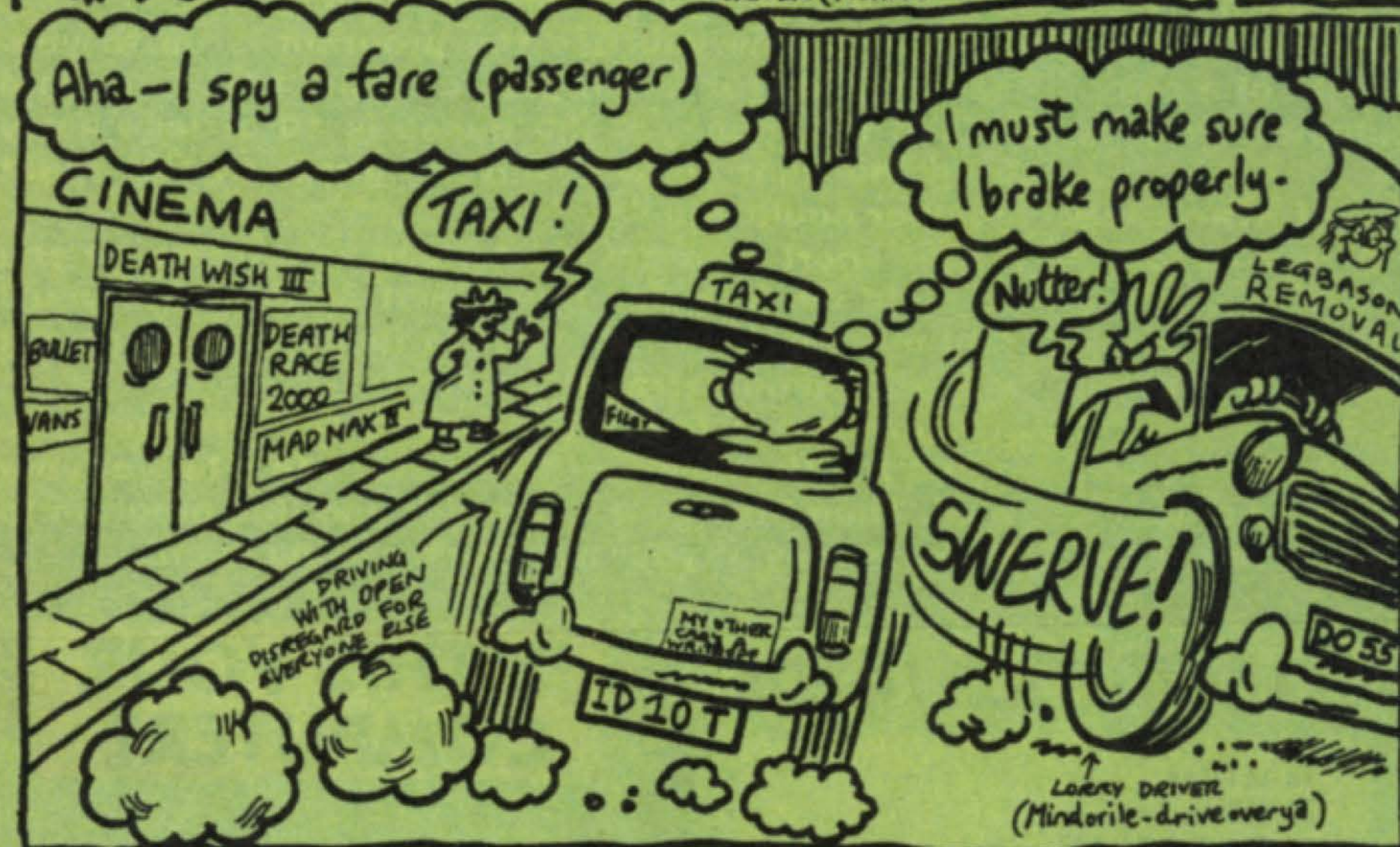
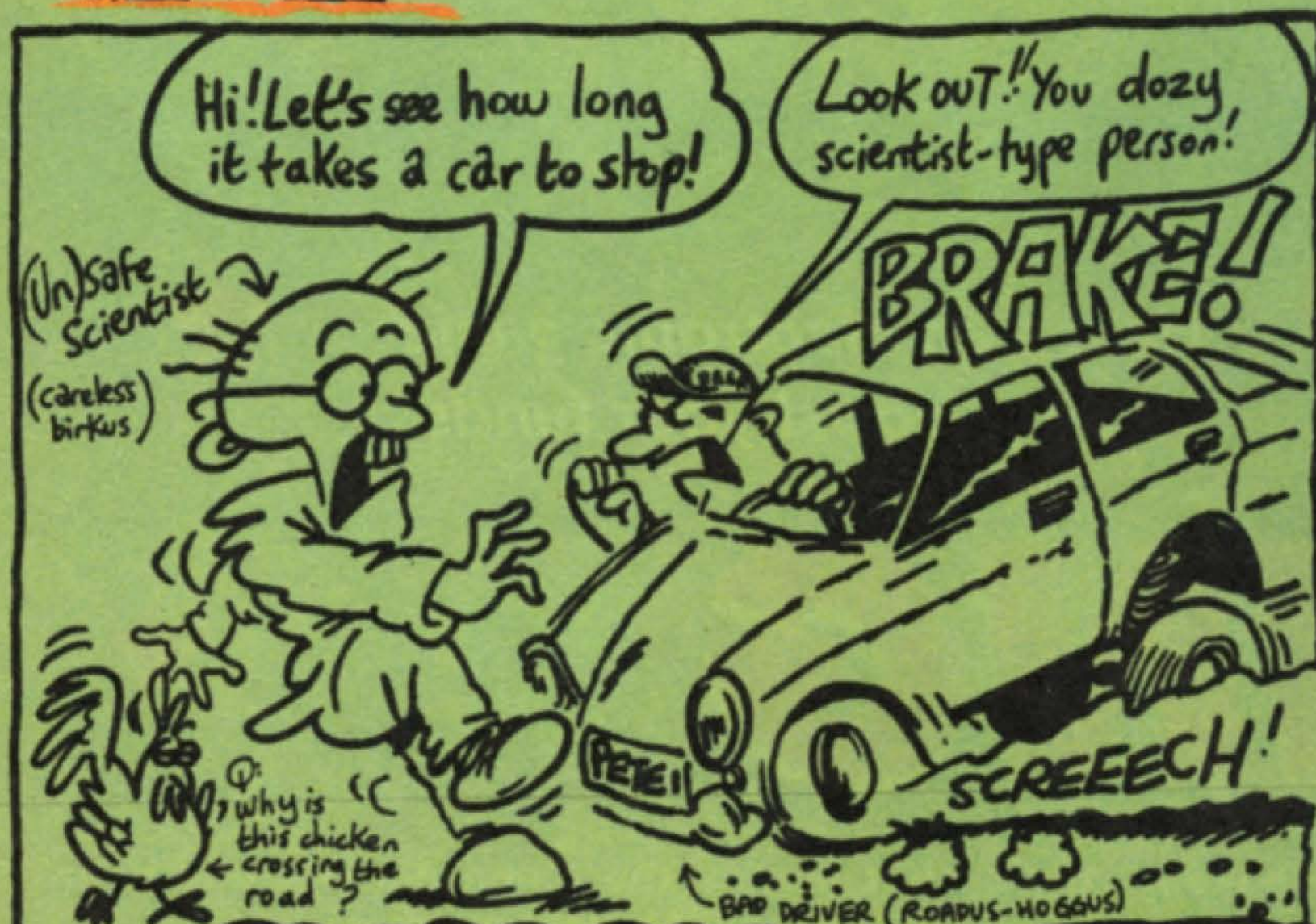
Commodore 64/128

Change these lines:

```
20 PRINT CHR$(147)
40 LET R=INT(RND(8)*10)+1
90 PRINT CHR$(147)
150 GET Z$:IF Z$<>" " THEN GOTO 150
```




The Safe



Scientist

STOP THAT CAR!



How long does it take to stop a car? It depends how fast it's going!

Let's look at braking distances.

Imagine you are a taxi driver. Ahead of you is a passenger who wants a lift. You must bring the car to a smooth stop right by your customer.

If you stop short the customer may go in another taxi. After all, who wants to ride in a poorly driven car?

It's even worse if you don't stop quickly enough. Your passenger could find himself getting a free ride in an ambulance! Type the program in and RUN it. Then see if you could stop in time.

The program is very simple to use. Just press the B key when you want to apply the brakes.

There is no need to hold the key down. Once it's been pressed the car will come to rest – eventually.

The stopping distances are based on

the highway code figures. They've been converted into a mathematical formula for your computer to use.

There are several investigations you could try. You can, for instance, work out the stopping distances you need and plot them on a graph against the speed. Also, find out if you need twice as much distance to stop if you travel twice as fast?

```

10 REM CAR SPEEDS
20 REM THE SAFE SCIENTIST
70 CLS
90 LET S=(1+RND(6))*10
100 LET D=S*(S*S)/20
120 LET X=29:LET Y=10:LET A$="&":GOSUB
B 300
130 LET X=0:LET A$="m":GOSUB 300
140 LET Y=11:LET A$=" ":FOR X=0 TO 31
:GOSUB 300:NEXT X
150 FOR X=0 TO 28 STEP 4
152 LET Y=12:LET A$="!":GOSUB 300:LET
A$=STR$(350-X*12.5):LET Y=13:GOSUB 300
155 NEXT X
160 LET T=100:GOSUB 400
170 LET X=10:LET Y=18:LET A$="PRESS S
PACE":GOSUB 300
175 GOSUB 350:IF K$<>" " THEN GOTO 17
5
176 LET A$=" ":GOSUB 300
180 LET X=0:LET Y=6:LET A$="SPEED ="
STR$(S)+ " Press B for Brakes.":GOSUB 3
00
190 LET X=0:LET Y=10:LET XS=S:LET P=0
200 LET T=100:GOSUB 400
210 LET A$=" m":GOSUB 300

```

```

220 LET X=X+1:IF P>0 THEN LET XS=XS-2
.5
230 LET T=350/XS:GOSUB 400
240 GOSUB 350:IF (K$="B" OR K$="b") A
ND P=0 THEN LET P=INT(X/D/12.5)
250 IF X<31 AND X<>P THEN GOTO 200
255 LET N=X:LET X=0:LET Y=16
260 IF N<28 THEN LET A$="You stopped
short.":GOSUB 300
265 IF N=28 THEN LET A$="SPOT ON!! Go
od braking.":GOSUB 300
267 IF N>28 THEN LET A$="OH DEAR!! Yo
u went too far.":GOSUB 300
270 LET X=10:LET Y=18:LET A$="PRESS S
PACE":GOSUB 300
280 GOSUB 350:IF K$=" " THEN GOTO 70
290 GOTO 280
300 REM
310 PRINT TAB(X,Y):A$:RETURN
350 REM FETCH KEY NO WAIT
360 LET K$=INKEY$(0):RETURN
400 REM WAIT
410 LET T=TIME+T
420 IF T>TIME THEN GOTO 420
430 RETURN

```

IS THIS YOUR COMPUTER?

Amiga/PC (GW-Basic)

Change the following lines:

```

90 LET S=(2+INT(RND*6))*10
310 LOCATE Y+1,X+1:PRINT A$:RETURN
360 LET K$=INKEY$:RETURN
410 LET T=TIMER+T/250
420 IF T>TIMER THEN GOTO 420

```

Atari ST (Stos)

Change the following lines:

```

70 MODE0:KEY OFF:HIDE
90 LET S=(2+INT(RND*6))*10
310 LOCATE X+1,Y+1:PRINT A$:RETURN
360 LET K$=INKEY$:RETURN
410 LET T=TIMER+T/250
420 IF T>TIMER THEN GOTO 420

```

Spectrum

Change the following lines:

```

90 LET S=(2+INT(RND*6))*10
310 PRINT AT Y,X:A$:RETURN
360 LET K$=INKEY$:RETURN
410 LET T=PEEK(23672)+T/2:IF T>255 THEN LET
T=T-256
420 IF T<>PEEK(23672)+T/2 THEN GOTO 420

```

Commodore C64/128

Use Capital letters.

Change the following lines:

```

70 PRINT CHR$(147);
90 LET S=(2+INT(RND(0)*6))*10
310 POKE 211,X:POKE 214,Y:SYS58732:PRINT A$
360 GET K$:RETURN
410 LET T=TI+T*1.4
420 IF T>TI THEN GOTO 420

```

Amstrad CPC

Change the following lines:

```

90 LET S=(2+INT(RND*6))*10
310 LOCATE X+1,Y+1:PRINT A$
360 LET K$=INKEY$:RETURN
410 LET T=TIME+T*3

```

That's it for this month. Next month we'll learn all about attraction and repulsion – between magnets.

Part 7

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

Get your printer ready
and load it with paper.
Roland Waddilove's
database is now
nearing completion

PRINT the cards

PRINTING A SINGLE CARD

The lines we've added this month print out ALL the cards. But what you only want to print out one of them?

There's an easy change you can make.

Delete Line 7150, change Line 7080 and add a new Line 7085:

```
7080 INPUT "Enter card number: ";x
```

Then add this line:

```
7085 x=x*5-4
```

How the program works

Let's see how we get the printer to do its job. The first few lines just print messages to make sure you're ready. Lines 7180 to 7160 are the ones that print the cards. As we've seen before, each card is five lines long and is stored in the array C\$(). The cards start at C\$(1), C\$(6), C\$(11) and so on. If the first character is an asterisk (*) it means that you haven't yet typed anything on that card. So it doesn't need to be printed out. Line 7090 checks this. Lines 7110 to 7130 print the card if there is information on it.

The *Let's Compute!* Database that we've been building up over the last few months is nearly finished. We can type in information on the cards, read them on the screen and save them to disk or tape.

But there are still a couple more things it needs to be able to do to before it's complete. You may want to print out the cards or change something if you make a mistake.

This month we'll see how to print the cards.

There are lots of reasons why you may want to do this. You may want to give someone a copy of all your cards. You may want to carry the facts that are on the cards with you when you are away from your computer.

For instance, if you are a train spotter you may want to check whether a train you see is already on your cards. Unless you have a portable computer, the only way of doing this is to look at a list on paper.

If you've got a printer connected to your computer, the lines printed here will let you print your cards. Here's what to do:

LOAD the program you've built up so far and type in the new lines. Then SAVE the latest version of the program. Now you're ready to RUN it.

To see it work either load some cards you've saved before or type a few in. Then choose 7 from the menu and see your cards print.

NEXT MONTH...

You'll add the final part of the program. Then you'll be able to alter a card if you make a mistake.

CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6
What do you want to do?
Please type in a number (1- 9)

Print out the cards program

```
7000 REM ===== Print cards =====
7010 CLS
7020 PRINT "Print out the cards"
7030 PRINT "-----"
7040 PRINT
7050 INPUT "Get your printer ready
then press Return";K$
7060 PRINT
7070 PRINT "Printing...";
7080 X=1
7090 IF C$(X)="*" THEN GOTO 7140
7100 VDU2:PRINT
7110 FOR Y=0 TO 4
7120 PRINT C$(X+Y):VDU3
7130 NEXT Y
7140 X=X+5
7150 IF X<255 THEN GOTO 7090
7160 RETURN
```

This
program
works on
all home
micros

IS THIS YOUR COMPUTER?

BBC/Electron/Archimedes

The program works as shown.

Spectrum

Change these lines:

```
7090 IF C$(X) =B$ THEN GO TO 7140
7100 LPRINT
7120 LPRINT C$(X+Y)
```

Amiga (Amos)/AtariST (Stos)/PC (GW Basic)

Change these lines:

```
7100 LPRINT
7120 LPRINT C$(X+Y)
```

CPC

Change these lines:

```
7100 PRINT #8
7120 PRINT #8,C$(X+Y)
```

C64

Use capital letters.

Change or add these lines:

```
7010 PRINT CHR$(147);
7075 OPEN 4,4
7100 PRINT #4,CHR$(13)
7120 PRINT #4,C$(X+Y)
7160 PRINT#4:CLOSE 4:RETURN
```


BEST SELLING AMIGA BOOKS

AMIGA DOS INSIDE & OUT

£16.95

An in-depth guide to AmigaDOS and the CLI. Covers the insides of AmigaDOS from internal design up to practical applications. Versions 1.2 and 1.3 are covered and this revised edition also includes AmigaDOS 2.0

AMIGA C FOR BEGINNERS

£16.95

An introductory guide for the Amiga user that wishes to learn this popular language without a lot of technical jargon. Describes the C library routines and use of the most popular C compilers. Lots of programming examples.

AMIGA DESKTOP VIDEO GUIDE

£16.95

This is a book for everyone who wants to use the Amiga for video. You'll find information on video basics, video-tape equipment, video/Amiga interfaces, artwork and more.

AMIGA FOR BEGINNERS

£12.95

The ideal book for the first time user. A clear introduction to essential Amiga features including the Workbench diskette, AmigaBASIC and AmigaDOS. Recently revised to include Workbench 2.0 info.

AMIGA PRINTERS INSIDE & OUT

NEW

£29.95

A comprehensive guide with easy to follow examples and simple explanations to help you use your printer more effectively. Learn AmigaDOS commands for printer control and printing tips and tricks from the experts. Includes a companion diskette with essential printer utilities.

MAKING MUSIC ON THE AMIGA

NEW

£29.95

The Amiga has wonderful music talents. This book shows you how to take advantage of those capabilities, it starts with the fundamentals of music and progresses to the details of using software such as Music Construction Set, Aegis/Oxxi and AudioMaster and how to construct your own MIDI interface, includes companion diskette containing lots of music examples.

THE BEST OF TRICKS & TIPS

NEW

£24.95

The best selling book 'Tricks & Tips' and its follow up book 'More Tricks & Tips' have been replaced by this single volume. The companion diskette is included. A great collection of programming hints and techniques for all Amiga users.

OTHER BOOKS

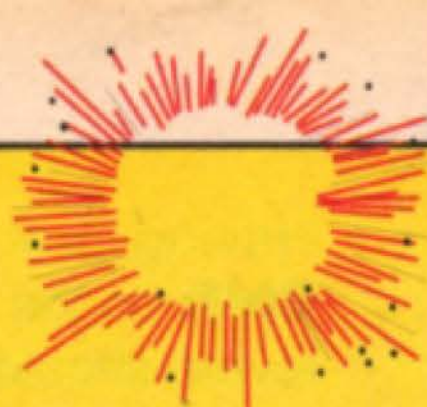
AmigaDOS Quick Ref. Guide	£8.95	Amiga Machine Language	£14.95
Amiga 3D Graphics Prog in BASIC	£16.95	Adv. System Programmers Guide	£29.95
Amiga System Programmers Guide	£29.95	Amiga C for Adv. Programmers	£29.95
Amiga Disk Drives Inside & Out	£24.95	More Amiga Tricks & Tips	£14.95
Amiga Graphics Inside & Out	£29.95	Amiga ROM Kernel Ref: Includes &	
Amiga ROM Kernel Ref: Libs. &		Autodoc	£28.95
Divs	£29.95	Kids and the Amiga	£13.95
Amiga Hardware Ref. Manual	£21.95	Advanced Amiga BASIC	£16.95
Elementary Amiga BASIC	£12.95	Using Delux Paint 2nd Ed.	£19.95
Amiga Applications	£16.95	First Book of Amiga	£14.95
Inside Amiga Graphics	£15.95	Second Book of Amiga	£14.95
Amiga DOS Ref. Guide (3rd ed)	£18.95	Amiga Programmer's	
Mapping the Amiga	£19.95	Guide (Compute!)	£16.95
Amiga Desktop Video (Compute!)	£16.95	Amiga DOS Reference	
AMIGADOS: A Dabhand Guide	£14.95	Guide (2nd Ed)	£6.95
Inside the Amiga with C	£19.95	Amiga Companion 2nd Ed.	£16.95
Amiga Programmer's		Becoming an Amiga Artist	£16.95
Handbook Vol. 2	£22.95	Amiga Assembly Language	
Desktop Video Production,		Programming	£12.95
Amiga & Mac II	£14.95	Amiga Desktop Video Workbook	
Amiga Basic Inside & Out	£18.95	(inc disk)	£29.95

All prices include postage within UK.

Add 10% postage for European orders and worldwide surface mail.

Add 30% for airmail outside Europe. Dealer enquiries welcome.

DTBS (Dept)
LC
Tel/Fax 0706 524304
18 Norwich Avenue, Rochdale, Lancs OL11 5JZ
Access/Visa Orders Accepted



**Challenge your
and see who has**

SHOOT

IS THIS YOUR COMPUTER?

The program
listed is for the BBC,
Archimedes and
Electron

Amiga

Change the following lines:

```
20 CLS:LET TU=1000:LET MT=0
30 DEF FNR(R)=INT(RND*R+1)
110 LET AS="INKEY$:IF AS="A" OR AS="a" THEN GOTO
230
120 IF AS="S" OR AS="s" THEN GOTO 200
130 IF AS="D" OR AS="d" THEN LET S1=S1-FNR(3):G
OTO 260
140 IF AS="M" OR AS="m" THEN GOTO 310
150 IF AS="K" OR AS="k" THEN GOTO 340
160 IF AS="O" OR AS="o" THEN LET S2=S2-FNR(3):G
OTO 370
610 LOCATE Y+1,X+1
```

PC

Make the changes listed above for the Amiga to Lines 30, 110, 120, 130, 140, 150 160 and 610.

Also change Line 20:

```
20 SCREEN1:CLS:LET TU=1000:LET MT=0
```

Atari ST (Stos)

Make the changes listed above for the Amiga to Lines 20, 30, 110, 120, 130, 140, 150 and 160.

Also change Line 610:

```
610 LOCATE X,Y
```

Spectrum

Make the changes listed above for the Amiga to Lines 20, 30, 110, 120, 130, 140, 150 and 160. Also change Line 610:

```
610 PRINT AT Y,X;
```

Commodore C64/128

Use Capital letters. Make the changes listed above for the Amiga to Lines 120, 130, 140, 150 and 160. Also change these lines:

```
20 PRINT CHR$(147);:LET TU=1000:LET MT=0
30 DEF FNR(R)=INT(RND(0)*R+1)
110 GET AS:IF AS="A" OR AS="a" THEN GOTO 230
550 PRINT CHR$(147);:PRINT"PLAYER 1 SCORED ";S1
610 POKE 211,X:POKE 214,Y:SYS58732
```

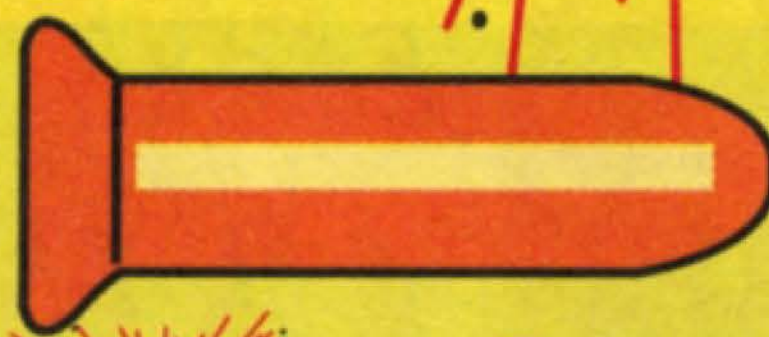
Amstrad CPC

Make the changes listed above for the Amiga to Lines 20, 30, and 610. Also change these lines:

```
110 IF INKEY(69)<>-1 THEN GOTO 230
120 IF INKEY(60)<>-1 THEN GOTO 200
130 IF INKEY(61)<>-1 THEN LET S1=S1-FNR(3):GOTO
260
140 IF INKEY(38)<>-1 THEN GOTO 310
150 IF INKEY(37)<>-1 THEN GOTO 340
160 IF INKEY(34)<>-1 THEN LET S2=S2-FNR(3):GOTO
370
```


**friend at this exciting game
the better shooting skills**

T'EM



This game and the description printed here are by David Johnston (11) from Glasgow. It's a two player game. Both try to shoot the same moving target.

IT'S YOUR PROGRAM

If you've written any programs, let us see them. You may soon see YOUR work in *Let's Compute!*

Don't worry if you can't make your programs work on all the computers we cover. Send us a program for any computer and our experts will convert any we print to work on all home computers.

Send them on disc or tape to *Let's Compute!*, Europa House, Adlington Park, Macclesfield SK10 4NP. Remember to tell us your name, address and age.

Also send us a description of the program. If you have a phone tell us the number. That way, if we're going to print it, we can let you know quickly.

Both players have a gun. Player one's gun is at the bottom and player two's is at the side.

See who can shoot the most moving targets. Don't fire too much because each time you fire your score goes down.

After one minute the screen will clear and the scores will appear. You will then be asked if you want another go. Answer Y or N.

HERE ARE SOME CHANGES YOU COULD MAKE:

- You could try making the game last longer or shorter by changing TU in line 20.

- Try altering the shape of the gun and the target by changing lines 30, 60 and 70.

- You can change the movement keys in lines 100 to 150.

- When you hit the target try making a sound by adding to lines 230 and 290.



