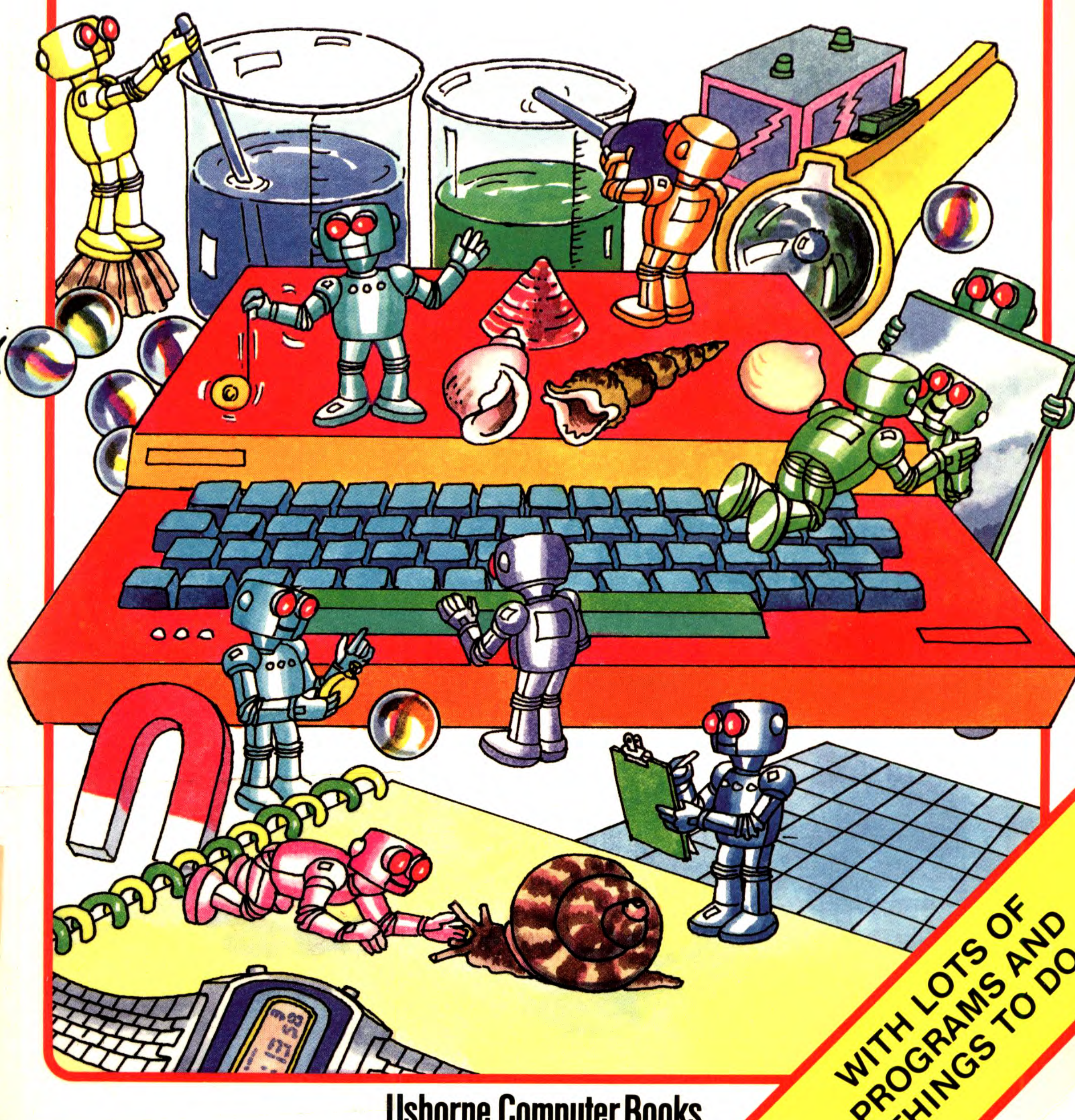
EXPERIMENTS WITH YOUR COMPUTER

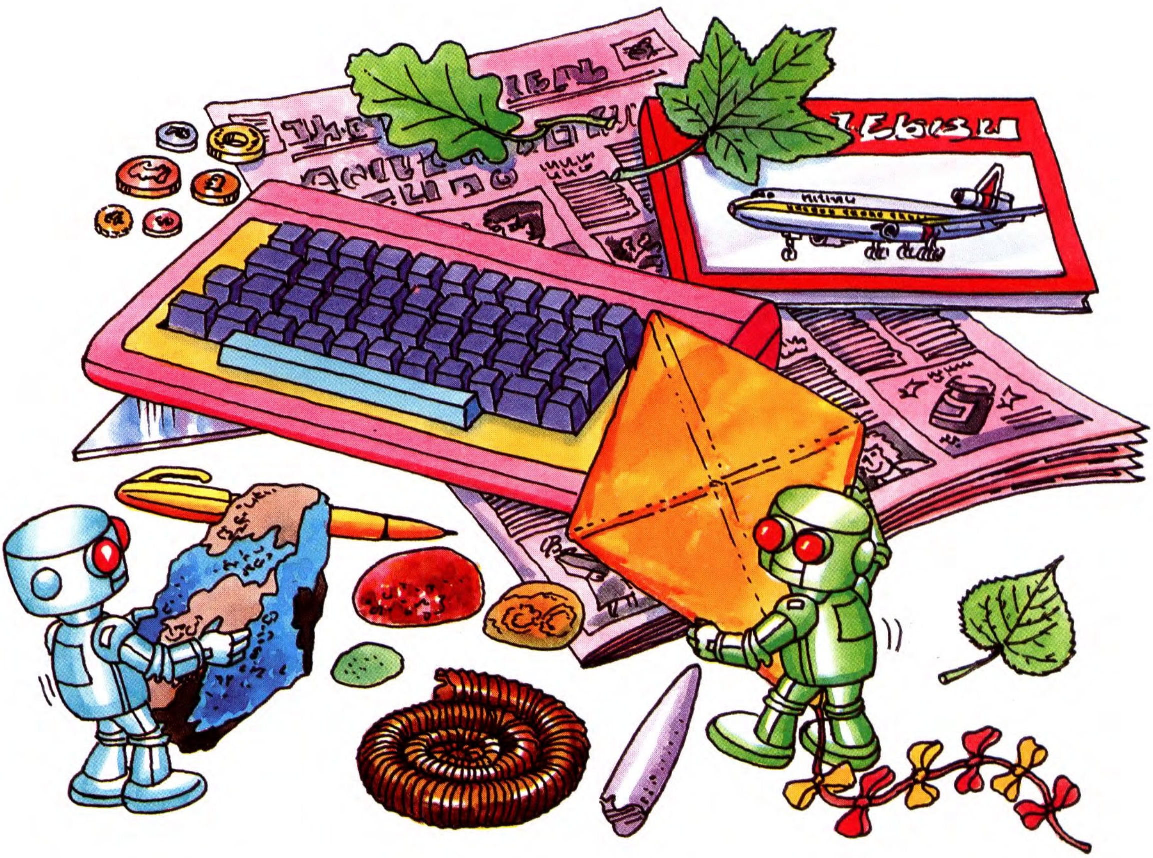


WITH LOTS OF PROGRAMS AND THINGS TO DO

Usborne Computer Books

EXPERIMENTS WITH YOUR COMPUTER

Helen Davies

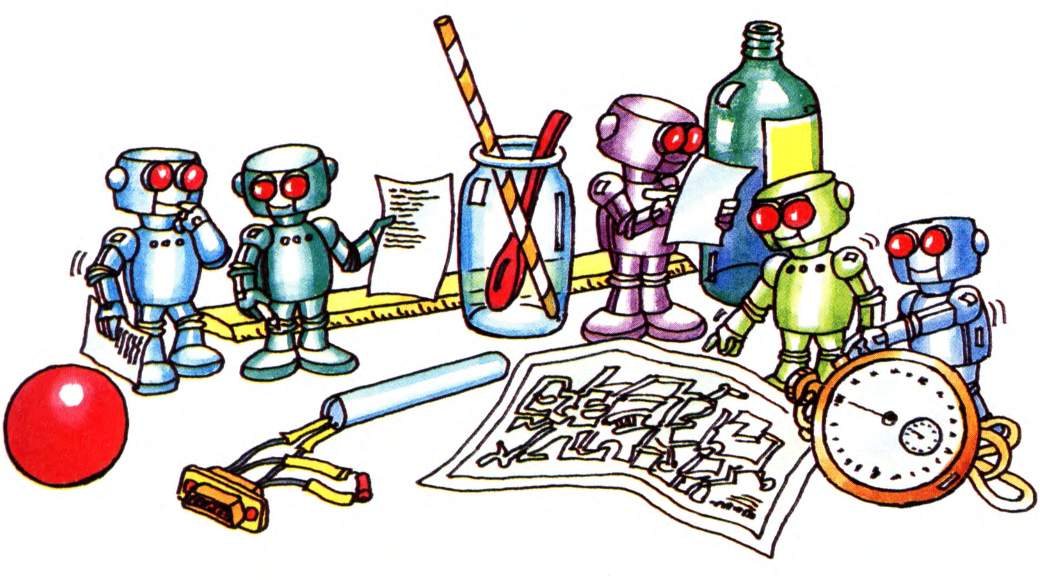


Designed by Graham Round and Kim Blundell

Programs by Chris Oxlade, Geoff Sheath, Bill Rivers, Philip Clarke and Chris Smith

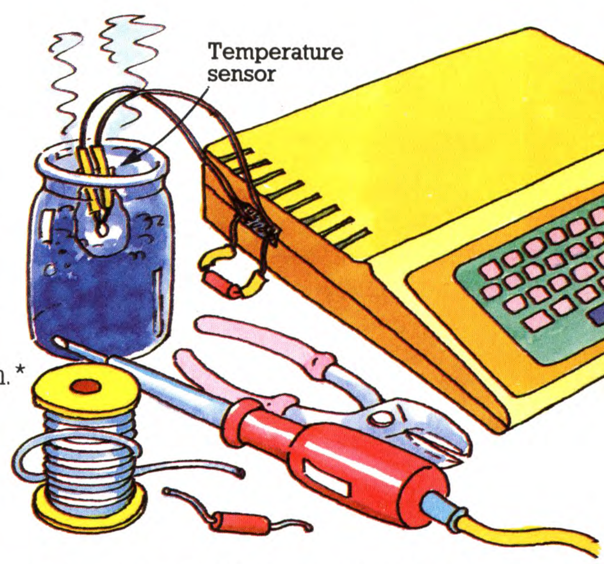
Illustrated by Graham Round, Chris Lyon and Jeremy Banks

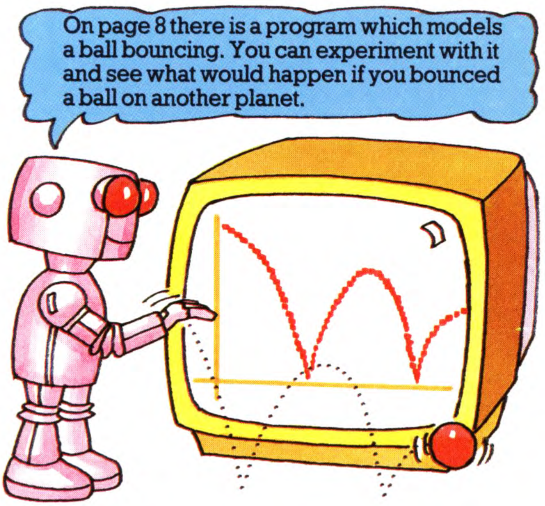
Contents



About this book

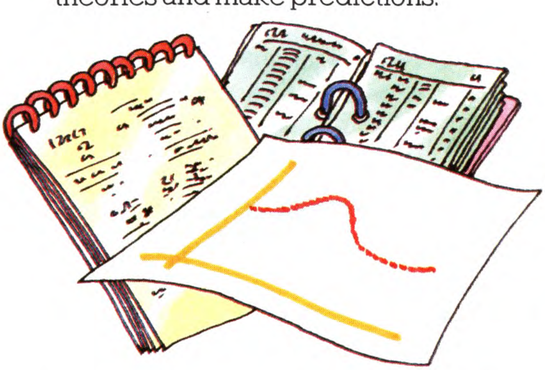
This book contains lots of fun programs which enable you to use your computer to do experiments and analyze information. The programs are simple examples of how scientists, economists and other researchers use computers. They are designed for home computers and will run on the Commodore 64, VIC 20, TRS-80 Colour Computer (32K), Apple II, BBC (B), Electron and Spectrum.*

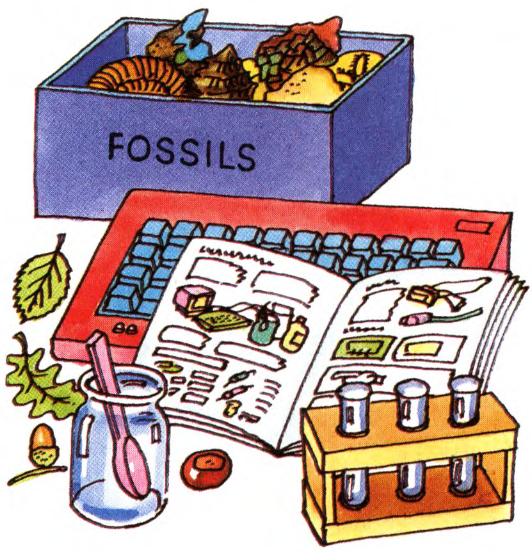




On pages 24-31 there are step-by-step instructions for building temperature and light sensors for your computer.* Sensors are electronic devices which measure things like temperature or light and convert the measurements into electrical signals. Using sensors a computer can take readings directly from an experiment and tell you what is happening.

Some of the programs are themselves experiments. You can alter different factors in the program and see what happens. These programs are called models. They enable you to test theories and make predictions.





Other programs are designed to deal with results you collect from experiments or surveys. They help you analyze the data and also store and display it in useful ways.

You can find out how to type in and use the programs on page 6. On pages 46-47 there is a graphics routine which you can add to several of the programs.

***The sensors on pages 24-31 only work with computers which have an "analogue" port, that is, the Commodore 64, VIC 20, TRS-80 and BBC.**

Computers in experiments and research

The programs used by scientists and other researchers are often extremely complex and are run on powerful computers. There are some examples of how such programs are used in the pictures on these two pages. The programs in this book are much simpler than those described here but they do similar jobs.

Modelling the big bang

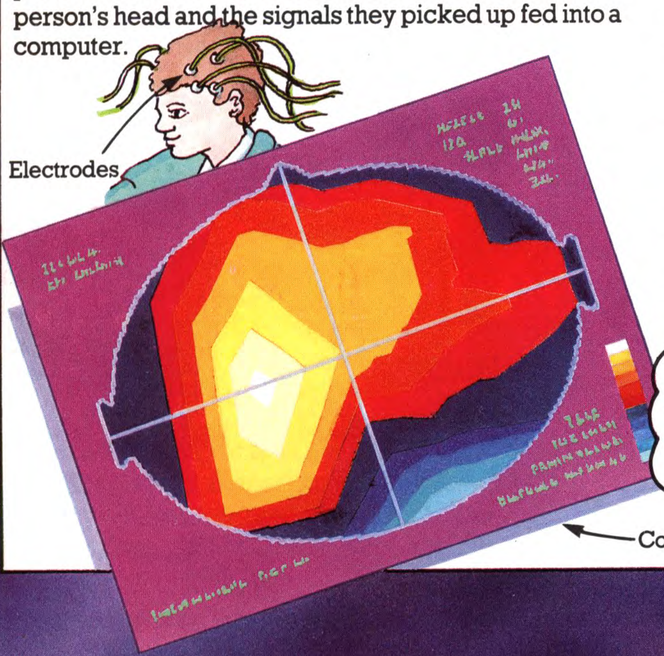
Computers are used to model events which are too vast or complex to be studied in reality. For instance, physicists in the USA are using one of the world's most powerful computers, a Cray-1, to model the birth of the Universe.

The model is a program based on physicists' theories that the Universe was born in a massive explosion called the big bang. The program uses mathematical equations to represent what happened during the first second of the explosion and traces the structure of the universe from then until now.

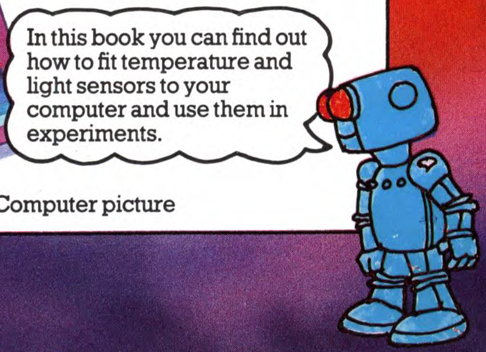
By comparing the model's predictions with actual observations scientists can see whether their theories seem right.

Sensors in brain research

This is a computer picture of someone thinking. To produce it sensors called electrodes were fitted to a person's head and the signals they picked up fed into a computer.

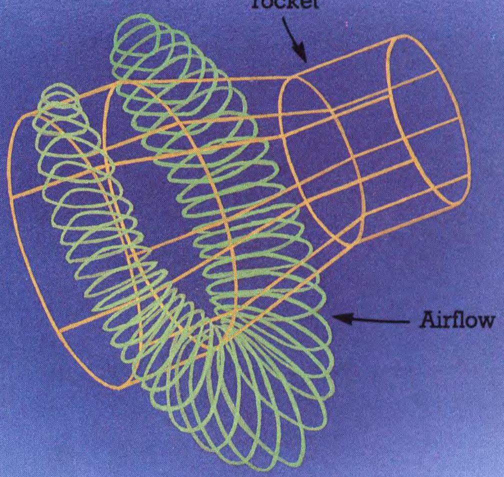


The different colours show areas of high and low electrical activity in the brain. Pictures like this help researchers understand which parts of the brain are used for different activities. They are being used to investigate why some people have reading difficulties, and also to diagnose brain tumours and epilepsy.

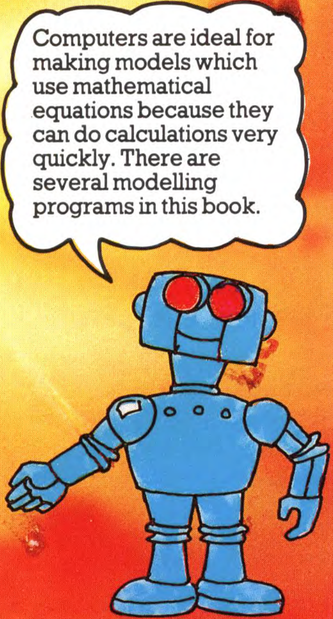


Designing a rocket

Computer models are also used to develop and test new products and machines, such as space craft. The picture below was produced by a computer modelling a rocket taking off into space.



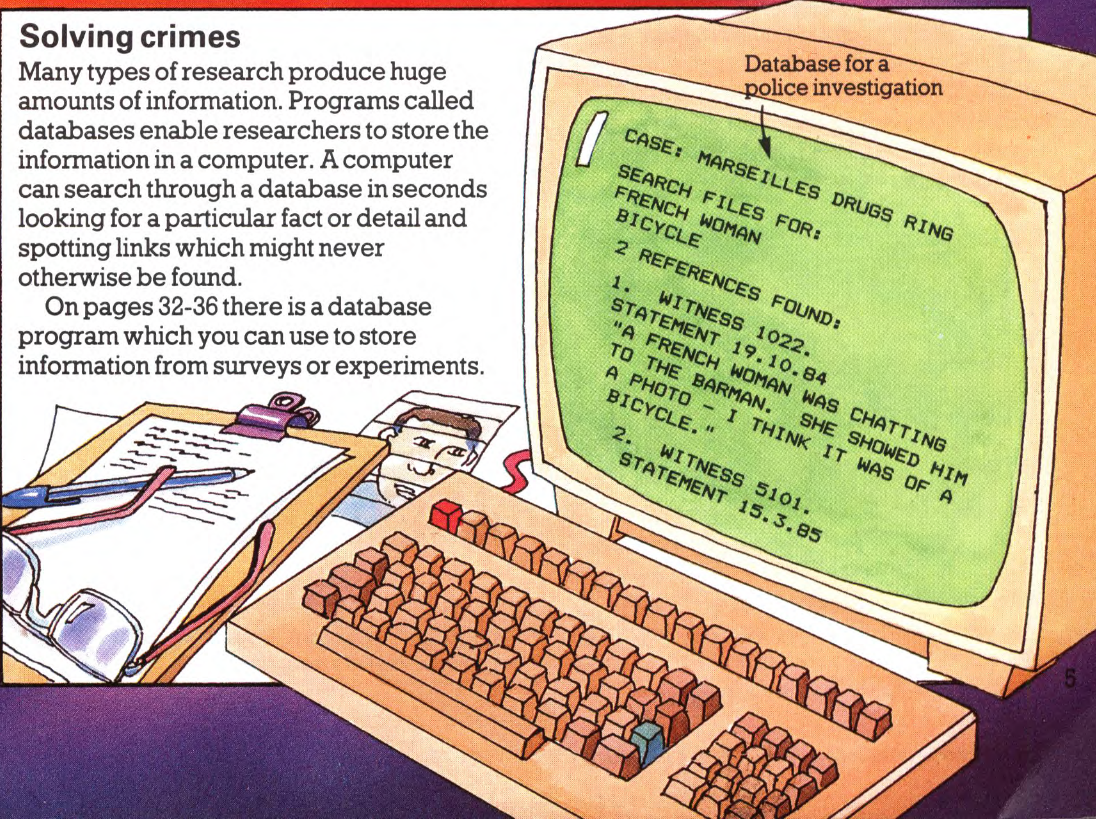
The green shape shows the flow of air around the rocket when it is travelling at just below the speed of sound. By experimenting with the model, space engineers were able to predict the effect of altering the rocket's angle of attack (that means the angle at which it pierces the air).



