INFORMATION HANDLING

SPECIAL

MICRO SCOPE

Newman College with MAPE

Contents

We are grateful to IBM for their sponsorship of this Special.

Editorial team Heather Govier, Bryan Weaver (ILEA)
Senga Whiteman, Roger Keeling (MICRO-SCOPE)

ISBN 0-948-04800-X
Correspondence to the Editor: Newman College, Bartley Green, Birmingham, B32 3NT
Tel: 021 476 1181

MAPE (Micros and Primary Education) is open to individuals and institutions.
The current subscription of £8.50 p.a. (UK) includes direct mailing of **MICRO-SCOPE**.
Application forms from: Mrs. G. Jones, 76 Sudbrooke Holme Drive, Sudbrooke, Lincs.

Published by Castlefield (Publishers) Ltd

Individual copies from Castlefield (Publishers) Ltd, 12 Chater Street, Moulton, Northants, NN3 1UD.
Tel: (0604) 494660

Typeset by Castlefield Press, Northampton.
Printed by Biddles of Guildford.

MICRO-SCOPE INFORMATION HANDLING SPECIAL

Editorial

Information handling in schools is nothing new. The accessing, organisation and presentation of knowledge has always been the very nub of education. At its worst this has consisted of drill, rote learning and regurgitation of diverse facts of varying usefulness.

However the majority of teachers have striven to introduce children to a range of information handling techniques from discovery methods to higher order reading and writing skills. Microcomputers present an opportunity for a fresh look at the nature of information and the ways in which schools can develop pupils' abilities to use it. Traditional skills such as reading and sorting are still needed but new techniques must be added to these if pupils are to become efficient users of information presented on a screen instead of on paper. Among other strengths, the computer highlights the need for precision and accuracy in such things as spelling and the formulation of questions. Its major contribution comes from its power to sort vast quantities of data quickly and accurately and to produce a clear and readable display.

Children do need to know about the ways in which information can be structured and accessed, and the best teachers will introduce this knowledge in the context of material which is relevant, meaningful and interesting to their pupils. Such is the work described in the five articles which comprise this booklet. None of the authors are 'computer-experts' but all are experienced teachers of primary pupils who see the micro as a tool for the development of the primary curriculum.

Four distinct approaches to data manipulation are presented:

1. the use of a simple BASIC program which emulates punched cards (mentioned in Chapter 1 and described in greater detail in Chapter 3);
2. the use of a binary tree form of hierarchical database (Chapters 2 and 3);
3. the use of another hierarchical system which emulates Prestel or local viewdata (Chapter 4);
4. the use of a package which handles data in the form of tables allowing very flexible access (Chapters 1 and 5).

Whilst reference is made to specific items of software, much of this work could be done with alternative packages. The main objective of this booklet is to give some examples of ways in which creative teachers have used micros to help children handle information, in the hope of sparking off further good ideas in the reader.

We are indebted to IBM whose sponsorship enabled us to commission the five articles and to the authors whose prompt response and eloquent presentations left us with little work to do.

Heather Govier *September 1984*
Bryan Weaver

Personal data and information retrieval

From time immemorial (or nearly) children in our schools have investigated facts about themselves, their peers, their families, their teachers, and so on. Many a fond parent on Open Day has been stunned by a revelation of a domestic incident on reading a young offspring's 'diary'.

Apart from written expression, the use of barcharts and histograms to record the details about their friends and families – heights, weights, birthdays, colouring, and so on, has occupied children from an early age for a significant part of the time allocated to 'mathematical' learning. Whether they have learnt how to interpret or retrieve their recorded information in a different form, is another matter. Schools have been littered with graphical representations since mathematics went 'modern'.

Later in their primary school careers many children will have come across punched cards and perhaps learnt how to manipulate these to record the sort of information mentioned above.

The advent of the microcomputer has given a new impetus to this activity. It offers the opportunity to store a permanent record of facts easily and in a limited space. The record can be revised or updated and the information is stored in a way that allows rapid searching. It is not surprising, therefore, that a number of programs which enable children to record and access information are already available for micros, and they have been used in some cases for handling personal data.

Taking as a starting point Frank Gregory's 'Six-line program', we find a direct link with punched cards – in fact, it depends on them. A listing follows:

```
 10 REM***SAMPLE SIX-LINE RETRIEVAL
    PROGRAM***
 20 FOR I=1 TO 5: REM***NO. OF RECORDS***
 30 READ N$,A,B,C,D,E:REM***N$=NAME***
 40 IF A=0 AND B=1 THEN PRINT N$
 50 NEXT I
 60 DATA JACK,1,0,1,0,0
 70 DATA JILL,0,1,0,0,1
 80 DATA TOM,1,0,1,0,1
 90 DATA MARY,0,1,0,0,1
100 DATA SALLY,0,0,1,1,1
110 END
```

Here the data is taken from the punched cards:

	Boy	Under 10	My friend	Has brother	Has sister
Jack	1	0	1	0	0
Jill	0	1	0	0	1
Tom	1	0	1	0	1
Mary	0	1	0	0	1
Sally	0	0	1	1	1

By changing line 40 different sets of information can be retrieved. A considerable amount of work has been done using this program, some of it very sophisticated. It is, however, a very limited use of information retrieval and it employs a binary search. (One piece of work which uses a variant of the six-line program is described in Chapter 3.)

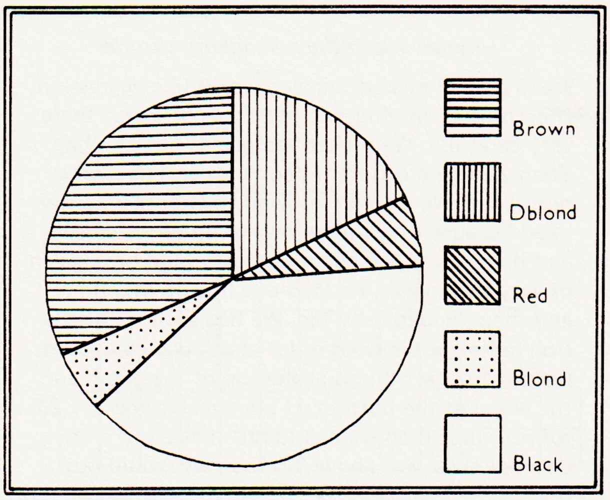
Other binary search programs are available (see Chapter 2). These have been found useful with young children in handling information about their peers and their teachers. The use of a powerful information retrieval program such as *LEEP* or *MICROSCAN* offers vastly more exciting prospects and the work of several primary schools will be discussed. Several schools have used personal data files which have dealt with such things as height, weight, shoe size, hair colour, etc. in effective ways. One school had a topic on 'Growth' and included in its data statistics of children in the first, second and third year junior classes. The program *LEEP* was used to store the information they gathered. They followed this up six months later with a further set of information when the children had changed classes. Their enquiries led them to investigate sex differences, the varying rates of growth within the three age groups, etc. Another school used *SEEK* with 6 year olds to create a file about the children in their class while in the same school older children used *LEEP* to work on both the staff (teaching and non-teaching) and also pupils in their own class.

An interesting thing occurred in another school where a single child insisted on creating her own file on the class, only to discover to her dismay that she could not ask the questions she had really wanted to have answered because her fields were not adequately organised. (She had declined the offer of assistance from her teacher.) The result of a great deal of work, including the typing in of the information, was a lesson for the child and incidentally for the teacher, since they realised how essential it is to ensure that the field structure is right. This is especially true in the case of *LEEP* as it is difficult, if not impossible, to make major alterations after the file has been constructed. This problem is reduced by the more flexible and powerful ILECC program *MICROSCAN.*

The child's problem mentioned above is also illustrated in the piece of work by fourth year juniors in another school which we will discuss at greater length. In this case, a file was remade in order that the omissions could be included.

These children took as their topic 'Myself'. This will strike a chord in those who know the Schools Council packages for Health Education. The 23 children in the class, boys and girls of almost equal number and of different ethnic origins, first made a profile of their own physical characteristics, height, weight, etc. Considerable discussion took place as to what fields to include. They wrote about themselves in notebooks which eventually contained their enquiries on their *LEEP* files as well. They then entered their statistics into a file (Wisely). Soon they were making enquiries and drawing pie charts and bar charts of the results. For instance, pie charts of hair colour were drawn, and one child showed a most interesting approach to the mathematical ratios involved in representing the groups of children with the same hair colour (see Fig. 1). Consider, too, the thought involved in deciding the colours – dblond presumably meaning 'darker blond' not 'dyed blond'. They made bar charts of shoe sizes, separating the boys from the girls for comparison and did the same for their heights (see Fig. 2). They also dealt with hair type, curly or straight, and some even sacrificed a lock of hair to illustrate the differences. Did you know that curliness goes with lighter colour? It was one hypothesis made. Whether they were right or wrong at least they were hypothesising and perhaps learning the need for larger numbers of statistics in order to form more correct judgements. Some of the enquiries made are reproduced.

An even more interesting file with its consequent enquiries came from their second profiles of themselves – their personalities and/or characters. Here they adopted an interesting approach, though not, of course, unknown before. Having spent a long time discussing what sort of things to include they arrived at a fascinating mixture of temperamental characteristics and abilities. They then assessed themselves on the basis of these, awarding a mark out of 10. Next they asked ten others (including their teacher) to give a rating. An average score was found. This was the mark entered in the Personality file (Wisely2). Two children's descriptions of the process are included (Fig. 3);



This is a Pie chart of the colour of people's hair.

Dblond	Dblond	Blond
Blond	4 64°	1 16°
Black	Black	
Red	9 144°	Red
Brown	Brown	1 16°
	8 128°	

I asked the computer for a file called WISELY taking out all the hair colours and field names on the printer, then I counted the numbers of the hair colours and gathered them in to groups. I divided 23 in to 360 because there are 23 people in the class, I found the answer which was roughly 16 then I multiplied it by each number of hair colour.

Figure 1 Child's pie chart of hair colour.

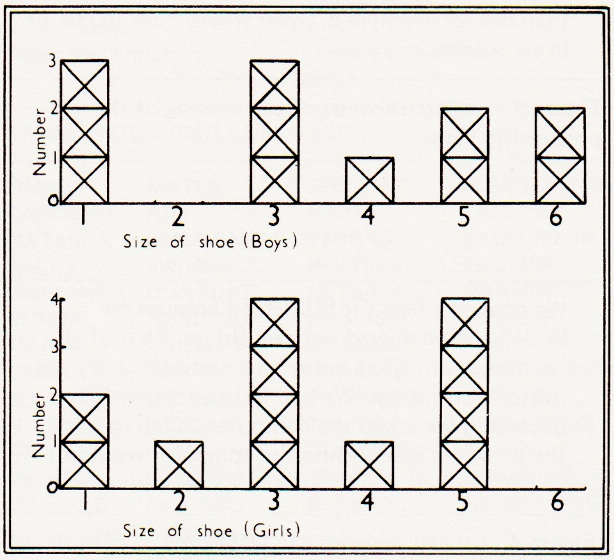


Figure 2 Bar chart of shoe sizes.

and a Personality profile is reproduced (Fig. 5). However, when they came to make their enquiries they found that they had omitted certain crucial fields, as one boy describes in

How we made the personality profile

We first of all picked thirteen words like aggression, cheerfulness, intelligence. We thought about these very carefully. We picked ten names out of a hat. Then we asked different people to give us marks out of ten. We weren't allowed to choose who we liked because they might have given us top marks and if we asked our enemies they would have given us very low marks. I added one line up at a time and then divided it by ten. We had another discussion to find out in what order to put our categories. We put the best things at the top of the graph and the worst at the bottom. If our average was say 2.5 for noisiness then we would put it as 2 but if the decimal place was above 2.5 then we would put it as 3.

* * * * *

First of all, we had a class discussion and picked thirteen categories for a file. When we'd finished our work, we had to pick ten names out of a hat or box, of people in our class. Then I would go to the ten people and ask them to give me marks out of ten for all of the categories. When I'd finished going to the ten people and getting the marks I would have to find out what the averages were. You found the average by adding up the ten marks from the first category and divided it by ten. You had to pick ten names out of a hat because if we asked our best friend she, or he, would give us high marks, but if you asked our worst enemy she, or he, would give us low marks. We made a graph. We wrote down the thirteen categories in a different order. We decided to put the good things first and the worst at the bottom. If the decimal place was for example 5.5 you would only have to colour in five squares, but if the decimal place was higher than five for example 5.7 you would have to colour in six squares.

Figure 3 Children's views of the making of the personality files.

We opened a new file WISELY3 because on WISELY2 we missed out two things. First of all we missed out what sex we were and secondly we missed out our age. We had to open a new file because if we asked the computer to tell us all of the girls ages and dislikes the computer wouldn't know what age and what sex the records were.

Figure 4 Crucial omissions in WISELY2.

Fig. 4. They needed, therefore, to create a further file (Wisely3), which is really a revision of Wisely2. With this they began their enquiries, for example – 'find the naughty boys', or 'find the naughty boys who are good at . . .'. Looking at the fields it is easy to see the interesting possibilities opening up. Looking through the children's books, too, it is evident that they found total involvement and the ethos of the class shows clearly. There is little doubt that the investigations made possible by a good information retrieval package are not achieved so easily by other means.

Significantly, much of the work was done away from the computer. Not only writing and drawing charts took place but also a great deal of discussion and co-operative learning at peer group level. Thus information retrieval can have social benefits as well as individual.

One or two suggestions may be in order here. All the evidence shows that a great deal of thought and preparation is necessary at the preliminary stages of file creation. This is true whatever retrieval package is used, not only in the cases of the more powerful programs. It is beyond doubt that children of junior age can cope with the concepts implicit in such work. It demands considerable guidance and effort from the teacher but what good learning doesn't? – and the rewards are great.

In addition, the temptation to use the trivial needs to be resisted. The personal statistics of children are not trivial to them, but they can be trivialised by mishandling. Therefore the enquiries must also be given serious thought and ideally should lead children to find excitement in hypothesising from their data. The thrill of finding that what you thought was the case, is in fact so, brings delight to children. The vital point to remember about using children's own data is that it is peculiarly theirs in a way that no other data is, and it should be accorded due respect.

In conclusion, it appears that only a first class program and substantial memory can provide the power to allow children to have worthwhile experiences of data retrieval, and so acquire an important skill in this information age.

Laurie Tate
Advisory teacher,
Inner London Education Authority

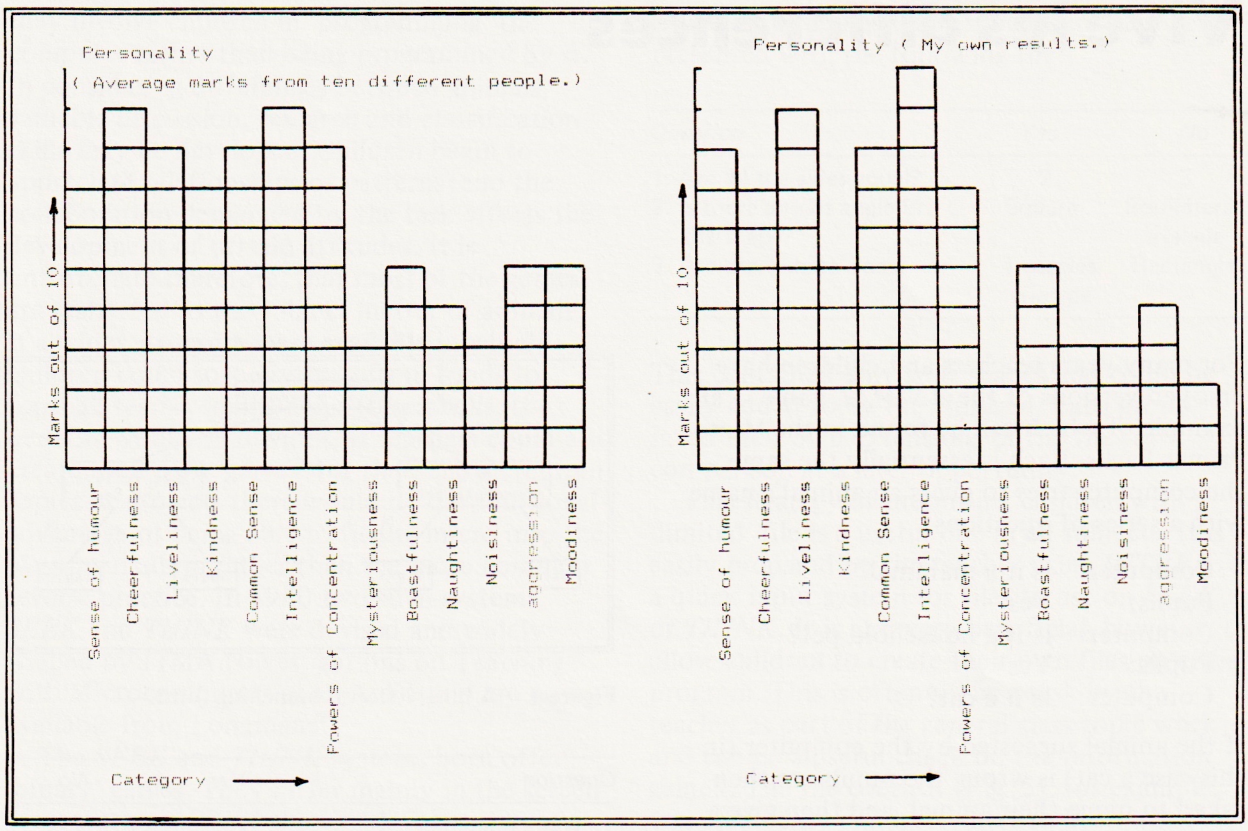


Figure 5 Bar charts of child's personality profile.