Ace game writer:
David Johnston

PLAYER 1		PLAYER 2	
Left	A	Up	K
Right	S	Down	M
Fire	D	Fire	O

These are the keys to use

```

10 LET SX=19:LET SY=21:LET EA=12:LET E
B=16:LET A=11:LET B=15:LET S1=0:LET S2=0
20 CLS:VDU 23;8202;0;0;0;:LET TU=1000:
LET MT=0
30 DEF FNR(R)=RND(R)
40 LET X=A:LET Y=SY:GOSUB 610:PRINT"A"
50 LET X=1:LET Y=1:GOSUB 610:PRINT"PLA
YER ONE'S SCORE ";S1;" "
60 LET X=1:LET Y=2:GOSUB 610:PRINT"PLA
YER TWO'S SCORE ";S2;" "
70 LET X=SX:LET Y=B:GOSUB 610:PRINT"<"
80 LET X=EA:LET Y=EB:GOSUB 610:PRINT"O
"
90 LET X=A:LET Y=SY-1:GOSUB 610:PRINT"
"
100 LET X=SX-1:LET Y=B:GOSUB 610:PRINT"
"
110 IF INKEY(-66)=-1 THEN GOTO 230
120 IF INKEY(-82)=-1 THEN GOTO 200
130 IF INKEY(-51)=-1 THEN LET S1=S1-FNR
(3):GOTO 260
140 IF INKEY(-102)=-1 THEN GOTO 310
150 IF INKEY(-71)=-1 THEN GOTO 340
160 IF INKEY(-55)=-1 THEN LET S2=S2-FNR
(3):GOTO 370
170 IF TU=0 THEN GOTO 550
180 LET TU=TU-1
190 GOTO 410

```

```

200 IF A=SX THEN GOTO 130
210 LET X=A:LET Y=SY:GOSUB 610:PRINT" "
:LET A=A+1:LET X=A:LET Y=SY:GOSUB 610
220 PRINT"A":GOTO 130
230 IF A=1 THEN GOTO 120
240 LET X=A:LET Y=SY:GOSUB 610:PRINT" "
:LET A=A-1:LET X=A:LET Y=SY:GOSUB 610
250 PRINT"A":GOTO 130
260 LET X=A:LET Y=SY-1:GOSUB 610:PRINT"
|"
270 IF EA=A THEN LET S1=S1+25:LET X=EA:
LET Y=EB
280 IF EA=A THEN GOSUB 610:PRINT"*":FOR
Z=1 TO 200:NEXT Z:GOSUB 610
290 IF EA=A THEN PRINT" ":LET EA=FNR(SX
-1):LET EB=FNR(SY-4)+3:GOTO 140
300 LET BV=0:GOTO 140
310 IF B=SY THEN GOTO 150
320 LET X=SX:LET Y=B:GOSUB 610:PRINT" "
:LET B=B+1:LET X=SX:LET Y=B:GOSUB 610
330 PRINT"<":GOTO 150
340 IF B=3 THEN GOTO 160
350 LET X=SX:LET Y=B:GOSUB 610:PRINT" "
:LET B=B-1:LET X=SX:LET Y=B:GOSUB 610
360 PRINT"<":GOTO 160
370 LET X=SX-1:LET Y=B:GOSUB 610:PRINT"
="
380 IF EB=B THEN LET S2=S2+25:LET X=EA:

```

```

LET Y=EB
390 IF EB=B THEN GOSUB 610:PRINT"*":FOR
Z=1 TO 200:NEXT Z:GOSUB 610
400 IF EB=B THEN PRINT" ":LET EA=FNR(SX
-1):LET EB=FNR(SY-4)+3
410 IF MT=4 THEN GOTO 430
420 LET MT=MT+1:GOTO 40
430 LET MT=0
440 LET D=FNR(4)
450 LET X=EA:LET Y=EB:GOSUB 610:PRINT"
"
460 IF D=1 AND EB=3 THEN GOTO 410
470 IF D=2 AND EB=SY-1 THEN GOTO 410
480 IF D=3 AND EA=SX-1 THEN GOTO 410
490 IF D=4 AND EA=1 THEN GOTO 410
500 IF D=1 THEN LET EB=EB-1
510 IF D=2 THEN LET EB=EB+1
520 IF D=3 THEN LET EA=EA+1
530 IF D=4 THEN LET EA=EA-1
540 GOTO 40
550 CLS:PRINT"PLAYER 1 SCORED ";S1
560 PRINT"PLAYER 2 SCORED ";S2
570 PRINT"ANOTHER GO";:INPUT AS
580 IF AS="Y" OR AS="y" THEN GOTO 10
590 IF AS="N" OR AS="n" THEN STOP
600 GOTO 550
610 PRINTTAB(X,Y);
620 RETURN

```


MORE MEMORY...

Here's a program to help boost your computer's power!

It's April, a month well known for its showers. So, while it's raining why not make your computer do something **REALLY** useful?

For instance, wouldn't it be a good idea to give it double the memory? That would let you play more powerful games.

So we asked our top programmer to write a program that would do just that. He went away, thought hard, punched a

few keys and returned with a huge smile on his silly face.

The program here is what he handed over.

He said he'd thought it out carefully and made sure it will work on all computers.

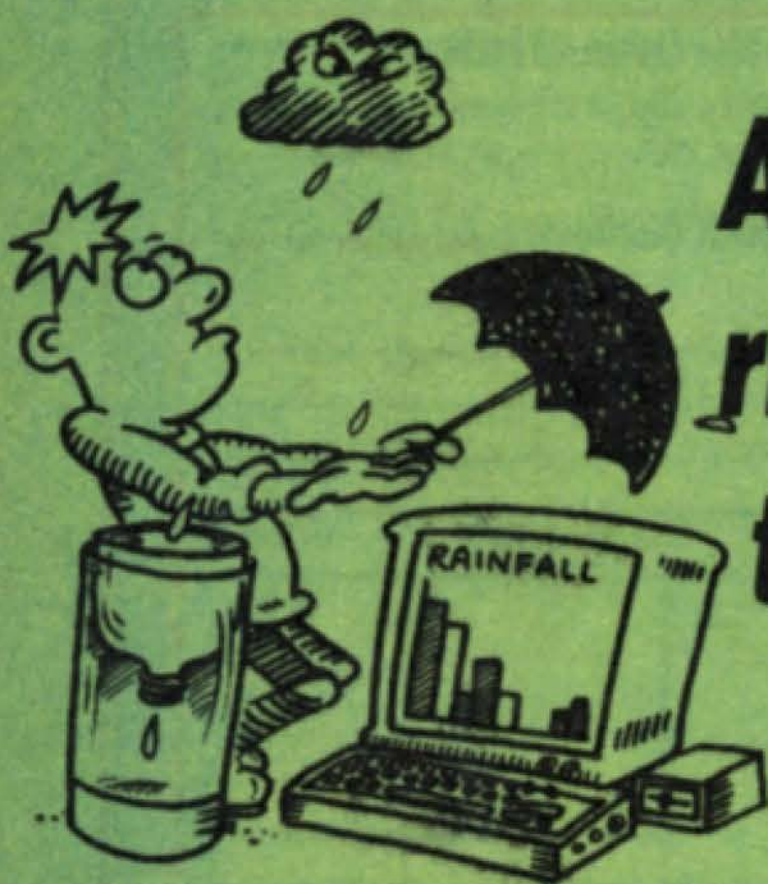
Just type it in and **RUN** it. And let the magic numbers do the rest!

How was this clever program put together? We'll go through it line by line next month.

```
10 LET M=0:DIM L(100)
20 LET M=M+1:READ L(M)
30 IF L(M)<>99 THEN GOTO 20
40 LET T=M*2:REM ** DOUBLES IT!!
50 FOR T=1 TO M-1
60 IF L(T)=13 THEN PRINT:GOTO 80
70 PRINT CHR$(L(T));
80 NEXT T
90 REM THE SECRET IS IN THE NUMBERS
100 DATA 13,73,84,32,73,83,32,73,77
110 DATA 80,79,83,83,73,66,76,69,32
120 DATA 84,79,32,87,82,73,84,69,32
130 DATA 65,13,80,82,79,71,82,65,77
140 DATA 32,84,79,32,68,79,85,66,76
150 DATA 69,32,84,72,69,32,77,69,77
160 DATA 79,82,89,46,13,89,79,85,32
170 DATA 72,65,86,69,32,66,69,69,78
180 DATA 32,84,82,73,67,75,69,68,46
190 DATA 13,65,80,82,73,76,32,70,79
200 DATA 79,76,33,13,13,99
```

Coming soon ... in

LET'S COMPUTE!



Are the weathermen right? Find out with the Pete's Project rainfall measurer

PACMAN:
Another great Let's Compute! game



DISCOVER
magnetism with
the Safe Scientist



LET'S COMPUTE!



**PAC
MAN**



*A super
game to
type in*

3 *Easy-to-
-enter*
CONTESTS

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

No 10 May 1991
A Database Publication



PARENTS & TEACHERS!
Turn to Page 26 for ways of developing
the National Curriculum content
of Let's Compute!

GUESS THE LETTER



**LET'S MAKE
MONEY**

Part 6



Use Colin Markwell's game to raise more cash!

Guess the Letter is a game of chance. It's great for any fund raising event.

Type in the program - very carefully. **SAVE** it and then **RUN** it.

For fund raising at a school fete or any other occasion you must first **LOAD** the game. Make sure Caps Lock or Shift Lock is pressed. Then let people guess the letter which the computer will pick at random. They are given five chances. At the end they are told what they have won - if anything!

Change lines 310 to 330 to suit your needs. **G** is the number of goes you've had and **W\$** is the message given with that score. If you want more than five goes, change the 5 in Lines 90 and 100.

This game is educational as well. It's a fun way to learn the alphabet.

If you aren't using it for fundraising you can get rid of some messages. Just delete Lines 310 to 340.

Of course, the program can be modified in lots of ways to suit your own needs. For example, you can change the messages, add more or put some sounds in it.

Happy Fund raising!



This program and the description printed here are by Colin Markwell (13) from Ipswich. It's based on the old **Guess the number I'm thinking of** Game. Colin's idea of using letters adds a new twist to it.

IS THIS YOUR COMPUTER?

BBC, Electron and Archimedes

The program works as shown

AMIGA and PC(GW-Basic)

Change these lines:

```
50 RANDOMIZE TIMER
70 LET C=65+INT(RND*26)
```

ST(STOS)

Change these lines:

```
50 LET Z=RND(TIMER)
70 LET C=65+RND(25)
```

Amstrad CPC

Change these lines:

```
50 RANDOMIZE TIME
70 LET C=65+INT(RND*26)
```

Spectrum

Change these lines:

```
50 RANDOMIZE
70 LET C=65+INT(RND*26)
130 PRINT:LET Q=CODE(AS)
```

Commodore 64/128

Use capital letters.

Change these lines:

```
40 PRINT CHR$(147);
70 LET C=65+INT(RND(0)*26)
```

```
10 REM GUESS THE LETTER GAME!
20 REM By C.Markwell
30 REM (c) Let's Compute!
40 CLS
50 LET S=RND(-TIME):REM Randomise
60 PRINT:PRINT:PRINT"The Let's Comput
e! Guessing Game"
70 LET C=64+RND(26):REM Choose Letter
80 PRINT:PRINT "O.K. Start guessing."
90 FOR G=1 TO 5:REM 5 GOES
100 PRINT:PRINT"Chance number ";G;" ou
t of 5 guesses."
110 PRINT
120 GOSUB 370:PRINT AS:REM GET LETTER
130 PRINT:LET Q=ASC(AS)
140 IF Q<65 OR Q>90 THEN PRINT"That's
a silly guess!":GOTO 100
150 IF G=5 THEN GOTO 180
160 IF Q<C THEN PRINT"It's further on
in the alphabet"
170 IF Q>C THEN PRINT "It's earlier on
in the alphabet."
180 IF Q=C THEN PRINT "You've got it r
ight! Congratulations!":GOTO 280
190 NEXT G
200 PRINT:PRINT "Oh dear. You failed t
o guess the"
210 PRINT "Letter (Which was ";CHR$(C)
;")"
220 PRINT:PRINT:PRINT"Do you want anot
her go (Y or N)?"
230 GOSUB 370:REM Get a Y or N
240 IF AS="Y" THEN RUN:REM Start Again
250 IF AS<>"N" THEN GOTO 230
260 STOP
270 REM Win!
280 PRINT:PRINT:PRINT"You've completed
the game in ";G;
290 IF G=1 THEN PRINT " Go!!!"
300 IF G>1 THEN PRINT " Goes."
310 IF G=5 OR G=4 THEN LET W$="your mone
y back."
320 IF G=3 OR G=2 THEN LET W$="twice as
much as you paid."
330 IF G=1 THEN LET W$="THE JACKPOT!"
340 PRINT:PRINT "You win ";W$
350 GOTO 220
360 REM Get a key
370 PRINT"ENTER A LETTER ";:INPUT AS
380 RETURN
```


ESC

EDUCATIONAL SOFTWARE CLUB

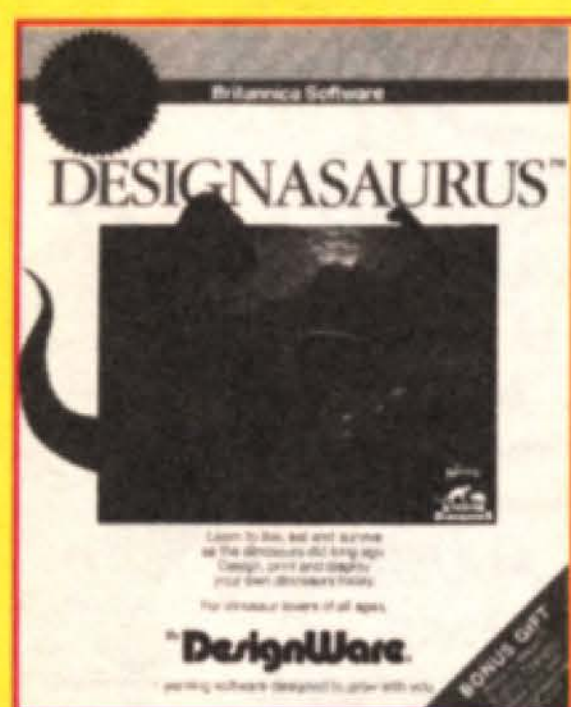
PHONE 0702 600557 • FAX 0702 613747

32A SOUTHCHURCH ROAD • SOUTHBEND-ON-SEA • ESSEX SS1 2ND

We Have Over 70 Top Class Educational Programs For The Amiga, IBM/PC & Atari ST.

All Ages Covered From 3 To Adult.

Contact Us Now For Your **Free Catalogue.**



Designasaurus: Print, Create & Survive like the dinosaurs in their own ecosystem. Ages 6+ Amiga or IBM/PC £29.95.

SPECIAL OFFERS

EARLY LEARNING MATHS: A really fun collection of 12 maths programs to quiz your kids. Ideal for 4 to 9 year olds. Atari ST/Amiga £14.95

LIZZY'S SPELLICOPTER: Fly Lizzy's Helicopter around and spell the words presented. Lots of speech and graphics make this the best Amiga Spelling Program. Ages from 4 to 9. Amiga Only £14.95

FUNTIME 1: A collection of 4 simple programs. Excellent graphic presentation for the Under 6's. Amiga/Atari ST £3.00

FUNTIME 2: Another set of 4 simple yet brilliant programs. Excellent graphic presentation. Great for the Under 6's. Amiga/Atari ST £3.00

BOX of 10 5¼" Branded Floppy Disks Only £4.50 Per Box

THIS IS NOT A MISPRINT

ONLY BY PURCHASING OVER 3 MILLION DISKS A YEAR CAN WE OFFER THE FOLLOWING SENSATIONAL PRICES...

Every diskette supplied by Dial a Disc is certified and tested 100% error free.

200 DSDD 3.5" ONLY £60.00

3.5" DSDD 135TPI

50 £22	100 £33	150 £48	200 £60	400 £115	500 £135
-----------	------------	------------	------------	-------------	-------------

3.5" HD ONLY
60p each

5.25" DSDD ONLY
23p each

5.25" DSHD ONLY
40p each

DISK STORAGE BOXES

3.5" 40 capacity.....	£2.95
3.5" 80 capacity.....	£3.45
3.5" 100 capacity.....	£3.95
only if bought with disks	

3.5" HD 100 FOR ONLY £57.50

OUR PROMISE IS SIMPLE

100% SATISFACTION OR MONEY BACK

DIAL A DISC

203 Southborough Lane, Bromley, Kent, BR2 8AR.
081-467 0131

ALL prices include VAT & delivery. All offers subject to availability. E/OE.

The object Adv



Up to now we've created an adventure landscape that can be moved around at will. Now let's move on to the next important part of your adventure manager - placing the objects.

Most adventures use objects. Remember the ones in The Golden Crown in February's issue of Let's Compute?

There you had objects like a large bucket and the cloak. You could pick them up and use them as part of the game.

Now we are going to put objects into OUR adventure. You can see them in the DATA in Lines 6000 to 6300. Add these to your program.

Note that the last DATA line contains a *rogue value*. It works like the one in the word list. This was explained in the March issue.

Remember, using this system means you can easily add things to your list. You don't have to tell the program how many things there are in it.



Look at the DATA lines. You can see that each object has a number in front of it. This is the number of the room the object will start in.

Some of the objects have zeros in front of them. This just means they will not appear in any room at the start. You can think of room zero as a sort of store room for your objects when you're not using them.

Notice that objects are not always things you can pick up and use. They can do other things. For instance, the red light in the air lock will eventually be changed to the green light.

The objects and room numbers are in DATA lines. Your computer now needs to READ them.

This is done in the same way as we did with the words. Type in lines 1080 to 1140.



These are now part of the setting up routine. Every time you run the program the objects will be put in the correct rooms.

The array OBJECT\$ contains the descriptions of the objects. OBJROOM holds the room numbers. So OBJECT\$(1) contains laser rifle and OBJROOM(1) contains 2, the room the laser starts in.

Like the room numbers, each object is numbered by its place in the list. For example, the laser is number 1, the space suit number 4 and the fixed cable number 14.

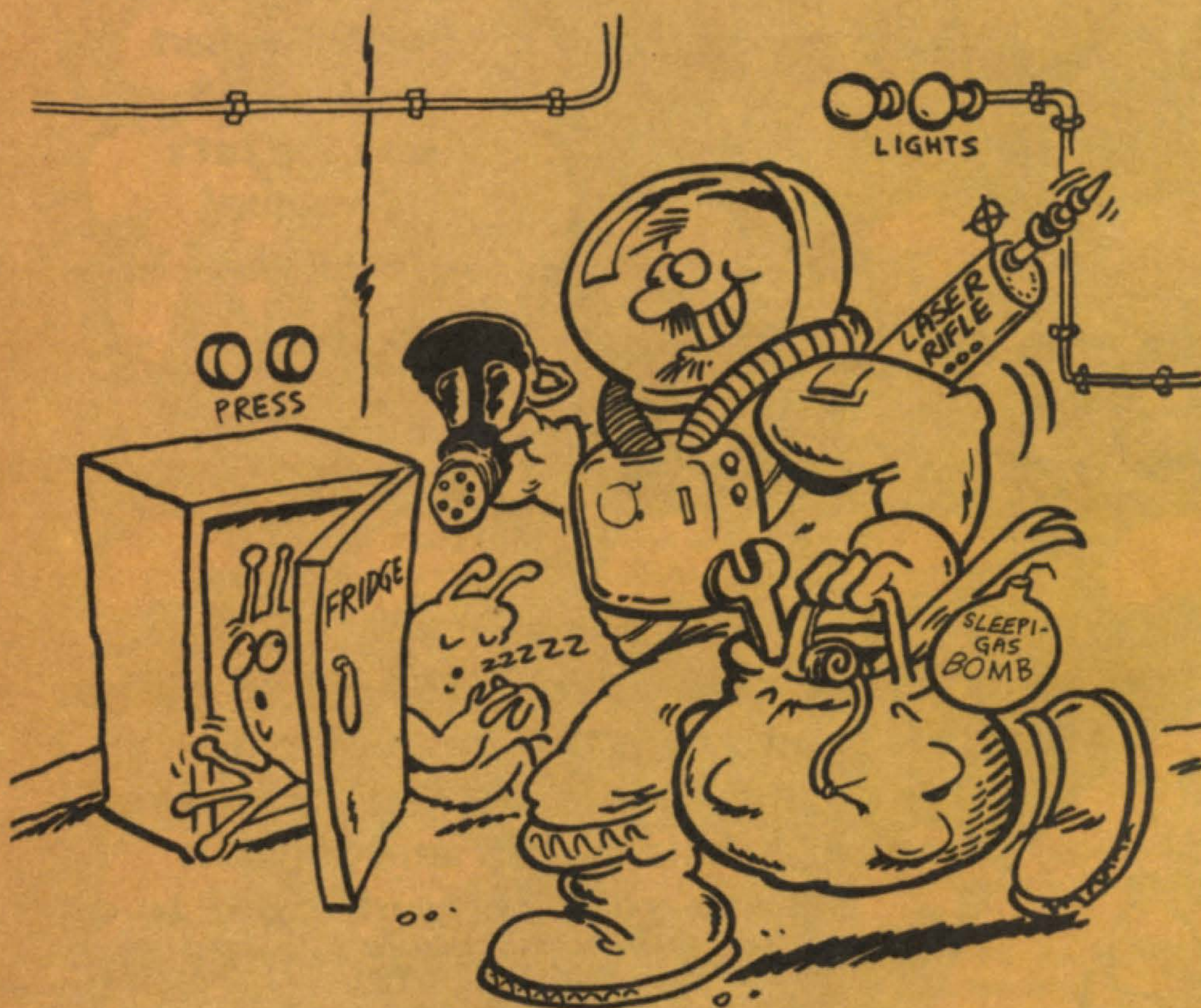
Next we want to display the objects as part of the

ect of an

Add a new dimension to
your adventure by placing
things to pick up

PART 3

venture!



IS YOUR COMPUTER HERE?

This program works on a BBC, Archimedes, Electron, CPC, Atari ST(Stos), Amiga and PC(GW-Basic). It will not work on a C64/128 or Spectrum.

```

35 IF W(1)>=1 and W(1)<=10 GO SUB 271
0:FOR MORE THAN 10 MOVEMENT WORDS (NOT ROOMS AS STATED IN APRIL) CHANGE THE 10 IN THIS LINE.
1070 REM NOW SORT OUT THE OBJECTS - FIRST COUNT THEM
1080 RESTORE 6000
1085 0=0:REM THIS WILL BE THE NUMBER OF OBJECTS
1090 READX,X$:REM THESE VARIABLES ARE JUST FOR COUNTING
1100 IF X>-1 0=0+1:GOTO 1090:REM NOT THE END OF THE LIST?
1110 DIM OBJROOM(0):DIM OBJECTS(0)
1120 REM NOW PUT THE OBJECTS INTO AN ARRAY
1130 RESTORE 6000:REM BACK TO THE START OF THE OBJECT LIST
1140 FOR X=1 TO 0:READ OBJROOM(X),OBJECTS(X):NEXT
2040 REM NOW TELL US WHAT YOU CAN SEE
2050 PRINT:PRINT "You can see :";
2060 FOR X=1 TO 0
2070 IF OBJROOM(X)=ROOM PRINTTAB(14);OBJECTS(X)
2080 NEXT X:PRINT
5990 REM OBJECTS START HERE
6000 DATA 2,lazer rifle
6010 DATA 4,sleepi-gas bomb
6020 DATA 1,red button
6030 DATA 1,green button
6040 DATA 0,space suit
6050 DATA 19,spanner
6060 DATA 18,crow-bar
6070 DATA 8,red light
6080 DATA 0,green light
6090 DATA 13,hungry alien
6100 DATA 0,sleeping alien
6110 DATA 12,gas mask
6120 DATA 16, loose cable
6130 DATA 0,fixed cable
6300 DATA -1,X
    
```

room description. Lines 2050 to 2080 do this. So type them in.

The lines search the **OBJROOM** array. As they do they check if any of the room numbers match the one you are in. If they do the description of the object is printed underneath the room description.

Now try **RUN**ning the program. You'll see our objects in your adventure. Why not put in some of your own?

You could put teachers into classrooms, tools into sheds or dragons into caves. Just use your imagination and scatter the map with all sorts of things!

● **At the moment the objects are just things in the rooms. Next month we'll find out how to carry them around. Then they'll really become part of the adventure!**

A CHALLENGE

So far your program displays the objects in a room. This is fine as long as there are some objects there!

If there aren't it prints:

You can see :

Can you alter the display routine yourself? It shouldn't print anything at all if there are no objects in the room. Or perhaps you could make it print:

You can see : nothing interesting

When



No.4

the rain drops in

*Have you got the
bottle to chart
the rainfall
with the Let's
Compute!
rain gauge?*

...