Solving crimes

Many types of research produce huge amounts of information. Programs called databases enable researchers to store the information in a computer. A computer can search through a database in seconds looking for a particular fact or detail and spotting links which might never otherwise be found.

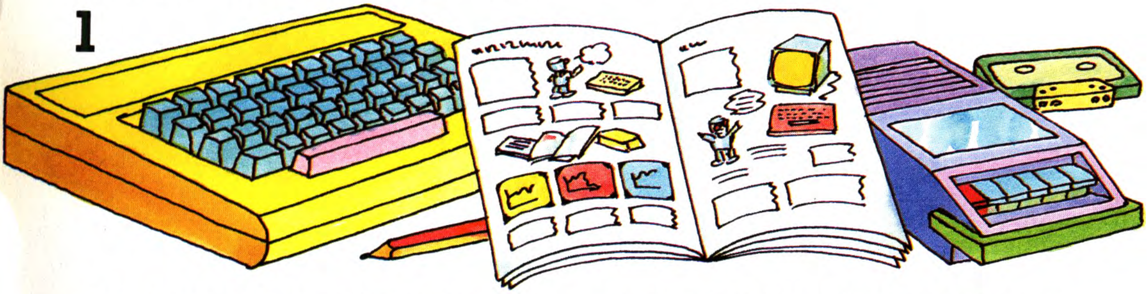
On pages 32-36 there is a database program which you can use to store information from surveys or experiments.



Using the programs in this book

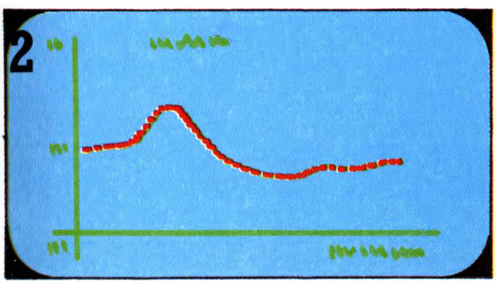
Before using the programs in this book read these hints on typing them in and running them. Some program lines need to be changed for different computers. These are marked with an asterisk and the conversion lines are given on pages 42-45. When you have typed a program and checked it is working, save it on tape in case you want to use it again.

1



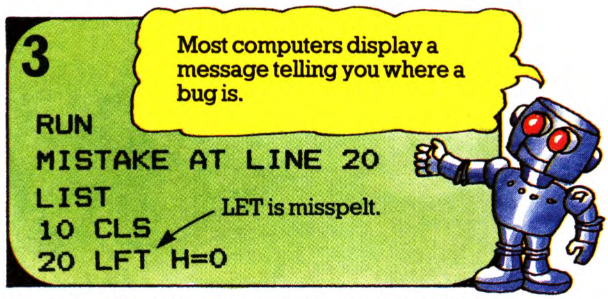
Type the programs exactly as they are printed. At the end of each line make sure there are no mistakes, then press RETURN (or ENTER on some computers).* When you come to a line with an asterisk turn to pages 42-45 and look at the conversions for your computer. If there is no conversion you can type the line just as it is in the program.

2



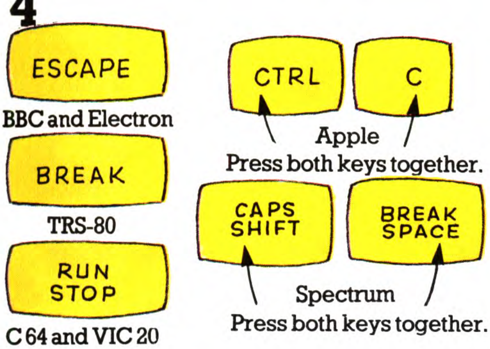
Some of the programs need to have a graphics routine added before you can use them. There are different versions of the routine for each computer and these are given on pages 46-47. Make sure you add the right one for your computer.

3



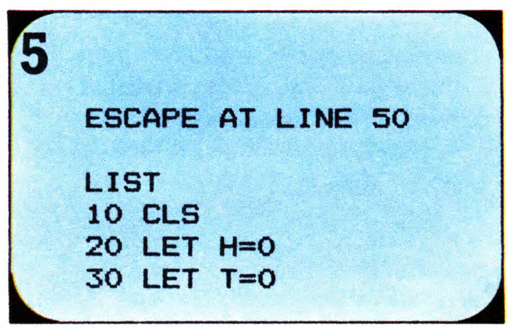
To make the computer carry out a program you type RUN. If the program does not work first time, there is probably a mistake (called a bug) in it. To debug a program, list the program lines on the screen, then retype or edit the lines with mistakes.

4



Sometimes you need to add or change program lines. You may have to stop the program running in order to do this. Press the "escape" key (shown above).

5

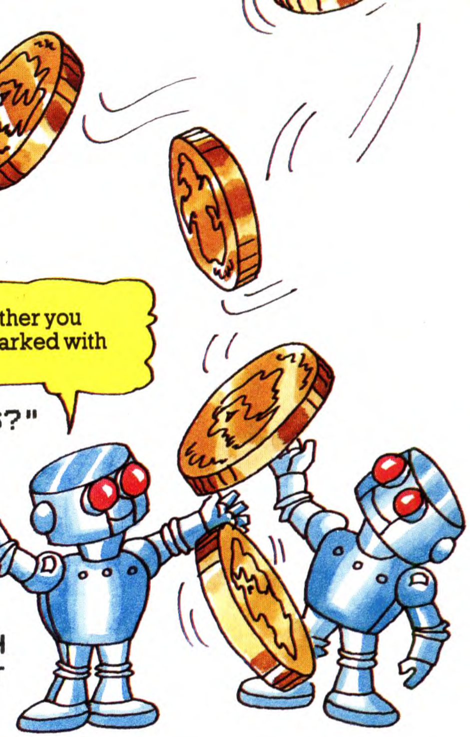


Then list the program and type in the new lines with their line numbers. To rerun a program type run again.

*If your computer has an ENTER key press it each time you see RETURN in this book.

Coin-tossing program

Here is a simple program to type in and try on your computer. It models a coin being tossed. You can experiment with it to see what results you get when you toss the coin different numbers of times.



Check pages 42-45 to see whether you need to alter any of the lines marked with asterisks.

```
★10 CLS
 20 LET H=0
 30 LET T=0
 40 PRINT:PRINT "HOW MANY TOSSES?"
 50 INPUT N
 60 FOR K=1 TO N
★70 LET X=RND(1)
 80 IF X<=0.5 THEN LET H=H+1
 90 IF X>0.5 THEN LET T=T+1
100 NEXT K
110 PRINT "NUMBER OF HEADS = ";H
120 PRINT "NUMBER OF TAILS = ";T
```

Running the program

When you run the program you tell the computer how many times to toss the coin. It models the tosses and shows how many heads and tails you got.

1

```
HOW MANY TOSSES?
?10

NUMBER OF HEADS = 8
NUMBER OF TAILS = 2
```

Try modelling the toss ten times and see what result you get. Then try a hundred or a thousand tosses.

2

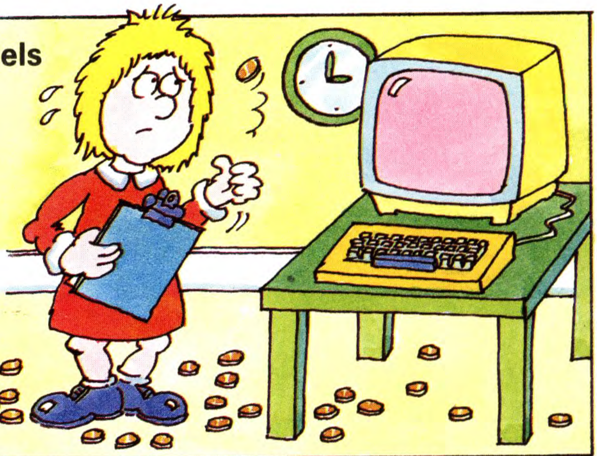
```
HOW MANY TOSSES?
?100

NUMBER OF HEADS = 48
NUMBER OF TAILS = 52
```

You should find the more times you model the toss the closer your result is to 50% heads and 50% tails.

More about computer models

Although this is a very simple program it shows how a computer model can save time and money doing real experiments.

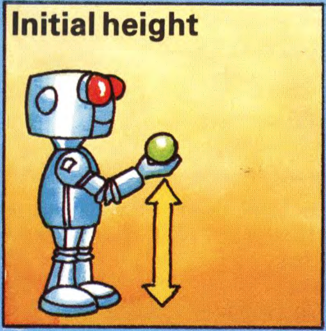
Time yourself tossing a coin a hundred times and writing down the result of each toss. How does your time compare with the computer's?



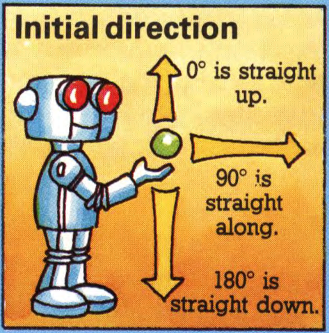
Bouncing ball

The program on this page models a bouncing ball. You can experiment with it and see what happens if you throw the ball from different heights, or throw it harder, or use a bouncier ball. You can even alter the force of gravity.

When you run the program you need to give the computer information about the ball as shown on the right.



This is the height above the ground of your hand when you let go of the ball. You could try 1 metre or 3 feet.

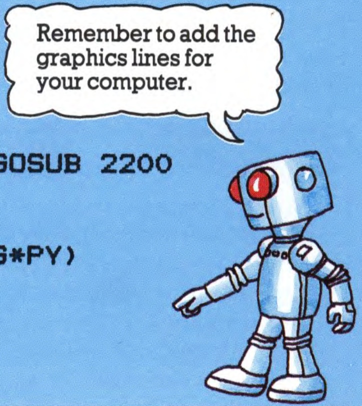


The direction of the ball is given as an angle, in degrees. For example, if you throw the ball up the angle is between 0° and 90°.

Bouncing ball program

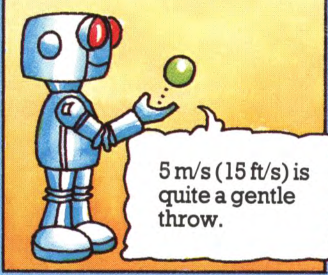
Type in these program lines, then turn to pages 46-47 and add the graphics routine for your computer. The program works out distances in metres. If you want it to use feet replace lines 20, 40, 50, 70 and 170 with those given at the bottom of this page.

```
★10 CLS:PRINT
 20 PRINT "INITIAL HEIGHT (M)":INPUT H
 30 PRINT "INITIAL DIRECTION (DEGREES)":INPUT A
 40 PRINT "INITIAL SPEED (M/S)":INPUT V
 50 PRINT "GRAVITY (M/S/S)":INPUT G
 60 PRINT "BOUNCINESS (0-1)":INPUT B
 70 LET T$="":LET Y$="5 M":LET X$="20 M":LET M$="2.5"
 80 GOSUB 2000
 90 LET DT=0.01:LET TB=20:LET NB=0:LET K=3.14/180
100 LET HS=V*SIN(A*K):LET VS=V*COS(A*K)
110 LET NX=0:LET NY=H
120 LET PX=NX:LET PY=NY:LET US=VS
130 LET NX=PX+HS*DT
140 LET VS=US-G*DT
150 LET NY=PY+((US+VS)/2)*DT
160 IF NY<=0 THEN GOSUB 200
170 LET Y=PY*200:LET X=PX*50:LET N=0:GOSUB 2200
180 IF X<900 AND NB<TB THEN GOTO 120
190 STOP
200 LET VS=SGN(US)*SQR((ABS(US)^2)+2*G*PY)
210 LET HT=(US-VS)/G
220 LET NX=PX+HS*HT:LET NY=0
230 LET VS=-VS*B:LET NB=NB+1
240 RETURN
```



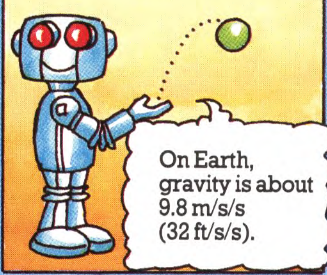

```
 20 PRINT "INITIAL HEIGHT (FT)":INPUT H
 40 PRINT "INITIAL SPEED (FT/S)":INPUT V
 50 PRINT "GRAVITY (FT/S/S)":INPUT G
 70 LET T$="":LET Y$="15 FT":LET X$="60 FT":LET M$="7.5"
170 LET Y=PY*66:LET X=PX*17:LET N=0:GOSUB 2200
```

Initial speed



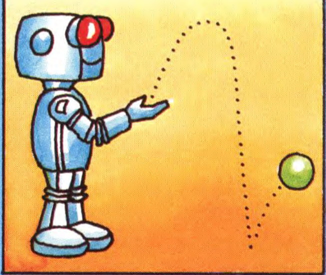
This is the speed of the ball the moment after it leaves your hand. It is measured in metres per second (m/s) or feet per second (ft/s).

Gravity



Gravity is the force which pulls things to the ground. It is measured in metres per second squared (m/s/s) or feet per second squared (ft/s/s).

Bounciness



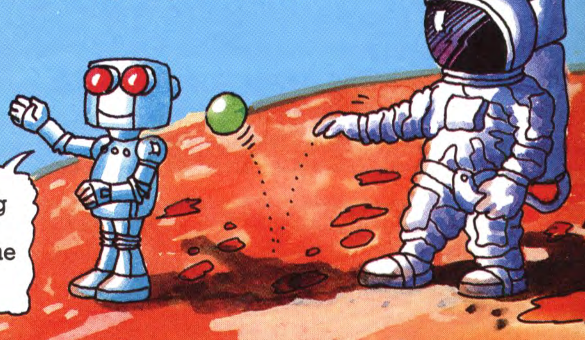
This is a measure of how bouncy a ball you are using. It can vary between 0 and 1. A hard rubber ball would be 0.9. Try 0.5 or 0.7.

Running the program

```
INITIAL HEIGHT (M)
?1.5
INITIAL DIRECTION (DEG)
?45
INITIAL SPEED (M/S)
?6
GRAVITY (M/S/S)
?10
BOUNCINESS
?0.7
```

Try the model with these values. Does it look like a bouncing ball? Run it again keeping everything the same except the direction. Try directions of 60°, 20°, 135°. Which makes the ball go furthest? Which gives the highest bounces?

Mercury experiment



On the planet Mercury gravity is only 3.7 m/s/s (12.1 ft/s/s). No-one has ever been there but you can use the model to find out what would happen to a bouncing ball on Mercury. Make gravity 3.7 (or 12.1) and keep everything else the same.

How the model works

Like the big bang model on pages 4-5 this program is a mathematical model. It works out the path of the ball using equations based on standard laws of physics. They were discovered by a physicist called Isaac Newton who lived between 1642 and 1727. The laws relate to the movement of objects and the pull of gravity.

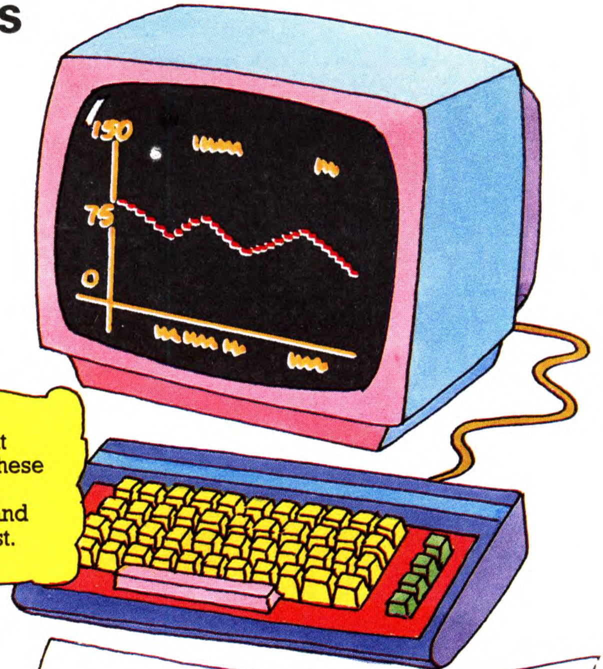
Ball bouncer's guide to the Universe

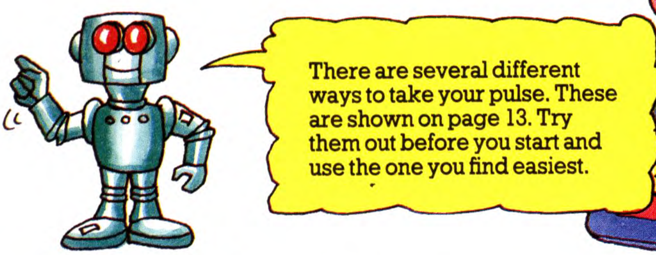
VENUS	8.8 m/s/s or 28.9 ft/s/s
MARS	3.7 m/s/s or 12.1 ft/s/s
JUPITER	26.5 m/s/s or 86.9 ft/s/s
SATURN	11.8 m/s/s or 38.7 ft/s/s
URANUS	9.1 m/s/s or 29.6 ft/s/s
NEPTUNE	12 m/s/s or 39.4 ft/s/s
PLUTO	0.4 m/s/s or 1.3 ft/s/s

Here are the values for gravity on some other planets so you can take your ball on a trip round the Universe.

Pulse rate experiments

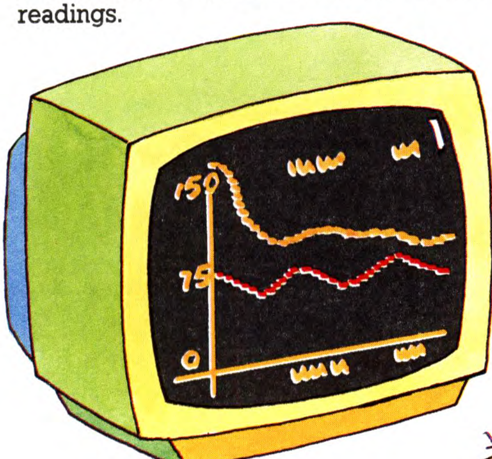
The program on the next two pages makes the computer store readings of your pulse rate and display them as a graph. You can use it to compare your pulse rate before and after exercise. You can find out about running the program opposite. Below there are some hints on how to use it.





Hints for using the program

Use the program to make a set of pulse readings before doing any exercise. The computer will display a graph of your normal pulse rate. Then do some exercise and give the computer another set of pulse readings.



Ideas for exercises

JOGGING

DOING PRESS-UPS

DOING SIT-UPS

CYCLING

SPRINTING OVER A SHORT DISTANCE

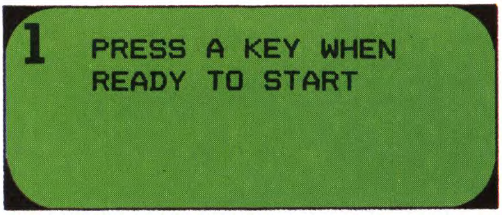
DOING STEP-UPS (CLIMBING ONTO A CHAIR AND OFF AGAIN)

Measure the amount of exercise you do. For instance, jog for a set length of time, e.g. ten minutes, or sprint over a measured distance or count the number of press-ups you do.

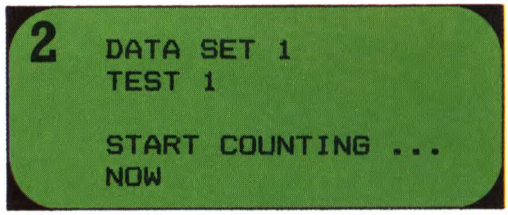
The computer will draw a second graph in a different colour so you can see how much your pulse rate has increased and how quickly it returns to normal.

The program allows you to make a third set of readings, so you can do a different kind of exercise and compare your recovery rate for that.

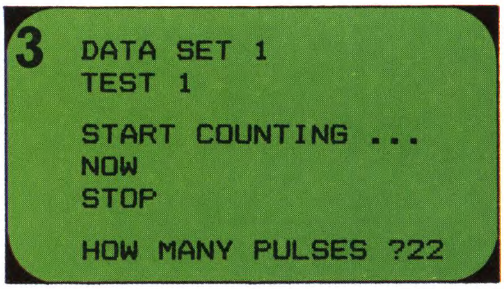
Running the program



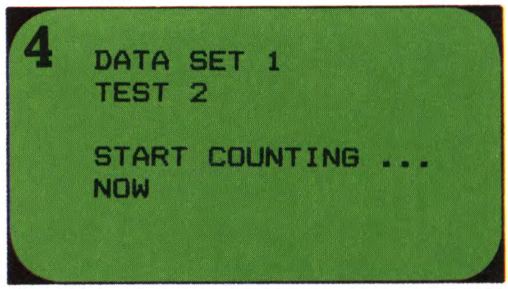
When you run the program this message tells you to get ready to take your pulse. Once you have pressed a key you have about 15 seconds to find your pulse.



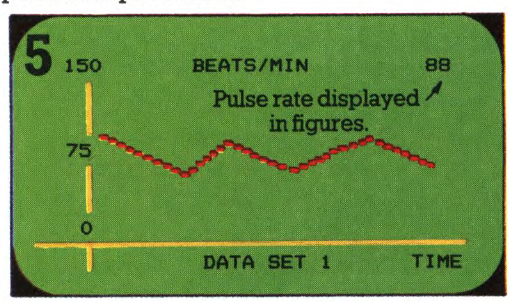
Start counting your pulse when the computer beeps and displays this message. After 15 seconds the computer beeps again and you should stop counting.