How many boys are good at football?
Five are good at football. Their names are Simon, Karl, Ramish, Motalib and Blair.

Filename wisely3
Your Enquiry . . . . Sex=m
OUTPUT to . . . . Screen Printer
Print Format 5
Fields for Output . . . . Fname Skill1 Skill2 Skill3

FNAME/SKILL1/SKILL2/SKILL3/

GERARDO	DRAWING	POGO STICK	ROBOTICS
MOTALIB	FOOTBALL	BMX	BADMINTON
ROCKY	BADMINTON	SWIMMING	ART
SIMON	FOOTBALL	SWIMMING	BADMINTON
KARL	SWIMMING	FOOTBALL	BADMINTON
BRIAN	BIKERIDING	RUNNING	FIGHTING
FILIPPO	DRAWING	LEGO	BMX
RAMISH	FOOTBALL	BMX	ROUNDERS
BLAIR	FOOTBALL	SWIMMING	BADMINTON
RAMISH	FOOTBALL	BMX	CH.CRICKET
BEN	ROBOTICS	MAKING BIKES	LEGO

The file is 26 records long
11 Records matched
END OF Micro LEEP

How many girls are good at netball?
Four girls are good at netball. They are Giovanna, Jeanette, Melanie and Heena.

Filename Wisely3
Your Enquiry . . . . Sex=f
OUTPUT to . . . . Screen Printer
Print Format 5
Fields for Output . . . . Fname Skill1 Skill2 Skill3

FNAME/SKILL1/SKILL2/SKILL3/

DIANA	MATHS	SPELLING	PUNCTUATION
CAROLINE	ART	MATHS	ENGLISH
NATASHA	SWIMMING	STORYS	BADMINTON
SALLY	SWIMMING	WRITING	ENGLISH
GIOVANNA	RUNNING	NETBALL	DRAWING
EMILIA	READING	ENGLISH	RUNNING
MARIA	ENGLISH	SCIENCE	TOPIC
SABRINA	MATHS	COOKING	PUNCTUATION
JEANETTE	NETBALL	SWIMMING	TENNIS
HEENA	NETBALL	TENNIS	BADMINTON
AMANDA	SWIMMING	RUNNING	GAMES
VEDIA	BADMINTON	TENNIS	SWIMMING
MELANIE	SKATING	NETBALL	ICE-SKATING

The file is 26 records long
13 Records matched
END OF Micro LEEP

Figure 6 Examples of enquiries.

Vive les differences

For many years teachers and children have enjoyed versions of the *ANIMAL* game – an example of which can be found in the Micro Primer Packs. Each is essentially the same – the computer tries to guess an animal's name with a series of questions:

Computer: Is it a mammal?
Pupils: Yes
Computer: Is it a household pet?
Pupils: Yes
Computer: Is it a cat?

If the animal suggested by the computer (in this case a cat) is wrong the pupils are then asked to name their animal, and then give a question to differentiate this new creature from the computer's guess. Thus, if a dog had been chosen the children would need to agree on a 'Yes/No' question to differentiate between it and a cat – a challenging task often resulting in much discussion and argument:

Pupil 1: Let's choose, 'Does it eat bones?' Dogs eat bones.
Pupil 2: My cat eats bones though!
Pupil 3: Yer! Cats eat bones.
Pupil 2: What about, 'Does it bark.' Cats don't bark.
Pupil 1: Or, 'Does it eat Whiskas?'
Pupil 2: Our dog eats Whiskas. You can't have that.
Pupil 3: We give our cat and dog 'Chum'. They both eat it.
Pupil 1: I think it will have to be, 'Does it bark?'
Pupil 3: That was yours Sarah. Okay?
Pupil 2: Yes, sure! Let's type it in.

ANIMAL-type structures in which there is a set of questions each having two possible responses (Yes and No) are known as binary trees. They can be found in two forms; namely a 'tree' down whose branches you search for information (Fig. 1), or a 'key' (Fig. 2) which must be carefully followed – the numbers guiding the reader to the next relevant questions or an object.

Binary trees are useful in many aspects of life – botanists search through them in their Floras to identify specimens; an amateur motor mechanic may seek help from one in a car handbook to diagnose a fault in his vehicle; children may even possess one of the new types of reading book adopting some 'choice' as to how the story should progress. (The latter are often 'binary' in structure, but several choices may also be possible).

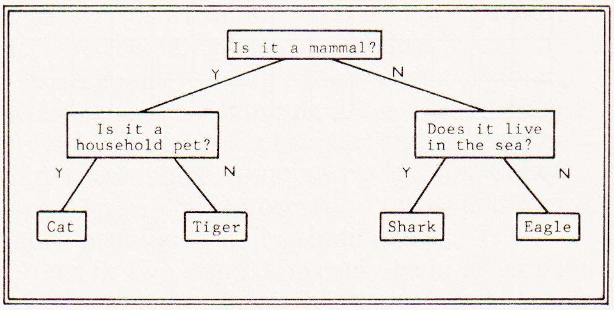


Figure 1 A binary tree in branching form.

Question	*Yes*	*No*
1. Is it a mammal?	2	3
2. Is it a household pet?	Cat	Tiger
3. Does it live in the sea?	Shark	Eagle

Figure 2 A binary tree in 'key' form.

The main advantage of placing binary trees onto a computer is that only one question need be displayed at a time. This avoids the distraction caused by viewing all the information at once, as when keys or trees are presented in other forms. Anyone doubting this should search for the name of some common flower e.g. a buttercup, in a popular amateur naturalists' handbook. Note how tempting it is to ignore the complex-looking 'key' and rush straight to the illustrations; yet this is too often a wasteful and unrewarding task. There are many flowers that look similar to the common buttercup and only by returning to the 'key' will we be able to differentiate between them.

Just as binary trees are commonly used in society, so classification – the process used to devise such structures – lies at the heart of learning in school. Indeed, noticing differences and similarities is so basic to concept formation that most learning cannot occur without it. Watching children use *ANIMAL* quickly reveals the educational potential of such games. First they involve children in 'programming' the computer rather than being programmed by it. In considering differences between animals, valuable discussion, research and classification skills may be developed. Children begin to appreciate relationships or patterns; and the collaboration demanded by the task effects the development of certain attitudes. It is unfortunate therefore, that most of the games are restricted to the subject matter of animals. If a wider structure were available in which children could sort anything from foods to football teams, sailing ships to seashells, the benefits would be obvious. If teachers could also create their own 'games' for their classes' current topics or projects there would be the additional advantage of being able to fit the micro into the class curriculum rather than the more common reverse practice. In 1980 two such systems, *SEEK* and *THINK* were devised and widely trialled by ITMA (Investigations on Teaching with Microcomputers as an Aid)[1] and are now available from Longmans[2].

The *SEEK* and *THINK* systems both offer 'binary' games. They differ mainly in the screen presentations, *SEEK* showing information as a branching tree around which the user can browse, and *THINK* retaining the text structure of the original *ANIMAL* game (Figs 3 and 4). Information stored on one system can, however, be used on the other. There are other differences too, which will be discussed later.

The original 'guessing game' of *ANIMAL* is retained in *SEEK* and *THINK* and can be used for any file created by the children or teacher.

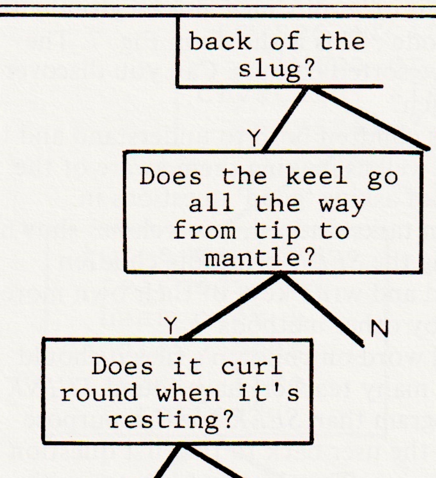
One class studying shape in mathematics were presented with the following file:

Question	*Yes*	*No*
1. Are all the sides equal?	2	3
2. Is there a right angle in the shape?	Square	Equilateral triangle
3. Are there three sides?	Isosceles triangle	Rectangle

This contained only four shapes and could be easily added to by the children. Each group saved their own extensions of the file for comparison at a later opportunity.

This strategy of presenting children with a 'limited' file is a useful one in all subjects. It is easily prepared by the teacher using *INTREE* – a quick input system for files to use on *SEEK* or *THINK*. It is also very beneficial, however, to allow children to create their own files on either program. This is often set as a task by the teacher as part of the general class topic work and can be a useful check on the information gained. The file overleaf (Fig. 5), for example, was created by children who had been working on leaves. Only a section is shown.

The child-created files also present opportunity for real observation to occur – actual objects being a popular choice of subject matter on which classification can be based. Let us join a group of children who are doing this. They are planning to teach the computer differences between some selected samples of cheese. The labelled samples are in front of them. Here is an extract from their conversation but you will need to imagine the tasting and grimacing that accompanies it:




```
>>Go through the file, finding an
object or adding a new object.

Is there a ridge or keel on the back of
the slug?   Yes
Does the keel go all the way from tip
to mantle?   Yes
Does it curl round when it's
resting?   Yes


At an object:

Is it 'BUDAPEST SLUG'?   Yes
```

Figure 3 A section of a file as seen with *SEEK*

Figure 4 A section of the same file as seen with *THINK*

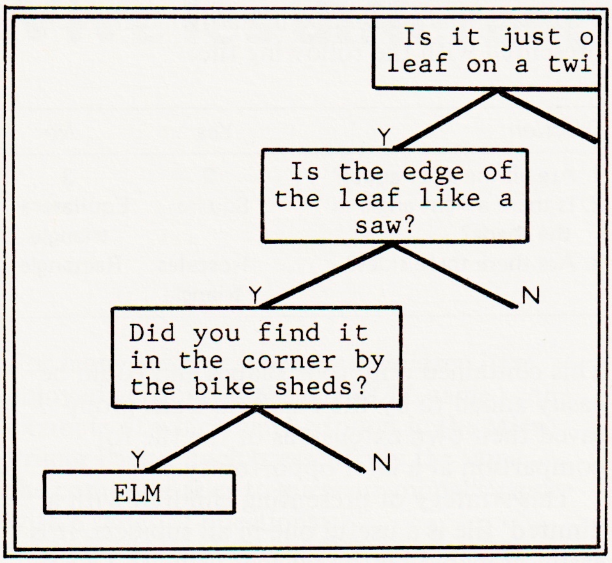


Figure 5

Pupil 1: I think we should start with the Smoked Gruyere first, it's softer and paler than the others.
Pupil 2: But this (Leicester) is also different – it's bright orange isn't it.
Pupil 3: Hey look at this – doesn't it crumble (Cheshire).
Pupil 1: My mum always buys Cheddar – look you can cut it for toast and it grates.
Pupil 4: Ugh!
Pupil 5: What's up with you?
Pupil 4: Ugh – it's that! (Danish Blue) I'll have to go and get a drink!
Pupil 2: I told you it was strong. We have it at home.
Pupil 4: It's mouldy!
Pupil 2: No it's not, it's good for you. You get it (blue veining) in other cheeses too like Stilton.
Pupil 3: Well what shall we start with?
Pupil 2: Let's take Leicester and Smoked Gruyere (type them in).
Computer: A question to give the difference?
Pupil 2: What shall we say?
Pupil 1: Is it soft and light coloured? (Yes for Smoked Gruyere).
Pupil 2: Is it orange? (Yes for Leicester).
Pupil 4: That's carrot coloured – Is it carrot coloured? (Yes for Leicester).'
Pupil 3: (Tasting) Does it taste like 'smokey bacon' crisps?
Pupil 1: No – its my turn – I'm going to have 'Is it soft and light coloured? (types in) . . . Yes for Smoked Gruyere, No for Leicester. OK?
Pupil 2: OK – now my turn.

This facility to use first-hand material is a very important advantage of *SEEK/THINK* over *ANIMAL*. It is a strategy that can occur across the curriculum – leaves in science, different types of clothing for project work, sample of papers in art and craft, and so on. Experience has also shown that older children are happier sorting concrete materials into groups with a computer than without one. Thus the programs can help make an important learning activity intellectually acceptable to a wider age range.

When children are extending files or creating ones of their own it is necessary for them to have a record of their results. Both programs offer this facility. One teacher commented:

> 'We are always being told to do discussion work and 'thinking' exercises, but the trouble is that we have nothing to show for it in the end. With *SEEK* we can get the children arguing and discussing differences. They get such a lot of good language practice and, when they have finished, I have something to show the headmaster and the parents. The children too get a change from 'having to write about it'! This is particularly good for the slow children. They can do the thinking without being inhibited by worries over writing, spelling, the things they find difficult!'

The print-out, in fact, is presented as a biological key. You can see this actual print-out in Fig. 6 (the results of a group sorting out clothes).

Current Key

Question	*Yes*	*No*
1. Has it a collar?	Shirt	2
2. Is it worn on the legs?	3	4
3. Are they worn on the feet?	Socks	Trousers
4. Is it worn on the head?	Hat	Jumper

Figure 6

A print-out of results may also be given with *SEEK*, in 'code'. This is shown in Fig. 7. The children have sorted colours. Can you discover which is which?

This helps children both to understand and to read keys as well as making them aware of the importance of asking 'good' questions in classification tasks. Indeed, research has shown that by using the *SEEK* printouts children learn to read and write keys of their own more easily than by other methods[4].

As a final word on children's files it should be said that many teachers have found *THINK* a better program than *SEEK* for this purpose as it throws the user back to the first question on each occasion. This does not happen with *SEEK*.

As well as the guessing game and the facilities to create original files, *SEEK* and *THINK* also offer a 'telling' option, i.e. where the program

Current Key for Colours

In this key are purple (A), black (B), red (C), silver (D), pink (E), gold (F), yellow (G), light green (H), blue (I) and dark greeny/brown (J)

Questions	*Yes*	*No*
1. Is it a dark colour?	2	5
2. Can you mix colours to make it?	4	3
3. is it the colour of blood?	C	I
4. Is it the colour of the sky?	B	9
5. Is it the colour of a face?	E	6
6. Is it expensive?	7	8
7. Is it worth about £60 an ounce?	F	D
8. If you mix it with red does it make orange?	G	H
9. Does yellow and blue make it?	J	A

Figure 7

Editor's interpretation: Question 2 implies the three primary colours and it looks as though a storm is approaching!

says 'It *is* terylene', 'It *is* a Yellow-Soled Slug.' This enables 'identification' of objects by the computer – an activity that is very different from the ones already discussed. Several files illustrating this use are published with the program. Fig. 8 shows a section of one, *POWDERS*, as it actually appears on the screen.