**Pete's Projects are always simple to make.
But this month's is the easiest yet.**

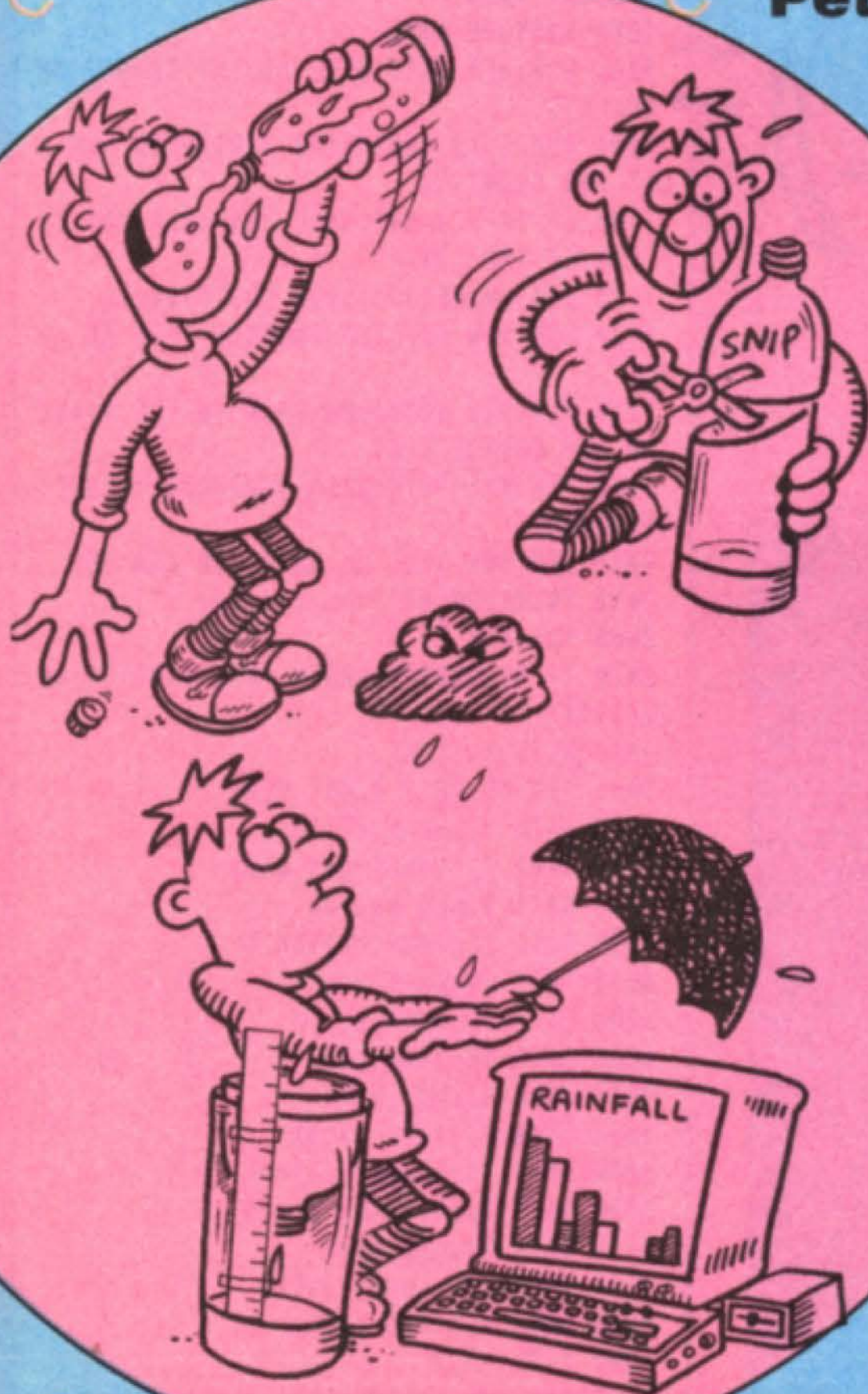
**It's a rain gauge. And it's just made
from a plastic bottle and a ruler.**

**What's your computer got to do
with that? Well what better way
to keep records than use it?**

**The program we've given here
draws a barchart from your
readings. But you can use other
programs to process your
results.**

**You could, for example, feed
them into a spreadsheet – just
like the real weathermen do!**

**Once you've made the gauge
take readings at regular intervals –
say daily. Then you can compare
yours with the ones that are often
shown on TV.**



MAKING THE RAIN GAUGE

First you need an empty plastic bottle. It must have straight sides like the one in our picture.

Now cut the top part off to form a funnel. It's a good idea to get your parents to help with that.

Turn the funnel you have made upside down. Push it into the other part of the bottle to make your gauge.

Now tape a ruler to the side of the gauge. The zero of the ruler should be above any bumps or shapes at the bottom of the bottle.

Pour water into the gauge up to the zero mark on the ruler. It's now ready to use.

Put it in your garden. Try to find a fairly shaded place so the rain doesn't evaporate. Now it's just a matter waiting. When it rains the gauge will collect it. You can check the depth by just reading the ruler.

WHY USE A FUNNEL?

At first it may seem pointless having a funnel into the base. After all, rain would fall into the bottom half of a bottle anyway.

The purpose of the funnel is to stop the water in the gauge evaporating.

As an experiment, make another gauge without the funnel. Stand it in your garden next to the one that has one. Check if the readings from both are the same.

USING YOUR COMPUTER

If you want to convert your numbers into a barchart here's a program to do it. Type it in, SAVE it and RUN it.

You'll first be asked *How many readings?* The most you are allowed to type in is about 35 – the exact number depends on which type of computer you have.

Next you type in the readings. One after each question mark that appears.

The biggest number that will be correctly displayed is about 25. If you type a larger one the biggest bar that will fit will be

drawn. Once you've typed in all your readings your barchart will be drawn. If there's enough room a gap will be left between each bar.

The numbers up the side are automatically chosen so the bars are as tall as possible.

Note that, although this program is intended to be used with Pete's rain gauge it has other uses. With it you can draw barcharts of any numbers you want such as the how much money you spend each day.

```
10 REM PETE'S PROJECT BAR CHART
20 H=30:W=40:VDU23,255,255,255,255,255,2
55,255,255,255:CS=CHR$(255):MODE4
30 PRINT "HOW MANY READINGS";:INPUT D
40 IF D>W-5 THEN PRINT "TOO BIG":GOTO 30
50 LET E=1:IF D<(W-5)/2 THEN LET E=2
60 DIM R(D):LET L=(W-D)/2
70 LET M=0:FOR I = 1 TO D
80 PRINT "READING ";I;:INPUT R(I)
90 IF R(I)>H-3 THEN LET R(I)=H-3
100 IF R(I)>M THEN LET M=R(I)
110 NEXT I
120 LET S=1:IF M<(H-4)/2 THEN LET S=0.5
130 IF M<(H-4)/4 THEN LET S=0.25
140 FOR I=1 TO D:LET R(I)=INT(R(I)*(1/S))
:NEXT I
150 CLS:PRINT
160 PRINT"THE LET'S COMPUTE! RAINFALL CHA
RT"
170 FOR I=H TO 4 STEP -4
180 LET X=L-2-D*(E-1):LET Y=I:LET Z$=STR$
(ABS(I-H)*S):GOSUB 260
190 NEXT I
200 FOR I=1 TO D
210 IF R(I)<=0 THEN GOTO 240
220 LET X=I+E+L-D*(E-1):LET Y=H-R(I):LET
Z$=CS:GOSUB 260
230 LET R(I)=R(I)-1:GOTO 210
240 NEXT I
250 PRINT:END
260 PRINT TAB(X,Y)Z$:RETURN
```

IS YOUR COMPUTER HERE?

BBC/Electron/Archimedes

The program works as shown.

Amiga

Change these lines:

```
20 LET H=20:LET W=80:LET CS=" ":COLOR 1
,2:CLS
220 LET X=I+E+L-D*(E-1):LET Y=H-R(I):LET
Z$=CS:COLOR,1:GOSUB 260
260 LOCATE Y+1,X+1:PRINT Z$:COLOR 1,2:RE
TURN
```

PC (GW-Basic)

Change these lines:

```
20 LET H=20:LET W=80:LET CS=CHR$(219):CL
S
260 LOCATE Y+1,X+1:PRINT Z$:RETURN
```

Amstrad CPC

Change these lines:

```
20 H=20:W=80:CS=CHR$(143):MODE 2
260 LOCATE X,Y:PRINT Z$:RETURN
```

ST(Stos)

Change these lines:

```
20 MODE 1:KEY OFF:CURS OFF:HIDE:H=20:W=
80:CS=" "
220 LET X=I+E+L-D*(E-1):LET Y=H-R(I):LET
Z$=CS:PAPER 15:GOSUB 260
260 LOCATE X,Y:PRINT Z$:PAPER 0:RETURN
```

Spectrum

Change these lines:

```
20 LET H=20:LET W=32:LET CS=CHR$(143):C
LS
250 PRINT:STOP
260 PRINT AT Y,X;Z$:RETURN
```

Commodore 64/128

Change these lines:

```
20 H=20:W=40:CS=CHR$(18)+" ":PRINT CHR$(
147)
150 PRINT CHR$(147);:PRINT
260 POKE 211,X:POKE 214,Y:SYS 58732:P
RINT Z$;:RETURN
```

NEXT MONTH:
Make Pete's nuclear
reactor puzzle

Last month we said that it would be really useful to have a program that doubles your computer's memory. And we printed the program – it's shown again on the right.

We didn't actually say that this program doubles the memory. It WAS the April issue after all!

So you typed it in? Don't worry! It wasn't such a waste of time after all. You can now use it to have some fun with your pals.

At the bottom of this page we explain how the program works. By changing the DATA numbers starting at Line 100 you can make it print ANY message you like.

The numbers are called the Ascii codes of the letters. Every letter has a number to represent it. Most computer instruction books give a list of the letters and their numbers.

But as usual, *Let's Compute!* makes it easy. Here's a program to print the Ascii codes for you whenever you like:

```
10 INPUT A$:PRINT ASC(A$):GOTO 10
```

On a Spectrum use CODE in place of ASC.

You'll notice that there are a lot of 13s in our DATA. These make your computer start a new line of type.

Try other numbers in the range 0 to 255. For example 12. On most computers that clears the screen.

Note that the last number in the DATA must be 99. Line 30 checks for it. More DATA is read until it is found. You could change it to 999 if you want to use code 99 anywhere else.

You can also make the program *interactive*. That means someone else can type the codes in and find out the secret message.

To do that change Line 20 to:

```
20 LET M=M+1:INPUT L(M)
```

If you make this change, Lines 100 to 200 now do nothing. Take them out if you want.

When the program is run, just type in your numbers. Type each one after a question mark appears – don't forget to press Return each time.

When you have typed them all in type 99. Your message will be displayed.

You've now got the basis of a coding and a decoding program. And you can go further. You can make the code harder to crack.

For example, you can add one to each number. Then change line 70 to take it off automatically:

```
70 PRINT CHR$(L(T)-1);
```

● **It's up to you to think of more sneaky tricks to make the code harder to crack.**

That clever program wasn't all it seemed to be!

```
10 LET M=0:DIM L(100)
20 LET M=M+1:READ L(M)
30 IF L(M)<>99 THEN GOTO 20
40 LET T=M*2:REM ** DOUBLES IT!!
50 FOR T=1 TO M-1
60 IF L(T)=13 THEN PRINT:GOTO 80
70 PRINT CHR$(L(T));
80 NEXT T
90 REM THE SECRET IS IN THE NUMBERS
100 DATA 13,73,84,32,73,83,32,73,77
110 DATA 80,79,83,83,73,66,76,69,32
120 DATA 84,79,32,87,82,73,84,69,32
130 DATA 65,13,80,82,79,71,82,65,77
140 DATA 32,84,79,32,68,79,85,66,76
150 DATA 69,32,84,72,69,32,77,69,77
160 DATA 79,82,89,46,13,89,79,85,32
170 DATA 72,65,86,69,32,66,69,69,78
180 DATA 32,84,82,73,67,75,69,68,46
190 DATA 13,65,80,82,73,76,32,70,79
200 DATA 79,76,33,13,13,99
```

Were YOU fooled?

HOW THE PROGRAM WORKS ...

Lines 10 to 30 read in all the numbers. They are stored in the computer memory locations L(1), L(2), L(3), and so on.

Line 40 does nothing. It was put there to make you think something was being doubled. Miss it out now.

Lines 50 to 80 print out the message.

Line 90 is a message for you. Your computer doesn't need it so you can miss it out if you want.

Lines 100 to 200 contain the numbers that will be printed as the message. Change them if you want. Remember, you must have a 99 as the last one.

But CAN memory be doubled?

It's impossible for any program to double the memory of a computer. Instead you need more memory chips.

But programs can sometimes be written to make it seem as if your computer has more memory. They are completely different for different kinds of computer.

Your computer reserves some of its memory for controlling disc drives and tape units. This is only needed for short periods.

For instance, say you want to play a game. Your computer uses some of its memory to work the tape or disc unit to load it. But once that's done it doesn't need it any more.

A program can be written so that the extra memory can be used for something else. It could be used to store the high score table.

So the useful memory can be extended by clever programming. But it can't be doubled. A short program like the one we printed last month couldn't possibly do it.

So, if you were fooled, watch out next April!


```

10 REM PACMAN
20 CLS:VDU 23;8202;0;0;0;:*FX11
30 DIM M$(18,17):DIM G(3,4):DIM GS(3)
:LET PWRC=20
40 LET PWRTIM=0:LET TMR=0:LET SC=0:LE
T L1=3:LET L2=3:GOSUB 980
50 GOSUB 1070
60 GOSUB 760:GOSUB 780:GOSUB 220
70 GOSUB 970:IF PWRTIM-TMR>0 THEN LET
TX=9:LET TY=8
80 IF PWRTIM-TMR<0 THEN LET TX=PX:LET
TY=PY
90 GOSUB 300
100 LET AS=STR$(SC):LET LSC=LEN(AS):LE
T Q=32-LSC:LET Z=1
110 GOSUB 940
120 GOSUB 220
130 LET Q=31:LET Z=3:LET AS=STR$(L1):G
OSUB 940
140 IF FOOD>0 AND L1=L2 THEN GOTO 60
150 IF L1<>L2 THEN LET L2=L2-1:GOSUB 1
070:GOSUB 1110
160 IF FOOD=0 THEN GOSUB 980
170 IF L1>0 THEN GOTO 50
180 GOSUB 950:IF K$<>" " THEN GOTO 180
190 LET Q=24:LET Z=10:LET AS="GAME OVE
R":GOSUB 940
200 LET Z=12:LET AS="PRESS ANY KEY":GO
SUB 940
210 GOTO 40
220 LET HIT=0:FOR L=1 TO 3
230 IF G(L,1)=PX AND G(L,2)=PY AND TMR
>PWRTIM AND G(L,4)=1 THEN LET HIT=1
240 IF G(L,1)=PX AND G(L,2)=PY AND TMR
<PWRTIM AND G(L,4)=2 THEN GOTO 260
250 GOTO 270
260 LET G(L,4)=3:LET GS(L)=CHR$(34):LE
T SC=SC+150
270 NEXT L
280 IF HIT=1 THEN LET L1=L1-1
290 RETURN
300 FOR L=1 TO 3
310 LET X=G(L,1):LET Y=G(L,2):LET GD=G
(L,3):LET GC=G(L,4)
320 LET Q=X:LET Z=Y:LET AS=MS(X,Y):GOS
UB 940
330 IF G(L,4)=3 THEN LET TX=9:LET TY=8
:IF X=9 AND Y=7 THEN LET GD=2:GOTO 350
340 GOSUB 450
350 IF GD=1 OR GD=3 THEN LET X=X+(GD-2
)
360 IF GD=0 OR GD=2 THEN LET Y=Y+(GD-1
)
370 LET XX=X:IF XX=1 THEN LET X=16
380 IF XX=17 THEN LET X=2
390 IF G(L,4)=3 AND G(L,1)=9 AND G(L,2
)=9 THEN LET GS(L)="*":LET GC=1
400 IF G(L,4)=2 AND TMR>PWRTIM THEN LE
T GC=1:LET GS(L)="*"
410 LET Q=X:LET Z=Y:LET AS=GS(L):GOSUB
940

```

```

420 LET G(L,1)=X:LET G(L,2)=Y:LET G(L,
3)=GD:LET G(L,4)=GC
430 NEXT L
440 RETURN
450 GOSUB 760:LET N=RND(6)
460 LET XX=-2:IF GD<2 THEN LET XX=2
470 LET FF=0:IF GD=1 OR GD=3 THEN LET
FF=1
480 IF FF=1 THEN IF M$(X+(GD-2),Y)="*"
AND US="*" AND DS="*" THEN LET GD=GD+XX:
RETURN
490 IF FF=0 THEN IF M$(X,Y+(GD-1))="*"
AND LS="*" AND RS="*" THEN LET GD=GD+XX:
RETURN
500 IF FF=1 AND US="*" AND DS="*" THEN
RETURN
510 IF FF=0 AND LS="*" AND RS="*" THEN
RETURN
520 IF GD=0 AND US="*" THEN GOSUB 640:
RETURN
530 IF GD=1 AND LS="*" THEN GOSUB 700:
RETURN
540 IF GD=2 AND DS="*" THEN GOSUB 640:
RETURN
550 IF GD=3 AND RS="*" THEN GOSUB 700:
RETURN
560 IF N>4 THEN RETURN
570 IF GD=1 OR GD=3 THEN GOSUB 610:RET
URN
580 IF LS="*" AND TX>X THEN LET GD=3:R
ETURN
590 IF RS="*" AND TX<X THEN LET GD=1
600 RETURN
610 IF US="*" AND TY>Y THEN LET GD=2:R
ETURN
620 IF DS="*" AND(TY<Y OR X=9) THEN LE
T GD=0
630 RETURN
640 IF LS="*" THEN LET GD=3:RETURN
650 IF RS="*" THEN LET GD=1:RETURN
660 IF TX>X AND N<5 THEN LET GD=3:RETU
RN
670 IF TX<X AND N<5 THEN LET GD=1:RET
URN
680 LET GD=3:IF N=5 THEN LET GD=1
690 RETURN
700 IF US="*" THEN LET GD=2:RETURN
710 IF DS="*" THEN LET GD=0:RETURN
720 IF TY>Y AND N<5 THEN LET GD=2:RETU
RN
730 IF TY<Y AND N<5 THEN LET GD=0:RETU
RN
740 LET GD=0:IF N=5 THEN LET GD=2
750 RETURN
760 LET OS=MS(X,Y):LET US=MS(X,Y-1):LE
T DS=MS(X,Y+1):LET LS=MS(X-1,Y)
770 LET RS=MS(X+1,Y):RETURN
780 GOSUB 950:IF K$<>" " THEN LET CS=K$
790 LET X=PX:LET Y=PY:GOSUB 760
800 LET Q=PX:LET Z=PY:LET AS=" ":GOSUB
940:LET M$(PX,PY)=" "

```

```

810 IF CS="A" AND US<>"#" THEN LET PY=
PY-1
820 IF CS="Z" AND DS<>"#" THEN LET PY=
PY+1
830 IF CS="N" AND LS<>"#" THEN LET PX=
PX-1
840 IF CS="M" AND RS<>"#" THEN LET PX=
PX+1
850 IF OS="." THEN LET FOOD=FOOD-1:LET
SC=SC+10
860 IF OS<>"+" THEN GOTO 900
870 LET FOOD=FOOD-1:LET SC=SC+100:LET
PWRTIM=TMR+PWRC
880 FOR L=1 TO 3:IF GS(L)="*" THEN LET
GS(L)="0":LET G(L,4)=2
890 NEXT L
900 LET XX=PX:IF XX=1 THEN LET PX=16
910 IF XX=17 THEN LET PX=2
920 LET Q=PX:LET Z=PY:LET AS="C":GOSUB
940
930 RETURN
940 PRINT TAB(Q,Z);AS;:RETURN
950 LET K$=INKEY$(0):RETURN
960 RETURN
970 LET TMR=TMR+1:RETURN
980 RESTORE
990 FOR A=1 TO 16
1000 READ AS:LET Q=1:LET Z=A:GOSUB 940
1010 FOR B=1 TO 17:LET M$(B,A)=MID$(AS,
B,1):NEXT B
1020 NEXT A
1030 LET Q=20:LET Z=1:LET AS="Score:
":GOSUB 940
1040 LET Z=3:LET AS="Lives:":GOSUB 940
1050 LET FOOD=127
1060 RETURN
1070 FOR L=1 TO 3:LET G(L,1)=L+7:LET G(
L,2)=9:LET G(L,3)=0:LET G(L,4)=1:LET GS
(L)="*":NEXT L
1080 LET PX=9:LET PY=13:LET X=PX:LET Y=
PY
1090 LET CS=" "
1100 RETURN
1110 FOR Z=1 TO 16:FOR Q=1 TO 17:LET AS
=MS(Q,Z):GOSUB 940:NEXT Q:NEXT Z:RETURN
1120 DATA "#####
1130 DATA "#+.....#.....+
1140 DATA "#.#.###.#.###.#.
1150 DATA "#.#.###.#.###.#.
1160 DATA "#.....#.#.....
1170 DATA "#.#####.#.#####
1180 DATA "#.....#.....#
1190 DATA "###.#.##.##.###
1200 DATA " ...#.#. #.#...
1210 DATA "###.#.#####.#.###
1220 DATA "#.....#.....#
1230 DATA "#.#.#####.#.###
1240 DATA "#.#.###.#.###.#.
1250 DATA "#.###.#.###.#.###
1260 DATA "#+.....#.....+
1270 DATA "#####

```

IS YOUR COMPUTER HERE?

BBC and Electron

The program works as shown

Archimedes and BBC A3000

Change these lines:

```

940 TIME=0:PRINT TAB(Q,Z);AS;:REPEAT U
NTIL TIME>4:RETURN:REM Change the 4 to s
peed up or slow down.