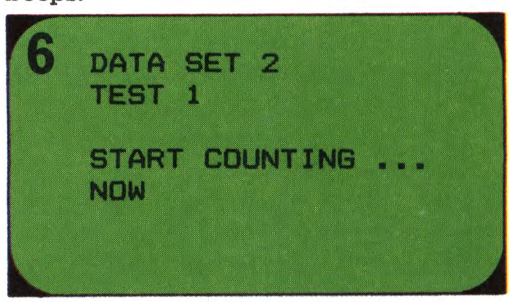
Type in the number of pulses you counted and press RETURN. The computer multiplies this by four to work out your pulse rate per minute.



When you have entered the reading, you have about 15 seconds to find your pulse again. Start counting when the computer beeps.



After you have taken ten pulse readings the computer displays a graph showing your pulse rate per minute against time.



Do some exercise and then take a second set of readings (called DATA SET 2). To make the computer start taking readings again press any key.

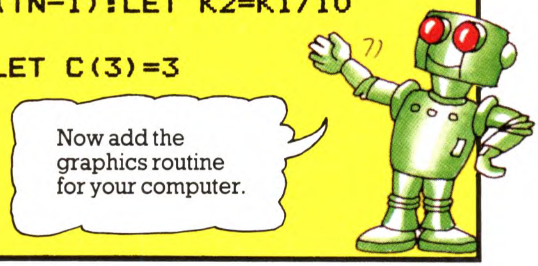
Pulse rate program

The main part of the program is given below and over the page. Type it in and then add the graphics routine on pages 46-47.

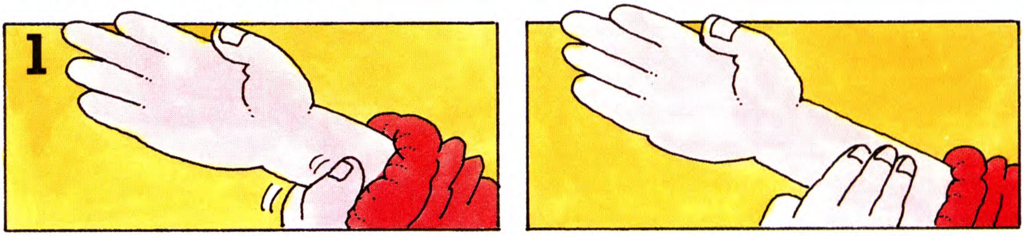
```
10 GOSUB 500
20 GOSUB 460
30 FOR S=1 TO 3:GOSUB 210
40 LET X$="TIME":LET Y$="150":LET M$="75"
50 LET T$="BEATS / MIN":GOSUB 2000
```

Listing continued over the page.

```
 60 FOR J=1 TO S
 70 LET DS=J:GOSUB 120
 80 NEXT J
 90 GOSUB 390
100 NEXT S
110 GOSUB 390:STOP
120 LET DC=C(DS)
130 LET M$="DATA SET "+CHR$(48+DS):GOSUB 2400
140 FOR K=1 TO TN-1
150 LET DY=(N(DS,K+1)-N(DS,K))/10
160 FOR W=0 TO 9
170 LET X=K*K1+W*K2-K1:LET Y=(N(DS,K)+DY*W)*6.6
180 LET N=N(DS,K):GOSUB 2200
190 NEXT W:NEXT K
200 RETURN
210 GOSUB 420:PRINT:PRINT
220 PRINT "PRESS A KEY WHEN"
230 PRINT "READY TO START"
240 GOSUB 390
250 FOR K=1 TO TN
260 GOSUB 420:PRINT:PRINT
270 PRINT "DATA SET ";S
280 PRINT "TEST ";K:PRINT:GOSUB 370
290 PRINT "START COUNTING...":GOSUB 350
300 GOSUB 440:PRINT "NOW":GOSUB 370
310 GOSUB 440:PRINT "STOP":PRINT
320 PRINT "HOW MANY PULSES ";:INPUT X
330 LET N(S,K)=X*4:NEXT K
340 RETURN
350 FOR T=1 TO PT:NEXT T
360 RETURN
370 FOR T=1 TO FT:NEXT T
380 RETURN
★390 GET I$
400 IF I$="" THEN GOTO 390
410 RETURN
★420 CLS
430 RETURN
★440 PRINT CHR$(7)
450 RETURN
460 LET TN=10:LET K1=1000/(TN-1):LET K2=K1/10
470 DIM N(3,TN):DIM C(3)
★480 LET C(1)=1:LET C(2)=2:LET C(3)=3
490 RETURN
★500 LET FT=7285:LET PT=900
510 RETURN
```

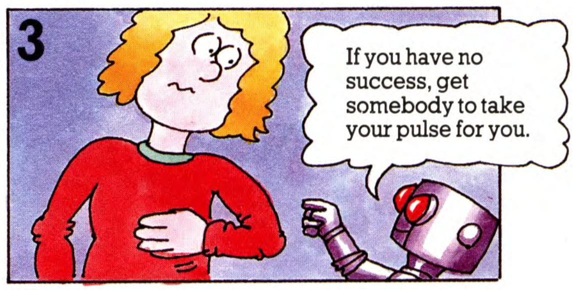


Ways to take your pulse



The most common place to take your pulse is at the wrist. Roll up the sleeve on your right arm and hold out your right hand with the palm facing upwards. Put the first, second and third fingers of your left hand on to your wrist, so they are pushing into the skin on the far side. There is an artery just under the skin at this point and you should feel a pulse under your fingertips.

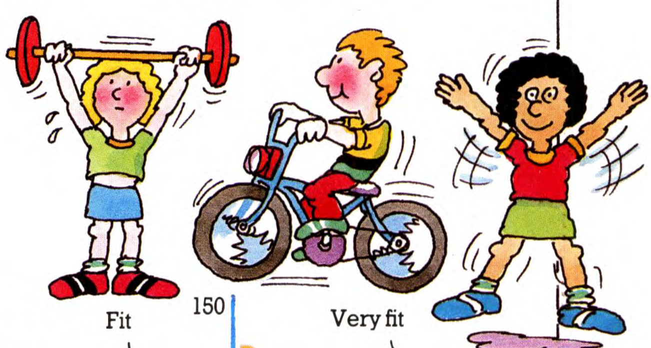
Another place to feel your pulse is your neck. Hold the fingers of your right hand on the lumpy part at the front. Then slide them round about 5cm to the left. There is a dip in the skin where you can feel your pulse.

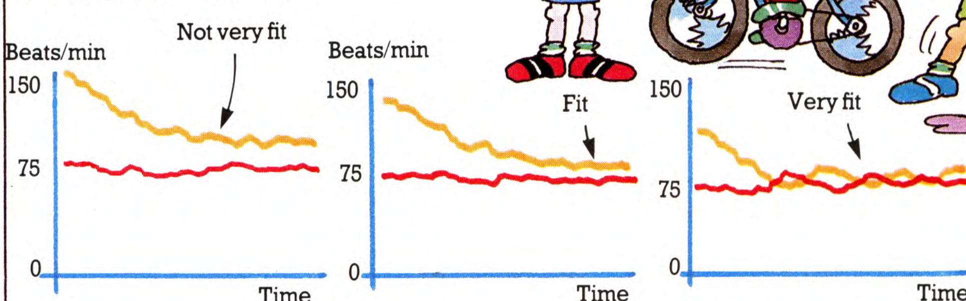


If you cannot find your pulse see if you can feel your heart beating. Hold your hand about three or four ribs up on the left-hand side of your body. You may be able to feel your heart beating at this point.

How fit are you?

The fitter you are, the quicker your pulse returns to normal after exercise. The graphs below show the recovery rate for different levels of fitness. Compare your graph with these to see how fit you are.





If you are doing regular exercise you can use the program to find out if you are getting fitter. Repeat the pulse rate tests at intervals over several months and each time note down the shape of the graphs. By comparing the shapes you can see whether your recovery rate is improving.

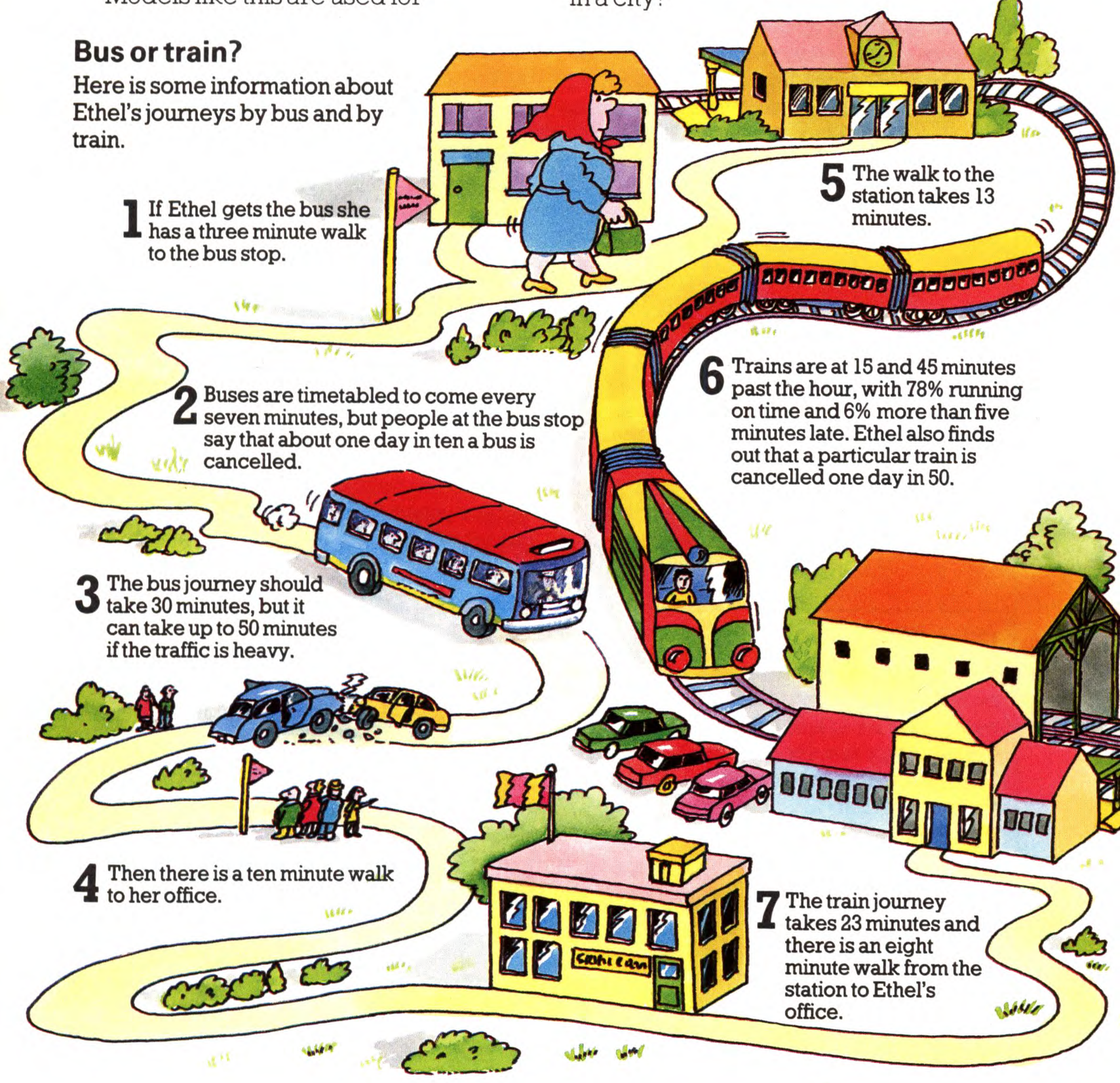
Ethel's journey

Ethel has just moved house and wants to find out whether it is better to go to work by train or by bus. The program opposite can help her. It models the journeys by bus and train over and over again, so she can see which is better over a period of six months or a year.

Models like this are used for "operational research", that is, finding out the most efficient way to do things. For instance, how should a self-service restaurant be organized so people do not have to queue for a long time? Or what is the best way to organize ships unloading cargo at a port, or traffic flow in a city?

Bus or train?

Here is some information about Ethel's journeys by bus and by train.



1 If Ethel gets the bus she has a three minute walk to the bus stop.

2 Buses are timetabled to come every seven minutes, but people at the bus stop say that about one day in ten a bus is cancelled.

3 The bus journey should take 30 minutes, but it can take up to 50 minutes if the traffic is heavy.

4 Then there is a ten minute walk to her office.

5 The walk to the station takes 13 minutes.

6 Trains are at 15 and 45 minutes past the hour, with 78% running on time and 6% more than five minutes late. Ethel also finds out that a particular train is cancelled one day in 50.

7 The train journey takes 23 minutes and there is an eight minute walk from the station to Ethel's office.

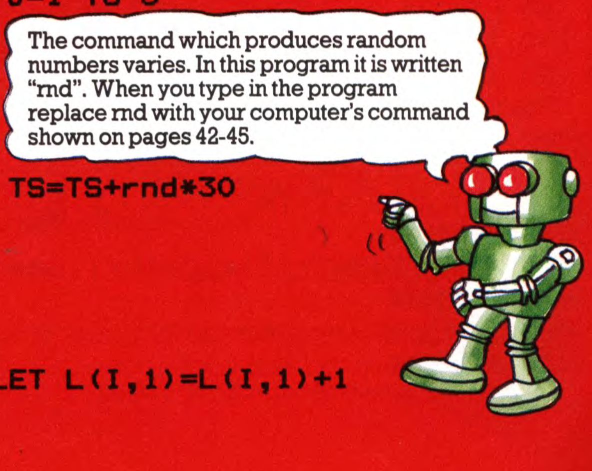
Ethel has tried both ways a few times, but almost every part of each route varies from day to day. Sometimes the bus is delayed, sometimes a train is cancelled.

To get a clear idea of which route is best she needs to compare the total times of lots of bus and train journeys. The only quick way to do this is to use the computer model.

Journey model program

This program models the two routes, using random numbers to cope with variations from day to day. For instance, on a particular day Ethel's wait for the bus can be between 0 and 7 minutes. So each time the computer simulates the bus journey it picks a random number between 0 and 7. It does a similar thing for the other stages of the journey and then adds the times together to give a total bus journey time. Obviously the time for one bus journey is not much use, but if you make the computer simulate lots of journeys, you can get a picture of bus journey times over several months and compare these with train journey times.

```
★10 CLS
 20 GOSUB 860:PRINT "JOURNEY MODEL INFORMATION"
 30 PRINT "(TIMES IN MINUTES)"
 40 PRINT:PRINT "BUS":PRINT
 50 PRINT "WALK TO BUS STOP":INPUT W1B
 60 PRINT "TIME BETWEEN BUSES":INPUT FB
 70 PRINT "SHORTEST BUS TIME":INPUT SB
 80 PRINT "LONGEST BUS TIME":INPUT LB:LET DB=LB-SB+1
 90 PRINT "PERCENTAGE OF BUSES CANCELLED":INPUT PB
100 PRINT "WALK FROM BUS STOP":INPUT W2B
110 PRINT:PRINT "TRAIN":PRINT
120 PRINT "WALK TO STATION":INPUT W3T
130 PRINT "TRAIN TIMES (HRS.MINS)":FOR I=1 TO 6
140 INPUT TT:GOSUB 840:LET T(I)=TT
150 NEXT I
160 PRINT "TIME TRAIN TAKES":INPUT TJ
170 PRINT "PERCENTAGE OF TRAINS ON TIME":INPUT PT
180 PRINT "PERCENTAGE OF TRAINS MORE THAN 5 MINUTES
    LATE":INPUT P5
190 PRINT "PERCENTAGE OF TRAINS CANCELLED":INPUT PC
200 PRINT "WALK FROM STATION":INPUT W4T
210 PRINT:PRINT
220 PRINT "TIME TO ARRIVE (HH.MM) ":INPUT A$
230 LET TT=VAL(A$):GOSUB 840:LET TA=TT
240 PRINT "TIME TO LEAVE (HH.MM) ":INPUT L$
250 LET TT=VAL(L$):GOSUB 840:LET TL=TT
260 PRINT "HOW MANY JOURNEYS":INPUT NJ
270 FOR I=1 TO 2:FOR J=1 TO 3
280 LET L(I,J)=0
290 NEXT J:NEXT I
300 FOR K=1 TO NJ
310 LET TS=TL
★320 LET OV=rnd*100
330 IF OV<2 THEN LET TS=TS+rnd*30
340 LET T=TS
350 GOSUB 610
360 LET T=TS
370 GOSUB 700
380 FOR I=1 TO 2
390 IF J(I)<=0 THEN LET L(I,1)=L(I,1)+1
```



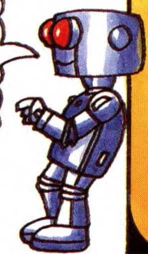
Listing continued over the page.

```
 400 IF J(I)>0 AND J(I)<=5 THEN LET L(I,2)=L(I,2)+1
 410 IF J(I)>5 THEN LET L(I,3)=L(I,3)+1
 420 NEXT I
 430 NEXT K
 440 PRINT:PRINT "RESULTS:"
 450 PRINT:PRINT "OUT OF ";NJ;" JOURNEYS"
 460 PRINT "LEAVING AT ";L$
 470 PRINT "TO ARRIVE AT ";A$
 480 PRINT
 490 FOR I=1 TO 2
 500 IF I=1 THEN PRINT "BY BUS"
 510 IF I=2 THEN PRINT "BY TRAIN"
 520 PRINT
 530 PRINT "ON TIME ";L(I,1)
 540 PRINT "LESS THAN 5 MINS LATE ";L(I,2)
 550 PRINT "MORE THAN 5 MINS LATE ";L(I,3)
 560 PRINT
 570 NEXT I
 580 PRINT "RUN THE MODEL AGAIN (Y/N)":INPUT A$
 590 IF A$="Y" THEN GOTO 210
 600 STOP
 610 LET T=T+W1B
★620 IF rnd*100<PB THEN LET T=T+FB
★630 LET WT=INT(rnd*FB)
 640 LET T=T+WT
★650 LET BJ=SB+INT(DB*(rnd+rnd)*0.5)
 660 LET T=T+BJ
 670 LET T=T+W2B
 680 LET J(1)=T-TA
 690 RETURN
 700 LET T=T+W3T
 710 LET I=0
 720 LET I=I+1:IF I=7 THEN PRINT "NO MORE TRAINS":STOP
 730 IF T(I)<T THEN GOTO 720
 740 LET TT=T(I)
★750 IF rnd*100<PC THEN LET TT=T(I+1)
★760 LET DL=0:LET R=rnd*100
★770 IF R>PT THEN LET DL=INT(rnd*5)
★780 IF R>(100-P5) THEN LET DL=INT(rnd*20+5)
 790 LET T=TT+DL
 800 LET T=T+TJ
 810 LET T=T+W4T
 820 LET J(2)=T-TA
 830 RETURN
 840 LET TT=INT(TT)*60+INT(100*(TT-INT(TT))+0.5)
 850 RETURN
 860 DIM L(2,3):DIM J(2):DIM T(6)
 870 RETURN
```

Using the program

When you run the program you need to type in the information about the two routes, as shown on the right. The information for Ethel's routes is given in the picture on page 14.

Over the page there are hints on using the program to model your own journeys.



1

```
JOURNEY MODEL INFORMATION
(TIMES IN MINUTES)

BUS
WALK TO BUS STOP
?3
TIME BETWEEN BUSES
?7
SHORTEST BUS TIME
?30
LONGEST BUS TIME
?50
PERCENTAGE OF BUSES CANCELLED
?10
WALK FROM BUS STOP
?10
```

2

BUS CANCELLATIONS = 1 in 10 or $\frac{1}{10}$

$\frac{1}{10} \times 100 = 10\%$

Multiply by 100 to convert to a percentage.

TRAIN CANCELLATIONS = 1 in 50, or $\frac{1}{50}$

$\frac{1}{50} \times 100 = 2\%$

The bus and train cancellations have to be given as percentages. You can see how to convert the figures to percentages above.

3

```
TRAIN TIMES (HRS.MINS)
?7.45
?8.15
?8.45
?9.15
?0
?0
```

You can type in up to six train times. Choose trains around the time you think Ethel needs to set out. If you do not want to enter six times, miss out the last ones by pressing 0, then RETURN.

4

```
ARRIVAL TIME(HRS.MINS)
?9.00
DEPARTURE TIME(HRS.MINS)
?8.05
```

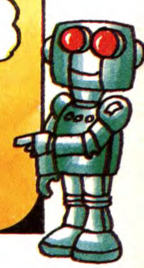
You also need to give the target arrival time and a departure time. Estimate the time Ethel should leave in order to get to work at 9 o'clock. You could try 8.00 or 8.05 a.m.

5

```
HOW MANY JOURNEYS
?100
```

10?.....100?.....1000?

Ten simulations are only a fortnight's travel – not enough to give an accurate picture. A thousand would take rather a long time to calculate.



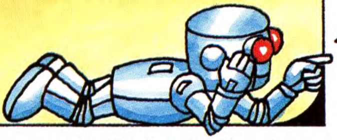
The computer must simulate enough journeys to give you a true picture of journey times. A hundred simulations are equivalent to nearly six months' travel and should give a reliable answer.

Looking at the results

Here are some sample results for Ethel's journey. Your results will probably not be exactly the same because the simulations vary every time the model is run.