Here children are given unnamed powders and apparatus by the teacher. The computer then guides them, through experiments, to their names. 'Identification' is also possible in many other areas of the curriculum, and files of leaves,

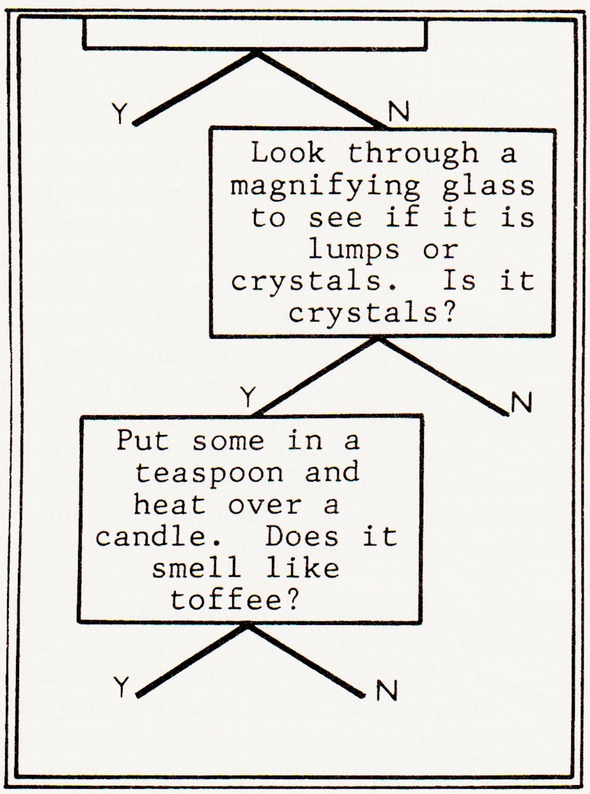


Figure 8

slugs, animals, rocks and polygons are published with the programs for such a purpose. Similar ones can easily be created by teachers themselves.

The 'Telling Mode' of the programs can also be used in other ways. Diagnosis, as described earlier, is an obvious use for teachers and pupils to explore. Travel (Fig. 9) was suggested by A. Paddle[5] and could create a lot of fun if used initially as a file to be added to and then as a means for other children, parents or teachers to choose a holiday.

Question	*Yes*	*No*
1. Do you want to go abroad?	2	3
2. Do foreign languages bother you?	Miami	Paris
3. Do you want to be near the sea?	Bognor	London

Figure 9

One use that the 'Telling Mode' can be put to is for allowing children to browse for information e.g. 'Find all the powders in the file that are crystals'; or 'Find all the slugs that have a keel.' With *SEEK* such information is found, literally, by browsing around the tree using the Y (Yes), N (No) or B (Go Back) options.

With *THINK* the information cannot be gained by browsing. Thus other facilities have been included. By using them pupils can not only list the objects and questions in a file but also the list of objects satisfying a positive or negative response to a question. Such uses for 'information retrieval' contrast well with other available systems. The 'Telling Mode', however, can also be used with the 'wait' facility in the program, as a game of logic. The teacher thinks of an object on the file, e.g. a leaf, and answers the computer's questions, e.g. 'Yes it is a simple leaf; it does have a serrated edge and it does have one side of it attached lower down the stalk than the other' – and so on. When the computer comes to the object it 'waits' so that the teacher can re-cap and the children guess what it might be. On re-pressing the final button the computer reveals the answer and further discussion can ensue as to whether the guesses by the children were, in the circumstances, reasonable ones.

Finally, in using only Y, N or B buttons, *SEEK* in the 'Telling Mode' can be used with a modified keyboard for young or handicapped users. One design is described in the program's supporting handbook.[6]

THINK and *SEEK*, therefore, provide a binary structure into which any information may be placed provided the user can respond to it with 'Yes" or 'No'. Because of the easy input and editing facilities, they can allow teachers and children to produce their own materials without any knowledge of programming. During development of this software we have been amazed by the wide range of uses for which it has been suggested e.g. 'If you listed your good qualities would it be longer than your list of bad ones?' is one question from a health education file; 'You see a large wardrobe in front of you. Are you going to open the door?' is part of a *THINK/SEEK* adventure – most adventures, after all, tend to adopt a binary structure. Other files have ranged from 'prediction' exercises in language to a simulation of the Battle of Waterloo with supporting maps. These are described in the book that accompanies the programs[6]. It is obvious, however, that uses for *SEEK* and *THINK* still remain unfound. Perhaps you or your children can discover one of them for yourself!

Jan Stewart
ITMA

References and Notes

1. The ITMA Collaboration can be contacted at either: The College of St Mark and St John, Derriford Road, Plymouth, (0752-784619) or The Shell Centre, Nottingham University, Nottingham (0606-56101).
2. *SEEK, THINK* and an 'input' program *INTREE* form Module 4 of the course Micros in the Primary Classroom, published for ITMA by Longmans.
3. Quoted from Chandler, D. (ed) (1983) *Exploring English with Microcomputers* C.E.T. (for MEP), pp. 64–65.
4. Stewart, J.G. (1981) *The Potential of the Microcomputer in Aiding the Teaching of Primary Science* Unpublished M. Phil. thesis – Nottingham University.
5. A. Paddle lectures at South Thames College, Wandsworth High Street, London SW18 2PP.
6. Stewart, J.G. (1983) *SEEK and THINK in the Classroom; Module 4, Micros in the Primary Classroom,* Longmans.

Two approaches to sorting information

In this article, I would like to explore two different approaches to handling data that I have used with lower junior children. The first involves a short BASIC program used in conjunction with punched cards; the second a hierarchical database. As I hope to show, the former method of handling data is by far the more suitable for use with young juniors.

My interest in exploring the possibilities of data handling stemmed from several concerns. First of all, I was keen to show the children how a micro could be of assistance in the adult world – I did not want the children to think that micros were limited to playing games. To this end I wanted to introduce the concept of data handling to my class. However, I was not particularly happy about using the program *FACTFILE* which had arrived as part of the MEP software package. Firstly it could deal with only a limited number of records and I was looking for a means of handling data that could be expanded if necessary. Secondly I had not found *FACTFILE* to be a particularly straightforward program to use – it was not always clear to me which of the many options in *FACTFILE* I should choose and I thought the language might prove a barrier in the children's attempts to understand what was going on when they were using the program. Thirdly I wanted to find an opportunity for the micro to make a real contribution to a school topic rather than merely engaging in data handling for the sake of it.

During the Summer Term 1983, my class of first and second-year juniors was engaged in a project on the Romans. One day a class discussion arose about Julius Caesar's assassination and we considered not only how the Romans might have tracked down 'Brutus & Co.' but also how nowadays modern technology would be used in a similar situation to home in on the villains.

This reminded me of an article in the Reader of the Micro Primer package by Frank Gregory in which he recounts his experiences of handling data by using a short BASIC program and punched cards. Something on the lines of his work seemed to be a good starting point for my exploration into data handling.

I thought I would base the work on an imaginary assassination of Caesar with the culprits being children in my class. The first step was to ask the children to devise Roman *alter egos* for themselves. For this they had not only to invent Roman names but also to decide upon physical characteristics, clothes worn and so on. Ultimately we arrived at a final list of characteristics for each 'Roman' which would ultimately form the fields for the data handling exercise:

A: was the assassin a man?
B: was he/she tall?
C: did he/she have brown hair?
D: did the murderer drive a chariot?
E: was the assassin wearing a red toga?
F: did he/she have a gold brooch?
G: was the murderer carrying a sword?
H: did he/she have a scar?
I: or a tattoo?
J: and, was he/she seen at the Forum on the Ides of March?

The children wrote brief descriptions of themselves based on these fields: e.g. 'Yes, I am a man; No, I am not tall; Yes, I have got brown hair . . . etc.'

I then demonstrated how difficult it was to sort out information just by reading it in the form of straightforward descriptions. I wrote information about several 'Romans' on the blackboard in the form of long sentences. I then asked questions like, 'Who has got brown hair, was not wearing a red toga and was carrying a sword'; The children were eventually able to work out the answers but they found it hard to remember and piece together all the different bits of information they read.

Next I showed the children how they could use punched cards to search information. They had never used them before but I dived in at the deep end and got the children to transfer the information about themselves onto the punched cards. For example, if the children had answered 'Yes' to a question about their 'Roman' selves then they cut away the hole at the edge of the card; if 'No' then the hole was left intact. Thus on the card shown (Fig. 1), Fabius was a man (A=Yes), was tall (B=Yes), did not have brown hair (C=No), did not drive a chariot (D=No), was wearing a red toga (E=Yes), did not

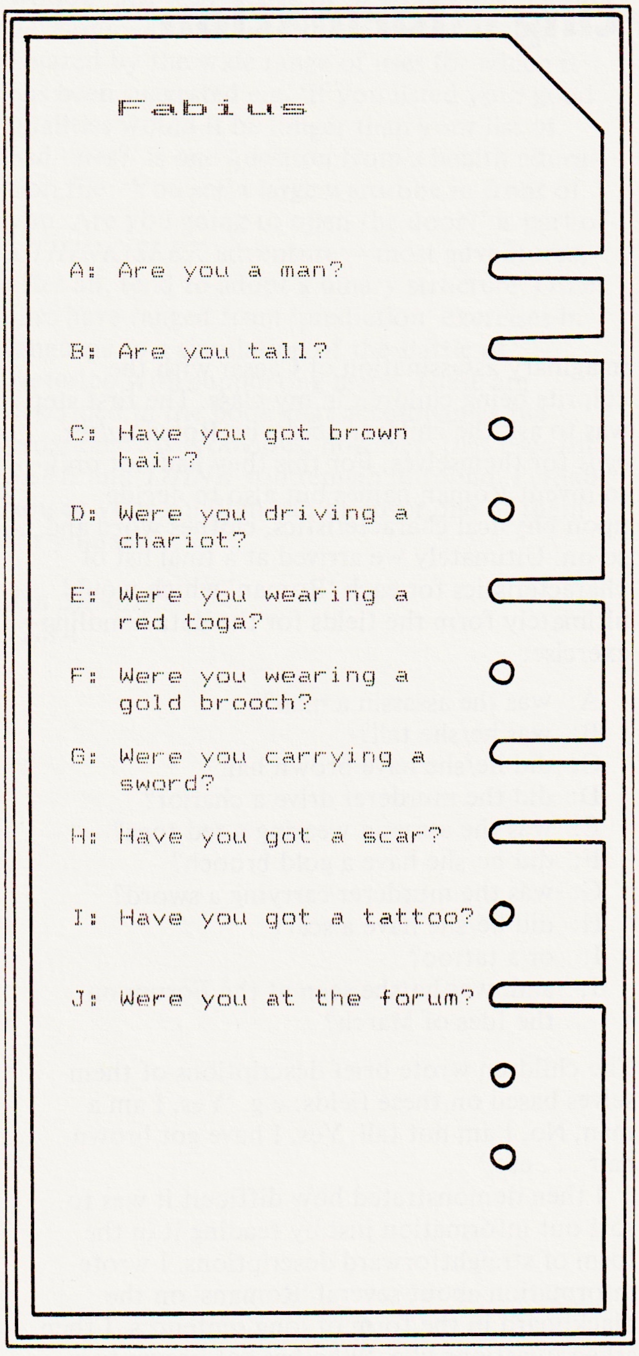


Figure 1

have a gold brooch (F=No), was carrying a sword (G=Yes), did have a scar (H=Yes) but not a tattoo (I=No) and was seen at the Forum (J=Yes).

Then I demonstrated some simple searches using the cards. For example, to separate the cards of the 'Roman men' from the 'Roman women' a plastic needle would be inserted through the stack of cards at question A and those cards with a slot cut there (i.e. A = 'Yes, I am a man') would fall out thus leaving the remainder of the punched cards skewered on the needle. This pile of cards could then be searched to determine, for example, which of the 'Roman men' had a scar, carried a sword and so on.

I asked groups of children to use the cards to solve simple problems I set them (e.g. 'Find out those Roman women with brown hair and a gold brooch') and also to devise simple problems for each other.

It was only when I felt the children could use the cards competently that I introduced them to the BASIC program I was going to use which simulated the card-sorts:

```
10 CLS
20 FOR suspect = 1 to 23
30 READ name$ A,B,C,D,E,F,G,H,I,J
40 IF A = 1 AND C = 0 THEN PRINT name$
50 NEXT suspect
60 DATA Celia, 0,0,1,1,0,0,0,1,0,1
70 DATA Billius, 1,0,0,0,1,1,1,0,1,0
80 DATA Silvia, 0,1,1,0,0,1,1,0,0,0
```

The hardest part of the work for me was trying to explain in as simple yet coherent a fashion as possible what the program did line by line:

10 CLS . . . this clears the screen;

20 FOR suspect = 1 to 23 . . . this, together with Line 50 tells the computer to read through 23 lines of data;

30 READ name$, A,B,C,D,E,F,G,H,I,J . . . this tells the micro that each line of data will contain a name and the answers to ten questions;

40 IF A = 1 AND C = 0 THEN PRINT name$ . . . this instructs the micro to print the name of anyone who answered Yes to question A and No to question C. (This line would have to be retyped each time the children wanted to find out something different about the 'Romans' in the data file.);

60 onwards . . . these lines contain the data about the Romans: a name, then a list of 1s and 0s – a 1 representing a Yes, an 0 standing for a No for each of the ten questions.

Initially I typed in some simplified lines of data and showed how the program worked. The children then typed in their own lines of data in pairs, each checking the other's typing to ensure no errors occurred which would make the program crash. Then, as with the punched cards, the children practised altering Line 40 to achieve simple searches.

Now that the children had been introduced to the two methods of searching data I explained the project to them. Caesar had been murdered and two 'Romans' in Class 2, one male, one female, were the culprits. Clues as to their identity would be revealed over the next few days and the class would have to use the punched cards and the BASIC program to discover 'Who done it'. I had already secretly determined who the culprits would be and so I was able to start hiding the clues around the classroom (e.g. 'A reliable witness says that a woman was seen driving a chariot away from the scene of the murder'). Once a clue had been found the children were eager to find out which suspects had now been eliminated. Sometimes the computer was used first, sometimes the punched cards, but in all cases one method was checked against the other. I wanted to show the children that the micro was merely carrying out an operation that they themselves could perform using the punched cards there was no black box mystique about it.

It is worth pointing out that much of the work undertaken by the class on data handling took place away from the micro: it served as a stimulus for many other activities. For example, the children wrote daily Roman newspapers giving the latest details of the hunt for the assassins and demonstrating how the information could be recorded in various forms (e.g. Venn diagrams). But apart from being a fruitful educational experience the project generated enthusiasm and enjoyment amongst my class. Indeed, when the culprits, Billius and Livia, were finally unmasked there was a universal plea that we undertake a similar piece of work again.

During the second half of the Autumn Term 1983 my class was following the BBC schools *Zig Zag* broadcasts on the theme of Detection. I wanted to incorporate the microcomputer into this topic and its ability to analyse data about 'criminals' once more seemed an obvious avenue to explore. But although many of the second-year children in my class were keen to repeat the punched cards exercise described above, I was interested in finding out whether other means of classifying and sorting data would generate an equally successful educational project. In particular I wondered whether I could use an hierarchical data handling program.

By this I am referring to a means of handling data in which each record is differentiated from the preceding one by some distinguishing feature. In form it is akin to the pattern made by the roots of a tree branching out as is illustrated in Figure 2:

In order to interrogate the database it is necessary to work through the various branches in sequence until the desired information is reached. Thus it is unlike databases such as the BASIC program and *FACTFILE* which allow you to go straight to the information you want. So, from a practical point of view, hierarchical databases seem rather limited in their application as they don't really allow a fast access to information. Nevertheless they do encourage you to think how each record is different from the others held on file. For me, then, the educational value in using a hierarchical database lies in its requiring the children to think about the characteristics of the data they type in rather than simply allowing the micro to sort it for them.