```

Amiga and PC(GW-Basic)

Change these lines:

```

10 SCREEN 1:RANDOMIZE TIMER:CLS
450 GOSUB 760:LET N=INT(RND*6)
940 LOCATE Z+1,Q+1:PRINT AS:RETURN
950 LET K$=INKEY$:RETURN

```

ST(STOS)

Change these lines:

```

10 MODE 1:KEY OFF:CURS OFF:HIDE
450 GOSUB 760:LET N=RND(6)
940 LOCATE Q, Z:PRINT AS:RETURN
950 LET K$=INKEY$:RETURN

```

Commodore 64

Leave out all the words: LET

Change these lines:

```

10 PRINT CHR$(147);:X=RND(-TI)
450 GOSUB 760:LET N=INT(RND(0)*6+1)
940 POKE 211,Q+1:POKE 214,Z+1:SYS 5873
2:PRINT AS;:RETURN
950 GET K$:RETURN

```

Amstrad CPC

Change these lines:

```

10 CLS:RANDOMIZE TIME
450 GOSUB 760:LET N=INT(6*RND+1)
940 LOCATE Q+1,Z+1:PRINT AS:RETURN
950 LET K$=INKEY$:RETURN

```

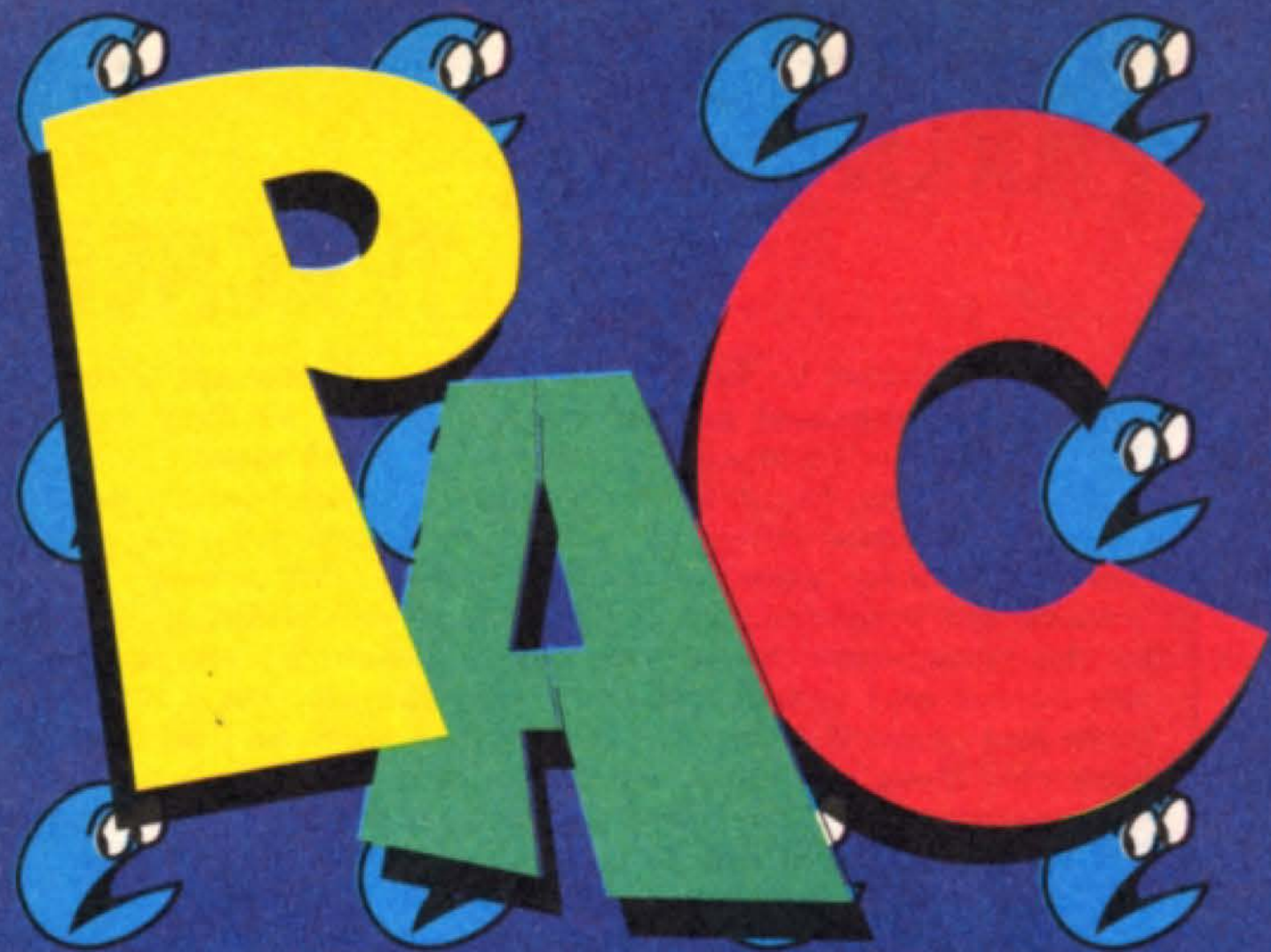
Spectrum

Change these lines:

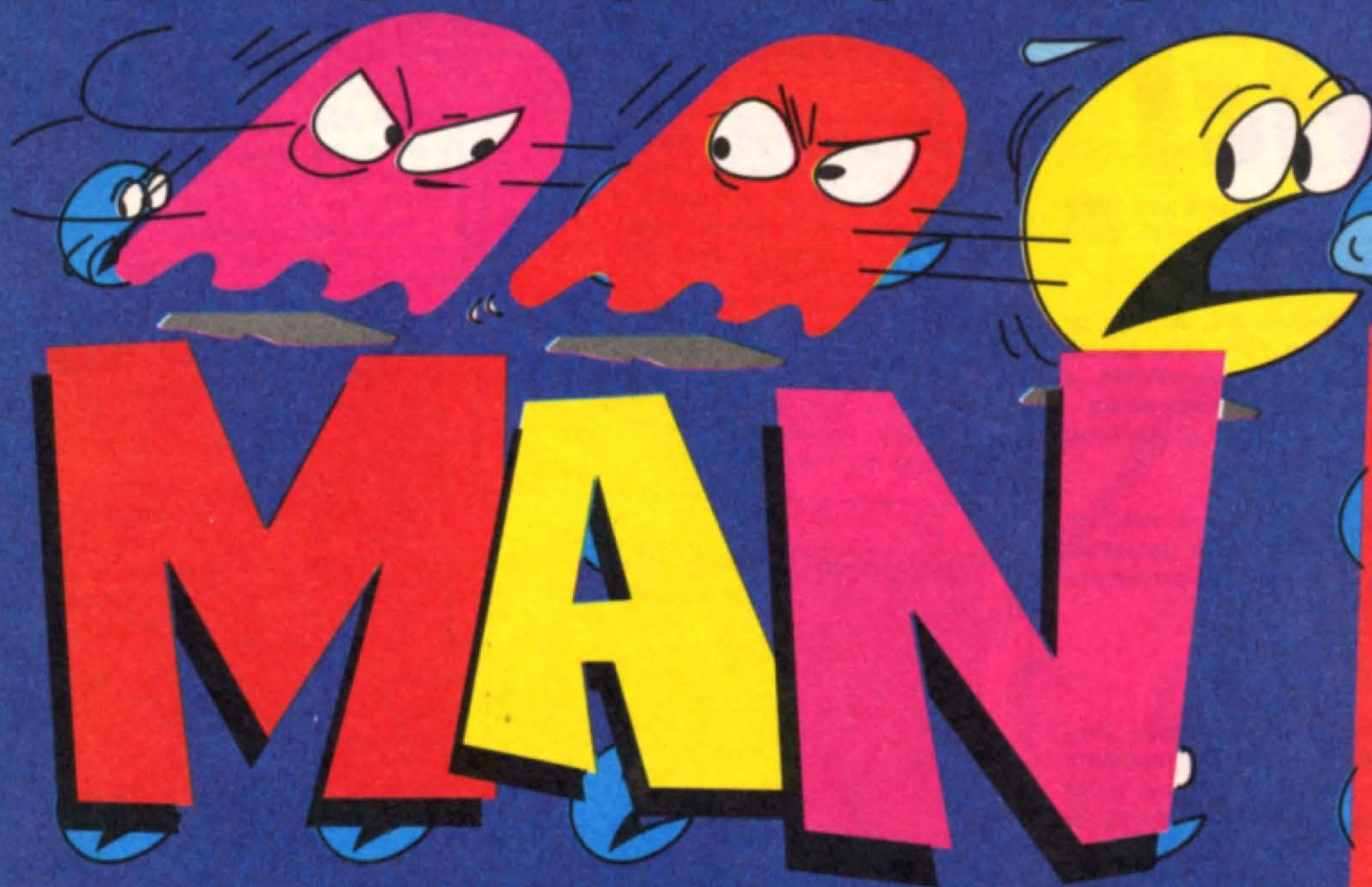
```

10 CLS:RANDOMIZE
450 GOSUB 760:LET N=INT(6*RND+1)
940 PRINT AT Z,Q;AS;:RETURN
950 LET K$=INKEY$:RETURN
1010 FOR B=1 TO 17:LET M$(B,A)=AS(B):NE
XT B

```


The letters 'PAC' are rendered in a large, bold, 3D style. The 'P' is yellow, the 'A' is green, and the 'C' is red. They are surrounded by several small, blue, cartoonish Pac-Man characters with large eyes and open mouths, appearing to be floating or swimming around the letters.

**David McLachlan
presents his own
version of one of
the all-time top
computer games**

The letters 'MAN' are rendered in a large, bold, 3D style. The 'M' is red, the 'A' is yellow, and the 'N' is pink. They are surrounded by several small, blue, cartoonish Pac-Man characters, some of which are eating the letters.

The classic Space Invaders started life in pubs and arcades. It was a hit long before computers became popular.

Then along came another game which was to knock it for six.

It's name was Pacman. And it became an instant success – the very first chase-around-the-maze game. Over the years there have been many copies of the original idea.

This special *Let's Compute!* version will run on most home computers. The changes needed for the popular ones are given. Pacman is represented by a letter C. The ghosts are asterisks – (*).

When you pick up the special power pill – (+) – the ghosts become harmless. They show this by changing into a letter O.

While they are like this you can eat them. If you do they change into quotes – (") – and return to their den. They are then reborn and chase you again.

The rules of Pacman

If you're too young, too old or have been on another planet you may not have heard of Pacman.

Here's how to play it.

The object is to move your chomping ball-shaped man around the maze. You munch pills and bonus items as you go along.

While you're doing it, stay out of the way of three ghosts! They'll continuously harrass you. Contact with one means certain death.

Try to pick up a special power pill. There's one in each of the four corners of the maze. The ghosts will then become harmless.

They're now edible and worth points. Eat them quick! They soon turn back to normal.

Keys to use

Up	A
Down	Z
Left	N
Right	M

*Make sure caps lock
or shift lock is pressed*

Sound advice

There are lots of things you can do to make this into YOUR version of Pacman. You could add colour, use different shapes, add extra objects to pick up or put some sounds into the program. Try putting in a click sound each time a pill is eaten. You could also add an explosion each time Pacman is eaten by a ghost. An increasing tone while the ghosts are edible gives you an idea how long you've got left.

Try to put that in too!

Part 8

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

BUILD YOUR OWN DATABASE

In his final article
Roland Waddilove
shows you how to
rub out your
mistakes

CHANGE the cards

Here it is: The last routine of the *Let's Compute!* database. Add it to the previous bits and you have a fully working mini-database!

This month's lines form another self contained subroutine. They start at Line 6000. What they do is let you to change what you have written on a card.

This is useful if you make a mistake and want to correct it. But that's not all!

Suppose you are using your database to store your friends' addresses. If one moves, the address will need changing.

Or you could be keeping a list of your videos or music cassettes on computer. If you wipe a tape and replace the recording with something else you need to change the information in your database.

LOAD the program that you've built up over the past few months. Then add the lines shown on the right. SAVE your new version and RUN it.

Type in a few cards so you can check the new bit. Now, to change a card just choose option 6 from the menu. Follow the instructions that will be printed on your screen and you can easily modify your data.

Your *Let's Compute!* database is now complete. But that doesn't mean you have to stop tweaking it. There are lots of improvements and new features you can add.

The idea is to use this program as a framework and create your own customised database.

HERE'S HOW IT WORKS

The information on the cards is stored in an array called *C\$(255)*. Each card is five lines long. If nothing has been written on a card the first line is simply an asterisk (*).

So Line 6020 checks for an asterisk and jumps over most of the lines in the subroutine if it finds one. If there is an asterisk nothing has been typed in so there's nothing to change!

If the card has been written on it is printed on the screen. The computer then asks if you want to change the information shown. If you say No, the program jumps over a bit of code and looks for the next card. But if you answer yes the program asks you which line of the card you would like to alter.

WARNING: Be careful what you enter

here. The program doesn't check if the number you have typed makes sense. If you type in something silly you might change what's been written on a completely different card.

Line 6110 is the one that asks which line you want to change. Then Line 6120 prints what's already there and 6130 give you the chance to type in a new one. The new line is stored in the array and the program continues with the next card.

It's a very simple system, but it works. You could try modifying the subroutine so that you can alter another line on the card at the same time. At the moment you can only change one. It's not a difficult problem to solve, but it's one I'll leave up to you!

CARD INDEX DATABASE

1. Load a box of cards
2. Save a box of cards
3. Read/search for a card
4. Write on a new card
5. Throw away old cards
6. Change something on a card
7. Print the cards
8. Sort the cards into order
9. Stop the program

Cards free=45/Cards used=6

What do you want to do?

Please type in a number (1- 9)

Change the cards program

```
6000 REM ===== Change card =====
6010 X=1
6020 IF C$(X)="*" THEN GOTO 6150
6030 CLS
6040 FOR Y=0 TO 4
6050 PRINT "Line";Y+1;":";C$(X+Y)
6060 NEXT Y
6070 PRINT
6080 INPUT "Change this card (Y or N)";K$
6090 IF K$<>"Y" AND K$<>"y" THEN GOTO 6150
6100 PRINT
6110 INPUT "Which line";Y
6120 PRINT "Line";Y;":";
6130 INPUT T$
6140 C$(X+Y-1)=T$
6150 X=X+5
6160 IF X<255 THEN GOTO 6020
6170 RETURN
```

This
program
works on
all home
micros

IS THIS YOUR COMPUTER?

**Archimedes, Electron,
BBC, Atari ST(Stos),
Amiga(Amos), PC(GW-Basic)**

The program works as shown.

Spectrum

Change this line:

```
6020 IF C$(X)=B$ THEN GOTO 6150
```

Amstrad CPC

In the January issue, Line 2080 –
which saves the cards – should have
been:

```
2080 WRITE #9,C$(X)
```

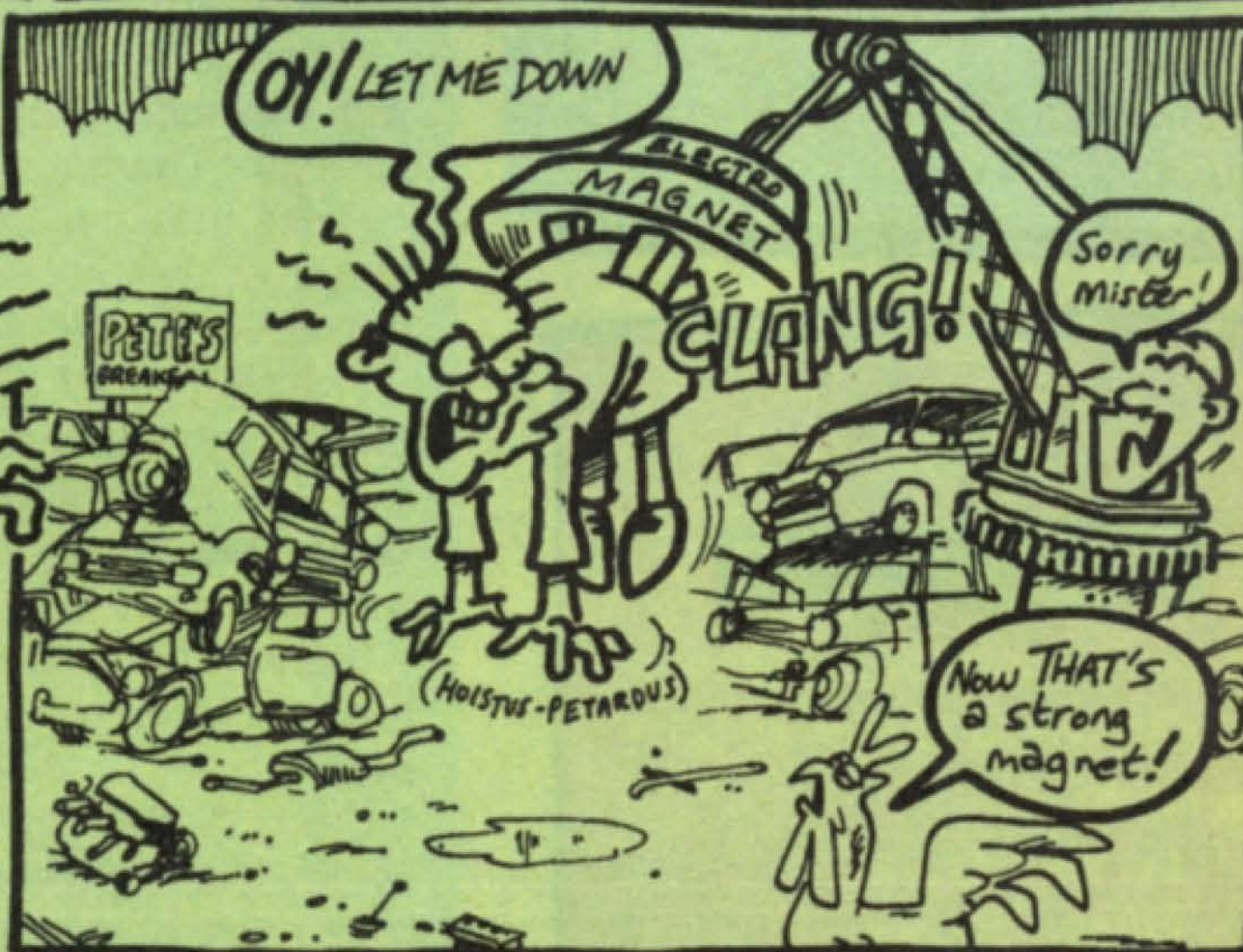
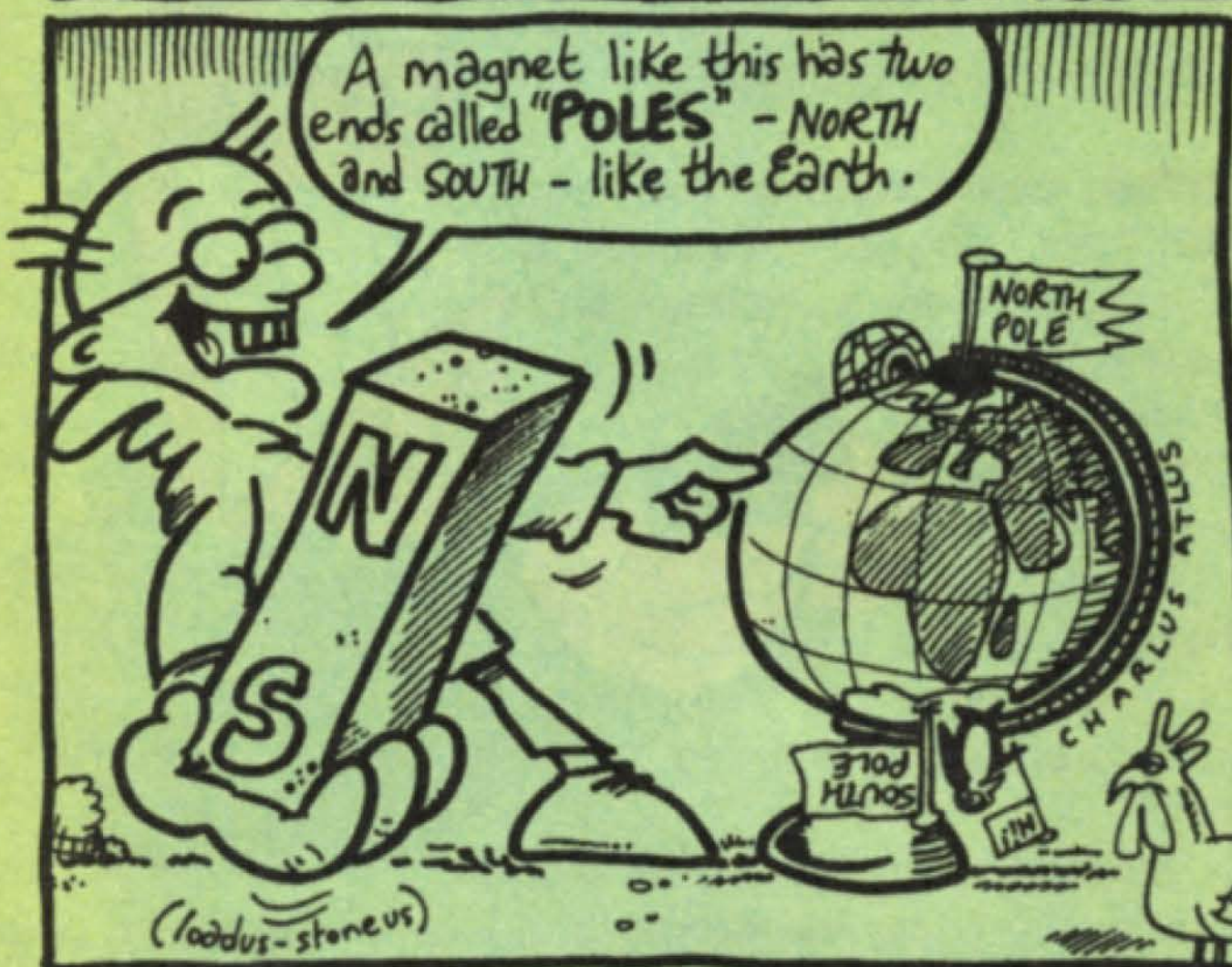
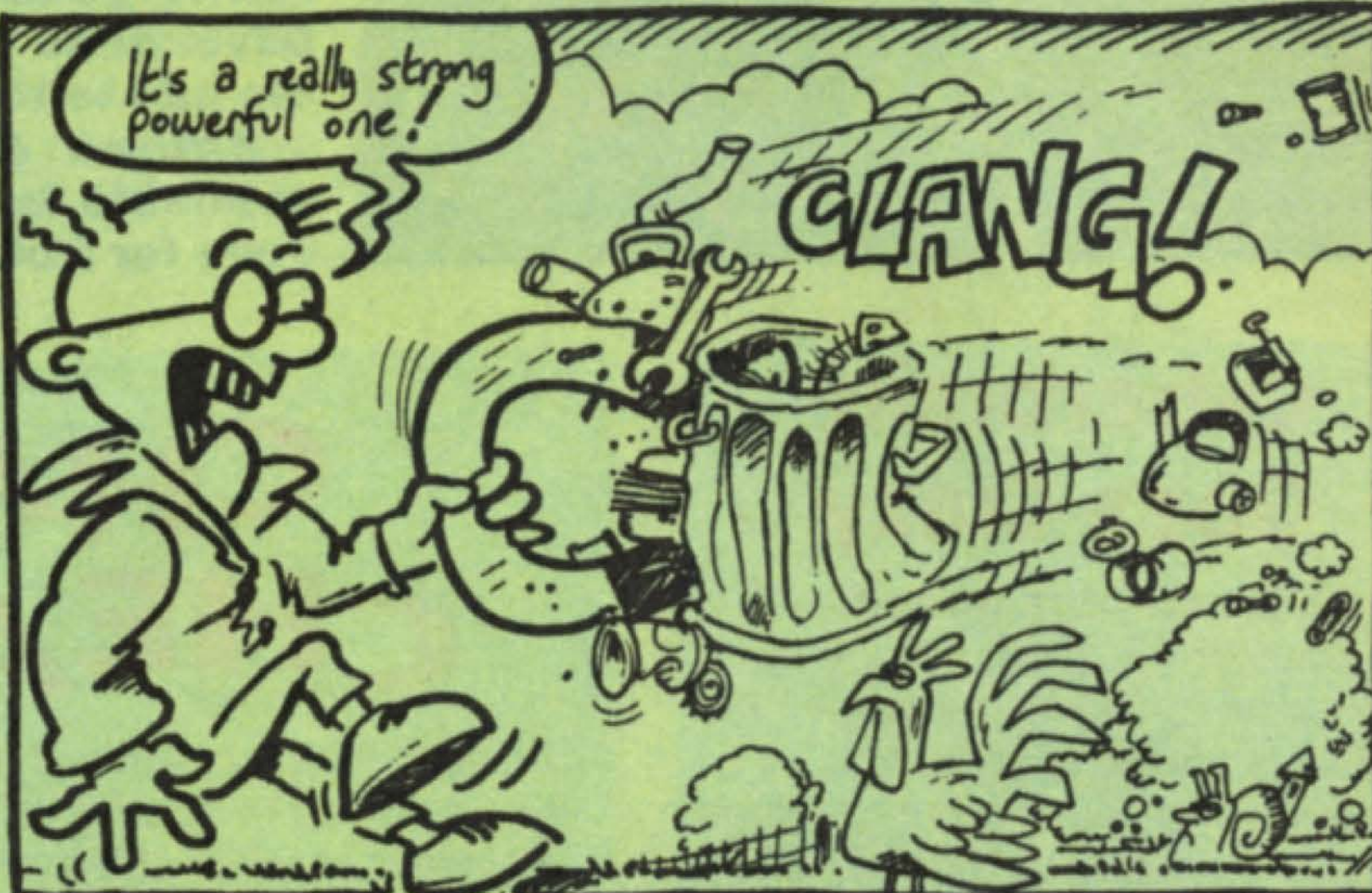
Commodore 64/128

Change or add these lines:

```
6030 PRINT CHR$(147);
6130 T$="":INPUT T$
6135 IF T$="" THEN T$="*"
```




The Safe



Scientist

ATTRACT OR REPEL

The Safe Scientist
plays with a magnetic model

Magnets attract some metal objects. But what happens if two magnets are put near each other? The program here is a simple investigation. It lets you find out just what DOES happen. Magnets have two ends. One is called the north pole and the other is the south pole. We need to find out what happens when poles are put near each other.

This computer simulation lets you do it without having to mess about with real magnets.

When you RUN the program two magnets are shown. They have red and blue ends - just like real ones.

Press the spacebar and one magnet will move. It will stop when it is in line with the other. That will move according to the laws of magnetism.

You can do the experiment as many times as you like by just pressing the spacebar. The poles of the magnets are arranged in a random way each time.