1

```
RESULTS

OUT OF 100 JOURNEYS
LEAVING AT 8.05
TO ARRIVE AT 9.00

BY BUS
ON TIME 39
LESS THAN 5 MINS LATE 31
MORE THAN 5 MINS LATE 30

BY TRAIN
ON TIME 0
LESS THAN 5 MINS LATE 0
MORE THAN 5 MINS LATE 100
```



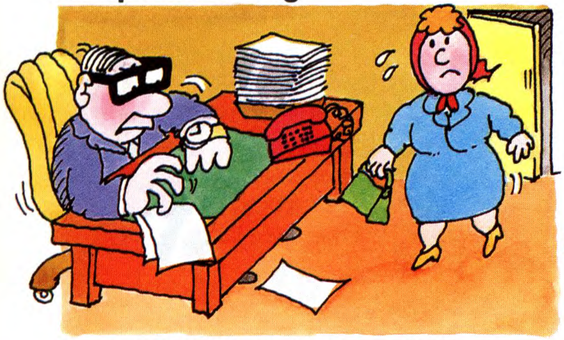
2

```
RESULTS

OUT OF 100 JOURNEYS
LEAVING AT 8.00
TO ARRIVE AT 9.00

BY BUS
ON TIME 63
LESS THAN 5 MINS LATE 25
MORE THAN 5 MINS LATE 12

BY TRAIN
ON TIME 93
LESS THAN 5 MINS LATE 2
MORE THAN 5 MINS LATE 5
```

By comparing the results for bus and train you can decide which is the best means of transport for the leaving time you gave.

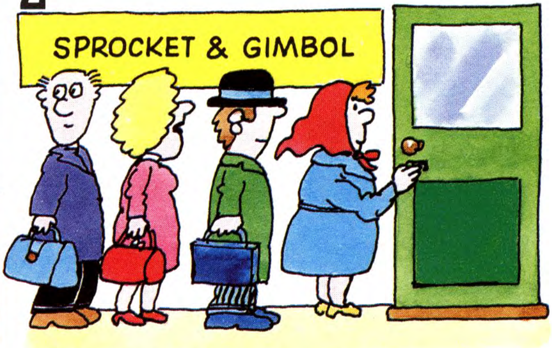
But is the same still true if you alter Ethel's leaving time? Run the model a few more times trying various departure times.

1 Experimenting



Assume Ethel's boss does not mind her being a bit late occasionally, but objects to very late arrivals or regular lateness. What time should Ethel leave? Should she travel by bus or train?

2

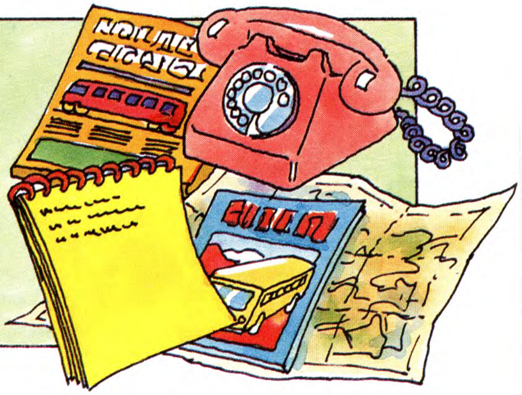


Ethel is put in charge of opening up the office and must never be late. Which means of transport should she use now? How much earlier does she need to leave?

Modelling your own journeys

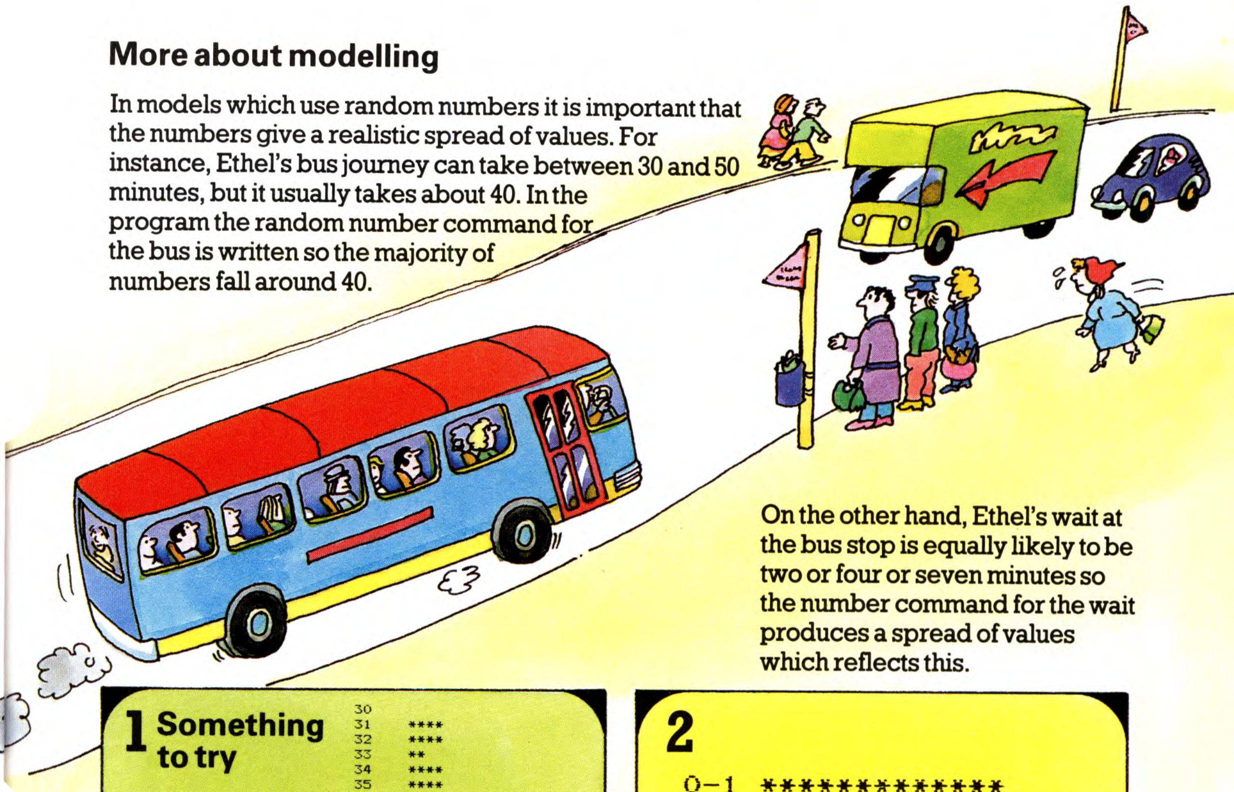
To find information for the model get hold of bus and train timetables and ring local transport services to find out about lengths of journeys and cancellations. Ask people at the station and bus stops too.

If you like, you can model just one route. To miss out the other route press RETURN each time the computer asks for information about it.



More about modelling

In models which use random numbers it is important that the numbers give a realistic spread of values. For instance, Ethel's bus journey can take between 30 and 50 minutes, but it usually takes about 40. In the program the random number command for the bus is written so the majority of numbers fall around 40.



On the other hand, Ethel's wait at the bus stop is equally likely to be two or four or seven minutes so the number command for the wait produces a spread of values which reflects this.

1 Something to try

```
30
31   ****
32   ****
33   **
34   ****
35   ****
36   ****
37   *****
38   *******
39   ******
40   ************
41   ***************
42   ******
43   *****
44   *****
45   ******
46   ******
47   ***
48   *
49   *
50
```

The program below generates 100 random numbers for Ethel's bus journey. It displays them on the screen as stars so you can see the spread of values. Try running the program a few times and see what pattern you get.

2

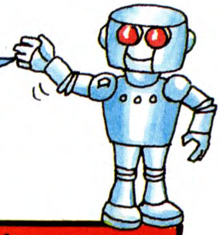
```
0-1 *************
1-2 ****************
2-3 ***************
3-4 ***********
4-5 *******************
5-6 ***************
6-7 ***********
```

If you replace lines 10, 20, 50, 60 and 120 with those in the box, the program generates Ethel's waits at the bus stop.

Compare the spread of values you get with the one for the bus journey times.

Bus journey program

```
 10 LET SB=30:LET DB=21
 20 LET L=21:LET B=29
 30 DIM F(L)
 40 FOR N=1 TO 100
★50 LET BJ=SB+INT(DB*(rnd+rnd)*0.5)
 60 LET P=BJ+1-SB
 70 LET F(P)=F(P)+1
 80 NEXT N
★90 CLS
100 PRINT:PRINT
110 FOR N=1 TO L
★120 PRINT TAB(0);B+N;TAB(5);" ";
130 IF F(N)=0 THEN GOTO 150
★140 FOR J=1 TO F(N):PRINT "*";:NEXT J
150 PRINT:NEXT N
```

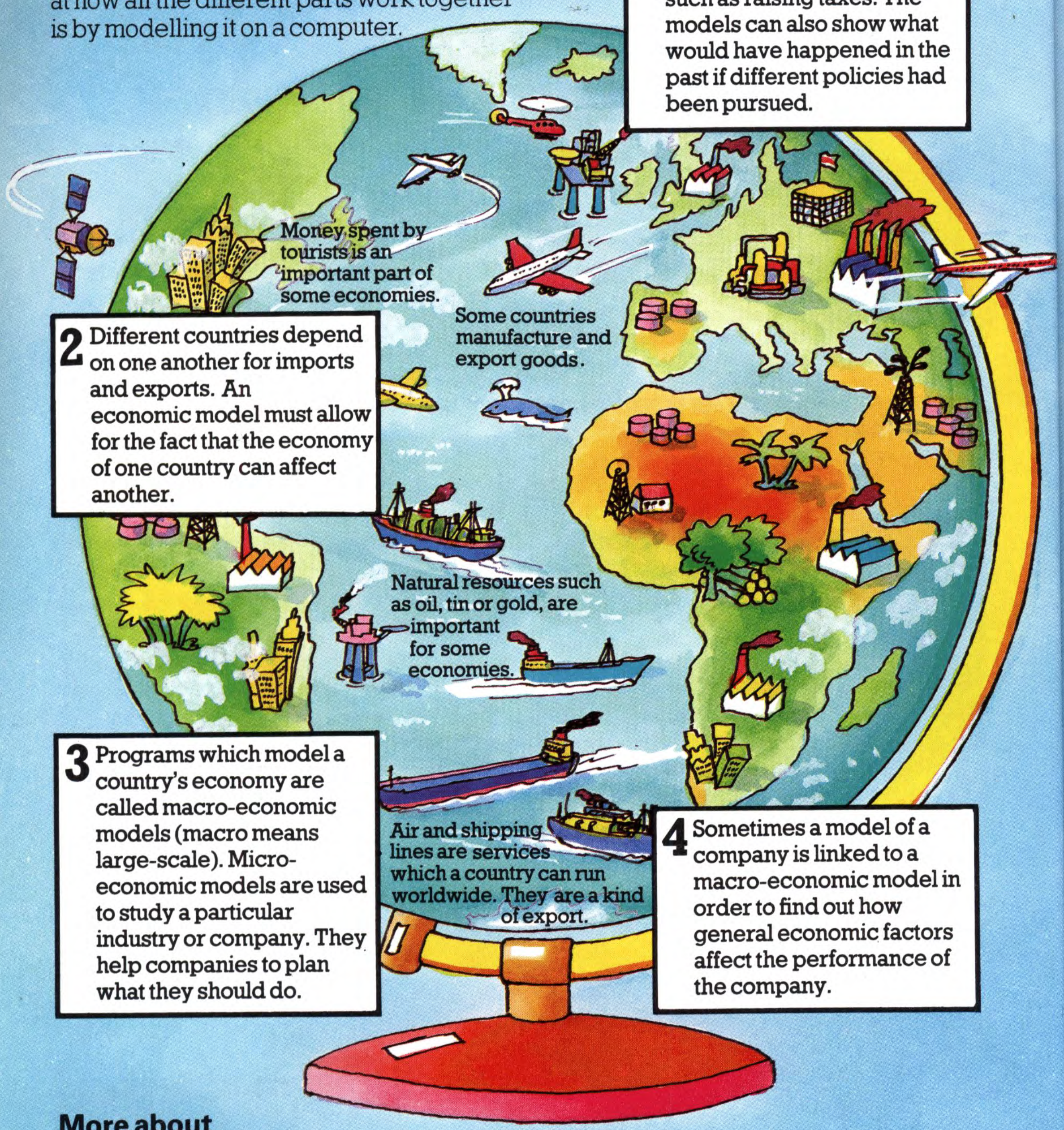


Changes for wait

```
 10 LET FB=7
 20 LET L=7:LET B=-1
★50 LET WT=INT(rnd*FB)
 60 LET P=WT+1
★120 PRINT TAB(0);B+N;
      "-";B+N+1;" ";
```

About economic models

Economics is the study of how wealth is produced and distributed in a country. The economy of a country is incredibly complex. The only way economists can look at how all the different parts work together is by modelling it on a computer.



1 Governments use economic models to forecast inflation or employment, and to predict the effect of policies such as raising taxes. The models can also show what would have happened in the past if different policies had been pursued.

2 Different countries depend on one another for imports and exports. An economic model must allow for the fact that the economy of one country can affect another.

3 Programs which model a country's economy are called macro-economic models (macro means large-scale). Micro-economic models are used to study a particular industry or company. They help companies to plan what they should do.

4 Sometimes a model of a company is linked to a macro-economic model in order to find out how general economic factors affect the performance of the company.

More about economic models

Economic models are based on theories. No-one understands exactly how the economy works and many economic events have never really been explained.

Different political groups use different models and often disagree about what the effects of a particular policy will be.

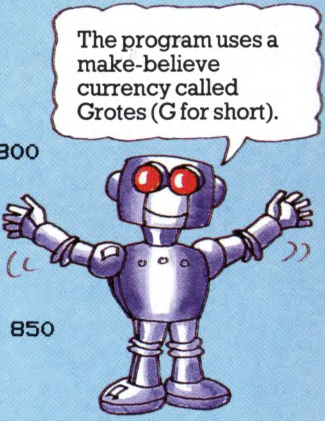
Running an airline

The program below is a very simple economic model of an airline company. You can use it to experiment with different business strategies and see which works best.

```
10 LET G$=" G.":LET E$="TOO MANY"
20 DIM A(5)
30 LET AC=50000
40 LET RC=10000
50 LET M=500000
60 LET A=0
70 LET YR=1
80 LET TC=M:LET VA=0
90 GOSUB 880
100 LET P$="MACRO-ECONOMIC PREDICTIONS FOR NEXT 5 YEARS**"
110 GOSUB 800
120 PRINT "INFLATION RATE (%)"
130 GOSUB 860:LET FR=RT
140 PRINT "INTEREST RATE (%)"
150 GOSUB 860:LET TR=RT
160 LET P$="EXCHANGE RATE**1. RISING*2. STEADY*3. FALLING*"
170 GOSUB 800
180 INPUT ER:IF ER<1 OR ER>3 THEN GOTO 180
190 LET ER=0.8+ER/10
200 GOSUB 880
210 LET P$="POLICY FOR YEAR AHEAD**":GOSUB 800
220 PRINT "YEAR ";YR
230 LET X=0:LET P$="*YOU WILL START THE YEAR WITH:**":GOSUB 800
240 GOSUB 740
250 LET P$="*HOW MANY AIRCRAFT DO YOU WANT TO BUY*":GOSUB 800
260 INPUT NA:IF NA<0 OR NA*AC>M THEN PRINT E$:GOTO 260
270 LET A(YR)=NA*AC:LET CS=NA*AC
280 LET P$="*HOW MANY CREW DO YOU WANT TO EMPLOY*":GOSUB 800
290 INPUT NC:IF NC<0 OR NC*RC+CS>M THEN PRINT E$:GOTO 290
300 LET CS=CS+RC*NC
310 LET P$="*HOW MUCH DO YOU WANT TO SPEND ON ADVERTISING*"
320 GOSUB 800
330 INPUT MV:IF MV<0 OR MV+CS>M THEN PRINT E$:GOTO 330
340 LET CS=CS+MV
350 GOSUB 880
360 LET P$="RESULTS FOR YEAR "+STR$(YR)+"**":GOSUB 800
370 LET X=TC:GOSUB 900:LET SC=TC
380 LET P$="START OF YEAR"
390 GOSUB 730:LET A=A+NA
400 LET BC=1000+2000*A
410 LET CS=CS+BC
420 LET X=CS:GOSUB 900
430 LET P$="*CASH SPENT    = "+STR$(X)+G$+"*":GOSUB 800
440 LET M=M-CS:IF M<0 THEN PRINT "YOU ARE BANKRUPT!":STOP
450 LET F=NC/2:IF A<F THEN LET F=A
460 LET LF=1.4+(MV/(RC*F*2+1))*3
470 IF LF>2 THEN LET LF=2
480 LET CR=RC*F*LF*ER*2
490 LET X=CR:GOSUB 900
500 LET P$="CASH RECEIVED= "+STR$(X)+G$+"*":GOSUB 800
510 LET IN=INT(M*TR)
520 LET X=IN:GOSUB 900
530 LET P$="INTEREST ON CASH INVESTED= "+STR$(X)+G$+"**":GOSUB 800
540 LET M=M+IN+CR
550 LET VA=0
560 FOR I=1 TO YR
570 LET A(I)=A(I)*0.8
580 LET VA=VA+A(I)
```

Listing continued over the page.

```
590 NEXT I
600 LET AC=AC*(1+FR):LET RC=RC+(1+FR)
610 LET TC=VA+M
620 LET X=TC:GOSUB 900:LET P$="END OF YEAR":GOSUB 730
630 LET DC=TC-SC:LET P$="PROFIT":IF DC<0 THEN LET P$="LOSS"
640 LET X=ABS(DC):GOSUB 900:LET P$="*"+P$+" = "+STR$(X)+G$+"*":GOSUB 800
650 LET Q$="START":LET R$="END"
660 IF DC<0 THEN LET Q$="END":LET R$="START"
670 LET P$="(CAPITAL AT "+R$+" OF YEAR*- CAPITAL AT "+Q$+" OF YEAR)"
680 GOSUB 800
690 PRINT:PRINT:PRINT "PRESS RETURN"
700 INPUT X$
710 LET YR=YR+1:IF YR<6 THEN GOTO 200
720 PRINT:PRINT:STOP
730 LET P$="CAPITAL AT "+P$+":*":GOSUB 800
740 LET X=M:GOSUB 900:PRINT "";X;G$;" CASH"
750 LET X=VA:GOSUB 900
760 LET P$="AIRCRAFT WORTH "+STR$(X)+G$+"*":GOSUB 800
770 LET X=VA+M:GOSUB 900
780 LET P$="TOTAL = "+STR$(X)+G$+"*"
790 GOSUB 800:RETURN
800 FOR L=1 TO LEN(P$)
★810 LET M$=MID$(P$,L,1)
★820 LET PS=POS(0)-28
830 IF (M$=" " AND PS>0) OR M$="*" THEN PRINT:GOTO 850
840 PRINT M$;
850 NEXT L:RETURN
860 INPUT RT:IF RT<0 OR RT>50 THEN GOTO 860
870 LET RT=RT/100:RETURN
★880 CLS
890 PRINT:PRINT:RETURN
900 LET X=INT(X/1000+0.5)*1000
910 RETURN
```



About the program

You are in control of a small airline and are planning your policy for the next five years. First you give the computer information about general macro-economic conditions which affect your business. These are the predicted inflation, interest and exchange rates. You can find out more about them below. Then you decide your strategies for each year.

Inflation rate

The inflation rate (the rate at which prices increase) affects the cost of buying new aircraft, paying the crew and other running costs.

Interest rate

The interest rate determines how much you earn on money you leave in the bank.

Exchange rate

The exchange rate indicates how much foreign currency you get in exchange for your money. It affects the number of tickets you sell abroad. A rising exchange rate means you sell fewer tickets because they are more expensive.

Running the program

1 Macro-economic predictions

Type in the predicted inflation, interest and exchange rates for the year ahead. You can invent the information or use real figures from newspapers.

```
MACRO-ECONOMIC PREDICTIONS FOR
NEXT 5 YEARS

YEARLY INFLATION RATE (%)
?8
YEARLY INTEREST RATE (%)
?10
EXCHANGE RATE
1 RISING
2 STEADY
3 FALLING
?1
```

2 Deciding policies

```
POLICY FOR YEAR AHEAD

YEAR 1
YOU WILL START THE YEAR WITH
500000 G. CASH
AIRCRAFT WORTH 0 G.
TOTAL = 500000 G.

HOW MANY AIRCRAFT DO YOU WANT
TO BUY
?3
HOW MANY CREW DO YOU WANT TO EMPLOY
?6
HOW MUCH DO YOU WANT TO SPEND
ON ADVERTISING
?50000
```

These are the decisions you have to make each year. On the right there is information about costs and other factors which affect your decisions.

1. At the start of the five years an aircraft costs 50,000 G. and the price goes up each year by the rate of inflation.

2. You need two crew for each aircraft. Crew and other running costs for one aircraft for a year are 10,000 G at the start of the five years. These costs also increase every year by the rate of inflation.

3. There is an automatic yearly cost of 1,000 G, plus 2,000 G for each aircraft, which you pay whether or not you run the aircraft.

4. You make interest at the predicted rate on any money you do not spend.

3 Results

Each year the computer works out your capital (that is, your money plus the value of your aircraft) at the start and end of the year. It also calculates how much profit (or loss) you have made.