Of the hierarchical packages available for me to use at school there was *ANIMAL* and *ANIMAL, VEGETABLE, MINERAL*. I was

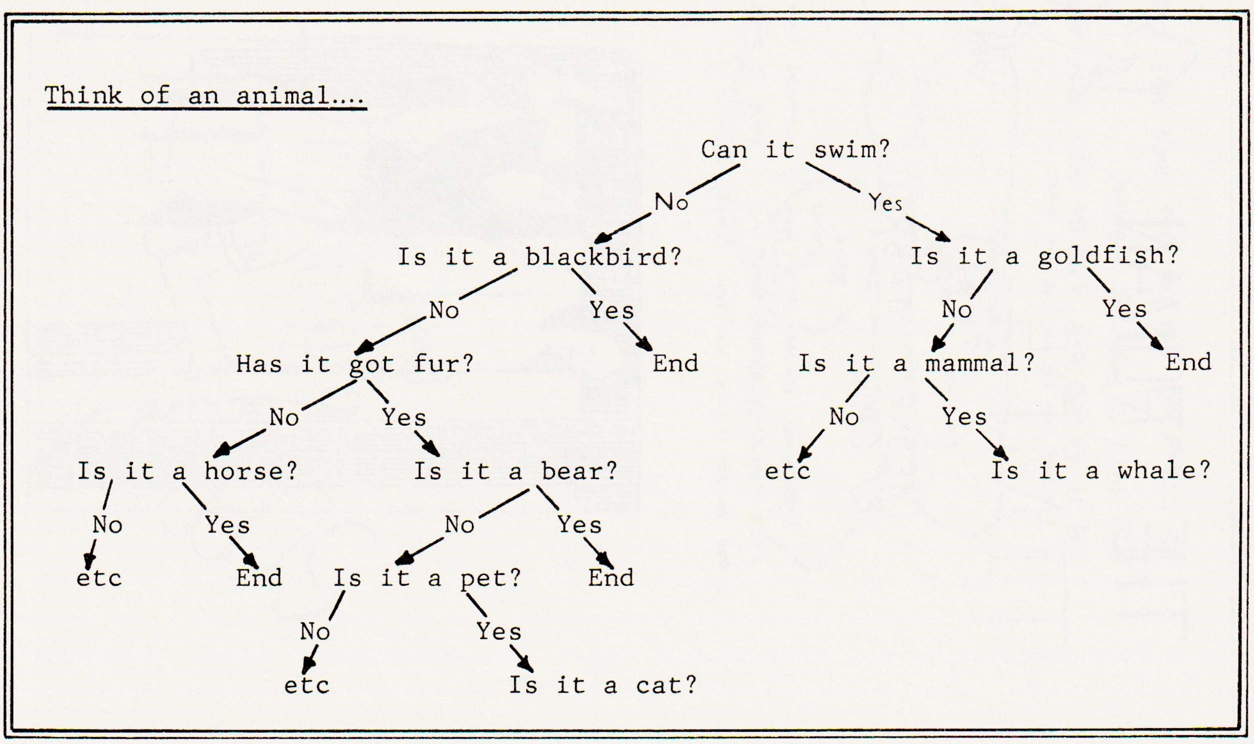


Figure 2

THE ROMAN TIMES

CAESAR STABBED BY GANG

Caesar was stabbed by a gang of murderers While in the senate people gathered around him and a gang of people stabbed him and he fell before his rival Pompey.

Ceaser Murdered

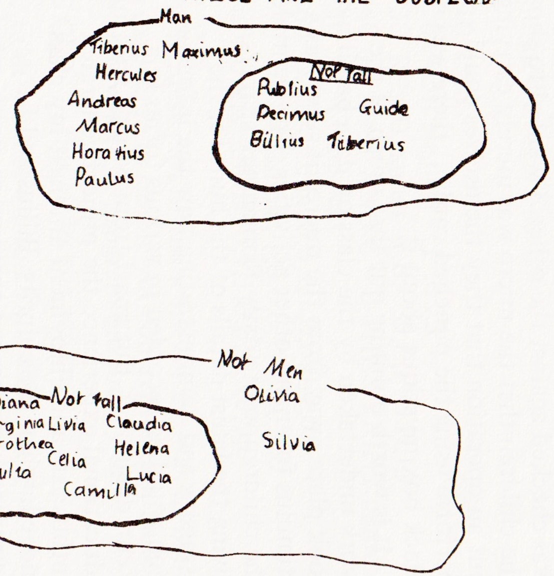
Ceaser was on his way to the senate. He thought some-thing might happen to him. He looked very worried. Then when he was talking to the people of Rome some people came up and started hanging around behind him. Then his best friend Brutus stabbed him in the back.

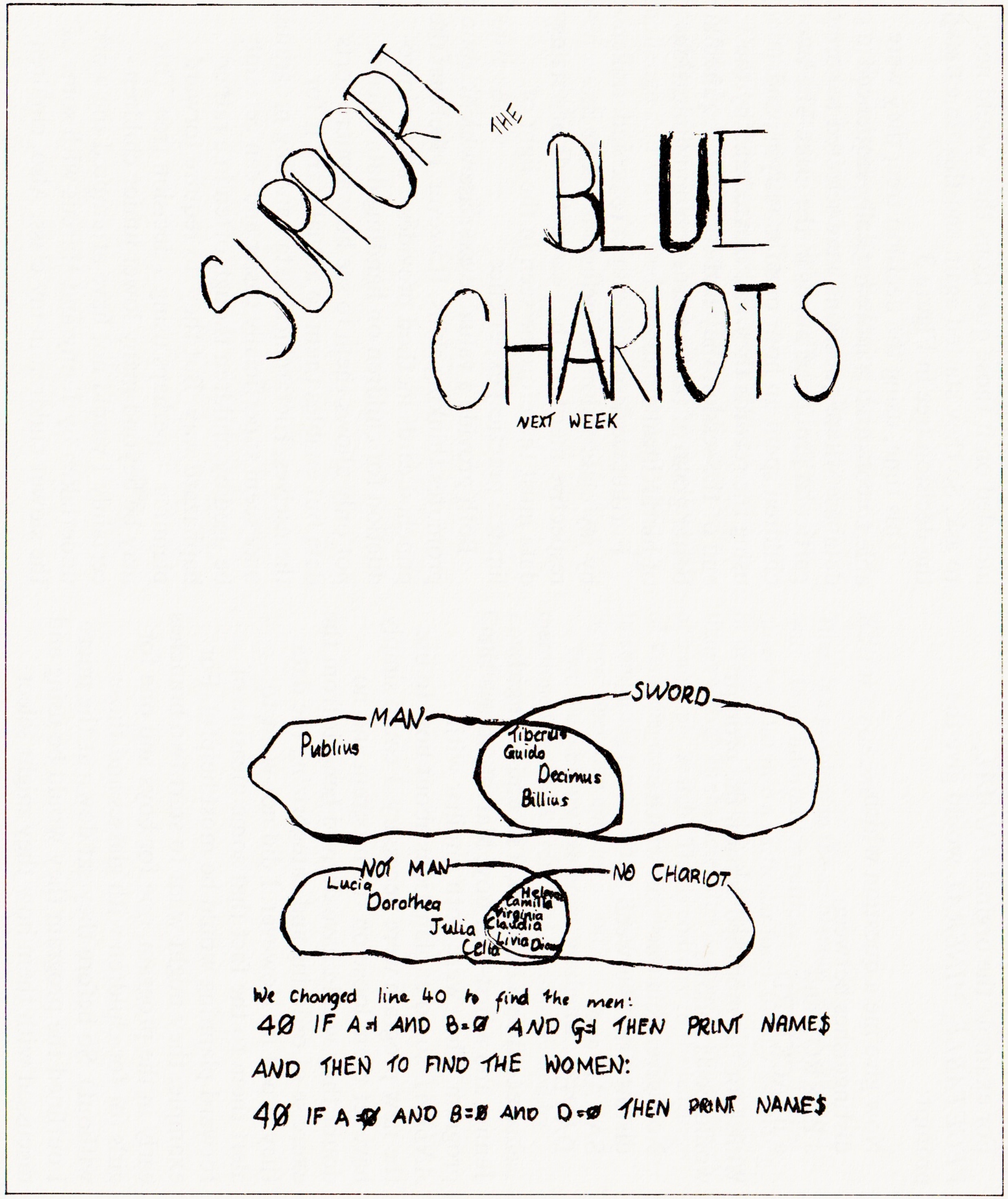
Our reporter has been talking to the people of Rome.

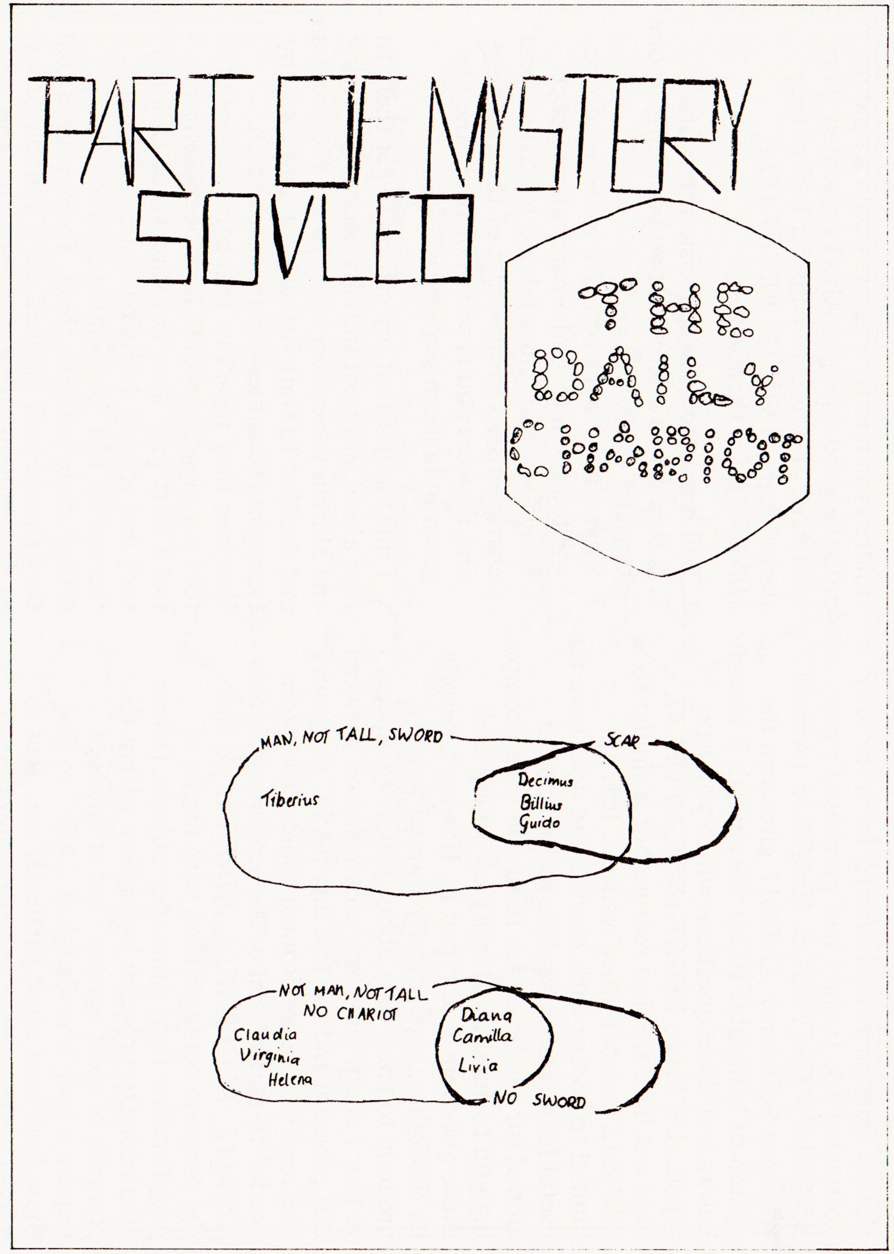
First we talk to Celia. Well hello, I was doing my daily shopping at Tescosus. Tescosus is a lovey shopping centre you must go there sometime. Well I went into the senate for a cupus of teaus. Then eh em I can't remember what happened next. We thankyou Celia.

One of the consuls said that legonaries were looking for a short man and a woman of medium hieght. Centurian Drusus appealed for witnesses to come forward.

THESE ARE THE SUSPECTS.




SUPPORT THE BLUE CHARIOTS
NEXT WEEK
MAN
SWORD
Publius
Tiberius
Guido
Decimus
Billius
NOT MAN
NO CHARIOT
Lucia
Dorothea
Julia
Celia
Helena
Camilla
Virginia
Claudia
Livia
We changed line 40 to find the men:
40 IF A=1 AND B=0 AND G=1 THEN PRINT NAME$
AND THEN TO FIND THE WOMEN:
40 IF A=0 AND B=0 AND D=0 THEN PRINT NAME$


PART OF MYSTERY SOVLED
THE DAILY CHARIOT
MAN, NOT TALL, SWORD
SCAR
Tiberius
Decimus
Billius
Guido
NOT MAN, NOT TALL
NO CHARIOT
Claudia
Virginia
Helena
Diana
Camilla
Livia
NO SWORD

already familiar with *ANIMAL* from the MEP Software Pack 1, and I had purchased a copy of *ANIMAL, VEGETABLE, MINERAL* (Arnold Wheaton Software). *ANIMAL* gives you the option of being able to list the animals it already knows and those questions which you have typed. *ANIMAL, VEGETABLE, MINERAL* does not have this last feature but is similar to *ANIMAL* in most other respects. It is not limited to coping with animals and does have the useful feature of being able to think of an animal (or whatever) for itself and then prompting you to guess its identity by asking back those questions you typed in. However, despite its flexibility, I specifically wanted to use a program to enter data about people ('suspects') so that I could make some comparison between this program and the five line BASIC program I had used during the Roman topic. I therefore needed to edit several of the screen prompts in *ANIMAL, VEGETABLE, MINERAL* so that the program could be used with names of people rather than animals or objects. (Minor alterations to program listings are within the compass of most teachers and it does not require a deep knowledge of BASIC to find those lines that cause particular messages to appear on the screen and then alter them.)

For example the orginal *ANIMAL, VEGETABLE, MINERAL* would give the prompt:

> Now give me a question which
> distinguishes between
> a PARROT and
> a BLACKBIRD

Whilst for the modified version the prompt would be as follows:

> Now give me a question which
> distinguishes between
> SARAH and
> DAVID

I selected two children, a second-year girl, Jenny, and a first-year boy, Marcus, to use the program. They were both familiar with *ANIMAL* so no explanations about how to use the new program were necessary. I could simply have let them loose on the program and no doubt they would have derived benefit from the exercise of ordering and categorizing the data they typed in. However, I did not want to alert them to the fact that some measure of forward planning would be most helpful. For example, they might want to start two branches early in the program, one for boys and one for girls (or for children with glasses and those without). So before they sat down at the micro I outlined the program they would be using and discussed with them how they might select children's names to type in: would a random order be good enough? Also I presented them with a list of points to consider before they started using *ANIMAL, VEGETABLE, MINERAL* for themselves:

1. Will it help to draw a decision tree first?
2. Is it easier to start with two boys/girls or one boy and one girl?
3. What clues would you give someone if you wanted them to find a particular suspect? Would you give clues related to that suspect's characteristics in the order of the questions and answers you typed in or could they be given in random sequences?

I must admit that my efforts to get them to plan ahead were not initially successful. Jenny and Marcus were most eager to use the program and decided to start virtually straight away by typing in the names of various children. However they soon saw that some degree of forward planning would have been advisable. For a start they were not sure which children they had already included and then found themselves lost for suitable questions to differentiate between them. They realized that they should have planned out a suitable order of the children, using a decision tree and included on it those questions they would need to ask. So they started again and this time drew the decision tree in Figure 3.

This time, using the decision tree, they were able to construct a more logically sequenced database where questions were selected to sort particular groups of pupils in the class. The two children spent an hour or so modifying and using the records they had created and by the end of the session they had certainly acquired a better grasp of the construction and limitations of hierarchical databases.

Further work on this has yet to be undertaken by my class and my conclusions about the respective merits of the two methods of sorting data must therefore be seen in the light of limited practical experience.