Try to come up with a rule. This means that you will be able to say what will happen BEFORE you run the program.

```

10 REM MAGNETS
20 REM THE SAFE SCIENTIST
30 DIM T(3):FOR I=1 TO 3:READ T(I):NE
XT I:LET SS="" :REM 10 SPACES
40 MODE1:VDU23;8202;0;0;0;0;VDU19,2,
4;0;
50 LET MAG1=RND(2):LET MAG2=RND(2)
60 LET Y=10
70 IF MAG1=1 THEN LET X=7:GOSUB 440:L
ET X=12:GOSUB 450
80 IF MAG1=2 THEN LET X=7:GOSUB 460:L
ET X=12:GOSUB 470
90 LET Y=3
100 IF MAG2=2 THEN LET X=22:GOSUB 440:
LET X=27:GOSUB 450
110 IF MAG2=1 THEN LET X=22:GOSUB 460:
LET X=27:GOSUB 470
120 LET X=14:LET Y=18:LET AS="PRESS SP
ACE":GOSUB 480
130 LET X=9:LET Y=20:LET AS="TO MOVE R
IGHT MAGNET":GOSUB 480
140 GOSUB 490
150 LET C=1:GOSUB 1000:LET X=14:LET Y=
18:LET AS=SS+" ":GOSUB 480
160 LET X=9:LET Y=20:LET AS=SS+SS:GOSUB
480
170 FOR N=3 TO 10:LET Y=N
180 IF MAG2=2 THEN LET X=22:GOSUB 440:
LET X=27:GOSUB 450
190 IF MAG2=1 THEN LET X=22:GOSUB 460:
LET X=27:GOSUB 470
200 LET C=1:GOSUB 1000:LET X=22:LET Y=
N-1:LET AS=SS:GOSUB 480
210 GOSUB 500
220 NEXT N
230 IF MAG1=MAG2 THEN GOSUB 280
240 IF MAG1<>MAG2 THEN GOSUB 360
250 LET X=14:LET Y=20:LET AS="PRESS SP
ACE":GOSUB 480
260 GOSUB 490
270 GOTO 40
280 FOR N = 7 TO 0 STEP -1:LET Y=10
290 IF MAG1=1 THEN LET X=N:GOSUB 440:L
ET X=N+5:GOSUB 450
300 IF MAG1=2 THEN LET X=N:GOSUB 460:L
ET X=N+5:GOSUB 470
310 LET C=1:GOSUB 1000:LET X=N+10:LET
Y=10:LET AS="" :GOSUB 480
320 GOSUB 500
330 NEXT N
340 LET X=15:LET Y=18:LET AS="REPULSIO
N":GOSUB 480
350 RETURN
360 FOR N = 7 TO 12:LET Y=10
370 IF MAG1=1 THEN LET X=N:GOSUB 440:L
ET X=N+5:GOSUB 450
380 IF MAG1=2 THEN LET X=N:GOSUB 460:L
ET X=N+5:GOSUB 470
390 LET C=1:GOSUB 1000:LET X=N-1:LET Y
=10:LET AS="" :GOSUB 480
400 GOSUB 500
410 NEXT N
420 LET X=15:LET Y=18:LET AS="ATTRACTI
ON":GOSUB 480
430 RETURN
440 LET C=2:GOSUB 1000:LET AS="N ":
GOSUB 480:RETURN
450 LET C=3:GOSUB 1000:LET AS=" S":
GOSUB 480:RETURN
460 LET C=3:GOSUB 1000:LET AS="S ":
GOSUB 480:RETURN
470 LET C=2:GOSUB 1000:LET AS=" N":
GOSUB 480:RETURN
480 PRINT TAB(X,Y);AS:RETURN
490 REPEAT UNTIL INKEY -99:RETURN
500 FOR D = 1 TO 200:NEXT D:RETURN
1000 COLOUR T(C):RETURN
1010 DATA 128,129,130

```

IS YOUR COMPUTER HERE?

BBC, Electron, Archimedes
The program works as shown:

Amiga

Change these lines:

```

40 PALETTE 3,.93,.2,0:COLOR 1,2:CLS
50 LET MAG1=INT(RND*2)+1:LET MAG2=INT
(RND*2)+1
480 LOCATE Y+1,X+1:PRINT AS:RETURN
490 WHILE INKEYS<>" ":WEND:RETURN
1000 COLOR,T(C):RETURN
1010 DATA 2,3,0

```

PC (GW-BASIC)

Change these lines:

```

40 CLS
50 LET MAG1=INT(RND*2)+1:LET MAG2=INT
(RND*2)+1
480 LOCATE Y+1,X+1:PRINT AS:RETURN
490 WHILE INKEYS<>" ":WEND:RETURN
1000 COLOR,T(C):RETURN
1010 DATA 0,4,1

```

ST(STOS)

Change or add these lines:

```

15 MODE 1:KEY OFF:CURS OFF:HIDE
40 PALETTE $0,$700,$007,$777:CLS
50 LET MAG1=INT(RND*2)+1:LET MAG2=INT
(RND*2)+1
480 LOCATE X,Y:PRINT AS:RETURN
490 WHILE INKEYS<>" ":WEND:RETURN
1000 PAPER T(C):RETURN
1010 DATA 0,1,2

```

Spectrum

Change or add these lines:

```

40 PAPER 7:CLS
50 LET MAG1=INT(RND*2)+1:LET MAG2=INT
(RND*2)+1
480 IF X>2 THEN PRINT AT Y,X-3;AS
485 RETURN
490 LET KS=INKEYS:IF KS<>" " THEN GOTO
490
495 RETURN
1000 PAPER T(C):RETURN
1010 DATA 7,2,1

```

Amstrad CPC

Change these lines:

```

40 MODE 1:INK 0,0:INK 2,6:INK 3,2:INK
1,26
50 LET MAG1=INT(RND*2)+1:LET MAG2=INT
(RND*2)+1
480 LOCATE X,Y:PRINT AS:RETURN
490 WHILE INKEYS<>" ":WEND:RETURN
1000 PAPER T(C):RETURN
1010 DATA 0,2,3

```

Commodore 64/128

Change these lines:

```

40 PRINT CHR$(147);
50 LET MAG1=INT(RND(0)*2)+1:LET MAG2=
INT(RND(0)*2)+1
480 POKE 211,X:POKE 214,Y:SYS 58732:PRI
NT AS:RETURN
490 GET KS:IF KS<>" " THEN GOTO 490
495 RETURN
1000 PRINT CHR$(18);CHR$(T(C));:RETURN
1010 DATA 144,28,31

```

NEXT MONTH we'll look at another computer model. We'll study the growth of a colony of bacteria. Ugh!

LET'S

99p

COMPUTE!

WIN!
more super prizes

PLAY!
our type-in game

SOLVE!
the nuclear mystery

LEARN!
Logo with the turtles

DETECT!
program errors

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

EUROPRESS
PUBLICATIONS



**BADGE PASSING
MADE EASY!**

PARENTS & TEACHERS!
Turn to Page 26

So far we've invented an adventure full of things waiting to be picked up. But what use are they if you can't take and carry them?

The first thing we need to do is get the computer to understand the names of the objects. You do this by adding them to the word list.

Add Lines 5060 to 5490 to your program. Words 11 and 12 give some words which mean TAKE and DROP. You could think of some more to add yourself. But there's a small snag!

You can see that some objects are easy to deal with. The laser rifle becomes word 50 - LASE and RIFE. Remember we are only using the first four letters of each word.

But the gas bomb and gas mask are a problem. If the player types TAKE GAS MASK or TAKE GAS BOMB the word numbers would be the same. The program would not know which object to deal with.

One way round this that we'll use for now is to leave out the word GAS. Now TAKE MASK or TAKE BOMB give different word numbers.

This shows that you have to think very carefully before you start. When you design your adventures look at just what players will try to type.

There will be the same problems with the buttons and lights. But as we can't carry these they will never be in the same room at once so we don't have to worry so much. We'll look at a better way of getting round the problem in a future issue of *Let's Compute!*

Now we need to give the program instructions on how to deal with the word combinations that might be typed. We call these conditions and actions.

If a condition is true then the program will carry out the listed actions. Type in Lines 6399 to 7000. This is the DATA.

Now type in Lines 1150 to 1220. These read the actions and conditions into their own arrays.

They come in pairs. Look at Line 6400. If words 12 and 50 are typed in the program will do action A, try to take object 01 - the laser rifle.

Like the room connection codes we must make sure that all our numbers have two digits. Remember, that's so we can scan through the list easily.

At the moment you can take one of three actions - A, B or C. As you can see in Lines 6400 to 6490, each is followed by a number.

The actions that are there so far are:

- Ann - Try to take object nn
- Bnn - Try to drop object nn
- C - List all the objects carried.

Each list of conditions and actions ends with a #.

Look through the table of conditions and actions and work out which words do what. You can see that the player will not be able to take the alien, the cable, the buttons or the lights.

Now we have to check to see if any of the words match the combinations in the condition table. Type in Lines 2999 to 3100.

They step through the condition table to see if the first word matches. If it doesn't the program goes to the next set of conditions. If the first word matches, it checks the second.

Look at Line 6460. There you'll find is the

Carry things Adventure

condition/action code for INVENTORY. That's the word which means *What am I carrying?* This is word 99. But we don't need another word so we follow it with 00. The program takes this to mean any word. You will find the inventory command very useful later on.

In future issues of *Let's Compute!* we'll be checking other things. For instance we'll look at what room we're in and whether we have a certain object.

At the moment we are only interested in matching the words. Now the program has both words matched it can go to the actions routine.

Type in Lines 3499 to 3590. These carry out the actions. They look at the first letter and use that at Line 3520 to GOSUB to that letter's routine.

So letter A causes a GOSUB to 3610, B to 3630 and C to 3650. You can see that it's very easy to add more letters with different actions to this system.

If you try to take an object the program checks two things. It makes sure it is there and that you aren't already carrying it. Then it lets you take it.

DROPPing first checks to see that you have the object. Room -1 is used for the inventory so carried objects have OBJROOM set to -1.

The program keeps track of where it is in the action with the variable P. At the moment this is not important because there is only one action in each set of codes.

Soon we will have several actions. The taking and dropping routines make sure they move this pointer along past the object numbers ready for the next action. Action C doesn't move the pointer because it has no numbers after it.

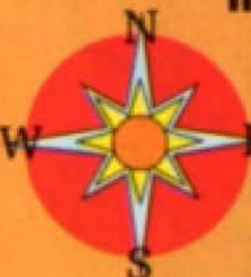
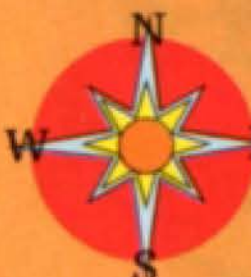
You will see the variable DIS in these routines. This tells the display routine how much to show. For example, the inventory list can be quite long so DIS is set to 2.

This means the list will be displayed but not the room description. Type in Line 1065 to initialise DIS and Lines 2015 and 2196 to put it into the display routine.

Finally, add Line 40. This puts the conditions/actions into the main program. Now you can move around the spaceship taking and dropping the objects we have listed in the condition/action table.

Alter your own adventure so you, too, can take and pick up objects. Your adventure is now really taking shape.

Next time we'll add more complicated conditions and some more actions. Then we will be able to push buttons and fix the cable.



s round your PART 4 nture!

*Here's how to pick
things up as you move
round the maze*

Answer to last month's challenge

Did you stop your computer printing *You can see* when the room is empty? There are lots of ways it can be done.

Here are two. Use one of the sets of line changes below to cure the problem.

The first method stops your computer printing anything at all if nothing is in the room. The second still prints *You can see* but follows it with the words *nothing useful*.

Both ways use what programmers call a flag. We've called ours ZZ. That way it's not likely to have the same name as anything else in the program.

It's set to zero first then changed to one if an object is found in the room. Then the program can print what's needed depending on whether ZZ is zero or 1.

Method 1: Doesn't print anything

```
2050 LET ZZ=0
2070 IF ZZ=1 AND OBJROOM(X)=ROOM THEN P
RINTTAB(14);OBJECT$(X)
2075 IF ZZ=0 AND OBJROOM(X)=ROOM THEN P
RINT "You can see: ";OBJECT$(X):LET ZZ=1
```

Method 2: Prints "You can see : nothing useful"

```
2045 LET ZZ=0
2070 IF OBJROOM(X)=ROOM THEN PRINTTAB(1
4);OBJECT$(X):LET ZZ=1
2080 NEXT X:IF ZZ=0 THEN PRINT "nothin
g useful"
2085 PRINT
```

IS YOUR COMPUTER HERE?

This program works on a BBC, Archimedes, Electron, CPC, Atari ST(Stos), Amiga and PC(GW-Basic). It will not work on a C64/128 or Spectrum.

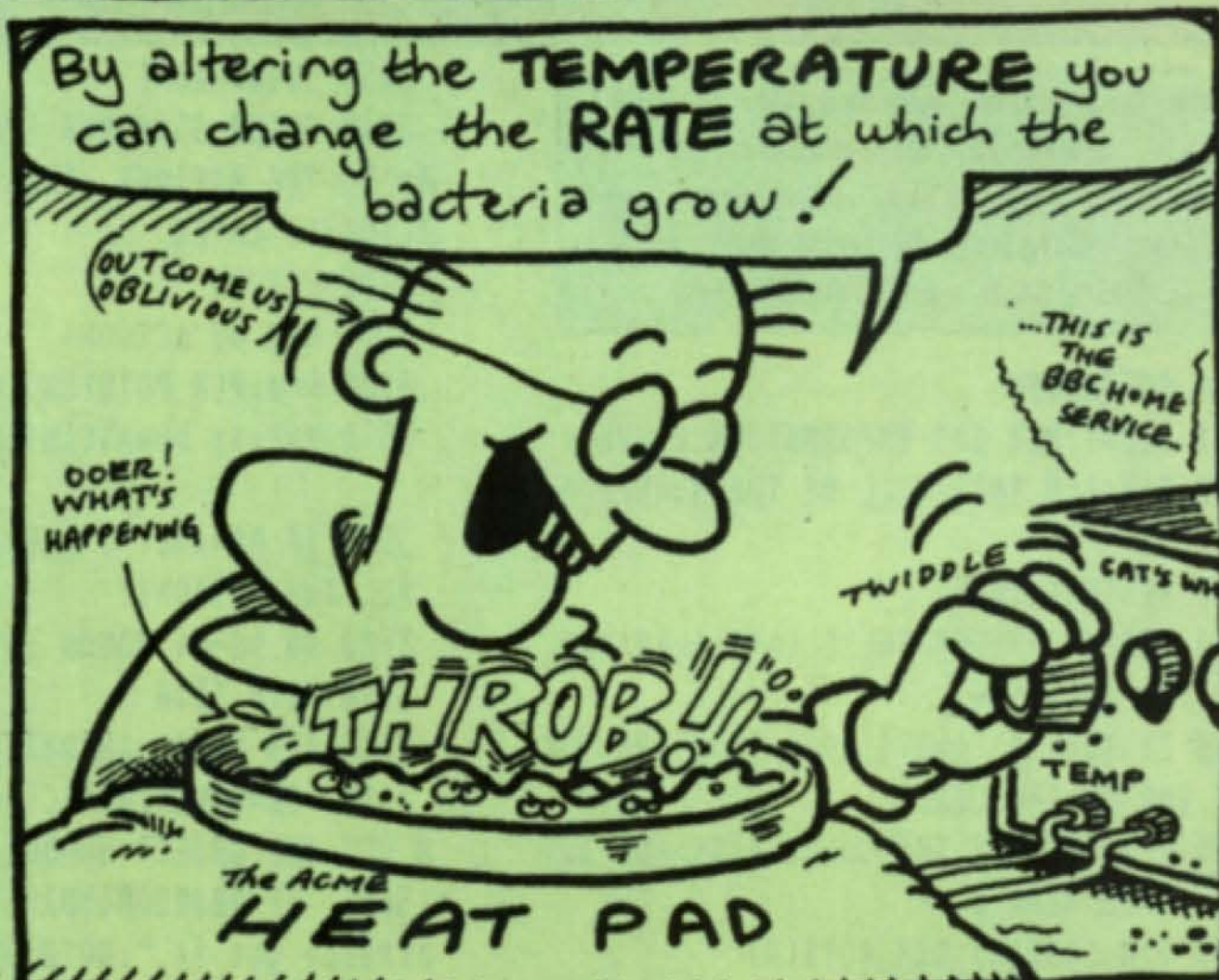
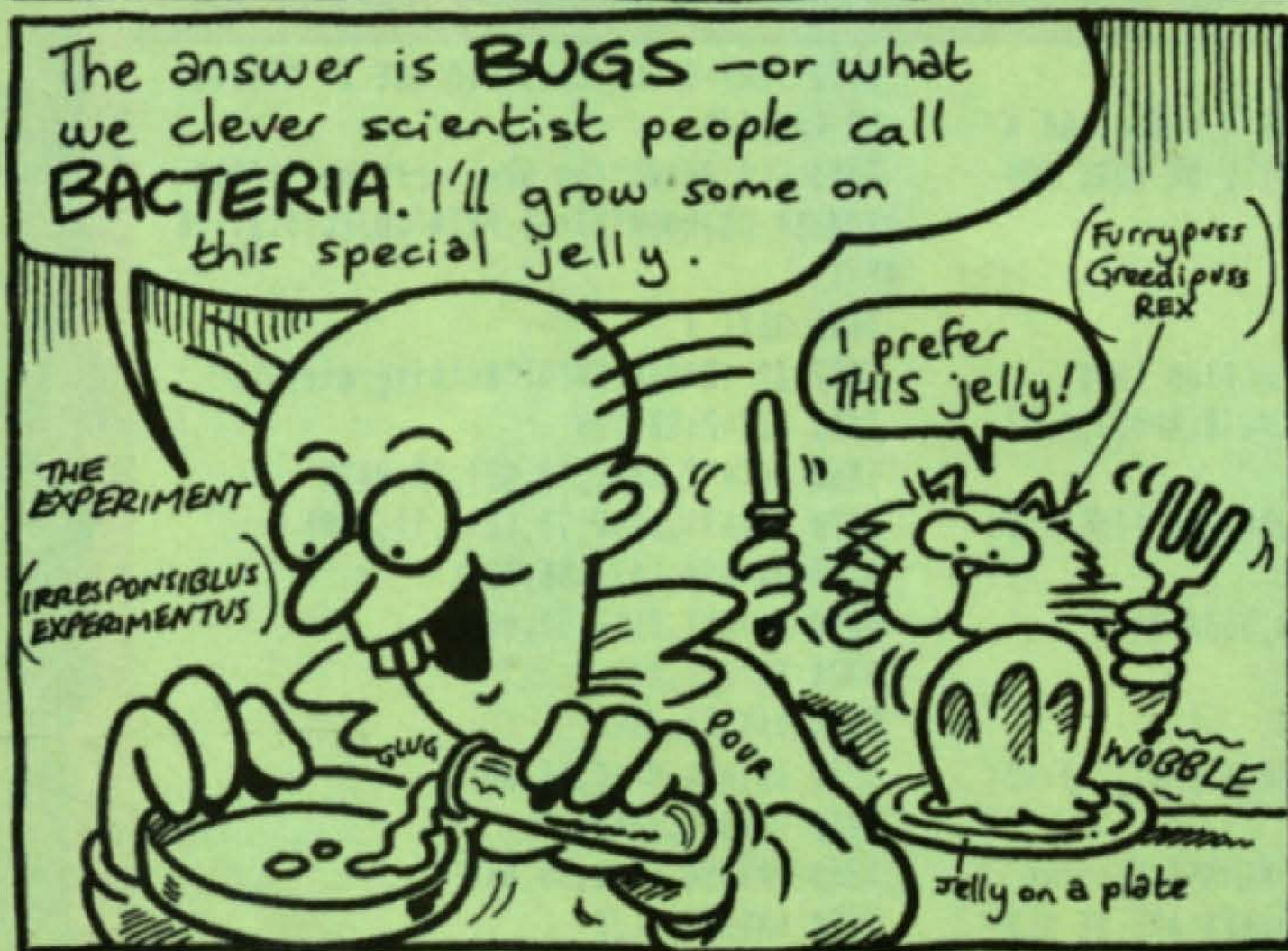
```
40 GOSUB3000
1065 DIS=1:REM SET INFORMATION DISPLAY
1150 A=0:REM THIS WILL BE THE NUMBER OF
ACTIONS
1160 RESTORE6400
1170 READX$,X$:REM THESE VARIABLES ARE
JUST FOR COUNTING
1180 IF X$<>"X" A=A+1:GOTO 1170:REM NOT
THE END OF THE LIST?
1190 REM NOW PUT THE CONDITIONS AND ACT
IONS INTO ARRAYS
1200 DIM CONS(A):DIM ACT$(A)
1210 RESTORE 6400:REM BACK TO THE START
OF THE LIST
1220 FOR X=1 TO A:READ CONS(X),ACT$(X):
NEXT
2015 ON DIS GOTO 2020,2191
2196 DIS=1:REM RESET DISPLAY TO NORMAL
2999 REM CONDITIONS/ACTIONS
3000 APOS=0:REM WHERE WE ARE IN THE CO
NDITION/ACTION TABLE
3015 WRDFLAG=0:REMTTHIS WILL TELL US T
HAT WORDS MATCHED
3020 APOS=APOS+1:IF APOS=A+1 RETURN:REM
HAVE WE BEEN THROUGH THE LIST?
3030 IF VAL (LEFT$(CONS(APOS),2))<>W(1
) GOTO 3020:REM DOES THE FIRST WORD MATC
H
3040 IF VAL (MID$(CONS(APOS),3,2))<>W(
2) AND VAL (MID$(CONS(APOS),3,2))>0 GOTO
3020:REM DOES THE SECOND WORD MATCH OR
IS THE CONDITION 0
```

```
3050 WRDFLAG=1
3060 GOSUB 3500:REM FOUND A MATCH, GO A
ND DO THE ACTIONS. THIS WILL BE MORE COM
PLICATED LATER.
3100 RETURN
3499 REM DO ACTIONS
3500 P=0:REM POINTER IN ACTION CODE
3510 P=P+1: DO=ASC(MID$(ACT$(APOS),P,P)
)
3515 IF DO=ASC("#") RETURN:REM FINISHED
ALL THE ACTIONS?
3520 ON DO-64 GOSUB 3610,3630,3650
3590 GOTO 3510
3609 REM CARRY AN OBJECT
3610 OB=VAL(MID$(ACT$(APOS),P+1,2)):RE
M GET THE OBJECT NUMBER
3613 IF OBJROOM(OB)=-1 MESS1$="You've
already got it.":GOTO3625:REM ARE WE ALR
EADY CARRYING IT
3615 IF OBJROOM(OB)<>ROOM MESS1$="It i
sn't here.":GOTO3625:REM NOT IN THIS ROO
M
3620 OBJROOM(OB)=-1:MESS1$="You take i
t.":REM MOVE THE OBJECT TO THE INVENTORY
3625 P=P+2:DIS=1:RETURN:REM MOVE THE PO
INTER TO THE NEXT ACTION
3629 REM DROP AN OBJECT
3630 OB=VAL(MID$(ACT$(APOS),P+1,2)):RE
M GET THE OBJECT NUMBER
3635 IF OBJROOM(OB)>-1 MESS1$="You hav
en't got it.":GOTO3645:REM ARE WE CARRYI
NG IT
3640 OBJROOM(OB)=ROOM:MESS1$="You drop
it.":P=P+2:RETURN:REM MOVE THE OBJECT T
O THE CURRENT ROOM
3645 P=P+2:DIS=1:RETURN:REM MOVE THE PO
INTER TO THE NEXT ACTION
3649 REM INVENTORY
```

```
3650 TEMP=0:REM CHECK TO SEE IF NOTHING
IS CARRIED
3653 CLS:PRINT"You are carrying:":FORX=
1TO0:IF OBJROOM(X)=-1 PRINTOBJECT$(X):TE
MP=1
3655 NEXT X
3657 IF TEMP=0 PRINT"Nothing useful."
3660 DIS=2:RETURN
5060 DATA12,TAKE,12,GET,12,BRIN
5070 DATA11,DROP,11,LEAV,11,DUMP
5200 DATA50,LAZE,50,RIFL
5210 DATA51,SLEE,51,BOMB
5220 DATA52,RED
5230 DATA53,GREE
5240 DATA54,SPAC,54,SUIT
5250 DATA55,SPAN
5260 DATA56,CROW,56,BAR
5270 DATA57,ALIE
5280 DATA58,MASK
5290 DATA59,CABL
5310 DATA61,LIGH
5320 DATA62,BUTT
5490 DATA99,INVE
6399 REM CONDITIONS AND ACTIONS IN HERE
6400 DATA1250#,A01#
6410 DATA1251#,A02#
6420 DATA1254#,A05#
6430 DATA1255#,A06#
6440 DATA1256#,A07#
6450 DATA1258#,A12#
6460 DATA9900#,C#
6465 DATA1150#,B01#
6470 DATA1151#,B02#
6475 DATA1154#,B05#
6480 DATA1155#,B06#
6485 DATA1156#,B07#
6490 DATA1158#,B12#
7000 DATA,X
```




The Safe



Scientist

BUGS

Watch the beasts multiply with the Safe Scientist's program!

Have you ever wondered why food goes rotten? And why it seems to go bad very quickly once it's started?

The answer is bugs or – to give them a more scientific name – bacteria. This month we're looking at how they grow.

Scientists watch colonies of bacteria growing on a specially prepared jelly. You may have tried that at school.

It is difficult though! You never know what extra nasty little bug has got onto your jelly.

There are health risks too! You have to take care that the bugs don't get onto you as well as the jelly. And it is a very slow, tricky process. You need lots of trays of jelly kept at different temperatures.

Your computer is an ideal tool to do these experiments for you. It takes away the risk and speeds things up.

Type the program in, SAVE it and RUN it. Your investigations can now begin!

You will first be asked to enter a temperature. Type one in and watch the colony grow from just a single bug. Soon 100 little beasts will be shown on the screen. The action will then freeze. The time it has taken will be shown at the top of the display.