Run the program a few times keeping inflation, interest and exchange rates the same whilst experimenting with different policies. Then keep your policies the same and see what happens if you alter the inflation, or exchange rate.

```
RESULTS FOR YEAR 1

CAPITAL AT START OF YEAR
500000 G. CASH
AIRCRAFT WORTH 0 G.
TOTAL = 500000 G.

CASH SPENT = 267000 G.
CASH RECEIVED =108000 G.
INTEREST ON CASH INVESTED = 23000 G.

CAPITAL AT END OF YEAR
364000 G. CASH
AIRCRAFT WORTH 120000 G.
TOTAL = 484000 G.

LOSS = 16000 G.
(CAPITAL AT START OF YEAR
- CAPITAL AT END OF YEAR)
```

This is the money you spent on aircraft and crew.

This is the money you made by selling tickets.

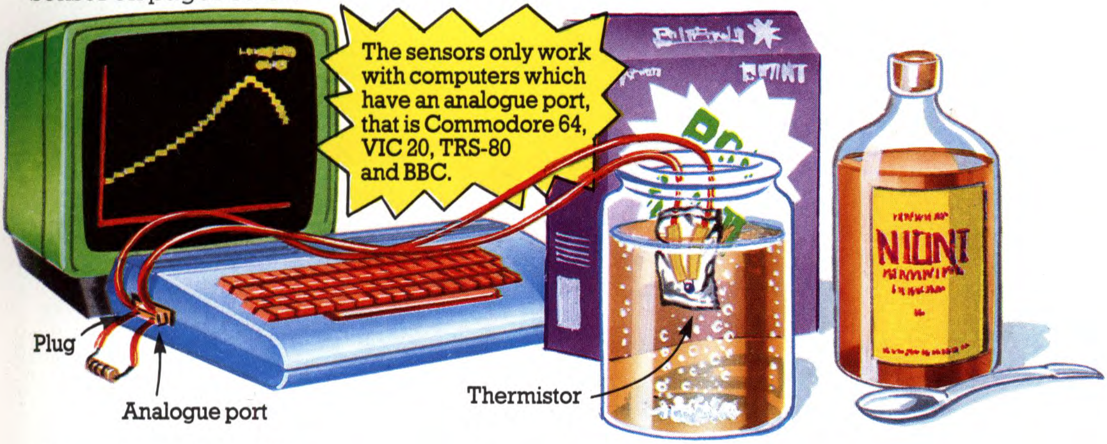
This is the interest on the money you did not spend.

Aircraft lose one fifth of their value each year.

Experiments with sensors

On the next few pages you can find out how to make sensors which enable your computer to display graphs of temperature or light changes. You can use the sensors in lots of experiments, such as finding out what happens to your body temperature during exercise, or when you are asleep or ill.

Below there are instructions for making the temperature sensor using an electronic component called a thermistor. You can find out how to make the light sensor on pages 30-31.

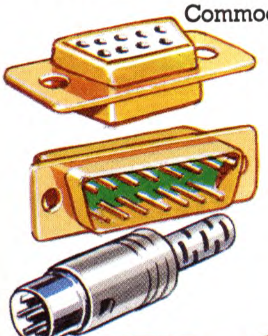


A thermistor allows different amounts of electric current to flow through it according to how hot it is. It converts temperature measurements into electric signals which the computer can recognize. You program the computer to translate the electric signals into temperature readings.

Things you need

This is the equipment you need to make the temperature sensor. You can buy thermistors, resistors and other equipment at electonics components shops or send off for them to an electronics supplier. Look in a hobby electronics magazine to find the address of a supplier.

A plug to fit the analogue port of your computer:



Commodore 64 and VIC 20
A D-type 9-way female (socket).

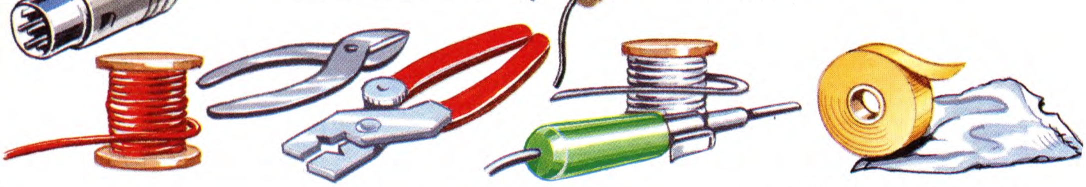
BBC
A D-type 15-way male (plug).

TRS-80
A 6-way DIN plug.

A type VA1067S thermistor. (If you cannot get this one, get any 100K thermistor.)

For the Commodore 64, VIC 20 and BBC – a 1M ohm (Ω), 0.25 or 0.5 watt resistor. This has brown, black, green stripes.

For the TRS-80 – a 100K ohm, 0.25 or 0.5 watt resistor. This has brown, black, yellow stripes.



Insulated, stranded copper wire, e.g. "bell wire".

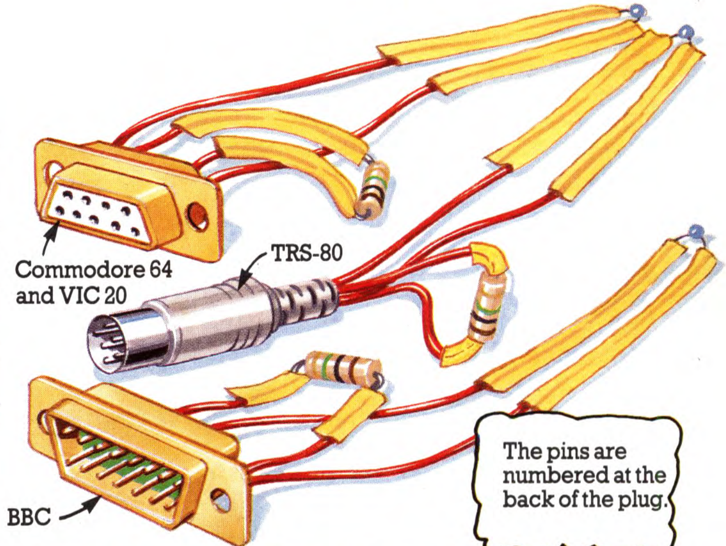
Scissors or wire cutters and strippers.

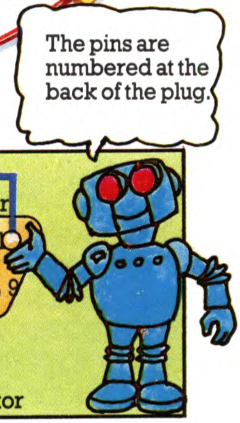
A soldering iron and some cored solder.

Insulating tape and a small polythene bag or some clingfilm.

Making the sensor

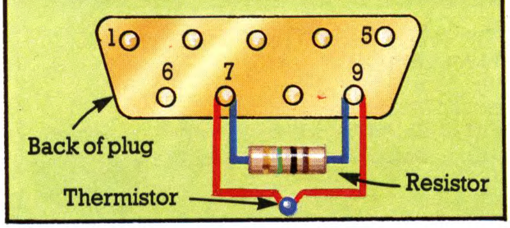
In the picture on the right you can see what your sensor will look like when it is made. You need to solder wires to pins at the back of the plug and then fit the components to the wires. There are hints on soldering on page 29. The diagrams below show which pins the components should be soldered to. Make sure you follow the right diagram for your computer.



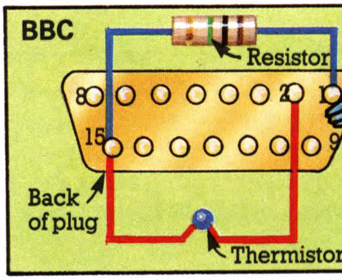


Wiring diagrams

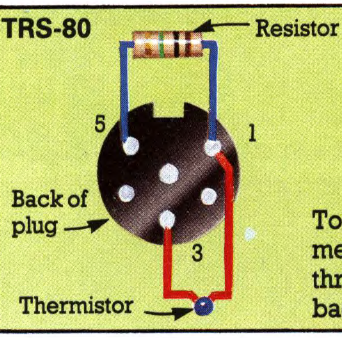
VIC 20 and Commodore 64

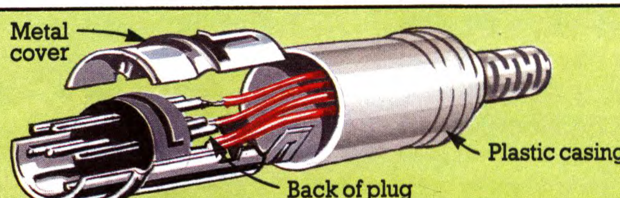


BBC



TRS-80





To wire the DIN plug take off the plastic casing and the metal cover underneath. Before soldering wires to the pins thread them through the plastic casing so you can put the plug back together when you have made the sensor.

Hints

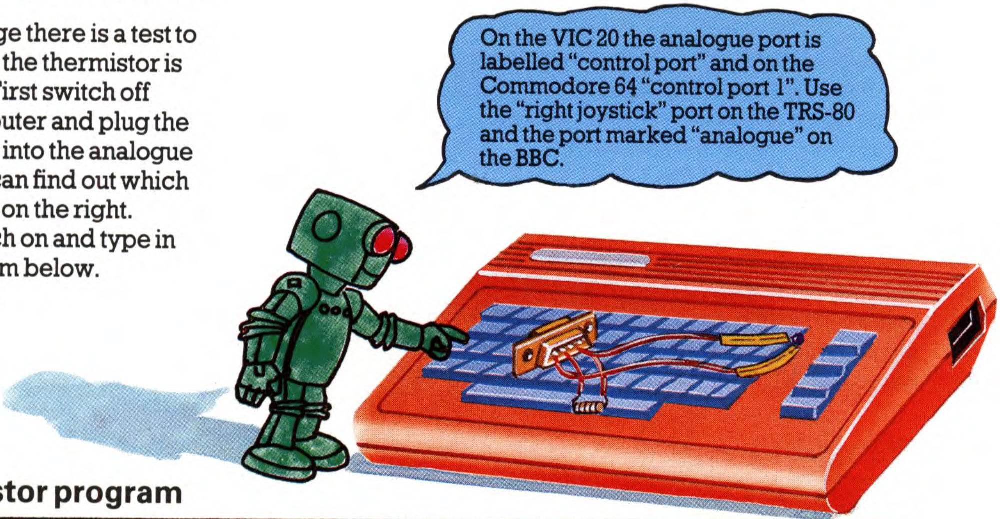
1. Use long wires (about 60cm) to connect the thermistor to the plug. This is so it will reach from the computer to an experiment. You can use much shorter wires for the resistor.

2. First solder the wires to the pins at the back of the plug, then solder the resistor and thermistor to the wires. Check carefully that you have connected each wire to the correct pin on the plug.

3. When you have made the sensor, wrap insulating tape round all the joints and bare wires. Otherwise wires may touch one another and cause a short circuit which may damage your computer.

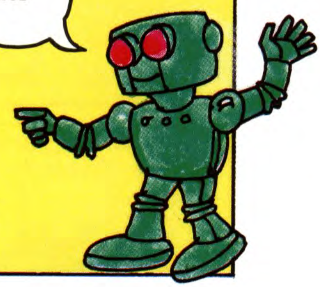
Testing the thermistor

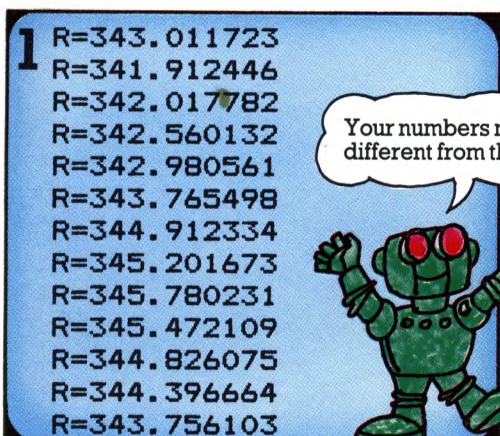
On this page there is a test to check that the thermistor is working. First switch off your computer and plug the thermistor into the analogue port. You can find out which port to use on the right. Then switch on and type in the program below.

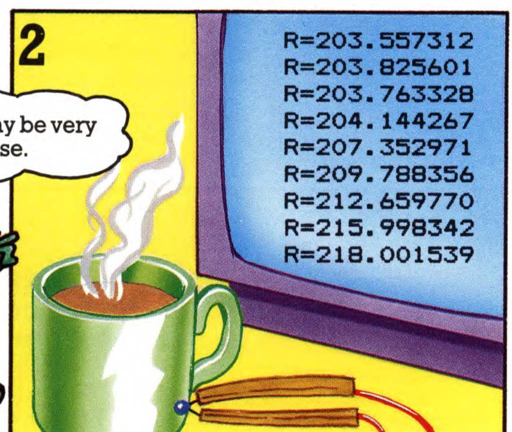


Thermistor program

```
  10 GOSUB 500
  20 GOSUB 1000
  30 PRINT "R = ";R
  40 GOTO 20
  500 LET K1=4000
  510 LET K2=1
  520 RETURN
★1000 LET TR=1.84*PEEK(54297)
★1010 LET R=1/(1/TR-1/1000)
★1020 LET T=1/((LOG(R/K2))/K1+1/273)-273
  1030 RETURN
```





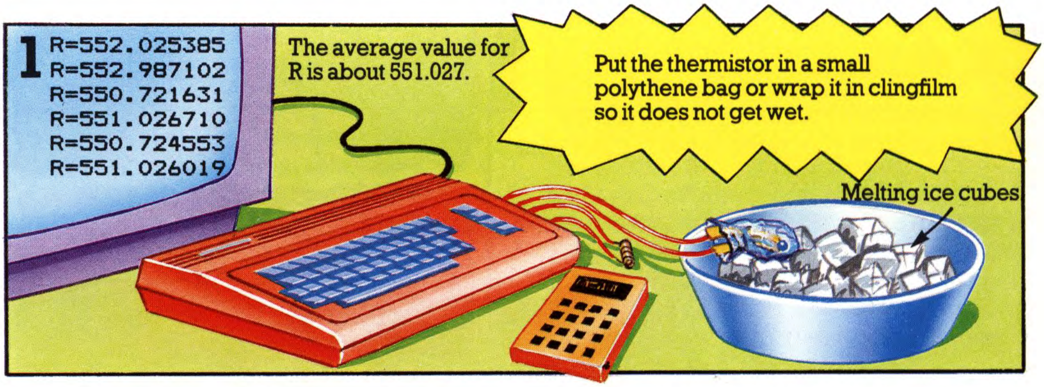


When you run the program a stream of large numbers appears on the screen. These are produced by the signals from the thermistor and are called R.

To check the signals are actually coming from the thermistor hold it against something hot. The values for R should go down, and then gradually up again when you take the thermistor away from the heat. If this does not happen, the sensor is not working. Check that all the connections are correct and firmly made, and no bare wires are touching one another.

Calibrating the thermistor

Before you can use the thermistor in experiments you need to translate the R values into temperatures. This is called calibrating the thermistor. Below you can find out how to calibrate the thermistor using the readings it gives at 0°C and at 80°C.



To find out what reading the thermistor gives at 0°C put it into a dish of melting ice. Make sure it is completely surrounded by the ice. The R values on the screen go up. After about ten minutes they should reach a steady level (though they will always fluctuate a bit). Press ESCAPE to stop the program.* From the numbers displayed on the screen pick out the most common value for R and round it off to 3 decimal places.

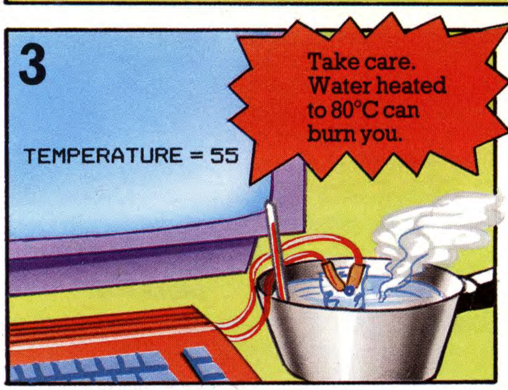
2

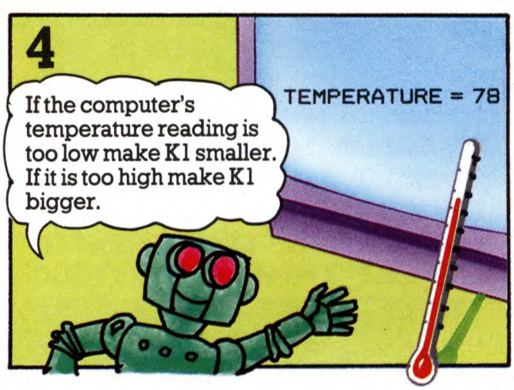
```
510 LET K2=
```

Type your value for R here.

```
30 PRINT "TEMPERATURE = ";INT (T)
```

List the program and retype line 510 making K2 equal your value for R. K2 is the bottom of the temperature scale. Then replace line 30 with the version given above. This makes the computer display temperature readings instead of R.





To find the thermistor reading at the top of the temperature scale put it in water heated to about 80°C. You can use a cooking thermometer to measure the water temperature. You need to match the computer's temperature reading to the thermometer's by altering the value of K1 in line 500 of the program.

To alter K1 stop the program and list it on the screen. Then experiment, making K1 bigger or smaller by two or three hundred and rerunning the program to check the computer's new temperature reading. Keep altering K1 until the computer's reading matches the thermometer's.

*The ESCAPE key may be marked differently on your computer. See page 6.

Doing experiments

Here are some ideas for experiments to do using the thermistor.

You can make the computer display a graph as well as temperature readings by adding the program lines at the bottom of the page and the graphics routine on pages 46-47.

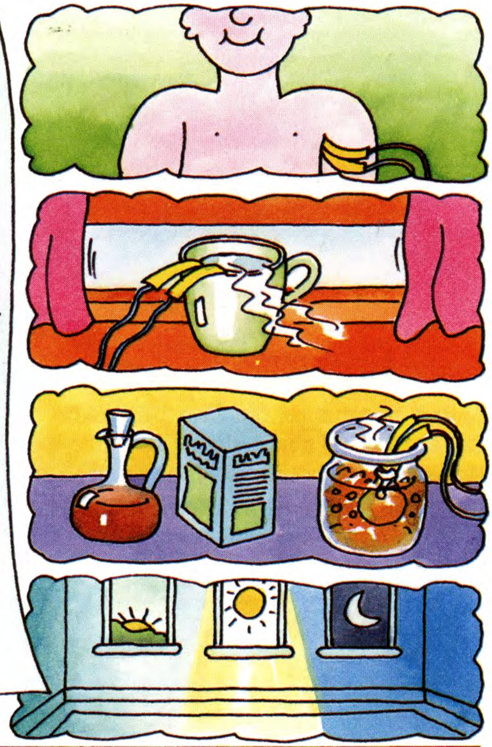
Ideas for things to do

1. Measure your body temperature. Grasp the thermistor tightly in your palm or put it under your armpit.

2. Put the thermistor into a cup of hot water placed in a cool draught and watch the shape of the cooling curve drawn by the computer. Remember to put the thermistor in a polythene bag so it does not get wet.

3. Use the thermistor to measure temperature changes in a chemistry experiment. For instance, mix about a tablespoonful of vinegar with four or five lumps of washing soda and measure the temperature change when the acid and salt react.

4. Plot a graph of variations in the temperature of a room over a whole day, or night. To do this the computer must draw the graph very slowly. You can find out how to alter the program to make it do so below.



Graph lines

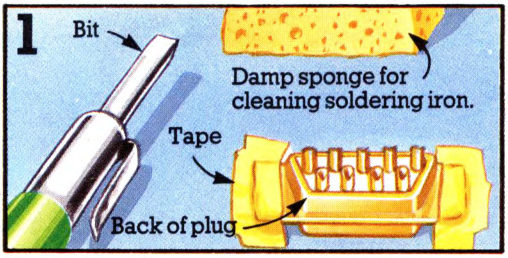
```
★30 LET FT=10:LET DX=10
 40 LET T$="TEMPERATURE":LET Y$="100":LET X$="TIME"
 50 LET M$="50":GOSUB 2000
 60 LET X=0
 70 LET SM=0:FOR J=1 TO 5
 80 GOSUB 1000:LET SM=SM+T
 90 NEXT J:LET T=SM/5
 100 LET Y=T*10:LET N=INT(T):GOSUB 2200
 110 FOR L=1 TO FT:NEXT L
 120 LET X=X+DX:IF X<1000 THEN GOTO 70
 130 LET M$="PRESS A KEY":GOSUB 2400
★140 GET I$
 150 IF I$="" THEN GOTO 140
 160 GOTO 40
```

The graph shows how the temperature varies over time.* The computer completes the graph in about a minute. To make it draw a new one press RETURN.

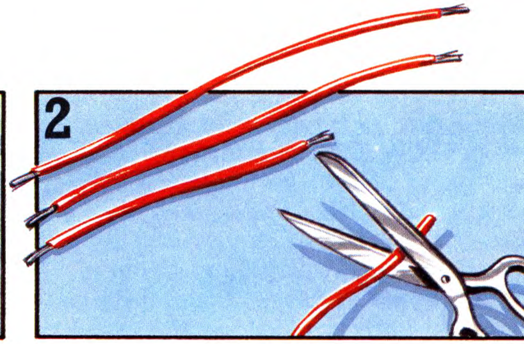
For some experiments you need the computer to draw the graph more slowly, so it shows temperature variation over, say, ten minutes or 12 hours. To make it do this list the program and increase the value of FT in line 30.

*On the TRS-80 the graph only shows temperature changes of 3°C or more.