Both provide a valuable exercise which promotes thinking skills. However, I feel that the punched cards method provides a better introduction for children on handling data since it not only allows them to see how a micro sorts data but enables them to actually try it for themselves. Hierarchical data handling programs may seem superficially easier and can certainly be used by children though often in a rather haphazard way. But they do require forward planning – perhaps using a decision tree. This may be beyond many lower junior children – certainly I would not have attempted the work undertaken by Jenny and Marcus with some of the slower children in my class. Also, children

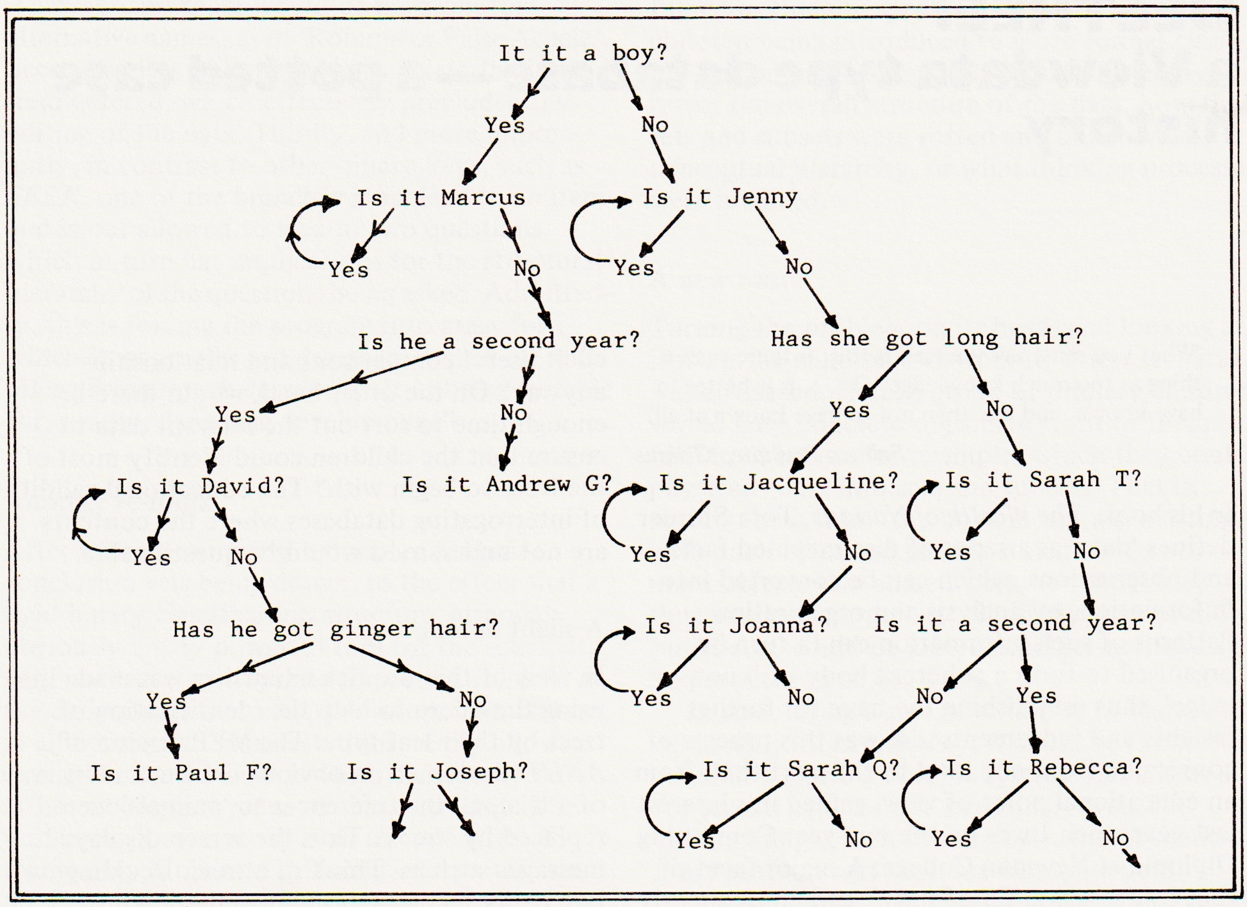


Figure 3 Suspects.

do make mistakes when entering data at the computer keyboard and it is not always easy to make alterations without disrupting the hierarchy of information.

Despite these caveats, programs like *ANIMAL, VEGETABLE, MINERAL* can provide junior children with a testing intellectual challenge and when used to complement the punched cards approach will provide many avenues of exploration for them in the field of processing information.

Charles Bake
Woodside Junior School
L B Croydon

Editor's comment

There is a problem in the ambiguous use of the word 'sort'. In an information retrieval sense, 'sort' implies ordering a list into a particular form, usually ascending or descending. The idea of producing a set of people/things with given characteristics is either a simple 'search' or a multiple 'search'. This is contrary to the language of Venn diagrams where we talk about 'sorting into sets' In computing terms finding the elements of a set is a 'search' operation and not a 'sort'.

Two more binary tree programs similar to those mentioned are *SEEK* from ITMA and published by Longmans and *DELTA* from Inner London Educational Computer Centre.

DEETREE: a Viewdata type database — a potted case history

> 'What you must ask yourself is this: is there such a thing as too much knowledge? Or . . . is it better to have known, and die, than not to have known at all?'
>
> *Salman Rushdie, 'Grimus'*

In his book, *The Wealth of Nations*, Tom Stonier defines 'data' as a series of disconnected facts and observations, which can be converted into 'information' by analysis and organisation. Patterns of such information can in turn be organised to form a coherent body of 'knowledge', thus establishing the basis for further insights and judgements[1]. It was this process of conversion from one level to the next that, from an educational point of view, gained my interest last year when I was on the one-year Computing Diploma at Newman College. A major facet of that course was a 'Case Study', and consequently I was able to follow up such an interest.

First steps

It had become quite obvious very early on in the course that Information Handling programs would bring a completely new dimension, and probably a completely new set of skills, into primary classrooms. Initially an investigation was planned to evaluate just one aspect, viz. how effectively junior children were able to make use of information retrieval packages. Children were to have access to ready-made databases about some fifty common trees, but with two objectives:

1. To be able to identify a tree from the careful observation of certain characteristics and, more importantly,
2. to begin to classify trees according to relationships they themselves had found.

At that time, there were no such databases available, to my knowledge, so after a visit to Kew Gardens and a quick survey of various reference books, a start was made. However, having only a layman's knowledge of trees, various doubts soon arose. On the one hand, the experts themselves often did not agree on identification 'keys', terminology or even the botanical classifications. How easily then would other teachers or children be able to add their own data? How many trees would be needed for children to elicit useful comparisons and relationships anyway? On the other hand, would there be enough time to sort out the relevant data to ensure that the children could identify most of the trees to begin with? The educational validity of interrogating databases where the contents are not understood would be questionable.

A slight detour

In view of this, a quick excursion was made into using the micro to help the identification of trees by their leaf-type. The MEP version of *ANIMAL* seemed an obvious choice, and by way of a trial, all the references to 'animals' were replaced by 'trees'. Thus the screen displayed messages such as 'Think of a tree'. Deciding on two easily recognisable and well-known trees as starting points for a binary classification structure was a slight problem. A small, random sample of (admittedly urban) ten year olds thought that acorn and conker trees would be best known by their peers! For the botanist, however, the first division for all flowering plants tends to be that of the two great classes 'gymnospermae' and 'angiospermae' which include 'conifers' and 'broadleaf trees' respectively. Moreover, since the great majority of gymnospermae have needle-shaped leaves, this fact invariably supplies the starting point for a binary key.

Eventually a binary database of some twenty trees was built up, beginning with a fir tree and an oak tree. Hence the first question was 'Are the leaves like needles?'. A 'yes' led to various conifers, whereas a 'no' guided the user, via the question 'Are the leaves simple?' to various broadleaf trees with simple or compound leaves.

After a few classroom trials, it was pretty clear that *ANIMAL* (or *'TREE'*) can provide a stimulating, dynamic opportunity for co-operative thinking skills and what has been termed 'creative problem solving', particularly when building a datafile. On the other hand, it does have some limitations when being used as a database for information retrieval. Firstly, the input is restricted to lower case letters and to a maximum of twenty-two characters, though this is not insuperable. Trees, however, are normally treated as proper nouns, eg. Field Maple or Red Oak or whatever and there would be no room for alternative names, as in 'Robinia or False Acacia'. Secondly, it is only possible to delete the last item entered, which effectively precludes any editing of the data. Thirdly, and more importantly, in contrast to other binary keys, such as *SEEK*, one of the branches must lead to an item and is not allowed to lead to two questions, which in turn has implications for the structural hierarchy of the questions being asked. Admittedly, this is forcing the program into areas for which it was really not designed – after all, the Micro Primer pack categorises it as being suitable for, simply, 'language development'!

Another re-think

After a 'brain-storming' tutorial, a tentative conclusion was being drawn, to the effect that a rigid binary classification structure, although obviously a very powerful tool for the scientist, may not be so appropriate for young children at the early stages of concept development. Nature, in all her myriad manifestations, rarely allows a straightforward black-or-white dichotomy but rather a complete range of shades of grey. The terminology evolved by man to cope with all this variety is open to differing interpretations, too, and exact scientific nomenclature can sometimes be too restrictive and inappropriate. For children of this age, categories tend to be comparative, rather in the sense that size and colours are. A 'narrow' leaf, for instance, would need to be seen to be narrow when compared to another leaf. (And is that 'narrow' leaf a typical example of its type anyway?) It might be more helpful to know what the available options are, rather than a strict 'Is it narrow? Yes or no'. Furthermore, a false decision on such a binary option can quickly lead down the wrong route, and whereas in a book it may be easy to glance over the next page and follow a parallel route, with the micro in its present form the user might well have to start all over again.

Above all, there were doubts as to how helpful, educationally speaking, a binary key is for children being introduced to more formal classification and categorisation. It does not easily reveal the overall structure of the data, how the sets and subsets were sorted and under what conceptual hierarchy, or what thinking processes were involved.

A new route

Turning the problem on its head, and looking at it from the children's viewpoint, a new scenario was envisaged. Here, a group of primary children would have concrete objects in front of them (different leaves, for example) which they could physically sort into sets and subsets. That is, sorting them from 'the bottom up', i.e. starting with one object and placing it next to one or more objects that share a common attribute, thus making a 'super-set'. Similarly, two or more 'super-sets' could be joined together to make a 'hyper-set' and so on.

In structuring a sort of data-tree or hierarchy of classifications this way, it could be far more valuable to have a flexible system that allowed for several options at a particular level. The children might feel, for example, that it made more sense to sort their leaves at one point according to whether they were lobed or prickly or toothed or smooth-edged. Such a method would have, hopefully, the advantage of being able to present the overall system of classification in a more accessible and intelligible manner than in following various binary decisions.

Prestel's Local Viewdata system was an obvious candidate, but as it was not readily available to many primary schools at the time, it was agreed that a similar 'multi-branching' program should be written and evaluated. The data could be organised and structured along the same sort of lines, with 'page 5', for example, offering three choices leading to 'pages 51, 52 and 53'. In the same way, five options on 'page 53' might lead to 'pages 531, 532, 533, 534 or 535' (Fig. 1).

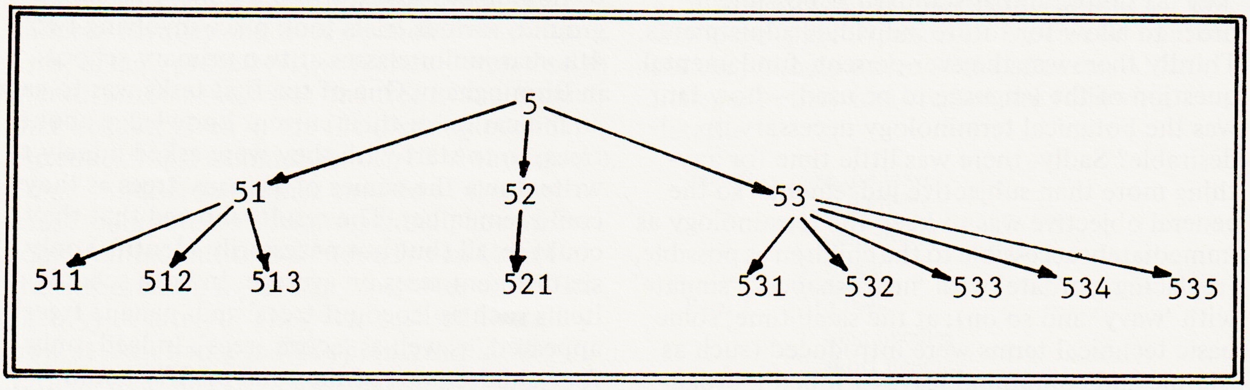


Figure 1

Using such a system, each page (or screen) of data could offer routings to another ten or so, each of which could lead to another ten pages if required. Although this method would only allow data to be 'searched through' rather than interrogated or queried, it was hoped that the new flexibility would help to bridge the gap between programs such as *ANIMAL* on the one hand, and *PQUERY* on the other.

Creating a Viewdata Database

Some time later, a program using the teletext mode of the BBC Model B had been written entitled *DEETREE*, in which the number of options on each page was limited arbitrarily to a maximum of ten. The name itself was really an amalgam of various connotations: the '-tree' part alluding to the multi-branching aspect, and 'Dee-' suggesting an abbreviation of 'data' at the same time as recalling its 'de-nary' rather than 'bi-nary' nature. In any case, it seemed to encapsulate the idea of data organised in a tree-like structure.

The Database itself
Bearing in mind that the initial users would be city children living in a mainly man-ordered if not man-made environment, there was quite a problem in deciding which trees were likely to be most common within the locality of each school. Luckily, however, help was at hand from an expert at the local Teachers' Centre, and eventually a database of over fifty trees was created.

It was also decided to restrict the data to leaf characteristics as the means by which the children were to identify the trees. These details were entered into a *QUEST* file to aid the initial sorting, and also copied onto individual index cards so that the 'data-tree' could be physically laid out on the floor, altered, re-written and refined before being committed to *DEETREE*.

At this stage, three aspects appeared to be crucial. Firstly the aim was to try to ask the right question at the right time and place in the structure, and secondly to keep the botanical 'key' as simple and as standard as possible in order to allow for future individual adjustments. Thirdly there was the ever-present, fundamental question of the language to be used – how far was the botanical terminology necessary or desirable? Sadly, there was little time for anything more than subjective judgements, so the general objective was to keep the terminology as immediately accessible to the children as possible, (replacing 'cordate' with 'heart-shaped', 'sinuate' with 'wavy' and so on); at the same time, some basic technical terms were introduced (such as 'simple and compound leaves', 'lobed', 'broadleaf' etc.) which could be built upon later.

The eventual datafile consisted of over seventy 'pages', but every tree could be reached within a maximum of six decision points, each level in turn narrowing down the possibilities available. A typical search might look like this:

Page 0 Are the leaves:
1 like needles,
2 like scales
3 or like neither – ie broadleaf?
* select number * –

Pressing 3 for the broadleaf option, we would get:

Page 3 Are the leaves:
1 simple
2 or compound?

Assuming we knew the difference, a 1 would now lead to:

Page 31 Are the leaves more or less:
1 lobed
2 heart-shaped
3 triangular or round
4 long
5 or oval?

If, perhaps more by default than anything, the leaf we have in front of us is best described by number 4, we would have:

Page 314 Is the margin:
1 toothed
2 or spiky?

Closer inspection does reveal that the leaf has pointed spikes along the edges so, by pressing number 2 we would arrive at:

Page 3142 It might be –
Sweet Chestnut.

At this point, reference to a suitable book would hopefully confirm the answer.

Classroom Trials

By this time, we were approaching the final three weeks of the school year with all those concurrent activities ranging from school trips to tidying out stockrooms. Against this background, various trials took place involving two 4th year junior classes at two primary schools in Birmingham. One of the first tasks was to gain an indication of their current knowledge about trees, so to start with they were asked merely to write down the names of as many trees as they could remember. The results showed that they could recall (but not necessarily identify) only six different trees on average. In both schools, items such as 'coconut trees' and 'banana trees' appeared, as well as 'acorn trees'. Indeed, only four children out of the total of fifty could name more than ten, the highest in fact being fourteen. The most well-known was the Oak, with Apple second and Willow third. Next came Sycamore and Beech, followed by Ash and, surprisingly, Pear. Bearing in mind the unscientific nature of these statistics, however, they must obviously remain only 'interesting' and by no means 'significant'.