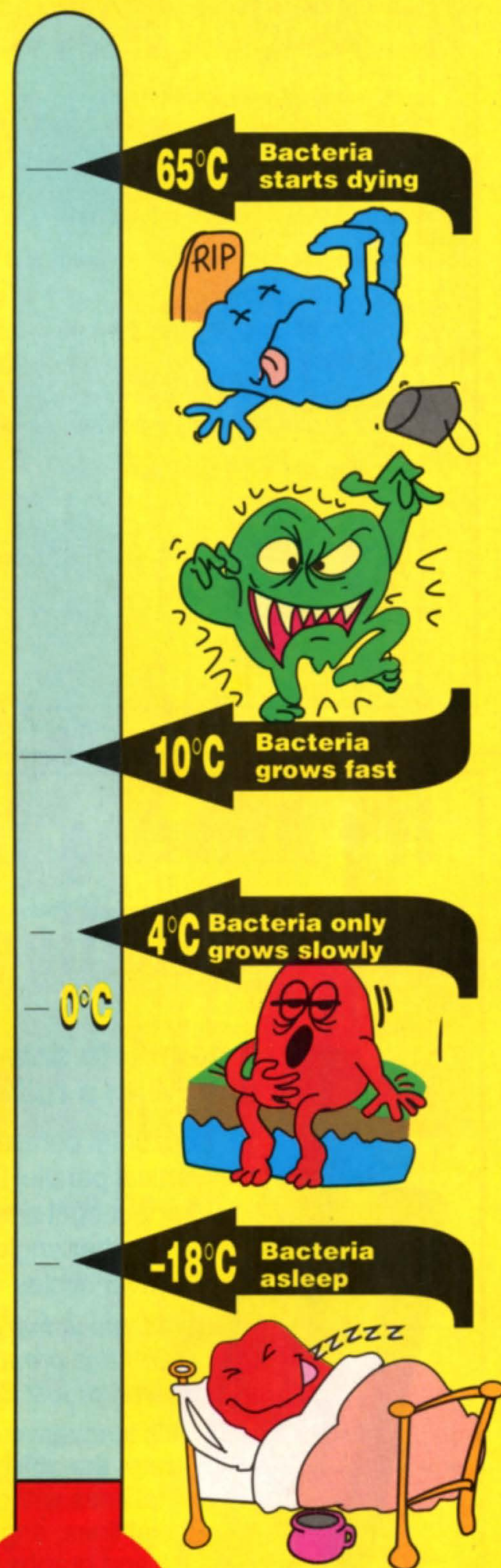
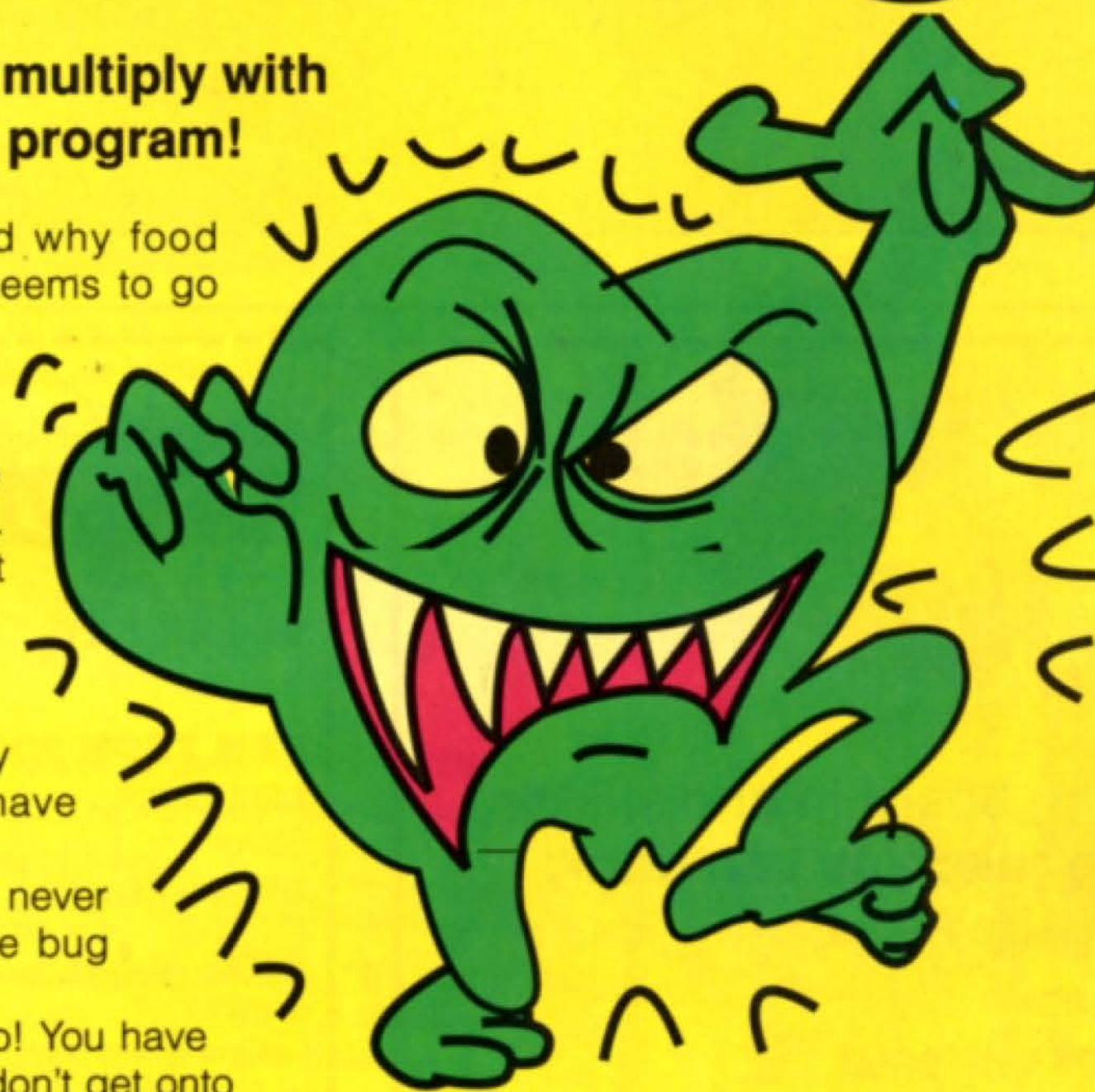
You'll notice that the colony grew slowly at first. Then you'll have seen an explosion of growth. I'll leave you to work out why that should be.

If the bacteria grew fast, think about a couple of ways we use to slow it down on food. For example, food keeps fresh longer when it's stored in a fridge.

In this program you only alter one factor – the temperature. Try to find out the temperature at which bacteria grow fastest.

This can be done by plain trial and error. But a good scientist will work in an ordered way to get to the answer.

The heart of the program is Line 170. This works out how long it should be before the next bacterium is created. Try altering it to get different effects.



Watch the bacteria grow with the listing on the next page

The Safe Scientist

```

10 REM BACTERIA GROWTH
20 REM THE SAFE SCIENTIST
30 REM LET'S COMPUTE!
40 DEF FNR(N)=RND(N)
50 MODE1:VDU23;8202;0;0;0;0;:LET W=39
:LET H=30
60 PRINT "What temperature (0 - 100)";
70 INPUT TP
80 IF TP>100 OR TP<0 THEN GOTO 60
90 CLS
100 PRINT "TEMP    TIME    BACTERIA"
110 GOSUB 270:LET BT=T
120 FOR N=1 TO 100
130 GOSUB 270:LET C=3:LET Y=1:LET X=1:
GOSUB 250:PRINT " ";TP
140 LET X=9:GOSUB 250:PRINT " ";(T-BT)/
100
150 LET X=21:GOSUB 250:PRINT " ";N
160 LET C=FNR(3):LET X=FNR(W):LET Y=FN
R(H-3)+3:GOSUB 250:PRINT "*"
170 LET D=(1.1^ABS(37-TP))/N*50
180 GOSUB 270:LET NT=T
190 GOSUB 270:IF T<NT+D THEN GOTO 190
200 NEXT N
210 LET X=12:LET Y=12:GOSUB 250:PRINT
"PRESS SPACE"
220 AS=GETS
230 IF AS<>" " THEN GOTO 220
240 RUN
250 COLOUR C:PRINT TAB(X,Y);
260 RETURN
270 LET T=TIME
280 RETURN
    
```

IS YOUR COMPUTER HERE?

BBC/Electron/Archimedes

The program works as shown.

Spectrum

Change these lines:

```

40 DEF FNR(N)=INT(RND*N)+1
50 CLS::LET W=31:LET H=20
220 LET AS=INKEYS
250 INK C:PRINT AT Y,X;
270 LET T=(256*PEEK 23673+PEEK 23672)*2
    
```

Amiga

Change these lines:

```

40 DEF FNR(N)=INT(RND*N)+1
50 CLS:LET W=39:LET H=20
220 LET AS=INKEYS
250 COLOR C:LOCATE Y+1,X+1
270 LET T=TIMER*100
    
```

Atari ST (Stos)

Change these lines:

```

40 DEF FNR(N)=INT(RND*N)+1
50 MODE1:KEY OFF:CURS OFF:HIDE:LET W=
39:LET H=30
220 LET AS=INKEYS
250 INK C:LOCATE X,Y
270 LET T=TIMER
    
```

PC (GW-Basic)

Change these lines:

```

40 DEF FNR(N)=INT(RND*N)+1
50 CLS:LET W=39:LET H=22
220 LET AS=INKEYS
250 COLOR C:LOCATE Y+1,X+1
270 LET T=TIMER*100
    
```

Commodore 64/128

Change these lines:

```

40 POKE 53280,0:POKE 53281,0:DEF FNR(
N)=INT(RND(0)*N)+1
50 PRINT CHR$(147);:LET W=39:LET H=20
90 PRINT CHR$(147);
220 GET AS
250 PRINT CHR$(C+152):POKE 211,X:POKE
214,Y:SYS 58732
270 LET T=TIMER
    
```

Amstrad CPC

Change these lines:

```

40 DEF FNR(N)=INT(RND*N)+1
50 CLS:LET W=39:LET H=30
220 LET AS=INKEYS
250 PEN C:LOCATE X+1,Y+1
270 LET T=TIME/3
    
```

It's Amazing It's Radical It's the Rolling Ruler



A complete drawing board in one simple unit. With a rolling ruler you can draw:

- ★ Horizontal parallel lines
- ★ Vertical parallel lines
- ★ Perfect right angles with ease
- ★ Any other angles from 1° to 359°
- ★ Clever circles

Available in two sizes 22cm (8in) and 30cm (12in) this great offer gets you **both** rolling rulers for just £5.95 (RRP £9.99)

It's innovative design gives the new rolling ruler the ability to draw vertical and horizontal lines without lifting it off the paper, parallel lines, angles, circles, musical staves and graphs with ease, making it ideal for the home, the office, schools, workshops and 3D design work.

Two rulers for just £5.95



READER OFFERS

RULER SET

£5.95 3605 ☐

PEN TECH 2000

Matt Black £14.95 3606 ☐
 Silver £14.95 3607 ☐
 Silver + Black £20.00 3608 ☐

Payment: Please indicate method (✓)

- ☐ Cheque/Eurocheque made payable to Database Direct
☐ Access/Mastercard/Eurocard/Barclaycard/Visa/Connect

No.

Expiry
Date

Signed

Name

Address

Post code

Daytime telephone number

Send to: Database Direct, FREEPOST,
 Ellesmere Port, South Wirral L65 3EB
 (No stamp needed if posted in UK)

Products are normally despatched within 48 hours of receipt
 but delivery of certain items could take up to 28 days

ORDERS AT ANY TIME OF THE DAY OR NIGHT

By phone: 051-357 1275 By Fax: 051-357 2813
 By MicroLink: MAG001 General Enquiries: 051-3572961
 Don't forget to give your name, address and credit card number



No.5

Safe

```

10 LET RT=0:LET CT=0:LET DT=0:LET LR
=0:LET RL=0
20 DIM U(3,3)
30 REM INPUT STRENGTHS
40 PRINT:PRINT"TYPE IN THE STRENGTHS
":PRINT
50 FOR R=1 TO 3
60 FOR C=1 TO 3
70 PRINT "ROW ";R;" COLUMN ";C
80 INPUT U(R,C):PRINT
90 NEXT C
100 NEXT R
110 REM CHECK ROWS
120 FOR R=1 TO 3
130 IF U(R,1)+U(R,2)+U(R,3)>30 THEN L
ET RT=RT+1
140 NEXT R
150 REM CHECK COLUMNS
160 FOR C=1 TO 3
170 IF U(1,C)+U(2,C)+U(3,C)>30 THEN L
ET CT=CT+1
180 NEXT C
190 REM CHECK DIAGONALS
200 FOR D=1 TO 3
210 LET LR=LR+U(D,D)
220 LET RL=RL+U(D,4-D)
230 NEXT D
240 IF LR>30 THEN LET DT=DT+1
250 IF RL>30 THEN LET DT=DT+1
260 REM PRINT RESULTS
270 IF RT=0 AND CT=0 AND DT=0 THEN GO
TO 330
280 PRINT "IT EXPLODED BECAUSE:"
290 IF RT>0 THEN PRINT RT;" OF THE RO
WS ARE TOO BIG"
300 IF CT>0 THEN PRINT CT;" OF THE CO
LUMNS ARE TOO BIG"
310 IF DT>0 THEN PRINT DT;" OF THE DI
AGONALS ARE TOO BIG"
320 PRINT:GOTO 340
330 PRINT"WELL DONE"
340 PRINT:PRINT "ANOTHER GO (Y/N)"
350 INPUT AS
360 IF AS="Y" THEN RUN
370 IF AS<>"N" THEN GOTO 340

```

**This program works on
all home computers**

Possible program projects

- At present it prints *1 ROWS ARE TOO BIG, 2 ROWS ARE TOO BIG* and so on. Stop it printing the S after ROW if there is only one.
- Make it tell you exactly which rows and columns don't add up to 30 or less.
- Put in some graphics. For example you could add an explosion routine in place of Line 280.

Here's a project that can be as easy or as hard as you want to make it. It's a model of a container to store uranium.

Radioactive materials like uranium are very dangerous. They have to be stored in lead containers.

Our picture shows one of these. It holds nine canisters of uranium.

The numbers on the canisters show the strength of the uranium inside. The whole lot is safe if all the rows, columns and diagonals add up to less than 30.

At the moment one column, two rows and one of the diagonals are likely to explode at any time. Can you rearrange them so they are safe?

Make a model so you can move the numbers around. It can be as simple as nine numbers on pieces of paper. Or you can use a cardboard box and tubes to make a more realistic model.

It's up to you how much effort and imagination you put into the project. All that's important is that you end up with the nine numbers shown below and they are able to be moved around.

The problem now is that each time you move a number you have to check that the new rows, columns and diagonals add up to less than 30. Don't worry. Pete has come up with an easy way.

The computer program on the left works it all out for you. The panel on the right explains how to use it.

See if you can arrange the canisters into a safe order. In case you have problems, the answer will be in next month's issue of *Let's Compute!*

8	10	14
9	6	7
11	12	13



win

Can you keep the uranium safe in Pete's nuclear puzzle?

numbers

USING YOUR COMPUTER

This program will tell you if your uranium pile is safe. Type it in, **SAVE** it and **RUN** it.

Then type in the numbers one at a time. Your computer will tell you which row and column the number will be put into.

If your canisters are safe a *Well done* message will be printed. But if any sets of numbers add up to more than 30 you'll be told how many are over the limit.

MAGIC MODIFICATIONS

You can use the program to check what maths experts call magic squares. One is shown below.

They're called magic because all the rows and columns add up to the same. In this case, 15. To use your program to check it, change

8	1	6
3	5	7
4	9	2

every >30 in the program to <>15. You'll find them in Lines 130, 170, 240 and 250.

You also need to change the message that's printed if it isn't right. Change the words **ARE TOO BIG** in Lines 290, 300 and 310 to **ARE WRONG**. This is because the program now checks for smaller as well as bigger.

SNAP!

Heetan Patel presents a super version of a classic game

Two words will randomly appear on the screen. If they are the same, one of the players has to quickly press his *fire* key.

The player who is the first to get five snaps wins. If you press fire and the words are not the same you lose a snap. There are six words that flash on the screen. You will see them in Lines 570 to 620. They appear in pairs and are randomly chosen. The words are the kind you often see in Let's Compute! If you want, you can change them into any other words you like.



This program and the description printed here are by Heetan Patel (12) from Coventry. It's a two player game. Both have to try to press a key first when a matching pair of words appears on the screen.

Here are some changes you can make

- Make snaps happen less often by changing the 5 in Line 170 to, say, 50.
- You can change the words (Lines 570 to 620). You can also add more words if you change E=6 in Line 90. Then put extra DATA lines starting at Line 630.
- You can make the game longer by making people have to get more snaps. This is in Lines 370 and 380. For example, to make the game last for eight snaps, change Line 370 to:

```
370 IF S1>7 THEN GOTO 400
```

- You can add colour and sound

The FIRE keys

Player 1 Player 2

Q P

You can only use them in Caps Lock

IS YOUR COMPUTER HERE?

BBC/Electron/Archimedes

The program works as shown.

Amstrad CPC/ST(STOS)

Change these lines:

```
130 DEF FNR(R)=INT(RND*R)+1
520 LOCATE X+1,Y+1
550 LET K$=INKEY$
```

Spectrum

Change these lines:

```
90 LET E=6:DIM A$(E,8):LET S1=0:LET S2=0
130 DEF FNR(R)=INT(RND*R)+1
500 IF H$="N" THEN STOP
520 PRINT AT Y,X;
550 LET K$=INKEY$
```

Amiga/PC(GW-Basic)

Change these lines:

```
130 DEF FNR(R)=INT(RND*R)+1
520 LOCATE Y+1,X+1
550 LET K$=INKEY$
```

Commodore 64/128

Replace CLS with PRINT CHR\$(147); on Lines 10, 80, 350, 410, 440, and 490.

Change these lines:

```
130 DEF FNR(R)=INT(RND(0)*R)+1
520 POKE 211,X:POKE 214,Y:SYS 58732
550 GET K$
```

HEETAN'S PROGRAM

```
10 CLS
20 LET Z=0
30 PRINT"HOW FAST DO YOU WANT THE GAM
E"
40 PRINT"(1 TO 50) 1 IS THE FASTEST"
50 INPUT M
60 IF M>50 OR M<1 THEN GOTO 50
70 LET G=100*M
80 CLS
90 LET E=6:DIM A$(E):LET S1=0:LET S2=
0
100 FOR I=1 TO E
110 READ A$(I)
120 NEXT I
130 DEF FNR(R)=RND(R)
140 PRINT"PLAYER1 :";S1;"      PLA
YER2 :";S2
150 LET A=FNR(25):LET B=FNR(20):LET C=
FNR(E):LET D=FNR(E)
160 LET L=B
170 IF Z>FNR(5) THEN LET Z=0:LET C=D
180 LET X=A:LET Y=B:GOSUB 520:PRINT A$
(C)
190 LET A=FNR(25):LET B=FNR(20)
200 IF L=B THEN GOTO 190
```

```
210 LET X=A:LET Y=B:GOSUB 520:PRINT A$
(D)
220 FOR I=1 TO G
230 GOSUB 550:LET B$=K$:IF B$="Q" OR B
$="P" THEN LET I=G
240 NEXT I
250 IF B$="Q" THEN GOSUB 280
260 IF B$="P" THEN GOSUB 310
270 GOSUB 550:GOTO 350
280 IF A$(C)=A$(D) THEN LET S1=S1+1
290 IF A$(C)<>A$(D) THEN LET S1=S1-1:I
F S1<0 THEN LET S1=0
300 RETURN
310 IF A$(C)=A$(D) THEN LET S2=S2+1
320 IF A$(C)<>A$(D) THEN LET S2=S2-1:I
F S2<0 THEN LET S2=0
330 RETURN
340 GOSUB 540
350 CLS
360 LET Z=Z+1:IF C=D THEN LET Z=0
370 IF S1>4 THEN GOTO 400
380 IF S2>4 THEN GOTO 440
390 GOTO 130
400 GOSUB 540
410 CLS
```

```
420 PRINT:PRINT"PLAYER 1 IS THE WINNER
"
430 GOTO 470
440 CLS
450 GOSUB 540
460 PRINT:PRINT"PLAYER 2 IS THE WINNER
"
470 PRINT:PRINT"ANOTHER GO Y OR N"
480 GOSUB 550:LET H$=K$
490 IF H$="Y" THEN CLS:LET S1=0:LET S2
=0:GOTO 140
500 IF H$="N" THEN END
510 GOTO 480
520 PRINT TAB(X,Y);
530 RETURN
540 FOR I=1 TO G:NEXT I
550 LET K$=INKEY$(0)
560 RETURN
570 DATA "CONTESTS"
580 DATA "LOGO"
590 DATA "GAMES"
600 DATA "PROJECTS"
610 DATA "CHEATS"
620 DATA "HELP"
```


LET'S COMPUTE!

99p

AARGH!

Make a monster

WIN!

*a trip to Alton Towers
– and much more*

PASS!

that computing badge

FLASH!

Spelling made simple

For **ALL** users of:
Electron, BBC, Amiga,
C64/128, Archimedes,
Spectrum, Atari ST,
PC, Amstrad CPC

EUROPRESS
PUBLICATIONS



**Conjure up
some fun!**

PARENTS & TEACHERS!
Turn to Page 26

Do you sometimes find it difficult to make decisions? Or just get bored thinking up answers? Then let your computer do the job for you.

Just put a list of possible answers to a question in the DATA lines at the end of this program. It will pick one at random when asked – and probably cause a lot of laughter in the process.

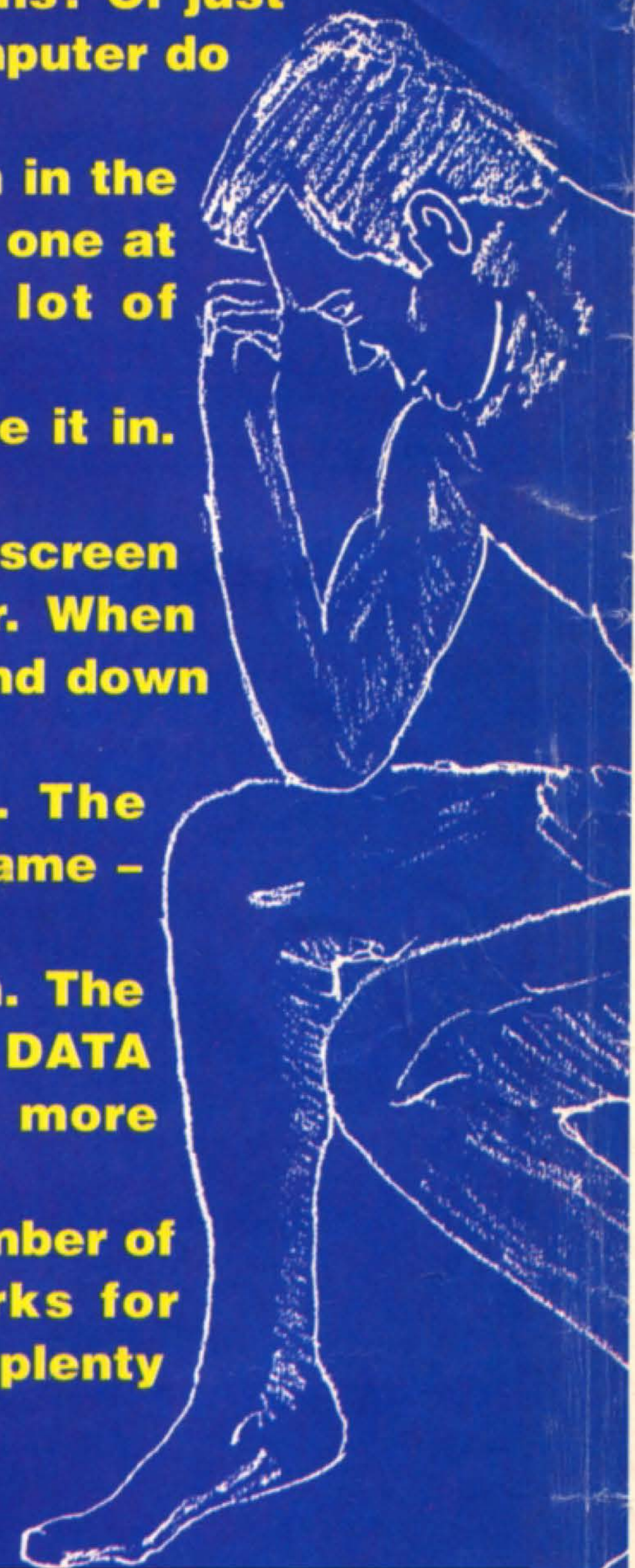
The program is really easy to use: Carefully type it in. Then SAVE it and RUN it.

All the possible responses will be printed on the screen and you will be asked to press Return or Enter. When you've done this a pointer will quickly move up and down to select an answer within a few seconds.

If you need more simply tap the key again. The pointer will move off to another – or maybe the same – answer.

You can alter the program to suit any situation. The answers you want to use should be placed in DATA statements starting at Line 380. You can have more than nine if you want.

Change the number 9 in Line 370 to the total number of answers in your DATA lines. The program works for between 2 and 18 responses – which should be plenty for all your needs.



This program, and advice on how to use it, are by Stephen Wade (16) from Bramhall.