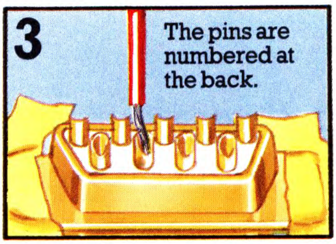
Hints on soldering



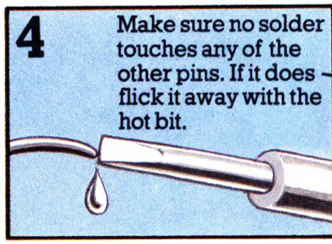
Plug in the soldering iron and let it heat up. Take care to prop it up carefully as the end part, called the bit, gets very hot. Tape the plug onto a flat surface ready for soldering.



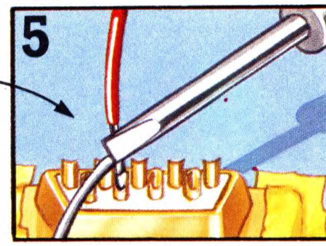
Cut two 60cm wires for the thermistor, and two shorter wires for the resistor. At the ends strip off about 1cm of the covering, then twist together the strands of bare wire and "tin" them (see below).



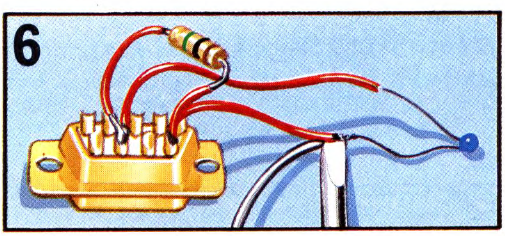
To solder a wire to a pin hold it in or against the back of the pin.



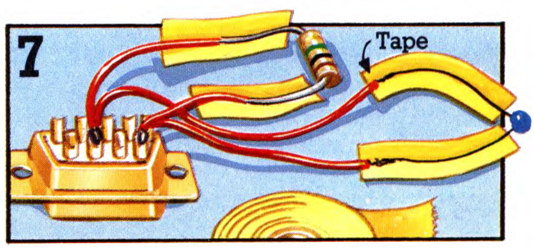
Touch the solder with the hot soldering iron so a drop of melted solder clings to the bit.



Then put the bit of the iron and the solder onto the joint and leave them until a blob of solder joins the wire to the pin.



To fit a component (e.g. the thermistor) to the wires, twist the leg of the component and the wire together as shown above. Then solder them.

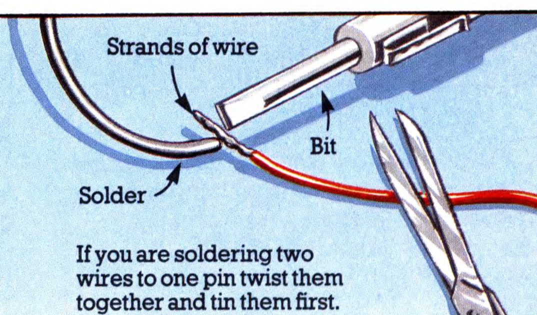


When you have made the connections, cover all the bare wires with insulating tape to prevent short circuits. The easiest way is to fold the tape lengthways over the wires.

How to tin wire

Tinning means coating wire with a thin layer of solder to hold the strands together. It makes the wire easier to solder.

To tin a wire twist the strands together. Then stroke them with the solder and the hot bit of the soldering iron.

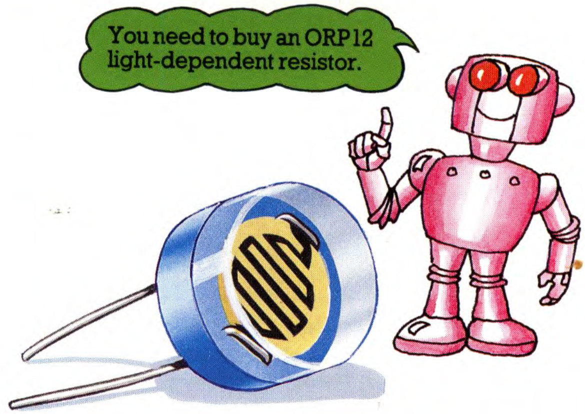


If you are soldering two wires to one pin twist them together and tin them first.

Making a light sensor

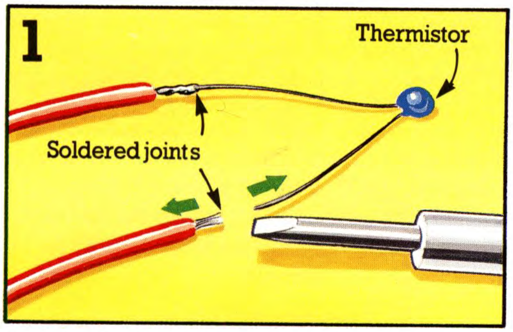
You can make the temperature sensor into a light sensor if you replace the thermistor with an LDR. LDR is short for light-dependent resistor. It is an electronic component which allows different amounts of electric current to flow through it according to how much light is falling on it.

The program for the light sensor is given opposite. You also need to add the graphics routine on pages 46-47.

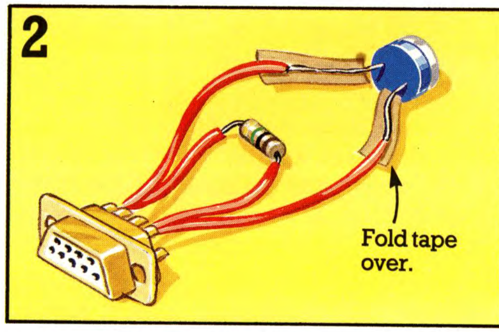


How to make the sensor

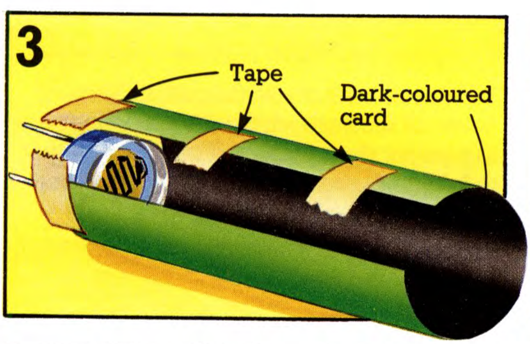
Below you can find out how to remove the thermistor, and solder the LDR in its place. If you want to build a separate light sensor buy another plug and resistor, and make the light sensor following the instructions on pages 24-25 for the temperature sensor.



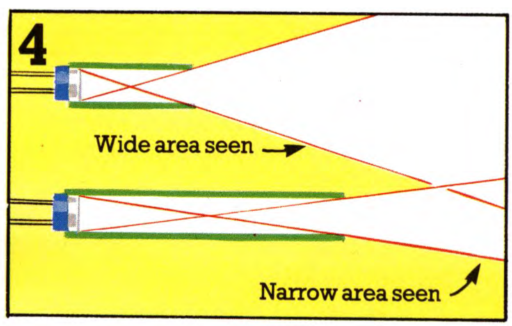
To take the thermistor off hold the hot soldering iron against the joint between the thermistor and the wire. As the solder melts pull the thermistor leg away.



Solder the LDR in place of the thermistor and wrap insulating tape round the new joints.



An LDR picks up light from all around. The light sensor is more useful if it only picks up light from the direction in which you point it. To make it do this tape a tube of dark-coloured card round the LDR.

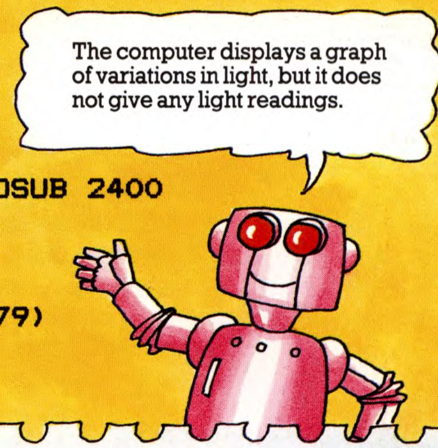


The length of the tube determines how large an area the light sensor "sees". A longer tube makes the area smaller so the light sensor is more sensitive.

Light sensor program

To use the light sensor type in the program lines below and add the graphics routine for your computer (see pages 46-47).

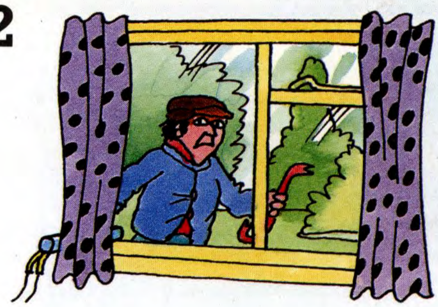
```
★10 LET FT=10:LET DX=10
 20 LET T$="LIGHT LEVEL":LET Y$="MAX"
 30 LET X$="TIME":LET M$="":GOSUB 2000
 40 LET X=0:LET N=0
 50 GOSUB 140
 60 LET X=X+DX:LET Y=LL
 70 GOSUB 2200
 80 FOR T=1 TO FT:NEXT T
 90 IF X<1000 THEN GOTO 50
100 LET M$="PRESS A KEY":GOSUB 2400
★110 GET I$
120 IF I$="" THEN GOTO 110
130 GOTO 20
★140 LET LL=1000-4*PEEK(54279)
150 RETURN
```



1 Things to do

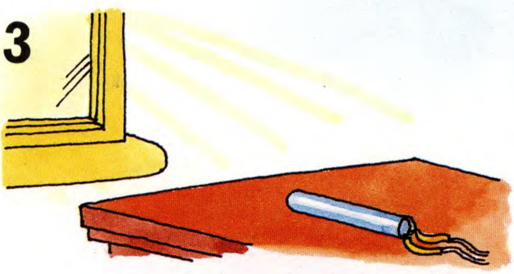
Point the sensor at different light sources and use the graph to compare how bright they are.

2



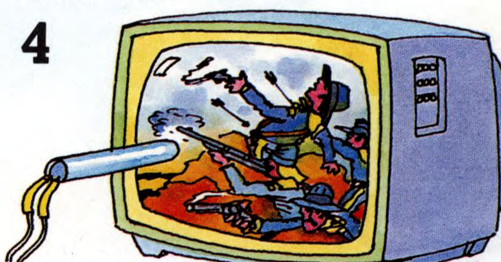
If you point the sensor at a window the graph will show when someone walks past because they will block out the light for a moment.

3



Use the sensor to record light changes in a room over a day in summer and in winter. You need to make the computer draw the graph very slowly by increasing the value of FT in line 10 of the program.

4



Move the sensor over a TV screen and see how much variation you get in the graph.

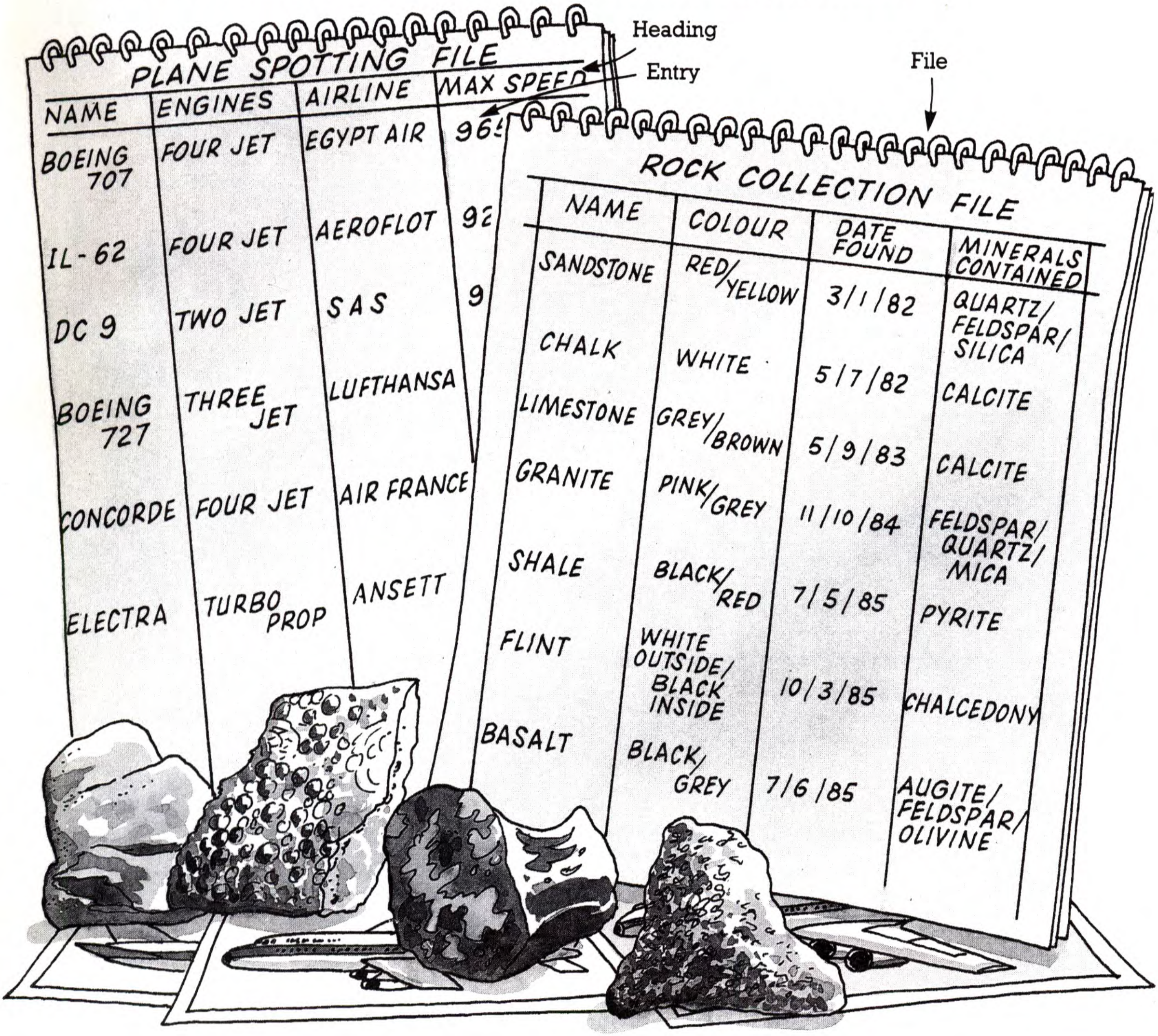
Storing information

In this part of the book there are programs to make your computer store and analyse information you collect from experiments and surveys.

The program described on the next few pages is called a database. It allows you to store details of a survey or the results of experiments. You can also use it to keep a record of a collection or hobby, such as plane spotting. The program listing is given on pages 35-36. Below there are some hints on how a database works.

About the database

The database program stores information in files. A file is like a page of a notepad. It is divided into four "columns" with different information stored in each column.



You give each of the columns a heading. Then you can make "entries" by giving the computer information to store in each column.

Once the computer has a file with several entries it can search through picking out particular pieces of information, such as rocks containing the mineral calcite or aircraft with four jet engines.

Using the database

Here are some hints on running the database program. When you have made a file, save it on tape so you do not lose it when you switch off the computer or rerun the program. You can make as many different files as you like, though you can only load and use one at a time.

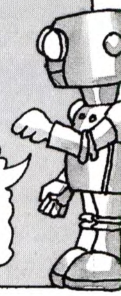
1 The menu

```
MAIN MENU

1. LOAD FILE
2. MAKE NEW FILE
3. MAKE ENTRIES
4. VIEW ENTRIES
5. ALTER ENTRY
6  SAVE FILE
7  SEARCH FILE

SELECT CHOICE
?
```

When you run the program the computer displays a "menu" showing the tasks it can do. You choose a task by typing its number and pressing RETURN. When you are using the program you can go back to the menu by pressing RETURN.

2 Loading a file

```
LOAD FILE

FILE NAME?ROCKS

PRESS PLAY ON TAPE
```



If you want to look at a file you have already made and stored on tape, choose option 1 to load it into the computer. You need to give the computer the name of the file, then press PLAY on the tape recorder.

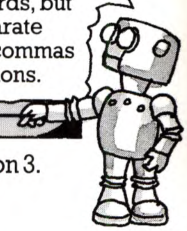
3 Making a new file

```
MAKE NEW FILE

TYPE IN HEADINGS

HEADING 1
?NAME
HEADING 2
?COLOUR
```

Option 2 is for making a new file. First you give the computer the four column headings. Each one must be no longer than ten letters or spaces, so you may need to abbreviate them.

4 Making entries

```
MAKE ENTRY

NOW ON ENTRY 2

NAME
?SANDSTONE
COLOUR
?RED/YELLOW
```



To make entries in a file choose option 3. The computer displays the column headings one by one and asks for information to put in them.

5 Viewing entries

```
VIEW ENTRIES

NOW ON ENTRY 2

NAME        CHALK
COLOUR      WHITE
DATEFOUND   5/7/82
MINERALS    CALCITE
```

Option 4 allows you to look through the entries in the file one by one. You press the space bar to move onto the next entry.

6 Altering entries

```
ALTER ENTRY

NOW ON ENTRY 2

NAME
?CHALK
COLOUR
?WHITE/YELLOW
```

You can alter the entry you are viewing by pressing RETURN to get back to the main menu, then choosing option 5. This allows you to retype the whole entry.

7 Saving the file

```
SAVE FILE

FILE NAME?ROCKS

PRESS RECORD
AND PLAY ON TAPE
```

This message may be different or may not appear on your computer (though you still have to do it).

It is a good idea to save a file as soon as you have made it. Otherwise you may forget and lose it by rerunning the program or switching off the computer. To save a file you type in a name for it, then press RECORD and PLAY on the tape recorder. Write filenames on the cassette label in case you forget them.

8 Making a search

1. Option 7 makes the computer search through the files for information. On the right you can find out how to tell the computer what information to look for.

2. You also need to tell the computer what to display when it finds the information. For instance, if it is searching for three jet engines, you must tell it whether to display the names of aircraft, the airlines flying them or their maximum speed – or all three.

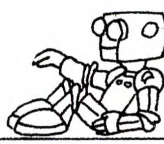
```
SEARCH FILE

1  NAME
2  ENGINES
3  AIRLINE
4  MAX SPEED

SUBJECT OF SEARCH
?2
ENGINES?TWO JET

INFO REQUIRED
?1
?3
?
```

First type the number of the column the information is in. Then type what you want the computer to look for.

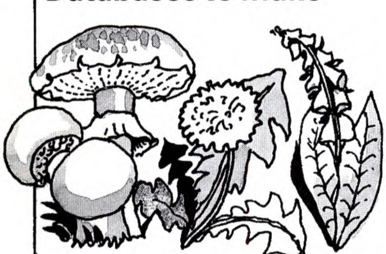
Press RETURN to miss out a column.

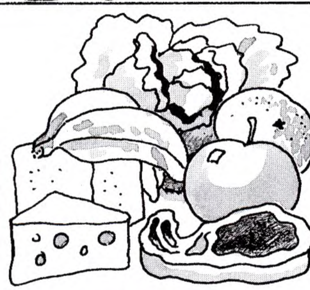
3. When the computer has searched the files it displays the information entry by entry. You need to press the space bar to move on to the next entry.

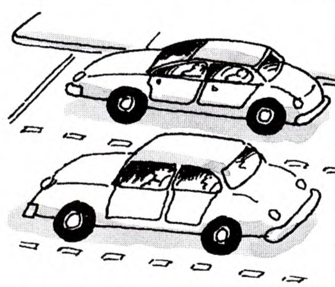
```
SEARCHING FOR:
ENGINE     TWO JET

ENTRY 3    DC 9
           SAS

PRESS SPACE

SEARCH OVER
1 ENTRY FOUND
```



Databases to make







If you are interested in plants or fungi you could make a database showing their common names, what species they belong to and where and when you saw them.

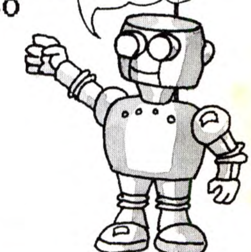
A food database would help you check whether you are eating a balanced diet. You could record which vitamins, protein or fat different foods contain.

Do a cars survey and make a database recording age of car, make, number of passengers and sex of driver. You could find out things like whether women drive newer cars than men.

Database program

This is the program for the database. The lines are quite long and complicated so type them carefully checking each line is correct before you press RETURN. Remember to make the conversions necessary for your computer (see pages 42-45).

```
 10 GOSUB 830
 20 LET T$="MAIN MENU":GOSUB 710
 30 PRINT:PRINT "1. LOAD FILE":
    PRINT "2. MAKE NEW FILE"
 40 PRINT "3. MAKE ENTRY":PRINT "4. VIEW ENTRIES"
 50 PRINT "5. ALTER ENTRY":PRINT "6. SAVE FILE"
 60 PRINT "7. SEARCH FILE":PRINT:
    PRINT "SELECT CHOICE"
 70 GOSUB 750:LET CH=KP-48
 80 IF CH<1 OR CH>ML THEN GOTO 20
★90 ON CH GOSUB 500,780,110,460,160,500,180
100 GOTO 20
110 IF LAST=NR-2 THEN RETURN
120 LET LAST=LAST+1
130 LET T$="MAKE ENTRY":GOSUB 710
140 LET TP=LAST:GOSUB 740
150 LET IP=LAST:GOSUB 680:RETURN
160 LET T$="ALTER ENTRY":GOSUB 710:GOSUB 740
170 LET IP=TP:GOSUB 680:RETURN
180 LET T$="SEARCH FILE":GOSUB 710
190 FOR I=1 TO LP:LET P(I)=0:
    LET R$((NR-1)*LP+I)=""
200 PRINT TAB(1);I;TAB(6);R$(I):NEXT I
210 PRINT:PRINT "SUBJECT OF SEARCH:"
220 LET A$="":INPUT A$:IF A$<"1" OR A$>"4"
    THEN GOTO 250
★230 PRINT R$(VAL(A$));":";:INPUT W$
240 LET R$((NR-1)*LP+VAL(A$))=W$+" ":IF W$>""
    THEN GOTO 220
250 PRINT "INFO REQUIRED:"
260 LET A$="":INPUT A$:IF A$<"1" OR A$>"4"
    THEN GOTO 280
270 LET P(VAL(A$))=1:GOTO 260
280 LET T$="SEARCHING FOR:":GOSUB 710
290 FOR I=1 TO 4
★300 IF R$((NR-1)*LP+I)>"" THEN
    PRINT TAB(0);R$(I);TAB(12);R$((NR-1)*LP+I)
310 NEXT I:PRINT:PRINT
320 LET C=0:FOR I=1 TO LAST:LET FL=0
330 FOR J=1 TO LP
340 LET S$=R$(I*LP+J)
350 LET F$=R$((NR-1)*LP+J)
360 IF F$="" THEN GOTO 420
370 LET F=0:LET LF=LEN(F$):LET LS=(LEN(S$)-LF)+1
380 FOR K=1 TO LS
```



Listing continued over the page.