During the following sessions there were discussions on trees in general, their uses, their different parts and distinctive features. A selection of very dissimilar leaves was brought in to encourage not only more careful observation as to any similarities or differences but also the use of more accurate language, including some technical terminology where relevant. The trees on the school grounds were mapped and leaf samples collected for closer inspection. Back in the classroom the children seemed to find it quite hard to separate the leaves into sets and subsets, either because they had not had enough time to assimilate the salient features or because they were not accustomed to sorting and classifying. The exercise did, however, give rise to some very useful discussion.

On one occasion, three young seedlings were brought in, two of which were named (Oak and Beech) but the third had no such label and needed to be identified. In general the children followed a rather haphazard approach, flicking through the books at random. (The database on Trees had not yet been introduced.) However, that in itself did highlight some of the problems involved. Even if a book had some sort of index, it was of little value unless you had some name to go on or some idea of what the tree might be. The books themselves either had no key at all or one that was too technical and complicated, (with the exception, that is, of the *Oxford Clue Book on Trees*[2] which very much helped to fill the gap). The terminology used, the lack of good photographs or drawings, as well as the typicality of the actual specimens, could all set up barriers to easy identification. However, more by luck than by judgement, the remaining seedling was correctly identified as a Sweet Chestnut.

At this juncture, *DEETREE* with its large datafile on Trees was introduced to help the children identify the trees on their own campus. This, in fact, proved very successful, even though at both schools there was one tree in the grounds that was not given in the file (bearing out that old educational axiom that 'there's always one!'). Some children felt rather cheated at this because 'computers know everything', but others soon pointed out that somebody had to put the information in, and any mistakes could (in theory) be altered. They also realised that they now at least had a starting point, in that if various trees had been eliminated, they could make reasoned guesses at a solution of the unknown. In other words, a micro may not supply the answer required but can, nevertheless, still be a helpful and valuable tool in reaching that answer.

It was also interesting to note, as has been stated elsewhere, how much the micro can motivate the less able children; in this case, to read aloud along with the others and soon get to know the vocabulary concerned. Not that it is always plain sailing by any means, and to some children the terminology is not as immediately clear as one might expect. For one group, anything sharp and pointed was 'needle-like', so since Holly leaves often have stiff prickly points they first tried to follow the 'needle-like' route – obviously unsuccessfully.

On the whole, the various groups of children were able to identify the trees on their respective grounds quite speedily, and the maps were detailed accordingly.

Sometimes the *Clue Book* was used alongside *DEETREE* and the datafile but it did not seem to be as motivating or successful by comparison. With the book, the children sometimes appeared to get stuck, possibly not being so conversant with the format or the language, whereas the more limited and repetitious terminology of *DEETREE* guided them through almost automatically to a possible solution. Nonetheless, the *Clue Book* remains by far the most useful and attractive book available, with its clarity of graphics and text, and indeed it is apparently the only one that attempts a botanical key specifically for primary children (and which, incidentally, is also not binary in its structure!)

Two children's files

A parallel aim of this study was to investigate how children might set up their own databases, and one such attempt by an able group of five resulted in the following structure. Eleven leaf samples had been gathered and grouped according to three main criteria: whether they were 'prickly', 'compound' or, to use their word, 'single'. They seemed quite quickly to get the idea that these questions had to appear on the first page (page 0), and that by careful numbering they should arrive at the answers. After a little help, they had created the following datafile:

Page 0	Are the leaves 1 prickly 2 compound 3 single?
Page 1	Are they 1 long needles 2 short needles 3 scaly and small 4 or with sharp prickles?

Page 11 It might be: Scots Pine
Page 12 It might be: Norway Spruce
Page 13 It might be: Cypress
Page 14 It might be: Holly

Page 2 Are the leaves
1 long and fat
2 slim with large teeth
3 slim with no teeth?

Page 21 It might be: Horse Chestnut
Page 22 It might be: Rowan
Page 23 It might be: Robinia

Page 3 Are the leaves
1 lobed
2 finger shaped
3 hairy
4 or small?

Page 31 It might be: Oak
Page 32 It might be: Sycamore
Page 33 It might be: Hazel
Page 34 It might be: Silver Birch

Another, not very able group took well over two hours with quite a lot of help to classify the twenty different modes of transport they had thought of. These were written down on small pieces of card and placed in various arrangements on the floor. It seemed easier for them to think in terms of similarities rather than differences when grouping the cards into sets and categories. Indeed, very careful attention must be paid to the pre-sorting of both the data and the routes involved. However, two advantages of *DEETREE* did become quickly apparent: firstly in its flexibility and secondly in its ability to be used at the intellectual level of the children. Sometimes, for example, they wanted a binary Yes/No classification, and at other times more options were required. Furthermore, there was a great variety in the language used and the types of features they considered most distinctive. Evidently sound provides the salient distinction between a horse and a donkey, whereas physical attributes decide between a camel and an elephant, and seating capacity between cars and buses. The other children in the class were quite impressed by the speed and accuracy of the finished product, which also incidentally increased that group's confidence and self-esteem.

Page 0 Is it
1 natural
2 or man-made

Page 1 Has it got
1 four legs
2 or two

Page 11 Has it got
1 humps
2 a trunk
3 or neither

Page 111 It could be: a camel
Page 112 It could be: an elephant

Page 113 Does it go
1 neigh
2 heehaw

Page 1131 It could be: a horse
Page 1132 It could be: a donkey

Page 12 It could be: man

Page 2 Does it travel
1 on land
2 in the air

Page 21 Does it carry
1 loads
2 or people

Page 211 It could be
1 a van
2 or a lorry

Page 212 Has it got
1 2 wheels
2 or 4
3 or more

Page 2121 Has it a motor
1 yes
2 no

Page 21211 It could be: a motorbike
Page 21212 It could be: a bike

Page 2122 Does it seat more than 20 people
1 yes
2 no

Page 21221 It could be
1 coach
2 bus
3 or a one-deck bus

Page 21222 It could be
1 minibus
2 or a car

Page 2123 It could be: a train

Page 22 Does it go into space
1 yes
2 no

Page 221 It could be
1 a rocket
2 or a spaceship

Page 222 Does it have
1 a motor
2 or hot air

Page 2221 It could be: a plane
Page 2222 It could be: a balloon

Sadly we had no more opportunity to explore the possible differences between a 'rocket' and a 'spaceship', or even to include some sea-going transport.

Looking back

It was very disappointing not to have had more time in the classroom, with children of different ages and abilities working with the program and especially creating their own files. However, there were enough indications that viewdata systems do have a powerful role to play at the primary level in terms of a 'dynamic reference book'. (And linking them to a laser disc system will mean an entirely different ball-game!) They are very productive in the use of the language and the thinking skills involved in the corporate creative problem solving required. They allow the children more opportunity of perceiving the overall structure of an area of organised information, and the manner in which the data has been grouped and re-grouped into its various categories. At the very least, children should gain an insight into the importance of the man-made divisions that the hierarchical strategies use in the classification of knowledge. By bridging the gap between the manipulation of concrete objects into sets and supersets, and the manipulation of abstract categories, the children should hopefully come to an appreciation of the value and of the limitations of man's attempt to impose order on the superabundant vagaries of the world he lives in.

But, on the other hand, if that very brief experience at those two schools is anything to go by, the organisation of the information into a form ready to be used by the micro, invaluable though that may be, did take a great deal of time. There is very little visible output for a large amount of input. Whether in the long run, teachers, and children, will think that the results warrant such an expenditure, is a moot point. One can only hope that, as such systems become more readily available and as the relevant skills and techniques become more commonplace, their educational value will become more greatly appreciated. After all, as Seymour Papert said on 'Talking Turtle':

> 'What's good for thinking is good for thinking – whether we're a five year old or a sophisticated scientist.'

It is beginning to look more and more as though the advent of the micro into primary education is adding greater depth to that old, old maxim:

> 'It's the thought that counts . . .'

Barry Wake
Primary Support Team
Birmingham Educational Computing Centre

References

1. Stonier, T. (1983) *The Wealth of Information.* London, Methuen (p. 19).
2. Allen, G. and Denslow, J. (1970) *The Clue Books – Trees.* Oxford, OUP.

The program *DEETREE* and the accompanying datafiles will be available free to BBC users on MAPE Tape 2 due out in November 1984.

ANIMAL – MEP Micro Primer Software Pack 1, version 1.0 (1982).

PQUERY – Newman College, version 2.1, 480Z (1982).

QUEST – AUCBE, version 1.2C (1983).

PARACHUTE: processing results from primary science experiments on a microcomputer

'What makes a good parachute?' I asked a class of third-year juniors one week. I wanted them firstly to talk about what actually was meant by a successful parachute, and secondly to experiment with various possibilities in parachute construction. I expected them to spend the week making model parachutes with which they could try out hypotheses, test variables, and try to reach some kind of conclusion about what the characteristics of a 'good' parachute were.

Reaching conclusions from experimental data seems to me to be a crucial missing element in otherwise good primary science. It is fairly obvious why this is so: young children's observations are not very exact, so false conclusions may often be drawn if there are only a few experiments. On the other hand, if the number of experiments conducted is sufficient to reduce the effects of false readings, the numbers themselves become too large to handle. The data-processing capabilities of the microcomputer can, I believe, dramatically change this. It is possible for children to make a large number of observations, store them in a data retrieval system and then, confidently, quickly and accurately, test hypotheses about what they have found.

In the classroom there was an RML380Z micro, with twin disk drives and a printer, and a data handling package, micro-*LEEP*. This program was used for the initial collection of results and testing; later we were able to transfer the file to a newer program that we were testing, *DATA PROBE*, which has graphic display features that I believe make it particularly appropriate for the primary school classroom.[1] About half of the children had used data-handling programs before in school.

Our opening discussion was on how a 'good' parachute should perform. There was a little confusion in a few children's minds at first – they saw 'good' in terms of an exciting, dramatically fast fall. However, when I asked them how they would like a parachute strapped to their own backs to perform, we all quickly agreed that what was needed was something that would allow us to land safely, and this meant a slow descent and a soft landing. 'The more gently it goes down, the better,' offered one child.

It was clear that we could easily measure the time that it took for a parachute to descend a fixed distance – several of the children had stop-watch facilities on their electronic wrist watches. But how could we measure the softness of landing? Two ways were suggested by the children. The first idea was to position a tray of sand for the parachute to land in. The children said that they could measure the depth of the depression as an indicator of softness of landing. But it was soon obvious that it would be very difficult to make sure that the tray of sand was always placed underneath the parachutes, which were to drop out of a first floor window. The second suggestion – the method we adopted – was to embed a stud-type paper clip into a piece of plasticine stuck onto the bottom of the parachute's load. The clip was left protruding exactly one centimetre when the parachute was launched, and, since this was the first part to hit the ground, the number of millimetres it had been driven into the plasticine offered a measure of the force of landing.

The discussion moved on to consideration of the different factors that might influence the design of our 'good' parachute. The first factor proposed was that the bigger the area of the chute, the softer the landing would be. This was discussed and fourteen children finally agreed with this statement, none disagreed, and seven were unsure. Another child suggested that the heavier the weight carried, the faster the descent would be, but only ten of her peers agreed, and eleven disagreed. Some pointed out that if the parachute was too light, it would sail off in the wind. It was agreed that we would have to have a weight great enough to ensure that the parachute went more or less straight down.

'What about the length of the strings?' asked another child. When I asked him, he reframed his question as another proposition to test – 'The longer the strings, the worse the parachute will perform.' Only three children agreed with this, and the rest were unsure how or whether string length would alter performance. There was more support (14 to 4, 4 don't know) for the hypothesis that the thicker the chute material, the faster the descent. But one child's suggestion that the material should not be colourful was dismissed by the class so completely that I was unable afterwards to get out of him the chain of reasoning that had led to this idea.

Somebody else suggested that the number of strings might have an effect, but she was alone in thinking that the more strings there were the better; three thought 'more meant worse', and the rest were undecided. I pointed out that some real parachutes had vents in them. Did this affect performance, I asked? Why did they have vents? All but one of the class thought that a hole in the chute would mean a faster descent. At this point, one boy pointed out that more than one of these factors might work together, and that instead of testing just one item at a time, we ought to test combinations. However, when we began listing the number of different possible combinations, it was agreed that this would not be possible, at least at first.

We split into five groups. Each group was to make a series of parachutes, each changing only one of the possible variables – the other dimensions were to remain constant. Thus each parachute was built to a standard pattern except for one characteristic. They all had a radius of 20 cm (except for those of group A, who were testing radius size); they all had six strings (except in group B); strings were 25 cm long (except group C); there was a weight of 25 g (except group D); and they had no vents (except group E, who tried out variable sizes, numbers and shapes of vents). The parachute canopies were made of polythene pedal-bin liners. Strings were stuck to the edge with masking tape, and supported a paper cup with one-gram plastic weights inside it. The impact-measuring device described above was under the paper cup. Each parachute was numbered with both the group letter and a number. Twenty-seven parachutes were made in this way over the week.

'The first day's test

Today we made parachutes and we had a plastic bag, string and a cup. We were split into groups. My group was C. We tested the length of the strings. Me, Felix and Berta made C1 (35 cm) and Alison and Cady made C2 (15 cm).

Me and Felix made the parachute and Berta went into the playground and timed the parachute and Felix hung it out of the window. C2 was Cady and Alison. They made one but the plastic burst so they made another one but they haven't tested it yet.'

The first-floor classroom window overlooked the playground, but there was a tall fence protecting the ground-floor windows a metre out from the wall. In order to drop the parachutes, we needed to drop them at a distance from the window. Despite offers from energetic pupils to lean out of the window, I produced parachute-release devices that were made from clothes pegs and lashed to a pole. By pulling a string, each parachute could be released well clear of the building in a controlled manner.[2]

There were still experimental problems. When there was any wind, we had to abandon testing – several parachutes were blown on to neighbouring rooftops. Merely recording the time was enough at the beginning, so many of the early tests did not include a measure of the impact of landing. There were other problems:

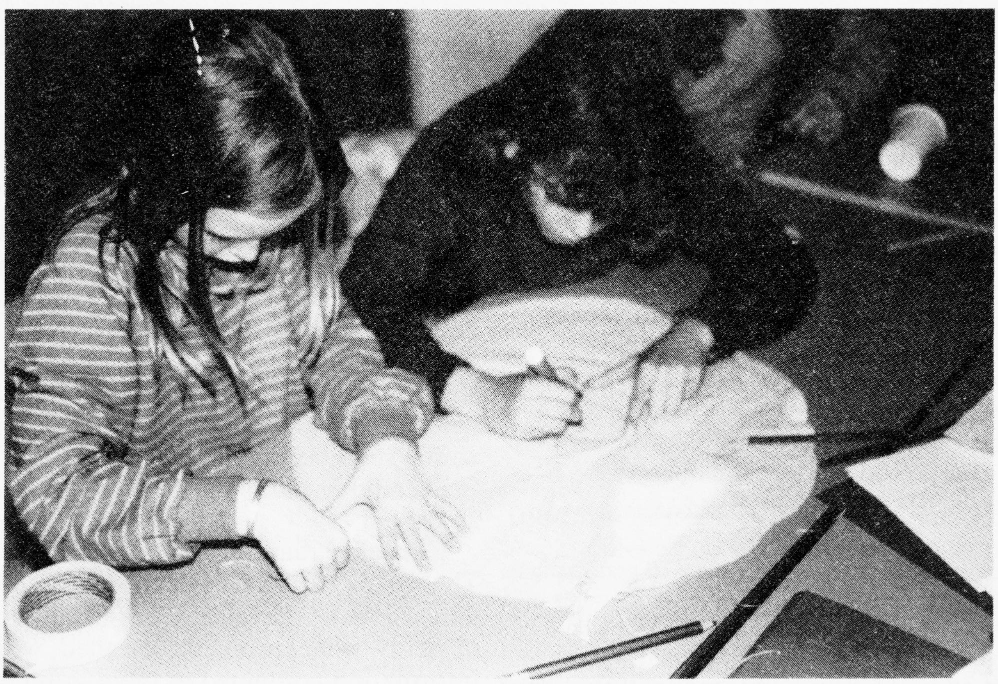


Figure 1 Making a parachute: one girl sticks on the strings, another writes on the identifying number.