It's a great way to get your computer to help you to reply to those tricky-to-answer questions.

```
10 REM Decision maker
20 REM by Stephen Wade
30 REM (c) Let's Compute!
40 CLS
50 PRINT "THE LET'S COMPUTE!"
60 PRINT "  DECISION MAKER"
70 PRINT "===== "
80 LET S=2:READ N
90 IF N>9 THEN LET S=1
100 IF N<2 OR N>18 THEN STOP
110 LET Z=(N*S)+1
120 FOR X=0 TO 24
130 FOR Y=3 TO Z STEP S
140 GOSUB 300:PRINT " ";
150 NEXT Y:NEXT X
160 FOR Y=3 TO Z STEP S:READ TS
170 LET X=0:GOSUB 300
180 PRINT TS;" ";:NEXT Y
190 LET X=0:LET Y=20:GOSUB 300
200 PRINT "PRESS RETURN OR ENTER"
210 GOSUB 360:GOSUB 300
220 I=1 TO 21:PRINT " ";:NEXT I
230 LET X=26:FOR Y=3 TO Z STEP S
240 GOSUB 300:PRINT "  "
250 NEXT Y:GOSUB 310
260 FOR A=1 TO Y+30
270 GOSUB 340:GOSUB 300:PRINT "  "
280 GOSUB 310:GOSUB 300:PRINT "<=="
290 NEXT A:GOTO 190
300 PRINT TAB(X,Y);:RETURN
310 LET Y=RND(N):LET Y=(Y*S)+1
320 IF Y<3 THEN LET Y=Y+S
330 RETURN
340 FOR I=1 TO 200:NEXT I
350 RETURN
360 INPUT AS:RETURN
370 DATA 9
380 DATA "Yes","Maybe","Ask me later"
390 DATA "I'll think about it"
400 DATA "If you're lucky"
410 DATA "Don't even ask","No"
420 DATA "Go away - I'm too busy"
430 DATA "Never!"
```

IS YOUR COMPUTER HERE?

BBC/Electron/Archimedes
The program works as shown.

Amstrad CPC/ST(STOS)
Change these lines:

```
300 LOCATE X,Y:RETURN
310 LET Y=INT(RND*N)+1:LET Y=(Y*S)+2
```

Spectrum

Change these lines:

```
300 PRINT AT Y,X;:RETURN
310 LET Y=INT(RND*N)+1:LET Y=(Y*S)+2
```

Amiga/PC(GW-Basic)

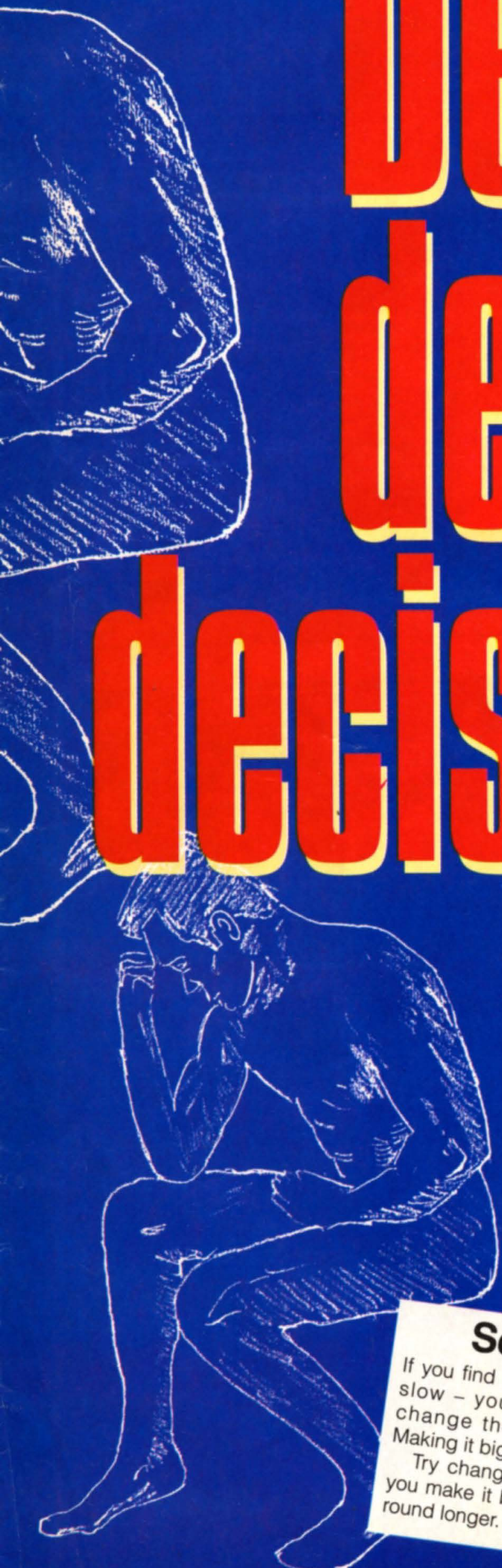
Change these lines:

```
300 LOCATE Y+1,X+1:RETURN
310 LET Y=INT(RND*N)+1:LET Y=(Y*S)+2
```

Commodore 64/128

Change these lines:

```
40 PRINT CHR$(147);
300 POKE 211,X:POKE 214,Y:SYS
58732:RETU
RN
310 LET Y=INT(RND(0)*N)+1:LET Y=(Y*S)+2
```

Decisions, decisions, decisions

***Make
your
mind up
with
Stephen
Wade's
unique
program***

Set the speed

If you find the pointer moves too fast – or slow – you can alter the speed. Just change the number 200 in Line 340. Making it bigger slows the pointer down. Try changing the 30 in Line 260 too. If you make it bigger the pointer will bounce round longer.

Line of fortune

At most fund-raising events you'll find a Wheel of Fortune. This program can be used as an electronic equivalent – the Line of Fortune.

Just change the DATA to pop groups, football teams or something else people will recognise. Then charge people to say which group or team will win.

Add action to your **PART 5** Adventure!

The excitement really starts as you USE the objects scattered round your maze

If you've been typing in the *Let's Compute!* adventure over the past few months you can already take and drop some of the objects. But the real thrill comes when you do something with them.

This means you need to add more actions and conditions to make the game more interesting.

Type in Lines 3060 to 3110. These let us have more complicated conditions than before.

You will see that the operation is similar to the way we tackled the actions. Each condition will have a letter and some numbers after it.



Again all the numbers must have two digits. And, just like the actions, it is really easy to add new conditions.

The **CFLAG** variable checks to see if the conditions are true. We only want to carry out an action if they are **ALL** true, so each will set **CFLAG** to one if it is false.

If the condition loop finds it has reached the # at Line 3075 all the conditions must be true. The program then goes on to perform the actions.

At the moment we'll just have one condition:

Condition Ann: Check if object *nn* is here or being carried.

Type in the new Line 3520 to add this. We're also going to add a new action:

Action Dnnmm: Move object *nn* to room *mm*.

Enter Lines 3669 to 3680 to put this in your program.

You will see that Line 3680 adds four to the action pointer - *P*. This is because there are two numbers instead of one.

This action does not affect the display so **DIS** is not set.

You want your player to be able to fix the loose cable and press the buttons in the control room. So you need to add Lines 5080 and 5085. These put the words you need into the program.

They include several ways of saying **FIX**: So **MEND** and **REPAIR** will also do the job. **PUSH** and **PRESS** also mean the same as each other.

You should always think of the different words players might try to use. Don't let them get bored hunting for the one and only right one!

Now we add the conditions and actions. Look at Line 6495. This first checks for words 20 and 59 - **FIX CABLE**.

If they're found, it uses condition **A** to check that the player is in the room with the spanner and the loose cable - Objects 06 and 13.

If that is true it then goes to action **D**. First it sets the loose cable to Room 0 - our store room. Then it sets the fixed cable - Object 14 - to Room 16.

So typing **FIX CABLE** will replace the loose cable with the fixed one. But your player must be in a room with the loose cable and the spanner - Room 16 - as the cable cannot be moved.



Type in Lines 6495 to 6505. The first of these sorts out what the program has to do when the player asks for the cable to be fixed.

The other two deal with pressing the buttons in the control room. Can you work out what effect they have? If you can't, just run the game, press the green button and go to the air lock.

If you use the conditions and actions you already have and add a word or two you should be able to put the alien to sleep. Have a go at that.

● Don't worry if you have problems: We'll reveal all next month!

IS YOUR COMPUTER HERE?

This program works on a BBC, Electron, CPC, Amiga and PC (GW-Basic)

It also works on an Atari ST using Stos. But if you're using Stos replace **CONS** with **CNS** throughout your program.

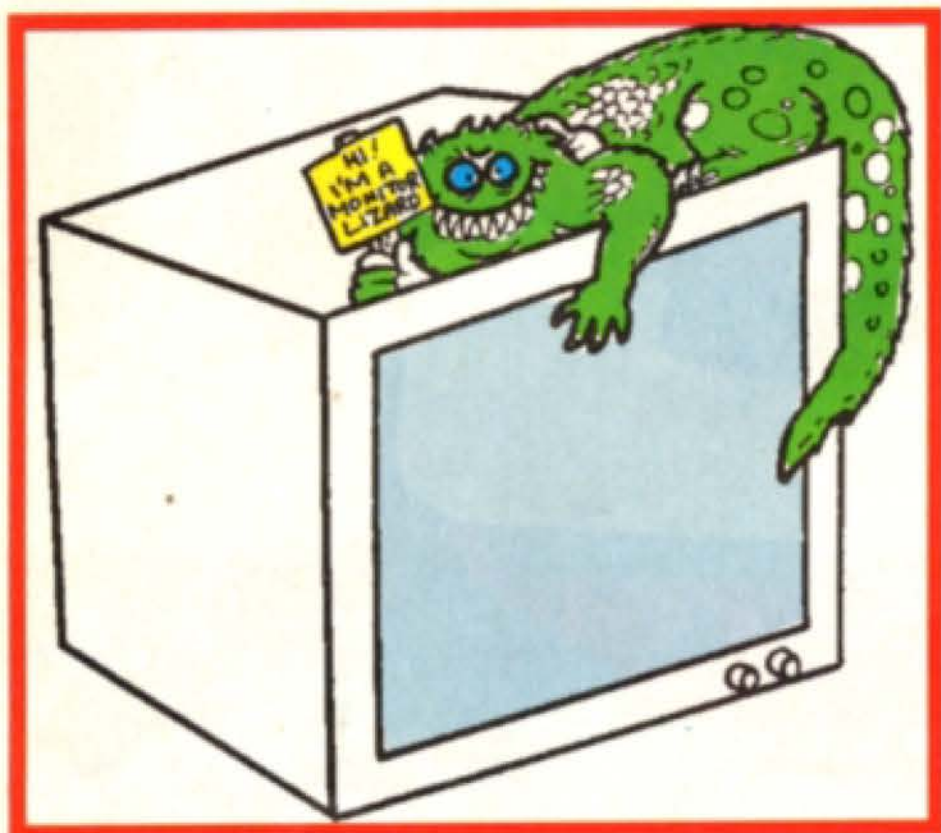
This adventure will not work on a C64/128 or Spectrum.

```
3060 P=4:REM POINTER IN CONDITION CODE
SET PAST WORD-NUMBERS
3065 CFLAG=0:REM CONDITION FLAG STAYS ZERO IF CONDITIONS ARE TRUE
```

```
3070 P=P+1: DO=ASC(MID$(CONS(APOS),P,P))
)
3075 IF DO=ASC("#") THEN GOSUB 3500:RETURN:REM IF ALL CONDITIONS TRUE, DO ACTION, GO BACK TO MAIN PROGRAM
3080 ON DO-64 GOSUB 3100
3085 IF CFLAG=1 THEN GOTO 3020:REM IF CONDITION NOT TRUE: GO TO NEXT SET OF CONDITIONS
3090 GOTO 3070:REM CONDITION TRUE, GOTO NEXT CONDITION
3099 REM CHECK IF OBJECT IS HERE OR CARRIED
3100 OB=VAL(MID$(CONS(APOS),P+1,2)):REM GET THE OBJECT NUMBER
3105 IF OBJROOM(OB)<>ROOM AND OBJROOM(OB)>-1 CFLAG=1:REM IF NOT HERE OR CARRIED
```

```
SET FLAG
3110 P=P+2:RETURN:REM MOVE POINTER TO NEXT CONDITION
3520 ON DO-64 GOSUB 3610,3630,3650,3670
3669 REM MOVE OBJECTnn TO ROOM nn
3670 OB=VAL(MID$(ACT$(APOS),P+1,2)):REM GET THE OBJECT NUMBER
3675 RO=VAL(MID$(ACT$(APOS),P+3,2)):REM GET THE ROOM NUMBER
3680 OBJROOM(OB)=RO:P=P+4:RETURN:REM MOVE THE OBJECT, RESET POINTER TO NEXT ACTION
5080 DATA20, FIX, 20, MEND, 20, REPA
5085 DATA21, PUSH, 21, PRES
6495 DATA2059A06A13#, D1300D1416#
6500 DATA2152A03#, D0900D0808#
6505 DATA2153A04#, D0800D0908#
```

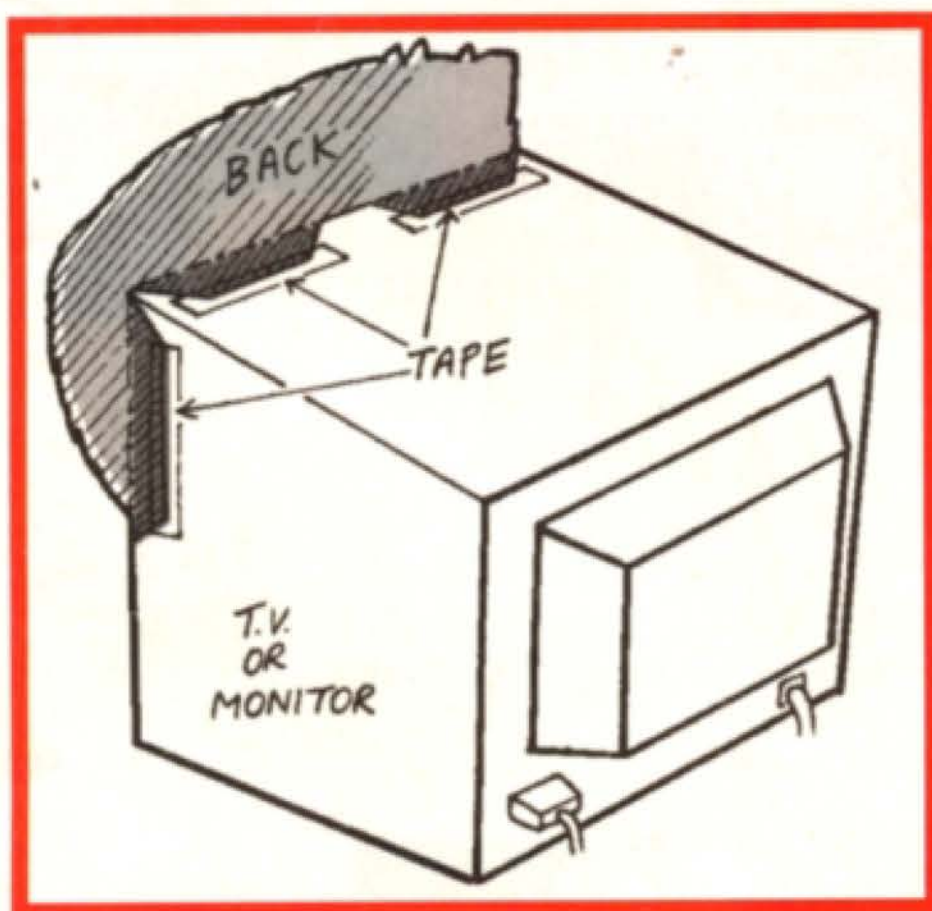

CUT IT OUT!



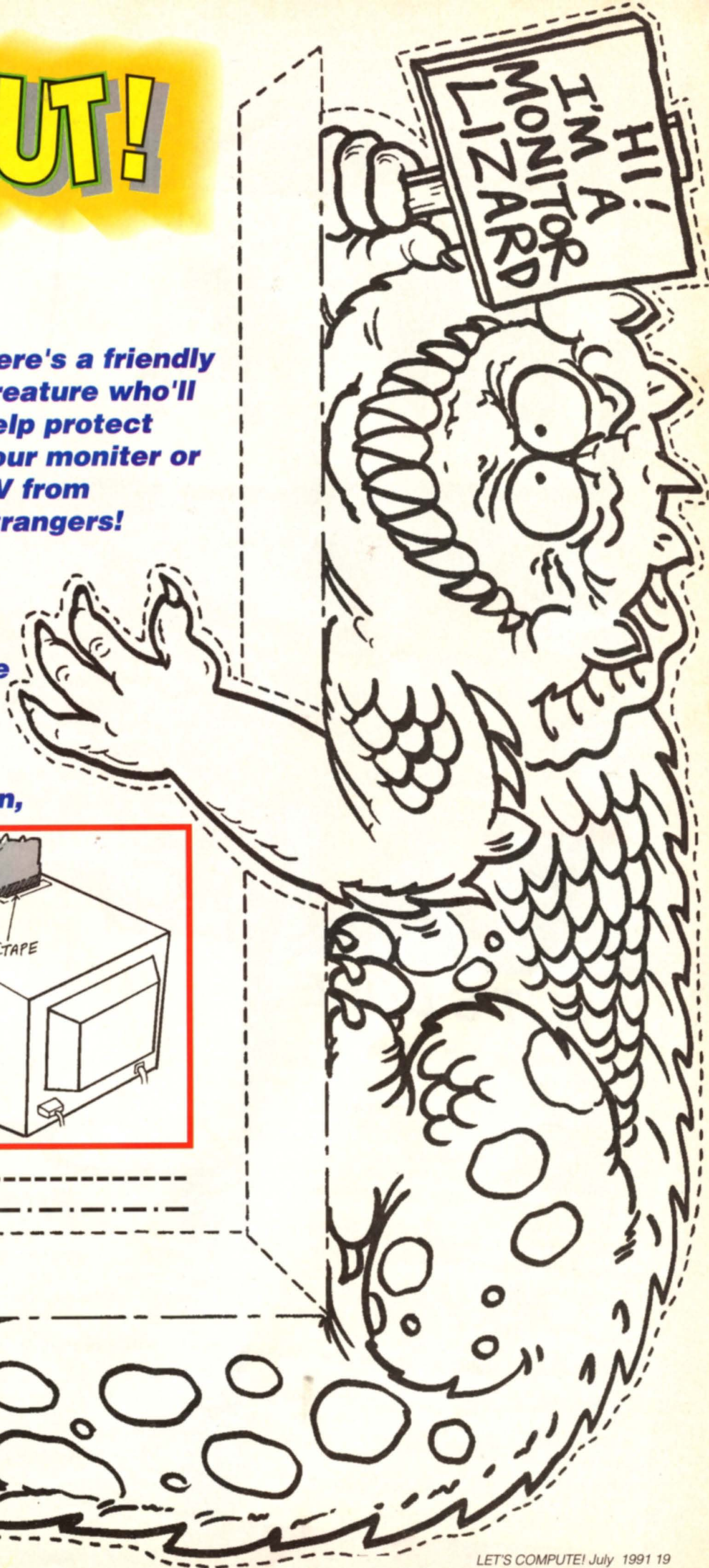
Here's a friendly creature who'll help protect your monitor or TV from strangers!

Stick this page on a piece of card or thick paper. Then cut out the lizard, carefully following the dotted line. Fold as shown, then stick it in place, using sticky tape or Blu-tack.

You can also colour it in using the most menacing colours you can find!



CUT
FOLD



Watch for another
CUT IT OUT!
monster next
month!

But don't make it look too simple. Look as if you're thinking hard. The showmanship is up to you!

HOW IT'S DONE

Right! You've made the cards, given them to a friend and got a few back. How do you know what number was written down?

No, you don't have to look at all the cards in your hand and try to spot the number that's on them all. It's much easier than that!

Just add up the first number on each card. You should find they add up to the number that your friend wrote down.

If they don't one of two things has happened:
Your friend still holds a card with the number on it
or you can't add up.

For a bit of practice you can make your computer ask the questions. Type in the extra lines given to convert the program into an electronic version. Make sure you add them to the original program, not the printing version.

Now, when you RUN the program, the cards will be shown one at a time. You think of a number and just answer each *Is the number on this card?* with a Y or N.

After all the cards have been displayed you'll be told the number you thought of.

[illegible]

Printing the cards

The bit of program you need to print the cards is different for the various types of computer. If you have a printer and want to print them alter the card making program, NOT the electronic version.
Add or change the lines given here for your own make of computer:

BBC/Electron/Archimedes

5 VDU 2
160 VDU 3

**Amiga (Amos)/ST (Stos)/
PC (GW Basic)/ Spectrum**

```

70 LPRINT SS;C;" ";
130 LPRINT:LPRINT

```

CPC

```
70 PRINT #8,$$;C;" ";
130 PRINT #8:PRINT #8
```

C64

```

5 OPEN 4,4
70 PRINT#4,SS;C;" ";
130 PRINT#4:PRINT #4
160 PRINT#4:CLOSE 4

```


MC

Making the cards

Type in the card making program. **SAVE** it and **RUN** it. Loads of numbers will scroll up the screen.

There are 192 of them altogether. And they need to be put on six cards - 32 numbers on each. As the numbers appear you'll see gaps separating them into the six groups.

So how do you make the cards?

If you've got a printer, make the changes shown in the *Printing the cards* panel. Then when you run the program they'll automatically be printed out.

Cut them out and either use them as they are or paste them on card.

If you haven't got a printer, write the groups of numbers on pieces of card.

To stop the numbers scrolling too fast on your screen you can insert a new Line 135:

```
135 INPUT ZZ$
```

The numbers will now stop after each group appears. Type any letter and press **Return** or **Enter** to get the next group.

The number of cards is set in Line 10. Change the $M=6$ to, say, $M=8$ if you want. The more cards you have, the more numbers will be on each. You can have any number of cards from 2 to 9.

If you try more you'll find the program still works. But the numbers go out of line as they go above 1,000.



The card making program

```
10 LET F=1:LET M=6:REM M IS THE NUMBER
OF CARDS (2 TO 9)
20 LET G=F:LET S=F:LET P=0
30 LET E=S+G-1
40 FOR C=S TO E:LET SS=""
50 IF C<100 THEN LET SS=""
60 IF C<10 THEN LET SS=""
70 PRINT SS;C;" ";
80 LET P=P+1
90 IF ABS(P/4-INT(P/4))<.01 THEN
PRINT
100 NEXT C
110 LET S=E+G+1
120 IF S<2^M THEN GOTO 30
130 PRINT:PRINT
140 LET F=F*2
150 IF F<2^M-1 THEN GOTO 20
```

This program will work on all home computers.

Note that on some keyboards there is no ^ as used in Lines 120 and 150. If it isn't on your computer use ↑.

On the ST (using Stos) replace P with P#.

Extra lines to make an electronic version

Add these lines to the card making program and your computer will ask the questions.

```
1 PRINT:PRINT"THINK OF A NUMBER":PRI
NT:PRINT
2 LET N=0
132 PRINT"WAS IT ON THAT CARD (Y/N)";
134 INPUT AS
136 IF AS="Y" THEN LET N=N+F:GOTO 139
138 IF AS<>"N" THEN GOTO 134
139 PRINT:PRINT
155 PRINT"THE NUMBER WAS ";N
```




PRIZE WINNERS

Famous Five Competition

In the May issue we offered a complete set of Famous five Books and five runners up prizes of the computer game The Famous Five on a Treasure Island from Enigma Variations, the winner of the set of 21 Famous Five Books was Jake Coleman, aged 10 from Doncaster and the runners up were; James Furneaux Smith, aged 8 from Birmingham; Kerry Lynch, aged 13 from Oxted; Philip Foster, aged 10 from Havant; Steven Mitchell, aged 15 from Wincanton and Jason Beard, aged 13 from Sheffield.

BugHunter Contest

Also in May, Minerva were offering 5 lucky readers a chance to zap the beasties with BugHunter I and II, the winners were: Geoffrey Styche, aged 14 from Camberley; James George, aged 13 from High Wycombe; Lorna James, aged 11 from St Whitchurch; Roland Walter, aged 12 from Battersea and Rachel Divall, aged 7 who comes from Bath.

SoundBlaster Contest

We asked our Let's Compute! members if they would like to make more noise with their favorite games and sound effects, we had three SoundBlasters from Siren Software on offer for four correct answers, the winners were; Tanya Webb, aged 13 from Dunstable; Daniel Eves, aged 10 from Wanstead and Tim Jay, aged 12 from Cawsand.

IS THIS YOUR COMPUTER?

```
10 CLS:INPUT"TYPE IN ANY NUMBER";N
20 FOR I=1 TO N:LET X=RND(1):NEXT I
30 INPUT "WHAT SPEED (TRY 1000)";S
40 READ N:DIM W$(N):LET C=0:LET D=0
50 FOR I=1 TO N:READ W$(I):NEXT I
60 LET R=INT(RND(1)*N+1):GOSUB 120
70 PRINT W$(R):FOR I=1 TO S:NEXT I
80 GOSUB 120:INPUT "WHAT WORD";G$
90 IF G$=W$(R) THEN LET C=C+1
100 IF G$<>W$(R) THEN LET D=D+1
110 GOSUB 120:FOR I=1 TO S:NEXT I:GOTO 60
120 CLS:PRINT"YOUR SCORE IS ";C;" OUT OF ";C+D:PRINT:RETURN
130 DATA 8
140 DATA "AHEAD","SOLVE","VASE","WEAK"
150 DATA "SUITE","ALBUM","SNOW","URGE"
```

BBC, Electron, Archimedes

The program works as shown

Commodore 64

Use PRINT CHR\$(147); in place of CLS in Lines 10 and 120.