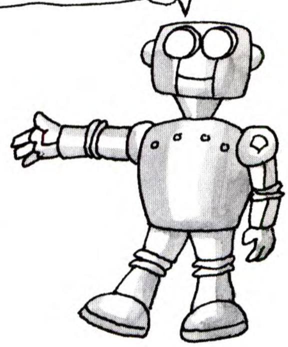
```
★390 IF MID$(S$,K,LF)=F$ THEN LET F=1:LET K=LS
 400 NEXT K
 410 IF F=0 THEN LET FL=1:LET J=LP
 420 NEXT J:IF FL=0 THEN GOSUB 620
 430 NEXT I
 440 PRINT:PRINT "SEARCH OVER":
     PRINT "FOUND ";C;" TIMES"
 450 GOSUB 750:RETURN
 460 LET TP=1:LET T$="VIEW ENTRIES"
 470 GOSUB 710:GOSUB 740:GOSUB 650:GOSUB 750
 480 IF KP=32 AND TP<LAST THEN LET TP=TP+1:GOTO 470
 490 RETURN
 500 LET T$="SAVE FILE":IF CH=1 THEN LET T$=
     "LOAD FILE"
 510 GOSUB 710:PRINT "FILE NAME:";:INPUT F$
 520 IF CH=6 THEN LET R$((LAST+1)*LP+1)="**"
★530 IF CH=6 THEN OPEN 1,1,1,F$
★540 IF CH=1 THEN OPEN 1,1,0,F$
 550 LET I=1
★560 IF CH=6 THEN PRINT#1,R$(I)
★570 IF CH=1 THEN INPUT#1,R$(I)
 580 IF R$(I)<>"**" THEN LET I=I+1:GOTO 560
★590 CLOSE 1
 600 IF CH=1 THEN LET LAST=INT(I/LP)-1
 610 RETURN
 620 PRINT "ENTRY ";I;:LET C=C+1:FOR K=1 TO LP
 630 IF P(K)=1 THEN PRINT TAB(12);R$(I*LP+K)
 640 NEXT K:PRINT:PRINT "PRESS SPACE":
     PRINT:GOSUB 750:RETURN
 650 FOR K=1 TO LP
★660 PRINT TAB(1);R$(K);TAB(15);R$(TP*LP+K)
 670 NEXT K:RETURN
 680 FOR I=1 TO LP:PRINT R$(I)
 690 LET A$="":INPUT A$:IF A$="" THEN LET A$="-"
 700 LET R$(IP*LP+I)=A$+" ":NEXT I:RETURN
★710 CLS:PRINT:PRINT
 720 PRINT "DATABASE":PRINT "--------":PRINT
 730 PRINT T$:PRINT:RETURN
 740 PRINT:PRINT "NOW ON ENTRY ";TP:PRINT:RETURN
★750 GET I$
 760 IF I$="" THEN GOTO 750
★770 LET KP=ASC(I$):RETURN
 780 LET T$="MAKE NEW FILE":GOSUB 710
 790 PRINT "TYPE IN HEADINGS":PRINT:FOR I=1 TO LP
 800 PRINT "HEADING ";I:INPUT A$
 810 IF LEN(A$)>10 THEN PRINT "TOO LONG":GOTO 800
 820 LET R$(I)=A$:NEXT I:GOSUB 850:RETURN
 830 LET NR=30:LET LP=4:LET ML=7
★840 DIM R$(NR*LP),P(LP)
 850 LET LAST=0:LET TP=1:RETURN
```

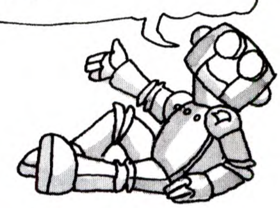
Looking at results

Experiments and surveys often produce lots of figures. To find out what the figures show, scientists summarize and test them in different ways. The programs below and over the page are simple examples of the things they do.

Averages program

```
★10 CLS
 20 LET LT=100
 30 DIM X(LT+1)
 40 LET N=1
 50 PRINT "VALUE ";N;" ";:INPUT A$
 60 IF A$="E" THEN GOTO 100
 70 LET X(N)=VAL(A$):LET N=N+1
 80 IF N<LT+1 THEN GOTO 50
 90 PRINT "NO MORE ROOM"
100 PRINT "END OF DATA"
110 LET N=N-1
120 GOSUB 150:GOSUB 210
130 GOSUB 250:GOSUB 290
140 STOP
150 FOR I=1 TO N-1:LET LW=X(I):LET P=I
160 FOR J=I+1 TO N
170 IF X(J)<LW THEN LET LW=X(J):LET P=J
180 NEXT J
190 LET T=X(I):LET X(I)=X(P):LET X(P)=T
200 NEXT I:LET X(N+1)=X(N)+1:RETURN
210 LET T=0:FOR I=1 TO N
220 LET T=T+X(I):NEXT I
230 PRINT "MEAN     = ";T/N
240 RETURN
250 LET MX=N/2:LET MD=X(INT(MX+1))
260 IF MX=INT(MX) THEN LET MD=(X(MX)+X(MX+1))/2
270 PRINT "MEDIAN   = ";MD
280 RETURN
290 LET SP=1:LET LP=1:LET MD=0
300 LET P=-1
310 LET P=P+1
320 IF X(SP)=X(SP+P) THEN GOTO 310
330 IF P>LP THEN LET MD=X(SP):LET LP=P
340 LET SP=SP+P
350 IF SP<N+1 THEN GOTO 300
360 IF MD>0 THEN PRINT "MODE     = ";MD
370 RETURN
```





An average is a way of summarizing figures. For instance, you may read about the average daily rainfall, or average earnings in an industry.

In fact there are three different kinds of average – the mean, median and mode – and they may give quite different pictures of the same set of figures.

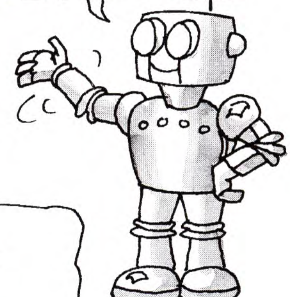
This program works out all three averages so you can compare them. The mean is calculated by adding all the numbers in a group and dividing by the number of numbers. The median is the middle number when you arrange them in order of size. The mode is the number which occurs most frequently.

Correlating program

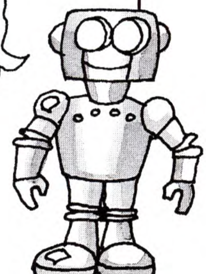
Correlating means finding out whether two sets of measurements are related to one another. For instance, is the amount a plant grows related to how much light it gets? Do people's ears get bigger as they grow older?

The program below makes your computer test measurements to see whether they are correlated. Opposite there are some ideas for measurements to collect and correlate.

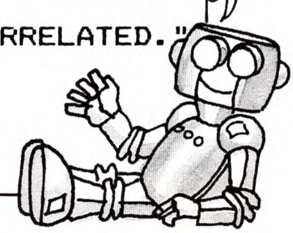
You type in the measurements in pairs, for instance, a person's age then their ear size.



The program tells you whether the sets of measurements are "positively" or "inversely" correlated. You can find out what these mean opposite.

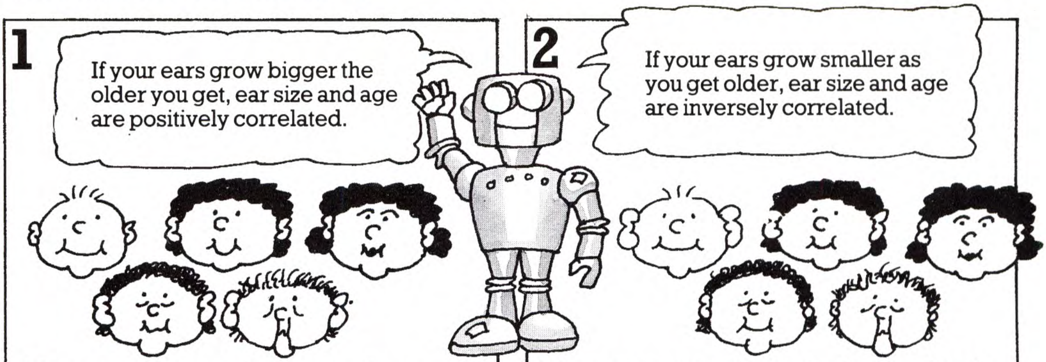


The program also says how much confidence you can have in the result. This depends on how many pairs of measurements you enter.



```
★10 CLS
 20 DIM A(100,2)
 30 PRINT "CORRELATION PROGRAM"
 40 PRINT
 50 PRINT "TYPE IN THE NUMBERS IN PAIRS"
 60 PRINT "TYPE  E"
 70 PRINT "WHEN YOU HAVE FINISHED"
 80 LET S=0:LET F=0:LET FS=0:LET FF=0
 90 LET SS=0:LET N=0
100 LET N=N+1
110 PRINT
120 INPUT "FIRST :";A$
130 IF A$="E" THEN GOTO 190
140 LET A(N,1)=VAL(A$)
150 INPUT "SECOND :";A(N,2)
160 LET F=F+A(N,1)
170 LET S=S+A(N,2)
180 GOTO 100
190 LET N=N-1:LET MF=F/N:LET MS=S/N
200 FOR Y=1 TO N
210 LET FS=FS+((A(Y,1)-MF)*(A(Y,2)-MS))
220 LET FF=FF+((A(Y,1)-MF)*(A(Y,1)-MF))
230 LET SS=SS+((A(Y,2)-MS)*(A(Y,2)-MS))
240 NEXT Y
250 LET R=FS/SQR(SS*FF)
260 LET D$="POSITIVELY"
270 IF R<0 THEN LET D$="INVERSELY"
280 LET C$="REASONABLY"
290 IF ABS(R)>0.7 THEN LET C$=" STRONGLY"
300 IF ABS(R)<0.3 THEN LET C$=" POORLY"
310 LET E$="SOME"
320 IF N<10 THEN LET E$="LITTLE"
330 IF N>60 THEN LET E$="STRONG"
340 PRINT "THE SETS ARE ";C$:PRINT D$;" CORRELATED."
350 PRINT "AS THERE ARE ";N;" PAIRS"
360 PRINT "YOU CAN HAVE ";E$
370 PRINT "CONFIDENCE IN THE RESULT"
380 STOP
```

More about correlations



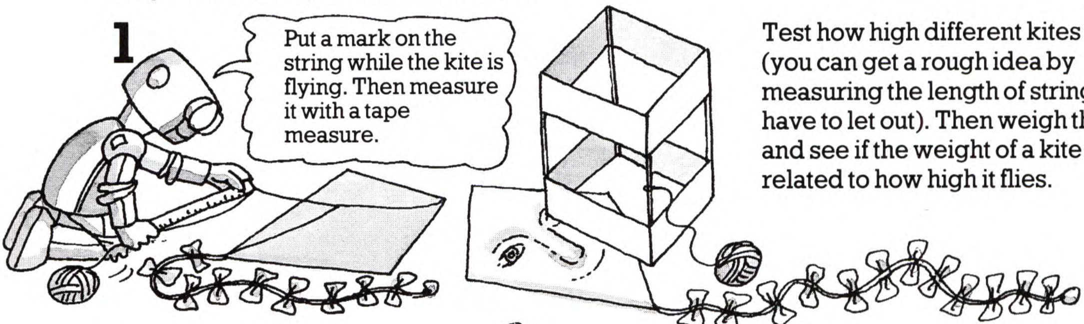
Measurements can be positively or inversely correlated. If they are positively correlated then, as one set of figures gets larger, so does the other.

Measurements are inversely correlated when one set gets smaller as the other gets larger. Try doing an age and ear size survey and see whether they are correlated.

Correlation experiments

Here are some ideas for things you can measure and test for correlation. If two things are correlated it often means one is caused by the other, but not always. Both may be caused by a third factor or the correlation may be a coincidence. For instance, you might find a correlation between people's heights and their results in a French test.

1



Test how high different kites fly (you can get a rough idea by measuring the length of string you have to let out). Then weigh them and see if the weight of a kite is related to how high it flies.

2

Is the quantity of water people can drink related to how tall they are? Measure your friends and have a drinking contest.

3

Most daily newspapers give figures which you can test for correlation. For instance, you could test whether the exchange rate is related to the interest rate, or to the daily temperature.

Likely or unlikely?

Before researchers accept that the results of an experiment or survey prove a theory, they need to find out what possibility there is of getting those results purely by chance. The program below is an example of a statistical test which checks the likelihood of a result occurring.

Imagine you are experimenting to find out whether a dice you have bought is correctly weighted (so that there is an equal chance of it falling on any of its six sides). Normally when you throw a dice the chances of getting a six are one in six. If you experiment and throw your dice 12 times you would expect to get two sixes. Say you get five sixes. Does this prove the dice is wrongly weighted? Or is it quite likely that the result happened by chance? Run the program to find out.

Chances program

```
★10 CLS
 20 PRINT:PRINT
 30 PRINT "HOW MANY TRIALS":INPUT N
 40 PRINT "HOW MANY TIMES WOULD"
 50 PRINT "YOU EXPECT THE RESULT"
 60 PRINT "TO OCCUR":INPUT E
 70 PRINT "HOW MANY TIMES DID"
 80 PRINT "THE RESULT OCCUR":INPUT R
 90 IF E>=N OR E<=0 OR R>N OR R<0 THEN
    PRINT "CHECK!":GOTO 30
100 IF R<E THEN LET TL=R+1:LET BL=-1
110 IF R>=E THEN LET TL=N+1:LET BL=R-1
120 LET P=E/N
130 LET FR=P/(1-P)
140 LET PP=(1-P)^N
150 LET MP=0:IF BL=-1 THEN LET MP=PP
160 FOR I=1 TO N
170 LET PN=((N-I+1)/I)*PP*FR
180 LET PP=PN
190 IF I<TL AND I>BL THEN LET MP=MP+PN
200 NEXT I
210 PRINT:PRINT
220 LET M$="MORE":IF BL=-1 THEN LET M$="LESS"
230 PRINT "CHANCE OF A RESULT"
240 PRINT "OF ";R;" (OR ";M$;") IS"
250 IF MP=0 THEN PRINT "TOO SMALL":GOTO 280
260 LET CH=1/MP:GOSUB 290
270 PRINT "1 IN ";CH;M$
280 PRINT:STOP
290 LET M$=""
300 IF CH>1E6 THEN LET CH=CH/1E6:
    LET M$=" MILLION"+M$:GOTO 300
310 IF CH>1E3 THEN LET CH=CH/1E3:
    LET M$=" THOUSAND"+M$:GOTO 310
320 LET CH=INT(10*CH+0.5)/10
330 RETURN
```

Running the program

```
1
HOW MANY TIMES DID
YOU DO THE EXPERIMENT
?12
```

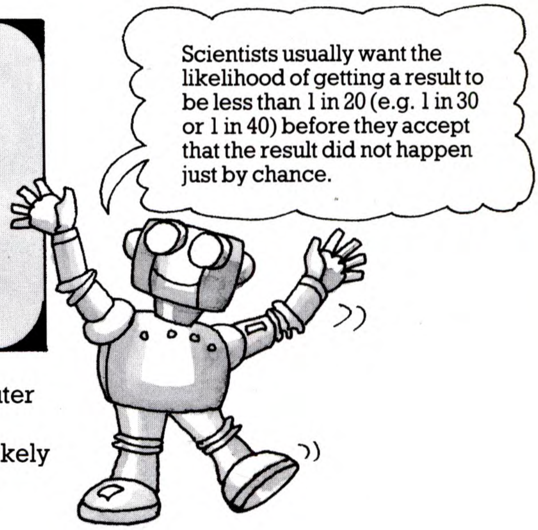
First type in the number of times you threw the dice.

```
2
HOW MANY TIMES WOULD
YOU EXPECT THE RESULT
TO OCCUR
?2
HOW MANY TIMES DID
THE RESULT OCCUR
?5
```

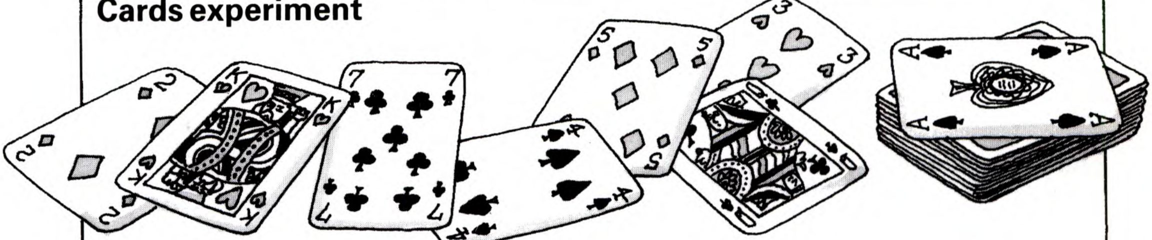
Then tell the computer how many times you expected the result (a six) to occur and how many times it actually did occur.

```
3
CHANCE OF A RESULT
OF 5 (OR MORE) IS
1 IN 27.5
```

Using the information you have given it the computer calculates that the likelihood of throwing five (or more) sixes is 1 in 27. So your result is quite an unlikely one and the dice may well be wrongly weighted.



Cards experiment



You can use the program to test the results of any experiment where you know the probability of the event you are investigating. For example, if a friend picks a playing card from a pack, your chances of guessing which suit it is are one in four. If you do the test 20 times you would expect to get five suits right. What are your chances of getting nine right suits, or 12? Use the program to find out.

Program conversions

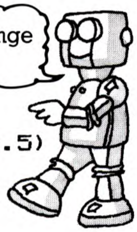
Here and on the next few pages there are conversion lines which enable you to adapt the programs for different makes of computer. To find a conversion line look in the section for your computer and find the page number and name of the program. If there is no conversion, you can type the line just as it is in the main program.

BBC

Page 11
Pulse rate program

```
390 LET I$=INKEY$(0)
500 LET FT=22000:LET PT=2000
```

Page 15
Journey model program




```
650 LET BJ=SB+INT
(DB*(RND(1)+RND(1))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*
    (RND(1)+RND(1))*0.5)
```

Changes for wait

```
50 LET WT=INT(RND(1)*FB)
```

Page 21
Running an airline

```
820 LET PS=POS-28
```

Page 26
Thermistor program

```
1000 LET V=1.8*ADVAL(1)/65520
1010 LET R=V/(5-V)
1020 LET T=1/((LN(R/K2))
       /K1+1/273)-273
```

Page 28
Graph lines

```
30 LET FT=100:LET DX=5
140 LET I$=INKEY$(0)
```

Page 31
Light sensor program

```
10 LET FT=100:LET DX=5
110 I$=INKEY$(0)
140 LET LL=1000-INT(ADVAL(1)/66)
```

Page 35
Database program

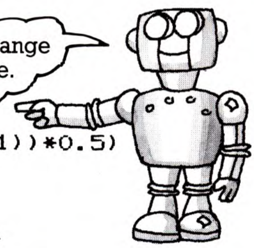
```
530 IF CH=6 THEN X=OPENOUT F$
540 IF CH=1 THEN X=OPENIN F$
560 IF CH=6 THEN PRINT#X,R$(I)
570 IF CH=1 THEN INPUT#X,R$(I)
590 CLOSE#X
750 LET I$=INKEY$(0)
```

Electron

Page 11
Pulse rate program

```
390 LET I$=INKEY$(0)
500 LET FT=7750:LET PT=1000
```

Page 15
Journey model program




```
650 LET BJ=SB+INT
    (DB*(RND(1)+RND(1))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*
(RND(1)+RND(1))*0.5)
```

Changes for wait

```
50 LET WT=INT(RND(1)*FB)
```

Page 21
Running an airline

```
820 LET PS=POS-28
```

Page 35
Database program

```
530 IF CH=6 THEN X=OPENOUT F$
540 IF CH=1 THEN X=OPENIN F$
560 IF CH=6 THEN PRINT#X,R$(I)
570 IF CH=1 THEN INPUT#X,R$(I)
590 CLOSE#X
750 LET I$=INKEY$(0)
```

VIC 20

Page 7
Coin-tossing program

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

Page 8
Bouncing ball

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

Page 11
Pulse rate program

```
420 PRINT CHR$(147)
440 POKE 36878,15:POKE 36876,200
442 FOR W=1 TO 100:NEXT W
444 POKE 36876,0
480 LET C(1)=2:LET C(2)=7:
    LET C(3)=5
500 LET FT=9000:LET PT=1000
```

VIC 20 continued

Page 15
Journey model program

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

Throughout the program change rnd to RND(1) as shown here.

```
650 LET BJ=SB+INT
    (DB*(RND(1)+RND(1))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*
   (RND(1)+RND(1))*0.5)
90 PRINT CHR$(147)
140 FOR J=1 TO F(N)/2:PRINT
    "*";:NEXT J
```

Changes for wait

```
50 LET WT=INT(RND(1)*FB)
```

Page 21
Running an airline

```
820 LET PS=POS(0)-13
880 PRINT CHR$(147)
```

Page 26
Thermistor program

```
1000 LET TR=1.84*PEEK(36872)
```

Page 28
Graph lines

```
30 LET FT=10:LET DX=20
```

Page 31
Light sensor program

```
10 LET FT=10:LET DX=20
140 LET LL=1000-4*PEEK(36872)
```

Page 35
Database program

```
710 PRINT CHR$(147):PRINT:PRINT
```

Page 37
Averages program

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

Page 38
Correlating program

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

Page 40
Likely or unlikely?