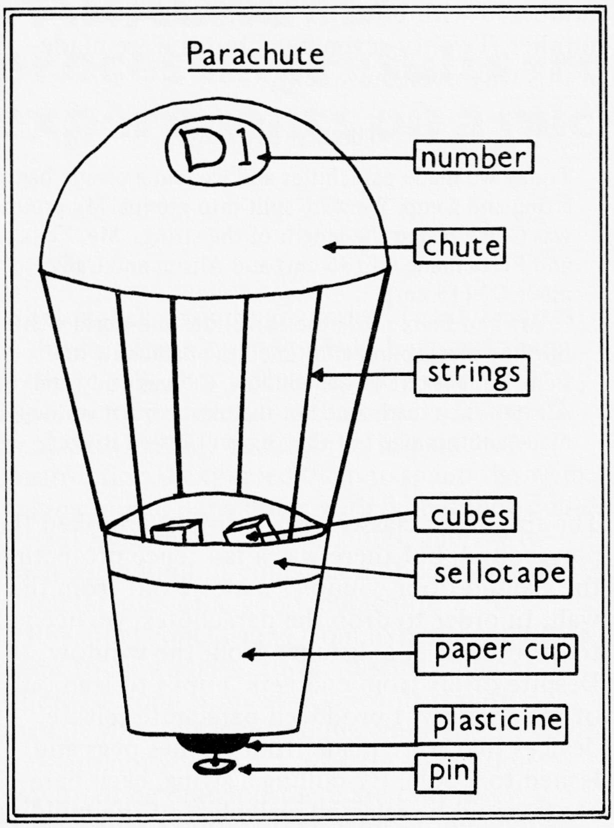


Figure 2 The construction of a test parachute. Note the impact-measuring device.

> 'Our parachute had six strings which kept on getting mixed up. It was quite hard to untangle it at first but I eventually did it. Then Mr Ross told us a new way of untangling the strings.
>
> The parachute floated quite well at first but afterwards it kept on getting tangled up. Sometimes it got caught on the wire but that was not very often.'

Another problem that I had not anticipated was the false sense of accuracy given by the children's digital watches, most of which recorded to two decimal points of a second. Differences in times that were too slight to be anything but observer bias were ascribed a quite unwarranted authority! Nevertheless, results were clearly emerging from these early experiments:

> 'Our group tested the weight. We used one gram cubes and put them in the cup of the parachute. Me, Tania and Nicola put 5 cubes in our parachute and Boris and Jan put 25 in theirs. What we wanted to find out [was] if more or less gram cubes made it go slower which is what we were aiming for. Our first time was 4.19 and our second 3.18. Boris and Jan's were 2.18 and 2.41. So it is the less the better so far.
>
> [next day]
>
> We have found out so far by testing a 10 gram weight and a 30 gram weight that the one with 10 grams in it went down slower. This is how I thought it would be. We are going to carry on testing different weights to see if the heaviest are the faster ones and if the lighter are the slower and better ones. I definitely think the heavier are faster and the lighter ones slower.'

Figure 3 Putting the test parachute in the jaws of the launcher.

Up to this point, each group had been recording its own results. We now decided to bring all the results together and keep them in a datafile using the microcomputer. The children who had experience of using the data processing programs were already very aware of the possibilities this would offer them for the quick and accurate sorting of the results.

> 'Today we put all our results into the computer and then asked it to give us all the times of the parachutes.'

It was almost as simple as this. A datafile can be imagined as a very large table of results. Each horizontal line of information contains the details of one experimental parachute flight – the number of the parachute, the number of strings it had, and so on, and the time it took to descend. The information is arranged in horizontal rows, each one known as a field. One field contains radii, another the number of vents, and so on.

To make a datafile two stages are required. The first is to specify the overall shape and dimensions of the table – principally the number of fields, what each is to be called, whether the information contained in them is numeric only or alphabetical characters, and how wide each field is to be (the number of characters reserved

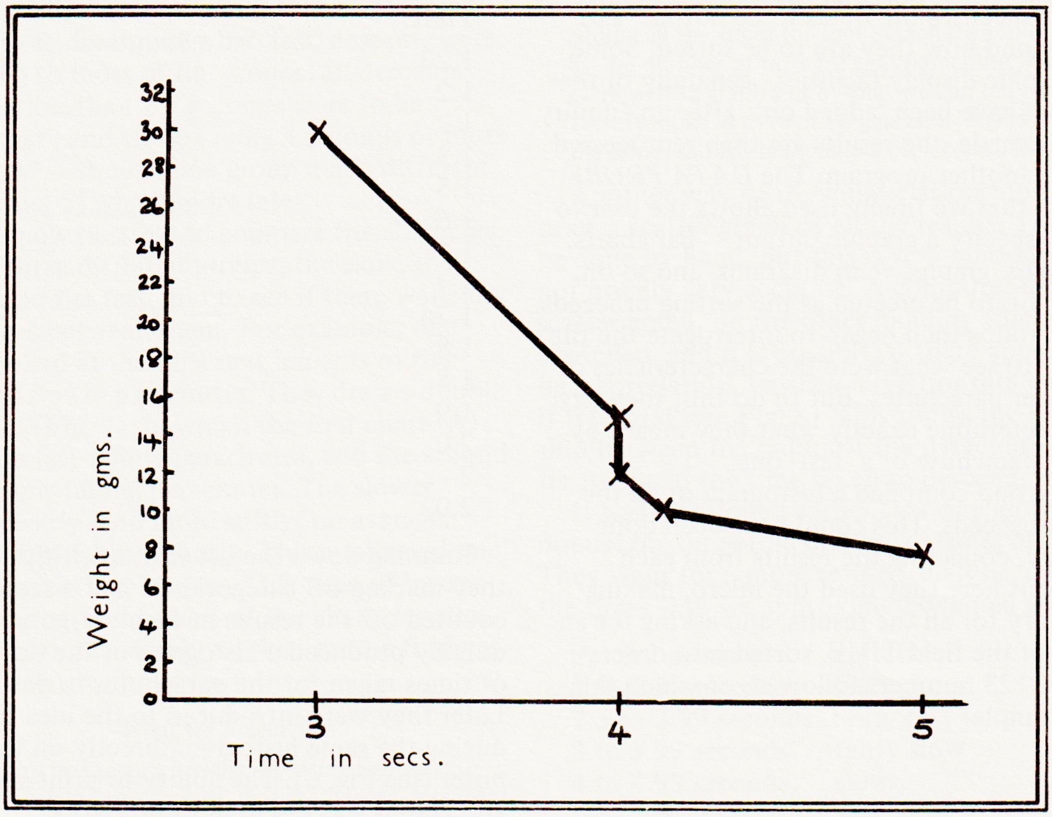


Figure 4 The average times of descent for each of several differently-weighted parachutes.

for each field must obviously be such as to accommodate the largest entry). This is a one-off process, and was done with the whole class. We created ten fields, though we later found that we didn't really use all the fields about the vents.

DATA PROBE File Information

File : PARA
Description : Parachute testing results: June 1983

Number of fields : 10

No.	Field Name	Type	Length	Notes
1	PARANO	Alpha	3	Distinguishing number
2	RADIUS	Numeric	2	in centimetres
3	NOSTRINGS	Numeric	2	number of strings
4	LENGTHST	Numeric	2	length of strings, cms.
5	WEIGHT	Numeric	2	weight of loads, grams
6	VENTNO	Numeric	1	number of vents
7	VENTSIZE	Numeric	2	total vent area, sq cms.
8	VENTSHAPE	Alpha	2	code for shape of vents
9	TIME	Numeric	5	of descent in secs
10	IMPACT	Numeric	1	in millimetres

Figure 5

The second stage is to enter all the results. This was done by the children keying in each experiment's details in response to being prompted with each fieldname. Because the fields were all very short, this was a quick job. In all we collected 123 records over the first three to four days. There are several advantages in the children keying in their own records: they become familiar with the QWERTY keyboard, and they realise that the information that eventually comes out is the exact information that they put in -- nothing is added!

The data file, when completed, was ready for interrogation. An enquiry is framed in terms of the field or fields the operator is interested in, and a condition that he wants fulfilled in that field. For example, to find all the parachutes with eight strings, one enquires

NOSTRINGS = 8

while to find those with less than seven strings, one asks for

NOSTRINGS<7

Combinations of fields can be asked for, as in

WEIGHT=25 AND RADIUS>30

or with

VENTNO=1 AND LENGTHST>25 OR LENGTHST<10

This form of Boolean logic is, in practice, fairly quickly picked up by most children of nine and ten. Having made an enquiry, the computer will match up all the records that meet the specification, rather as one might run one's fingers down a list looking for certain numbers. The user must now specify what he wants to know about the records that have been matched. Most data processing packages produce lists of information:

the user specifies the fields to be output, and whether and how they are to be sorted. Some have graphic display facilities , generally of the type that have been 'added on': after an enquiry has been made, the results are then reprocessed through another program. The *DATA PROBE* program that we finally used allows the user to directly specify a graphic output – bar charts, histograms, graphs, venn diagrams, and so on, and for this to be created as the sorting proceeds.

The groups then began to interrogate the file in order to see what were the characteristics of the slower parachutes. But to do this, they first had to determine exactly what they meant by a 'slow' parachute or a 'fast' one.

One group compiled a histogram of all the different speeds. This could have been done manually, collecting the results from each group, but here they used the micro, making an enquiry for all the results, and asking for an output of the field TIME, sorted into order. A list of 123 numbers followed, of which this is an example:

TIME
1.20
1.2
1.2
1.3
1.5
1.6
1.7
1.7
1.74
1.8
1.8
1.9
2.0
2
2.02

Running down the list with a felt-tip pen, they marked off categories of half-a-second, counted off the results in each category, and quickly produced a histogram of the distribution of times taken for the parachutes to descend. Later they were introduced to the idea of producing the same histogram directly on the computer (see Fig. 6). The ability to print such lists and graphs was essential to this work.

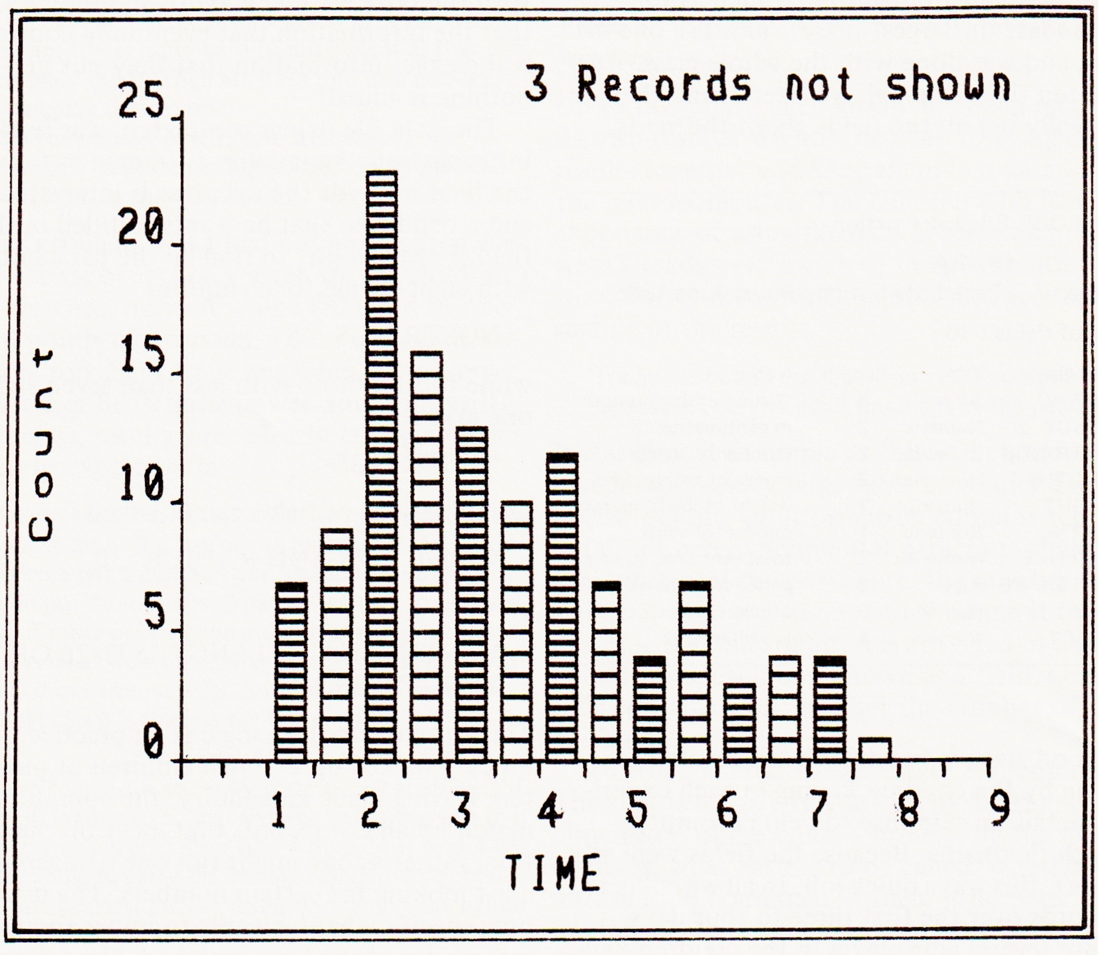


Figure 6 Histogram showing the range of times of descent recorded by 120 of the 123 parachutes (three were over 9 seconds).

Armed with the histogram, it was now relatively easy to determine what 'fast' descents were. According to most of the groups, all descents that took less than 2.5 seconds were to be judged 'fast', and those taking 4 seconds or more were 'slow' – though one group made different judgements, of which more later.

It was now possible to compare the characteristics of three different groups, the slow, the average and the fast, and to see if there were any differences between them. For example, one group looked at the different impacts of the faster and slower parachutes. They drew a double histogram (Fig. 7), in which the first chart (A) shows the fast-falling parachutes, and the second (B) the slow-falling parachutes. The slower 'chutes clearly land more softly, on average, than the fast-falling 'chutes. This confirmed one of the original hypotheses. This type of investigation was carried out on all the characteristics.

Other children simply worked on the findings of their own group.

'What we found out

Our group was testing weight. By using lighter and heavier weights [we found] that the lighter ones went slower and the heavier ones faster, as I thought. We worked out the average time for each parachute by adding all the times for that parachute and dividing by the number of tests for that parachute. We put the averages on a graph and by that we can tell by looking at it. It looks as though 10 gms and 12 gms are nearly the same, though ten gms is a bit slower'. (See Fig. 4.)

Another way of showing relationships was to draw a scatter graph of all the results. This could be done by hand, but this is very laborious and not usually very accurate. A computer-drawn scatter graph, on the other hand, is quickly compiled, and it is possible, if there appears to be a correlation, to draw a relationship curve on it freehand (see Fig. 8). This shows the relationship between the radius of the parachute (or its area) and the time taken to descend.