Spectrum

Use DIM W\$(N,20):DIM G\$(20) in place of DIM W\$(N) in Line 40

Use RND in place of RND(1) in Lines 20 and 60

Amiga, ST(Use Stos), Amstrad CPC, PC(Use GW-Basic)

Use RND in place of RND(1) in Lines 20 and 60

DO YOU WANT MORE?

There's a far more powerful, flexible version of this program in **Spell!** - a collection of FIVE programs to help you learn your spellings.

As well as being able to use your own words, it comes complete with 5,000 built-in ones.

More details of Spell are on Page 27

In a FLASH!

How well can you spell? Here's a program to test your skills

Have you ever had to learn a load of spellings ready for a test at school? Once you think you know all the words you need someone to read them to check you out. But what if there's no one available?

You need never face this problem again. Instead use your computer and this short program.

It's called **FLASHER**. Just type it in and **SAVE** it. Then you're ready to **RUN** it.

You'll first be asked to type in a number. Any number will do but try to use a different one each time you run the program. It's just used to make sure you don't keep getting the words in the same order.

Next you'll be asked for the speed. Different makes of computer run at different speeds but try 1000 first. If you find the program runs too fast for you try a bigger number – say 10,000.

Next, a word will flash on the screen and disappear. Try to type it in correctly. Note that to be right you must type it exactly as it flashed – for example, use capital letters if that was how it appeared on your screen.

Your score is displayed constantly so you'll know whether you are right or

wrong and how well you're doing. Each time you type in a word another will be shown. So watch carefully. You have to be alert and a good speller to score well.

The program has been set up with eight sample words. You can put your own words in place of them in Lines 140 and 150.

Put as many words as you like: Just change the number in Line 130 to say how many there are. Note that each word should be in quotes (") and separated by a comma(.). But don't put a comma on the end of lines.

If you can't fit all your words into the two lines you can have as many extra as you like.

For example:

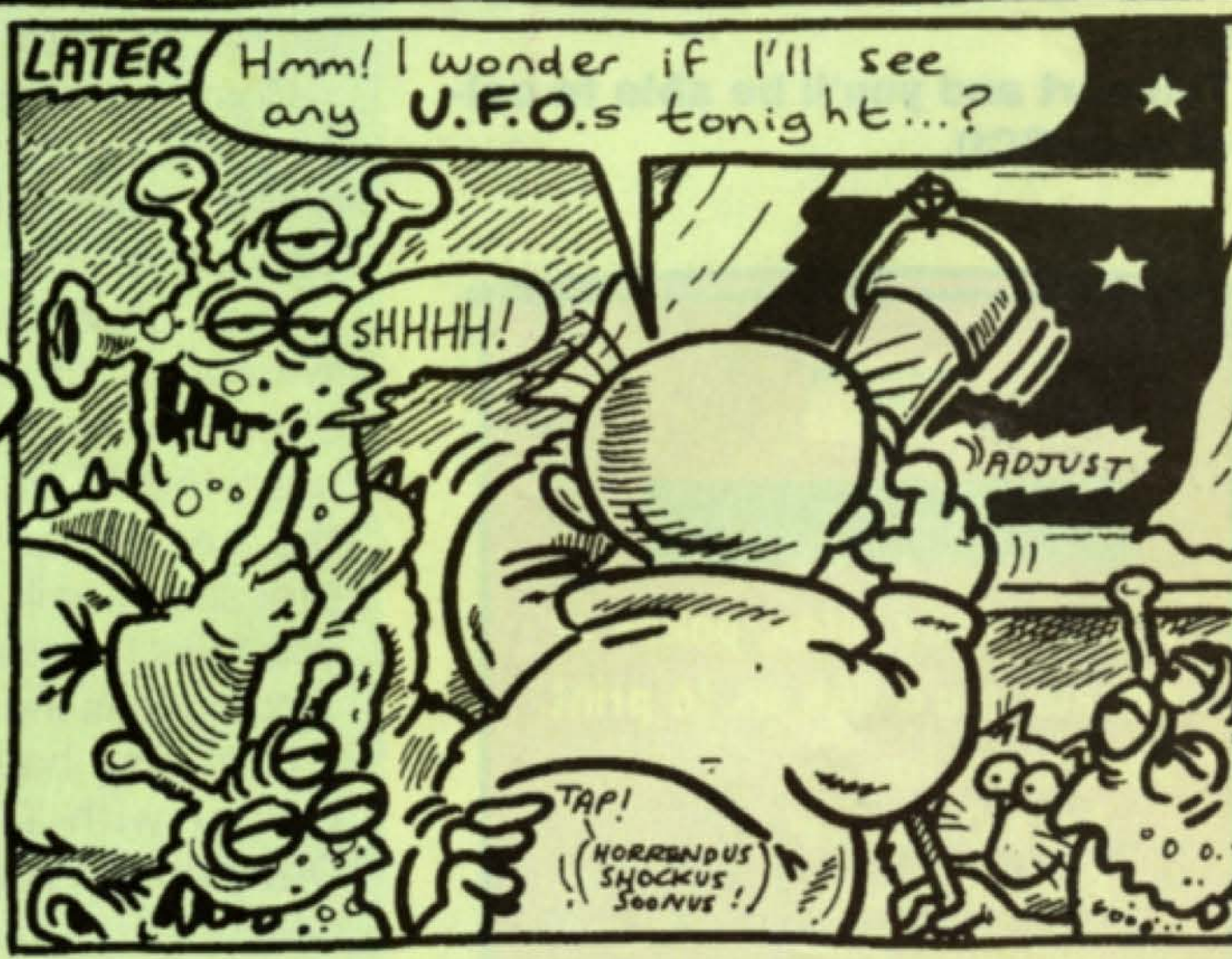
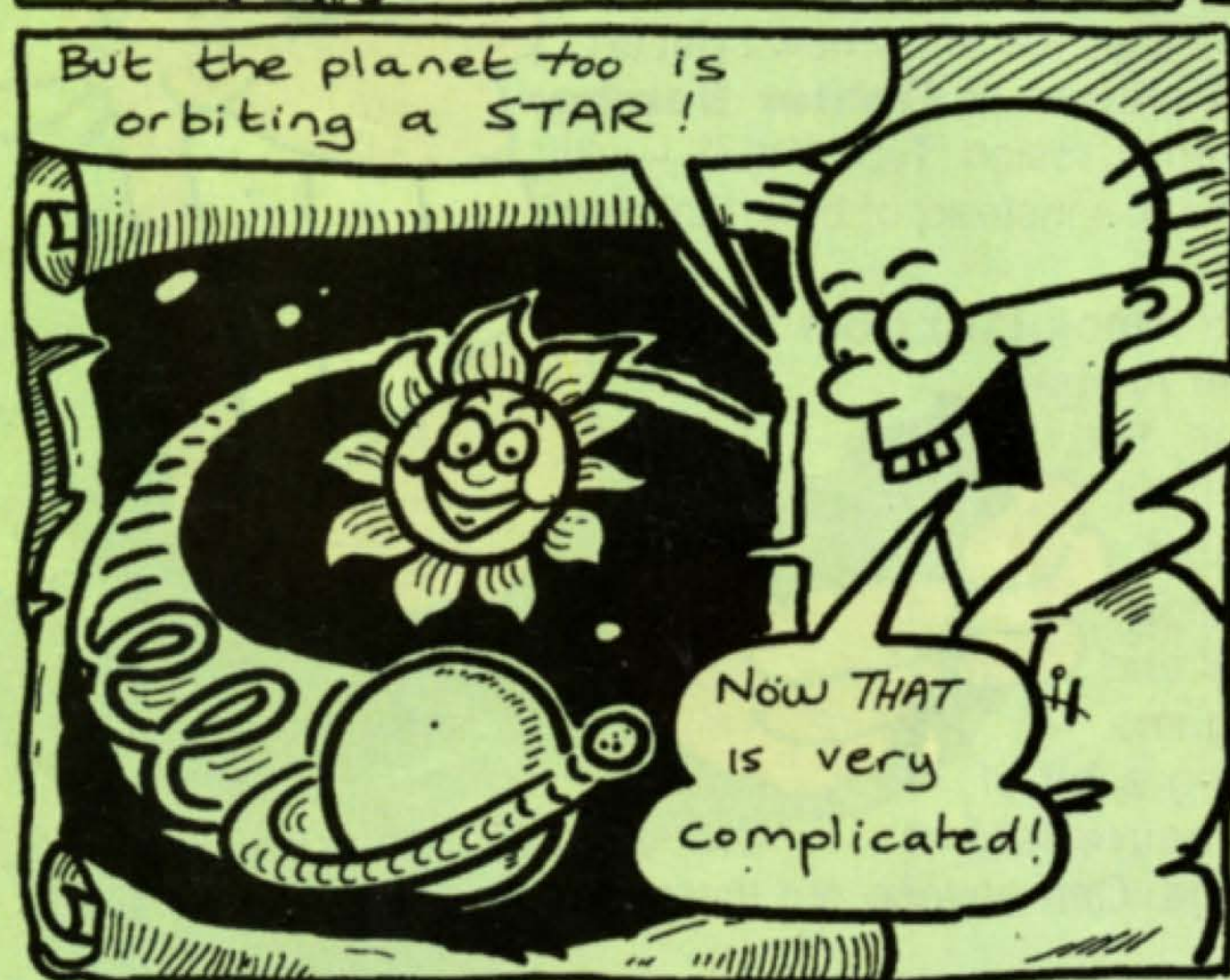
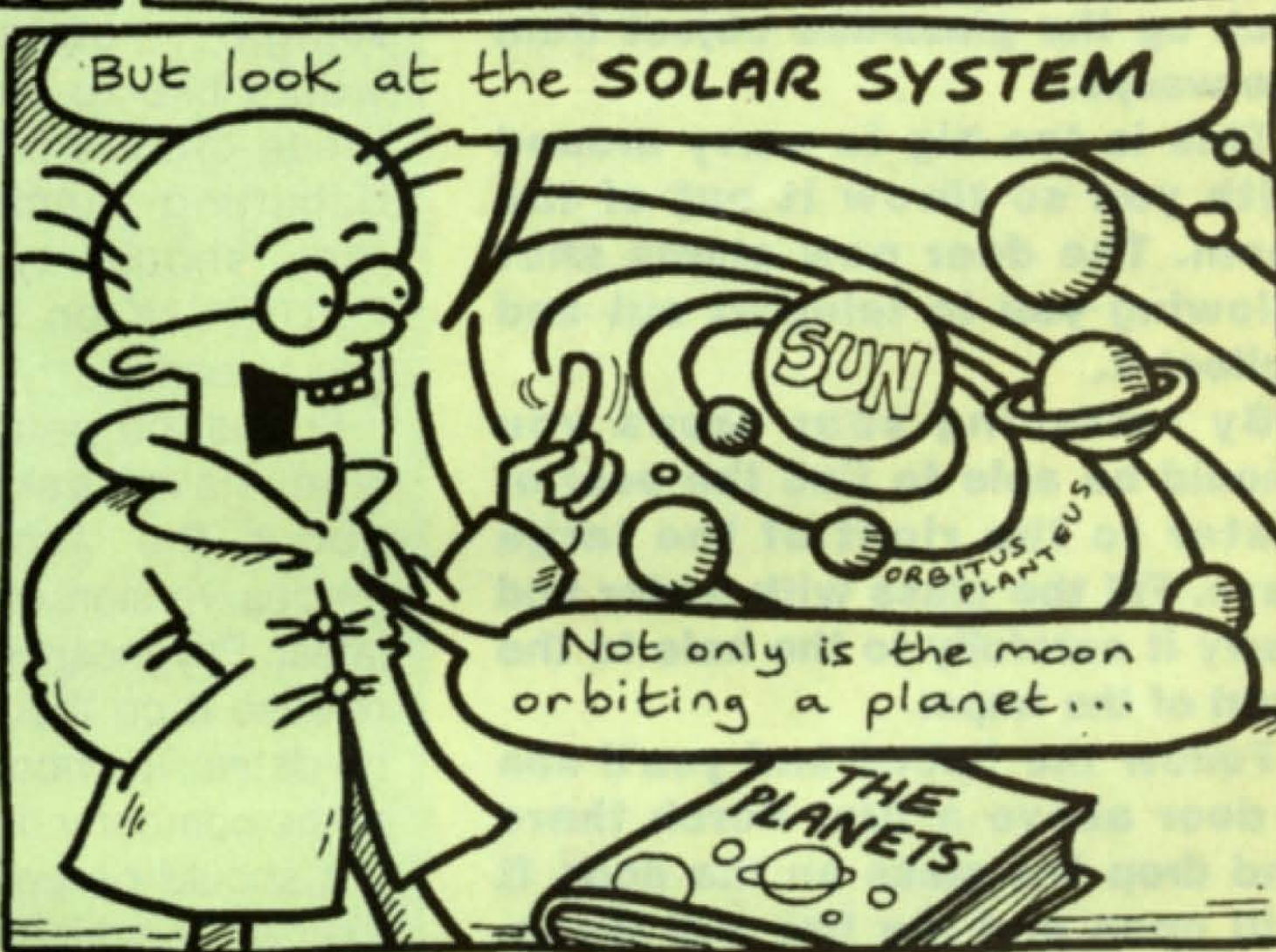
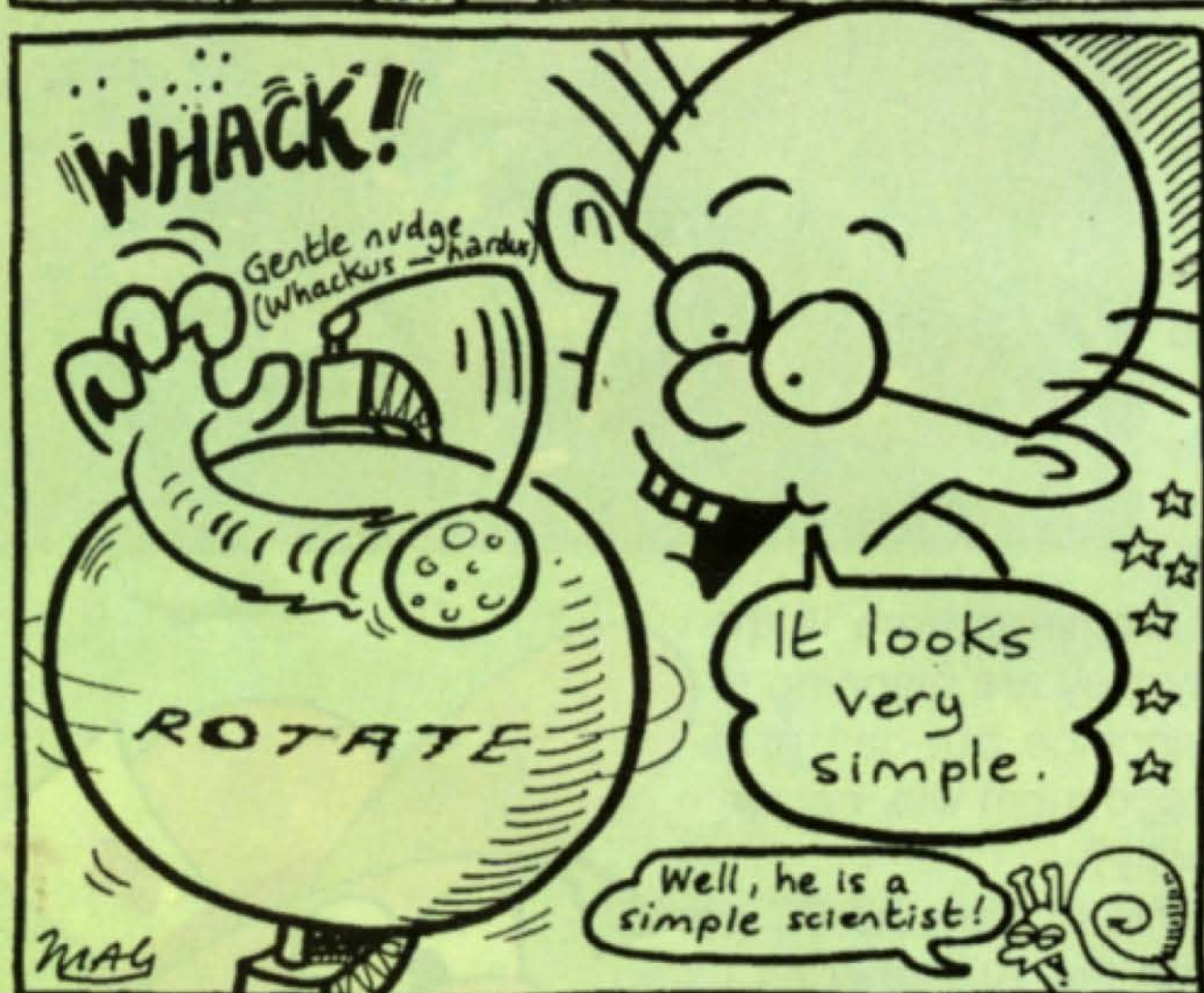
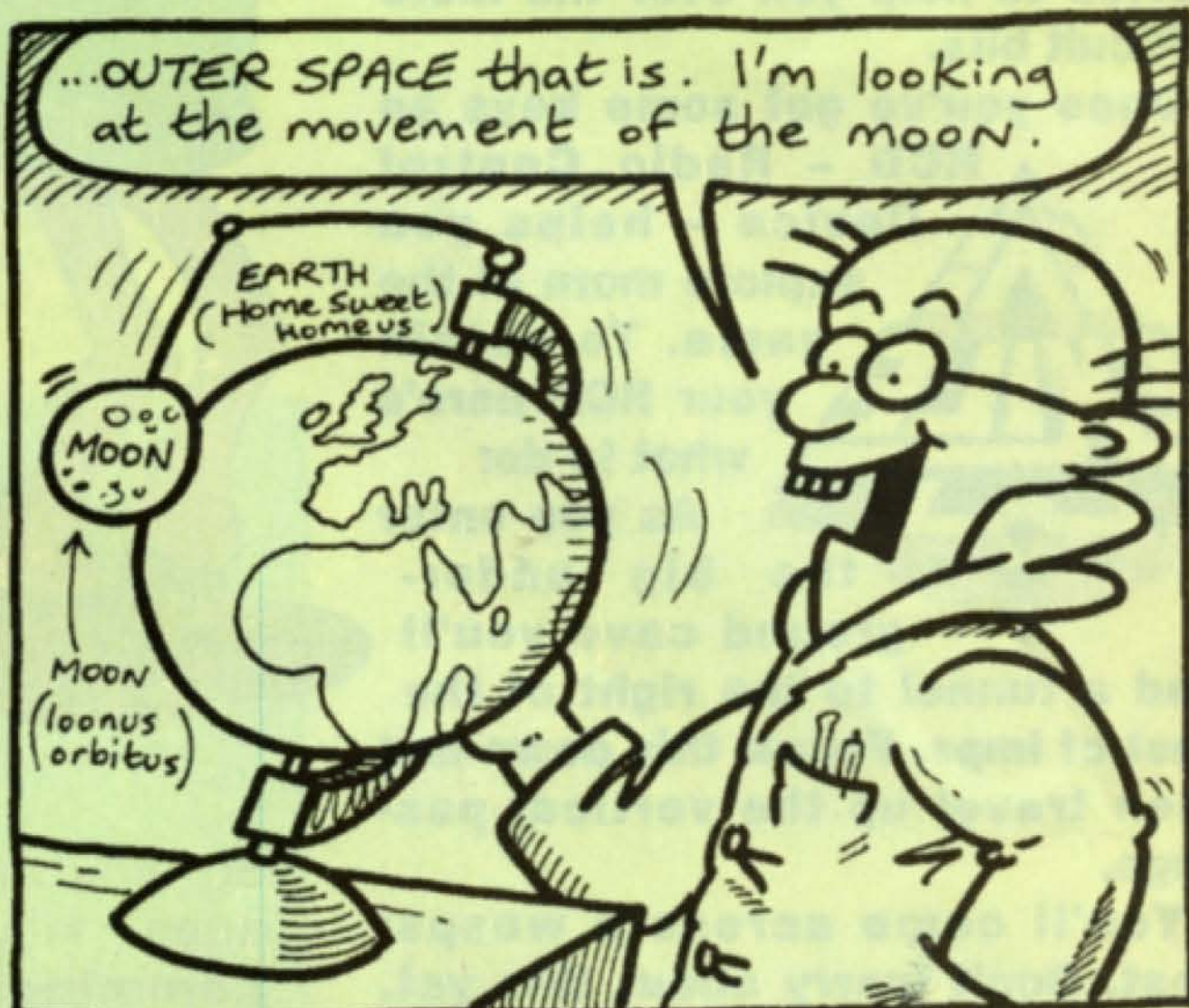
```
160 DATA, "SATISFACTORY", "ALTOGETHER", "P  
RESENT", "WELFARE"
```

Type it in now and see how it works.

You can also use this program to test yourself in lots of subjects such as French, Geography, German or Science. You can **SAVE** the program lots of times with different sets of words.



The Safe



Scientist

The Safe Scientist goes round in circles as he looks at orbits

WAA SPIN!

Hi! It's me again. This month I'm looking at the movement of a heavenly body.

In other words how a moon goes round a planet.

This may sound very simple. After all, it just goes round in circles. But this is only half the story!

You see, the planet is also orbiting a star. In the case of the earth, it is going round the sun. So the moon is spinning around an object which is also going round in circles.

The program here shows what is happening. Type it in, SAVE it and RUN it. Each time you press a key you'll see the new positions of the planet and moon.

You'll soon be able to see the surprising route the moon takes.

To save your computer having to work out all the positions of the planet and moon they are all stored in the DATA lines. This makes the program run faster.

It also lets you to experiment with your own ideas for orbits. Just change the numbers to get new, if unreal routes.

Try this!

If you just want the moon and planet on your screen you can wipe out the trail they leave. Add these lines to your program:

```
115 LET X=16-A+SGN(A):LET Y=11-B+SGN(B)
:LET SS=" ":GOSUB 140
116 LET X=X-C:LET Y=Y-D:LET SS=" ":GOSUB 140
```

```
10 REM MOON ORBIT
20 REM THE SAFE SCIENTIST
30 MODE1:VDU23;8202;0;0;0;0;
40 LET X=1:LET Y=0:LET SS="PRESS ANY
KEY FOR NEXT POSITION":LET COL=3
50 GOSUB 140
60LET X=16:LET Y=11:LET SS="S":LET CO
L=2:GOSUB 140
70 FOR N=1 TO 28
80 READ A,B,C,D
90 LET X=16-A+SGN(A):LET Y=11-B+SGN(B)
:LET SS="P":LET COL=1:GOSUB 140
100 LET X=X-C:LET Y=Y-D:LET SS="M":LET
COL=3:GOSUB 140
110 LET G=GET
```

```
120 NEXT N
130 END
140 COLOUR COL:PRINTTAB(X,Y);SS
150 RETURN
160 DATA 0,9,0,2,2,9,2,2,4,9,2,0,6,8,1
,-2,8,6,0,-2,9,4,-2,-2,10,2,-2,0,10
170 DATA 0,-1,2,10,-2,1,2,9,-4,2,2,8,-
6,2,0,6,-8,2,-2,4,-9,0,-2,2,-10,-2,0
180 DATA -10,-3,0,-2,-10,-2,2,-4,-9,0,
2,-6,-8,2,2,-8,-6,2,0,-9,-4,2,-2,-10,-2
190 DATA 0,-2,-10,0,-2,-2,-10,2,-2,0,-
9,4,-2,2,-8,6,0,2,-6,8,2,2,-4,9,2,0,-2
200 DATA 10,2,-2
```

IS THIS YOUR COMPUTER?

BBC/Electron/Archimedes

The program works as shown

Amstrad CPC

Change these lines:

```
30 CLS
110 WHILE INKEYS="" :WEND
140 INK COL:LOCATE X,Y:PRINT SS
```

ST(STOS)

Change these lines:

```
30 MODE 1:KEY OFF:CURS OFF:HIDE
110 WHILE INKEYS="" :WEND
140 INK COL:LOCATE X,Y:PRINT SS
```

Spectrum

Change these lines:

```
30 CLS
110 LET KS=INKEYS:IF KS="" THEN GOTO 110
310 STOP
140 INK COL:PRINT AT Y,X;SS
```

Amiga/PC(GW-Basic)

Change these lines:

```
30 CLS
110 WHILE INKEYS="" :WEND
140 COLOR COL:LOCATE Y+1,X+1:PRINT SS
```

Commodore 64/128

Change these lines:

```
30 PRINT CHR$(147);
110 GET AS:IF AS="" THEN GOTO 110
140 POKE 211,X:POKE 214,Y:SYS 58732
:PRINT CHR$(COL+17);SS
```

It's back down to earth next month. We'll be looking at your diet with a fun way to learn about food and energy