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

Commodore 64

Page 7
Coin-tossing program

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

Page 8
Bouncing ball

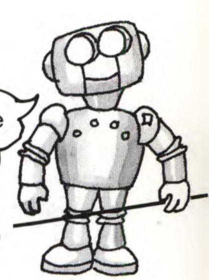
```
10 PRINT CHR$(147):PRINT
```

Page 11
Pulse rate program

```
420 PRINT CHR$(147)
440 POKE 54296,15:POKE 54277,0
442 POKE 54278,240:POKE 54276,33
444 POKE 54272,75:POKE 54273,39
446 FOR W=1 TO 100:NEXT W
448 POKE 54276,32:POKE 54296,0
480 LET C(1)=2:LET C(2)=7:LET
    C(3)=5
```

Page 15
Journey model program

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



Throughout the program change rnd to RND(1) as shown here.

```
650 LET BJ=SB+INT(DB*
    (RND(1)+RND(1))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*
   (RND(1)+RND(1))*0.5)
90 PRINT CHR$(147)
```

Changes for wait

```
50 LET WT=INT(RND(1)*FB)
```

Page 21
Running an airline

```
880 PRINT CHR$(147)
```

Page 35
Database program

```
710 PRINT CHR$(147):PRINT:PRINT
```

Page 37
Averages program

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

Page 38
Correlating program

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

Page 40
Likely or unlikely?

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

Apple

Page 7
Coin-tossing program

```
10 HOME
```

Page 8
Bouncing ball

```
10 HOME:PRINT
```

Page 11
Pulse rate program

```
390 LET I$="":IF PEEK(-16384)
    >127 THEN GET I$
420 HOME
480 LET C(1)=43:LET C(2)=
    46:LET C(3)=42
500 LET FT=9700:LET PT=1000
```

Page 15
Journey model program

Throughout the program change rnd to RND(1) e.g.

```
10  HOME
650 LET BJ=SB+INT
    (DB*(RND(1)+RND(1))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT
   (DB*(RND(1)+RND(1))*0.5)
90 HOME
120 PRINT TAB(1);B+N;TAB(5);" ";
```

Changes for wait

```
50 LET WT=INT(RND(1)*FB)
120 PRINT TAB(1);B+N;"-";
    B+N+1;"  ";
```

Page 21
Running an airline

```
880 HOME
```

Page 35
Database program

```
530 IF CH=6 THEN PRINT CHR$(4);
    "OPEN"+F$:PRINT CHR$(4);
    "WRITE"+F$
540 IF CH=1 THEN PRINT CHR$(4);
    "OPEN"+F$:PRINT CHR$(4);
    "READ"+F$
560 IF CH=6 THEN PRINT R$(I)
570 IF CH=1 THEN INPUT R$(I)
590 PRINT CHR$(4);"CLOSE"+F$
710 HOME:PRINT:PRINT
750 I$="":IF PEEK(-16384)>127
    THEN GET I$
```

Page 37
Averages program

```
10 HOME
```

Page 38
Correlating program

```
10 HOME
```

Page 40
Likely or unlikely?

```
10 HOME
```

TRS-80

Page 7
Coin-tossing program

```
70 LET X=RND(0)
```

Page 11
Pulse rate program

```
390 LET I$=INKEY$
440 SOUND 100,1
480 LET C(1)=4:LET C(2)=2:LET
    C(3)=1
500 LET FT=8700:LET PT=1000
```

Page 15
Journey model program

Throughout the program change rnd to RND(0) e.g.

```
650 LET BJ=SB+INT
    (DB*(RND(0)+RND(0))*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*(RND(0)
   +RND(0))*0.5)
135 IF N=15 THEN IF INKEY$=
    "" THEN GOTO 135
```

Type in this extra line.

Changes for wait

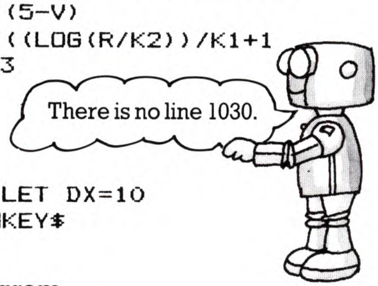
```
50 LET WT=INT(RND(0)*FB)
```

Page 21
Running an airline

```
820 LET PS=POS(0)-24
```

Page 26
Thermistor program

```
1000 LET V=JOYSTK(0)*0.0715+0.25:
     LET R=V/(5-V)
1010 LET T=1/((LOG(R/K2))/K1+1
     /273)-273
1020 RETURN
```



Page 28
Graph lines

```
30 LET FT=30:LET DX=10
140 LET I$=INKEY$
```

Page 31
Light sensor program

```
10 LET FT=30:LET DX=10
110 LET I$=INKEY$
140 LET LL=1000-16*JOYSTK(0)
```

Page 35
Database program

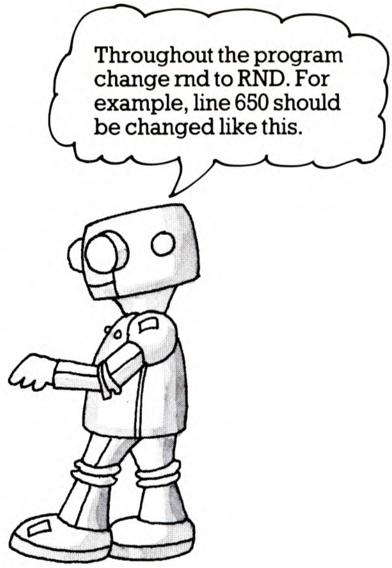
Letter O

```
530 IF CH=6 THEN OPEN "O",#-1,F$
540 IF CH=1 THEN OPEN "I",#-1,F$
560 IF CH=6 THEN PRINT#-1,R$(I)
570 IF CH=1 THEN INPUT#-1,R$(I)
590 CLOSE#-1
750 LET I$=INKEY$
```

Spectrum

Page 7
Coin-tossing program

```
70 LET X=RND
```



Page 11
Pulse rate program

```
390 LET I$=INKEY$
440 BEEP 0.1,5
480 LET C(1)=2:LET C(2)=6:LET C(3)=4
500 LET FT=1950:LET PT=300
```

Page 15
Journey model program

```
650 LET BJ=SB+INT(DB*(RND+RND)*0.5)
```

Page 19
Bus journey program

```
50 LET BJ=SB+INT(DB*(RND+RND)*0.5)
```

Changes for wait

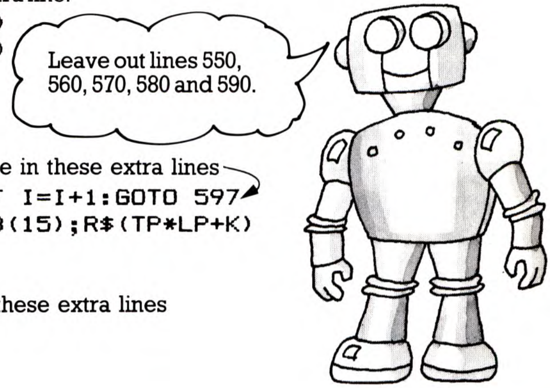
```
50 LET WT=INT(RND*FB)
```

Page 21
Running an airline

```
810 LET M$=P$(L)
820 LET PS=8-PEEK(23688)
```

Page 35
Database program

```
90 GOSUB 500*(CH=1)+780*(CH=2)+110*(CH=3)+460*(CH=4)+160*
   (CH=5)+500*(CH=6)+180* (CH=7)
225 PRINT A$            <---- Type in this extra line.
230 PRINT R$(VAL(A$))( TO 10);":";:INPUT W$:PRINT W$
265 PRINT A$            <---- Type in this extra line.
300 IF R$((NR-1)*LP+I)(1)<>" " THEN PRINT
    TAB(0);R$(I);TAB(12);R$((NR-1)*LP+I)
345 IF S$(LEN(S$)-1 TO LEN(S$))="  " THEN LET S$=S$( TO
    LEN(S$)-1):GOTO 345          Two spaces here.
355 IF F$(LEN(F$))=" " AND LEN(F$)>1 THEN LET F$=F$( TO
    LEN(F$)-1):GOTO 355
357 IF F$=" " THEN LET F$=""
365 LET F$=F$+" "
        (lines 345-365: Type in these extra lines.)
390 IF S$(K TO K+LF-1)=F$ THEN LET F=1:LET K=LS
515 PRINT F$            <---- Type in this extra line.
530 IF CH=6 THEN SAVE F$ DATA R$()
540 IF CH=1 THEN LOAD F$ DATA R$()
```




```
595 LET I=1 <---- Type in these extra lines
597 IF R$(I)( TO 2)<>"**" THEN LET I=I+1:GOTO 597
660 PRINT TAB(1);R$(K)( TO 10);TAB(15);R$(TP*LP+K)
695 PRINT A$
750 LET I$=INKEY$
770 LET KP=CODE(I$):RETURN   Type in these extra lines
805 PRINT A$
840 DIM R$(NR*LP,30):DIM P(LP)
```

Page 37
Averages program

```
55 PRINT A$
```

Graphics routines

On these two pages there are versions of the graphics routine which you need to add to the Bouncing ball, Pulse rate, Thermistor and Light sensor programs. Make sure you use the right version for your computer. The program lines are quite complicated with lots of numbers and symbols, so type carefully and check each line before pressing RETURN.

Apple

```
2000 HOME
2010 FOR I=1 TO 22:VTAB(I):PRINT "  !":NEXT I
2020 FOR I=1 TO 39:VTAB(23):HTAB(I):PRINT "-":NEXT I
2030 VTAB(23):HTAB(1):PRINT "O"
2040 VTAB(1):PRINT Y$;TAB(10);T$
2050 VTAB(11):PRINT M$
2060 VTAB(24):HTAB(34):PRINT X$;
2070 LET DC=43:RETURN
2200 VTAB(1):HTAB(34):PRINT N;"  "
2205 IF X<O OR Y<O OR X>1000 OR Y>1000 THEN RETURN
2210 LET QX=X/28:LET QY=Y/48
2220 VTAB(22-QY):HTAB(QX+3):PRINT CHR$(DC)
2230 RETURN
```

Type 13 spaces here.

```
2400 VTAB(24):HTAB(5):PRINT "             ";
2410 VTAB(24):HTAB(5):PRINT M$;
2420 RETURN
```

BBC and Electron

```
2000 MODE 1:VDU 19,3,2;0;:COLOUR2:CLS:GCOL0,2:DC=1
2010 MOVE 0,100:DRAW 1200,100:MOVE 50,50:DRAW 50,1000
2020 PRINT TAB(10,1);T$;TAB(0,1);Y$;TAB(33,30);X$
2030 PRINT TAB(0,14);M$;TAB(0,28);"O":RETURN
2200 IF X<O OR Y<O THEN RETURN
2210 GCOL0,DC:PLOT 69,X+50,Y*0.88+100
2220 PRINT TAB(33,1);N;"  "
2230 RETURN
2400 PRINT TAB(5,30);STRING$(13," ");
2410 PRINT TAB(5,30);M$;
2420 RETURN
```

Spectrum

```
2000 PAPER 0:CLS:INK 6:LET DC=2
2010 PLOT 0,15:DRAW 255,0
2020 PLOT 7,0:DRAW 0,175
2030 PRINT AT 0,0;Y$;AT 0,8;T$;AT 20,0;"O"
2040 PRINT AT 21,25;X$;AT 10,0;M$:RETURN
2200 INK 6:PRINT AT 0,27;N;"  "
2210 IF X<O OR Y<O THEN RETURN
2220 LET QX=X/4+8:LET QY=Y/6.25+16
2230 IF QX>255 OR QY>175 THEN RETURN
2240 INK DC:PLOT QX,QY:INK 6:RETURN
2400 PRINT AT 21,5;"             "
2410 PRINT AT 21,5;M$:RETURN
```

Type 13 spaces here.

TRS-80

```
2000 CLS(0)
2010 FOR I=0 TO 29:SET(1,I,2):NEXT I
2020 FOR I=0 TO 63:SET(I,28,2):NEXT I
2030 PRINT@9,T$;:PRINT@0,Y$;
2040 PRINT@507,X$;:PRINT@448,"O";:PRINT@192,M$;
2050 LET DC=4:RETURN
2200 IF X<0 OR Y<0 OR X>1000 OR Y>1000 THEN RETURN
2210 SET(X/16.5+2,27-Y/37.1,DC)
2220 PRINT@28,STR$(N);"  ";
2230 RETURN
```

Type 13 spaces here.

```
2400 PRINT@484,"             ";
2410 PRINT@484,M$;:RETURN
```

Commodore 64

Here press the C= and M keys together.

```
   5 GOSUB 3000
2000 POKE 53281,0:POKE 53280,0
2010 PRINT CHR$(158);CHR$(147)
2020 PRINT HM$;:FOR CT=1 TO SH-2:PRINT "[C=][M] "
     :NEXT
```

Press the C= and T keys together 38 times.

```
2030 PRINT "O [C=][T] "
2040 PRINT TAB(WS-5);X$;HM$;Y$;HM$;SPC(WS/5);T$
2050 PRINT HM$;LEFT$(CU$,(SH-3)/2);M$
2060 DC=2:RETURN
2200 IF X<0 OR Y<0 OR X>1000 OR Y>1000 THEN RETURN
2210 QX=X/XK:QY=Y/YK
2220 ZX=INT(QX/2):ZY=INT(QY/2)
2230 BX=INT(QX-ZX*2):BY=INT(QY-ZY*2)
2240 QD=2↑(2-2*BY+BX):SN=BL+ZX-LC*ZY
2250 PN=PEEK(SN):FOR CT=0 TO 15
2260 IF QQ(CT)=PN THEN PN=QQ(CT OR QD):CT=15
2270 NEXT:POKE SN,PN:POKE SN+DM,DC
2280 PRINT HM$;SPC(WS-5);STR$(N);"  ":RETURN
```

Type 13 spaces here.

```
2400 PRINT HM$;LEFT$(CU$,SH-1);SPC(3);"             ";
2410 PRINT HM$;LEFT$(CU$,SH-1);SPC(3);M$;:RETURN
3000 DIM QQ(15):FOR I=0 TO 15:READ QQ(I):NEXT
3010 HM$=CHR$(19):CU$="":FOR I=1 TO 25:CU$=CU$+CHR$(17)
     :NEXT
3020 LC=40:BL=1905:DM=54272
3030 XK=13:YK=22.2:SH=25:WS=40
3040 RETURN
3050 DATA 32,126,124,226,123,97,255,236,108,127,225,
     251,98,252,254,160
```

VIC 20

Use the Commodore 64 version, but change these lines.

Press the C= and T keys together 20 times.

```
2000 POKE 36879,8
2030 PRINT "O [C=][T] "
3020 LC=22:BL=8121:DM=30720
3030 XK=24:YK=24:SH=23:WS=22
```

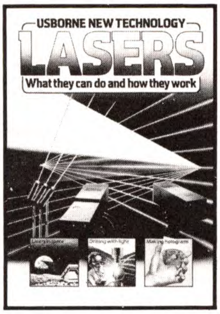
Index

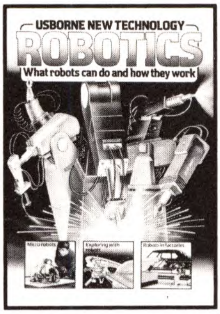
First published by Usborne Publishing Ltd,
20 Garrick Street, London WC2E 9BJ, England.

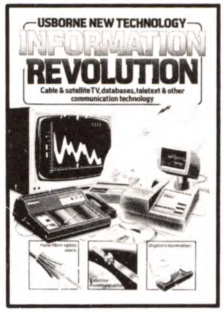
 Printed in Spain D.L.B. 1912-1985

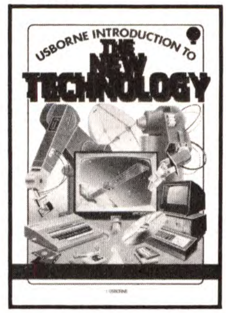
Other Usborne Books

There are hundreds of colourful Usborne books for all ages on a wide range of subjects. Titles which may be of particular interest to you are:



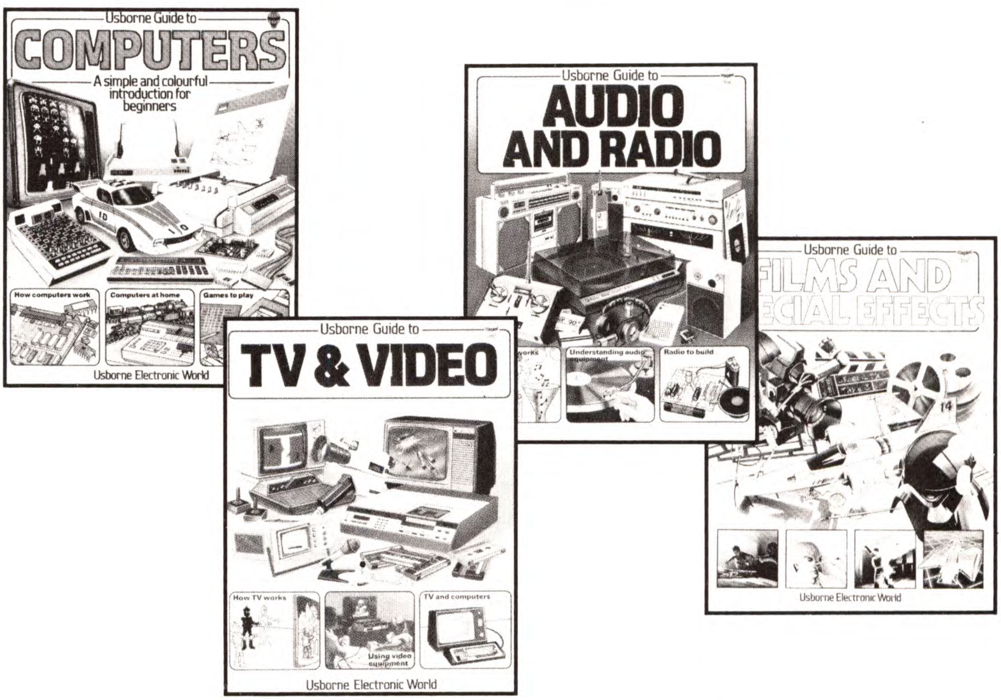






This exciting new series takes a serious look at what is happening *now* in the world of new technology. Many people think that such things as lasers, robots, databases and interactive TV belong only to the world of science fiction but, as these brilliantly illustrated books show, many of them are already in use and affecting our everyday lives. The books take a straightforward approach to these apparently difficult subjects, making them easy for everyone to understand.

Page size: 240 × 170 mm 48 pages



This up-to-the-minute series on electronic technology explores the worlds of computers, TV and video, audio and radio and, in a new title, films and special effects. In a clear visual way, the books describe the very latest equipment and show what it does and how it works. They also explain much of the confusing technical jargon which usually surrounds these subjects. There are fascinating sections on what computers can do for us and how they do it, how TV and video cameras can turn an ordinary scene into a pattern of electronic signals that can be stored on tape, and how a recording studio works. *Audio & Radio* also contains instructions for building a simple radio.

Page size: 276 × 216 mm 32 pages

Usborne Computer Books

"Highly recommended to anyone of any age." Computing Today
"Without question the best general introduction to computing I have ever seen." Personal Computer World
". . . perhaps the best introduction around . . . outstanding . . ." Educational Computing
"These books are outstanding . . . they make all other young people's computer books look meretricious." Times Educational Supplement

Guide to Computers
Understanding the Micro
Computer and Video Games
Computer Jargon
Computer Graphics
Inside the Chip
Computer Programming
Practise Your BASIC
Better BASIC
Machine Code for Beginners
Practical Things to do with a Microcomputer
Computer Spacegames
Computer Battlegames
Write Your Own Adventure Programs
Creepy Computer Games

New Titles

Programming Tricks & Skills Professional tips and tricks for better programming.

Experiments With Your Computer An exciting introduction to scientific investigation with a microcomputer.

Expanding Your Micro A detailed guide to add-ons and interfaces.

Write Your Own Fantasy Games A step-by-step guide to writing fantasy games, with program listing.

Weird Computer Games, Computer Spy Games Short listings to run on most main home computers.

Mystery of Silver Mountain, Island of Secrets Unique adventure game books containing full program listings.

ISBN 0-86020-791-9

9 780860 207917

ISBN0 86020 791 9

£2.25