It was mentioned earlier that one group did not use the three-fold division of times of descent. They went for a more refined analysis, dividing the times into six approximately equal groups:

up to 2 seconds:	very fast
2 to 2.49 seconds:	fast
2.5 to 2.99 seconds:	fairly fast
3 to 3.99 seconds:	fairly slow
4 to 4.99 seconds:	slow
over 5 seconds:	very slow

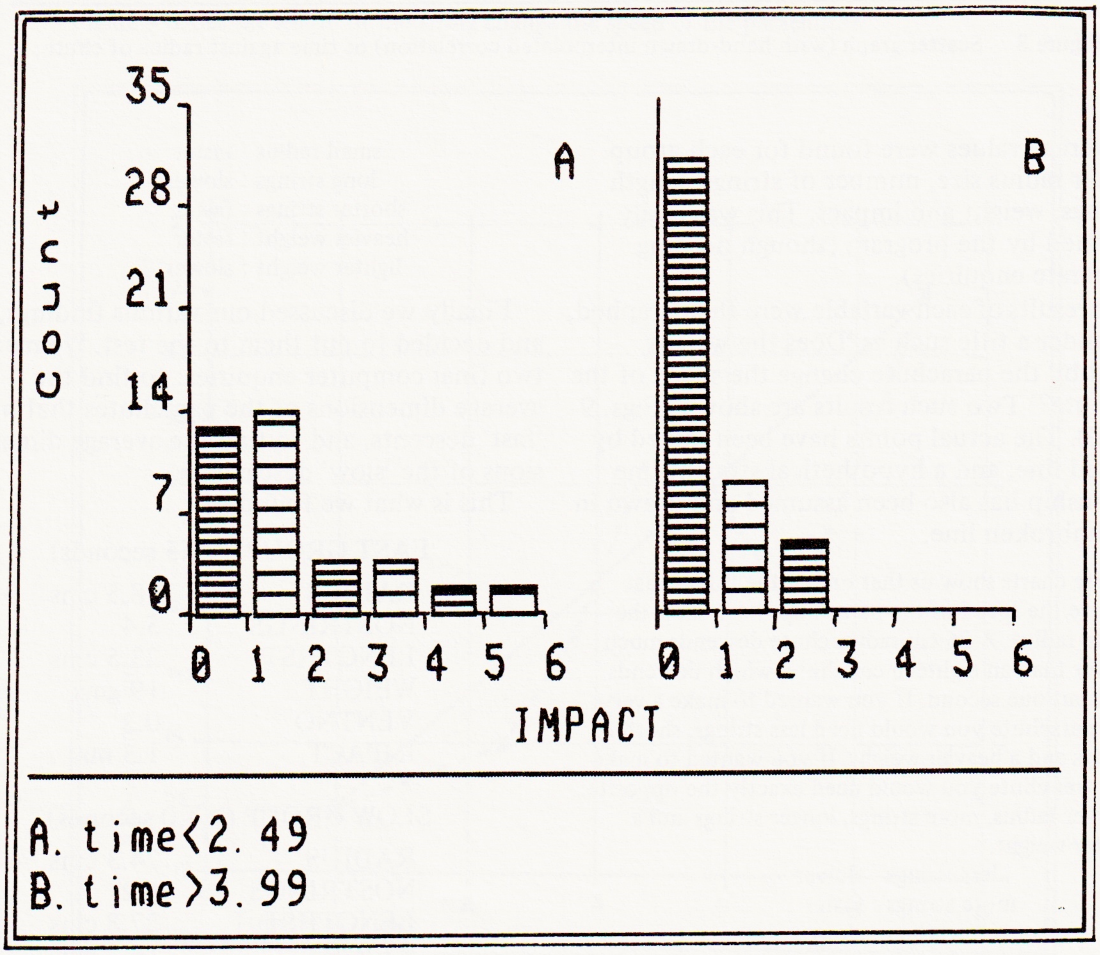


Figure 7 Twin histograms comparing the impact of fast and slow parachutes.

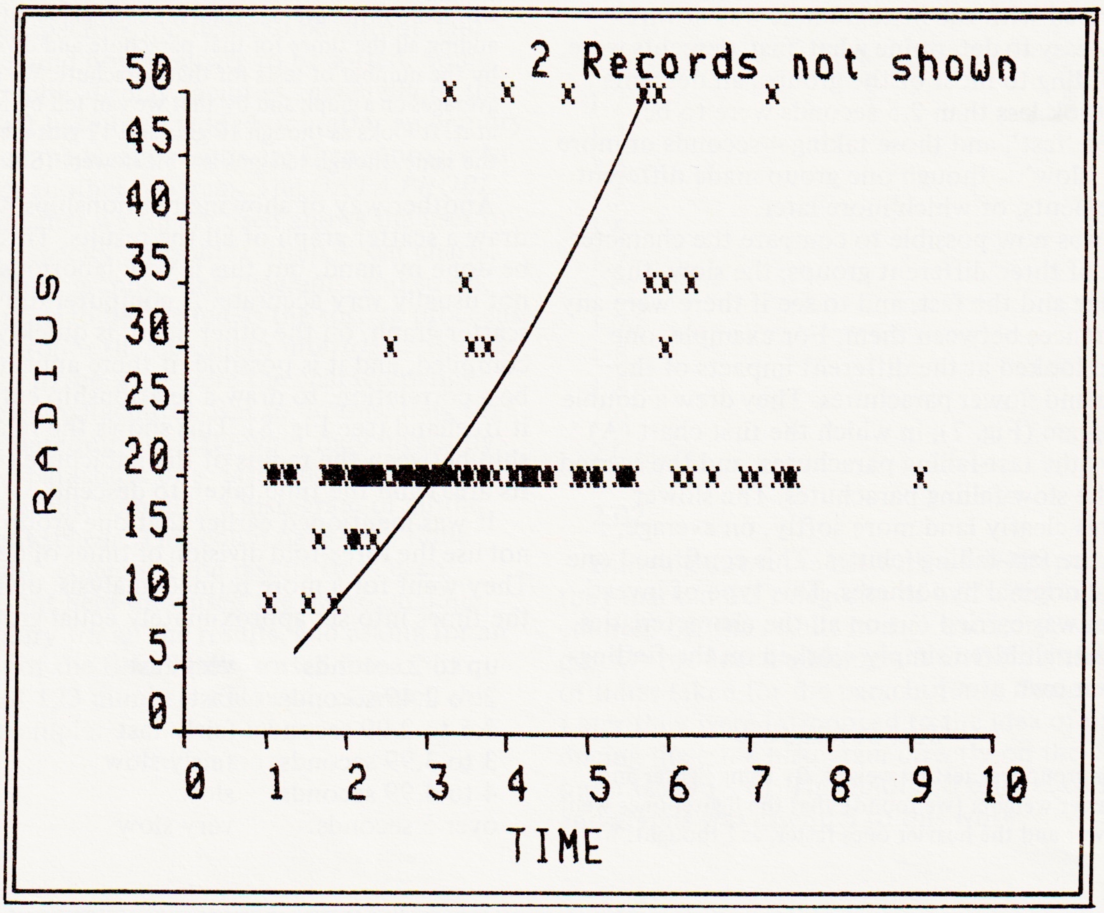


Figure 8 Scatter graph (with hand-drawn interpolated correlation) of time against radius of chute.

The average values were found for each group, for their radius size, number of strings, length of strings, weight and impact. This was easily performed by the program (though needing six separate enquiries).

The results of each variable were then graphed, each under a title such as 'Does the weight carried by the parachute change the speed of the parachute?' Two such results are shown (Figs. 9 and 10). The actual points have been joined by a dotted line, and a hypothetical straight-line relationship has also been assumed and drawn in as an unbroken line.

'These charts show us that one of the things that change the speed of the parachute the most is the chute radius. A 26 cm radius chute descends much slower than an eighteen cm chute, which descends in about one second. If you wanted to make a very fast parachute you would need less strings, shorter strings and a heavier weight. If you wanted to make a slower chute you would need exactly the opposite: a larger radius, more strings, longer strings and a lighter weight.

less strings : slower
more strings : faster
impact is less when : slower
impact is more when : faster
large radius : slower
small radius : faster
long strings : slower
shorter strings : faster
heavier weight : faster
lighter weight : slower'

Finally we discussed our various findings, and decided to put them to the test. We made two final computer enquiries: to find the average dimensions of the parachutes that made 'fast' descents, and to find the average dimensions of the 'slow' descenders.

This is what we found:

FAST GROUP (<2.5 seconds)

RADIUS	18.5 cms
NOSTRINGS	5.4
LENGTHST	23.5 cms
WEIGHT	19 gms
VENTNO	0.3
IMPACT	1.3 mm

SLOW GROUP (>4.0 seconds)

RADIUS	24.3 cms
NOSTRINGS	6.2
LENGTHST	27.8 cms
WEIGHT	16.2 gms
VENTNO	0.1
IMPACT	0.4

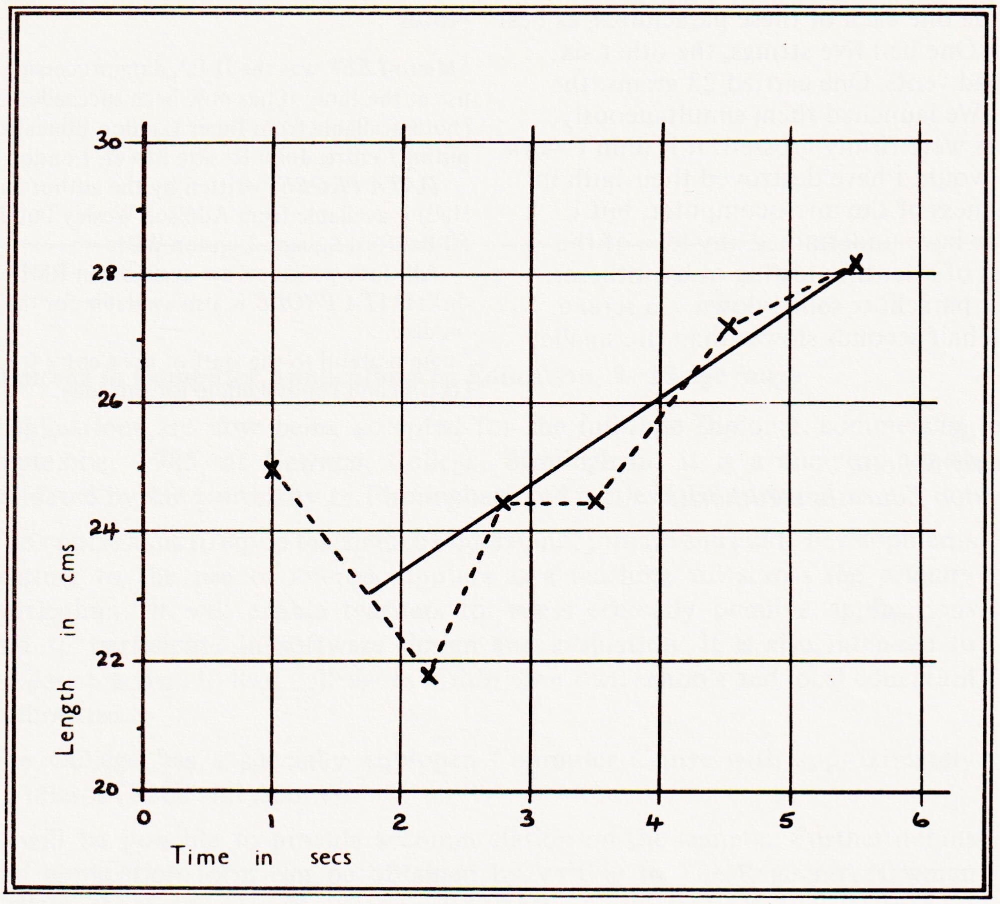


Figure 9 Does the length of the string change the speed of the parachute?

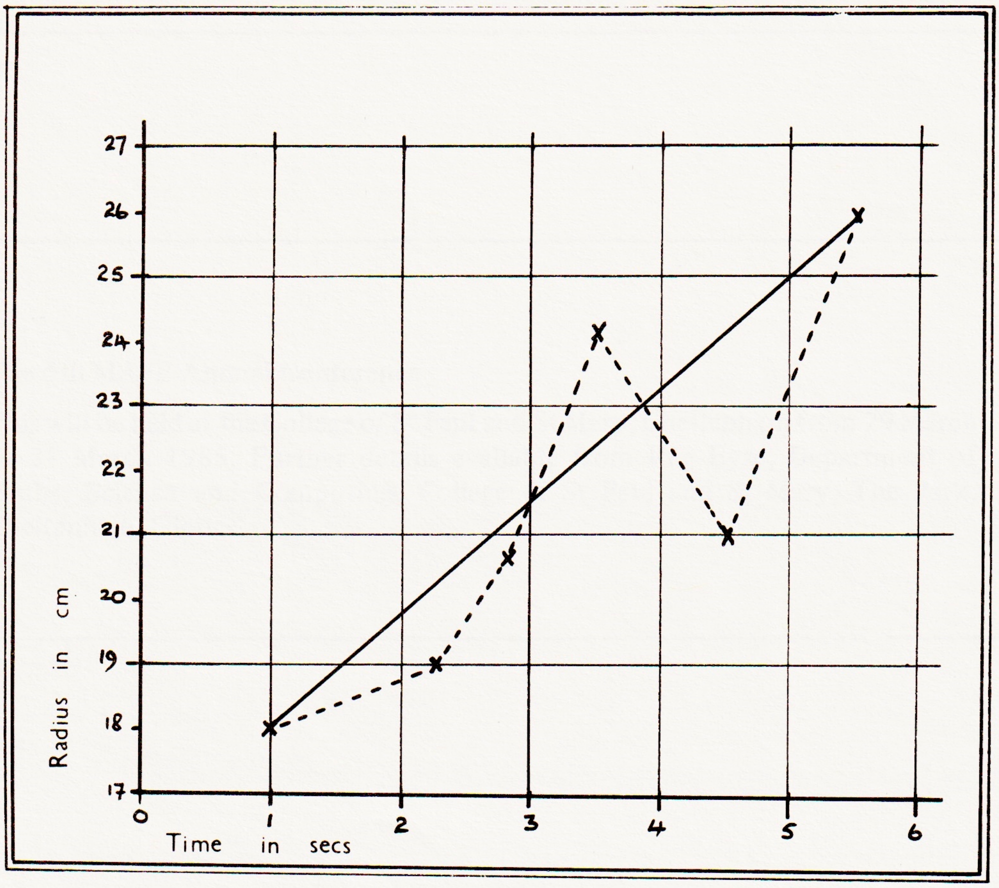


Figure 10 Does the radius of the chute make the parachute fly slower?

We made one each of these parachutes, as best we could. One had five strings, the other six. Neither had vents. One carried 23 grams, the other 28. We launched them simultaneously. My fingers were firmly crossed: if it didn't work not only would I have destroyed their faith in the usefulness of the microcomputer, but I would also have undermined any idea of the usefulness of scientific testing of hypotheses. The larger parachute sailed down – a serene one and a half seconds slower than the smaller parachute.

Alistair Ross
Fox Primary School,
Inner London Education Authority

Notes

[1] Micro-*LEEP* was the ILEA data-processing package in use at the time: it has now been succeeded by *SCAN* (both available from Inner London Educational Computing Centre, John Ruskin Street, London SE5).

DATA PROBE (written by the author and Malcolm Hall) is available from Addison Wesley Publishers Ltd, 53 Bedford Square, London WC1.

All three programs are available in RML versions on disk; *DATA PROBE* is also available for the BBC micro on disk.

[2] I am grateful to the staff of the Centre for Life Studies (ILEA) and Paddy Paddle for this idea.

Diploma in Computer Applications to Education, 5–13 age range

Applications are now being accepted for the full-time Diploma, commencing September 1985, at Newman College, Birmingham. It is a one-year course validated by the University of Birmingham and carries DES approval.

The course aims to equip teachers to understand, initiate and guide developments relating to the use of microcomputers as a teaching aid across the primary curriculum. It will enable teachers to assess critically possible applications and to participate in software design and evaluation. It is also intended to prepare teachers to lead colleagues within their own schools and local education authorities.

The College has a specially equipped Computer Centre with approximately 35 micros (RML and Acorn).

It will be possible to provide accommodation on the campus. Further details and application form can be obtained by writing to The Registrar, Newman College, Bartley Green, Birmingham B32 3NT.

The 5th MAPE Annual Conference

This will be held at the College of St Paul and St Mary, Cheltenham from 29 March to 31 March 1985. Further details available from Reg Eyre, Department of Maths, Science and Computing, College of St Paul and St Mary, The Park, Cheltenham, Gloucs.

Published by Castlefield (Publishers) Ltd.,
12 Chater Street, Moulton, Northants. NN3 1UD.
Tel: (0604) 494660.

£1.75