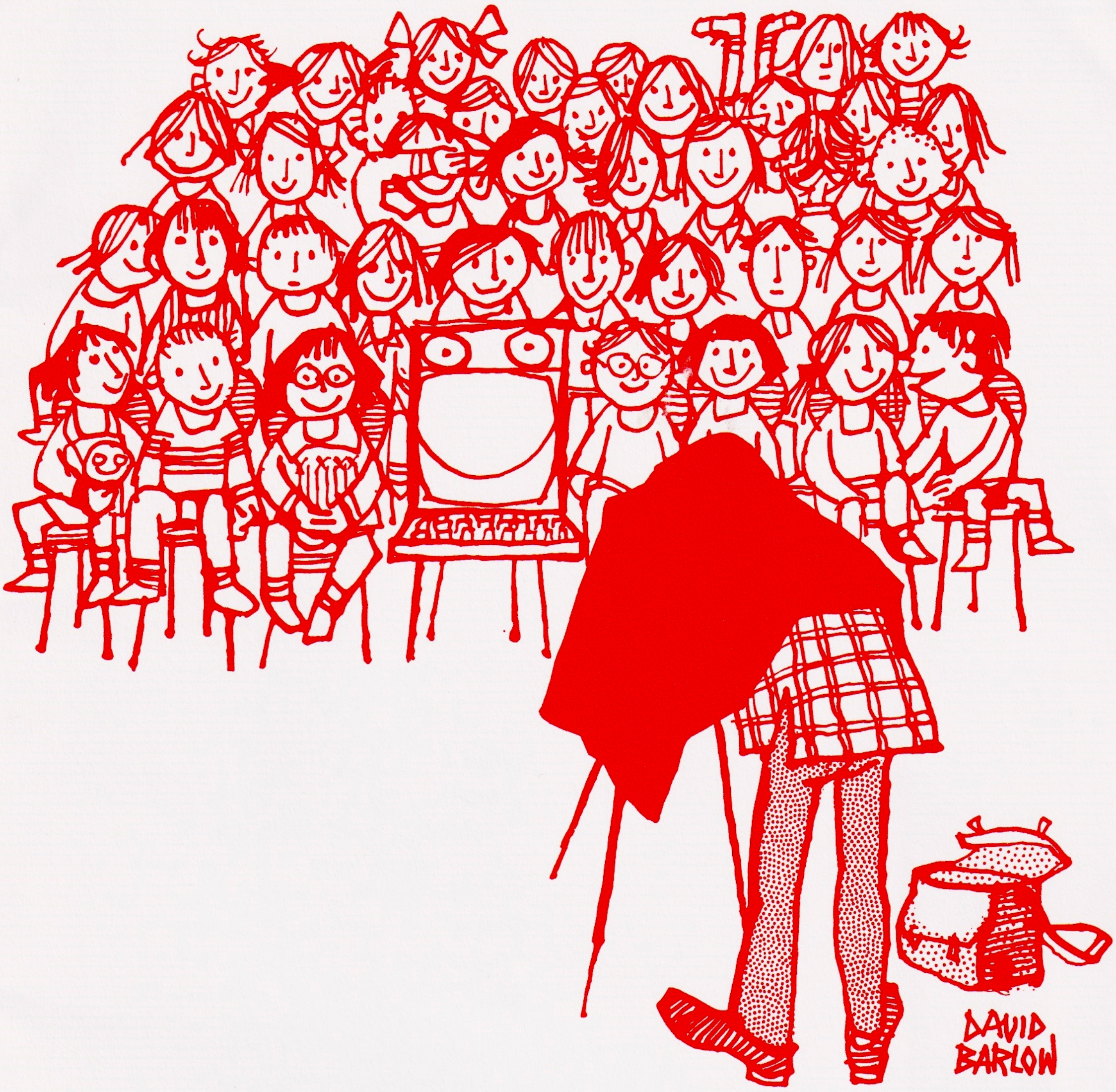
16

AUTUMN 1985



MICRO SCOPE

Newman College with MAPE

Contents

Editor Senga Whiteman
Assistant Editor Roger Keeling
Review Sub-editors Margaret Smith
Barry Wake
Design David Barlow

ISSN 0264-3847

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 £10.00 p.a. UK, £14.00 p.a. overseas, includes direct mailing of **MICRO-SCOPE**. Application forms from: Mrs G. Jones, 76 Sudbrooke Holme Drive, Sudbrooke, Lincs LN2 2SF.

Published by Castlefield (Publishers) Ltd.

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

Typeset by The Castlefield Press, Northampton.
Printed by Heyford Press, Moulton.

Editorial

I would like to take this opportunity to salute the MEP National Primary Project team, while the Project is still in existence! In terms of personnel the team is small in number, and they work from a limited amount of space in Winchester. However, their contribution to teacher education has been enormous.

The DTI scheme was launched in 1982, and the National Primary Project was set up in September 1983. One of the Project's aims was to plan and present short residential courses for those who have responsibility for the in-service and initial training of primary teachers. In practice, the NPP has produced a wide range of INSET packs upon which courses have been based.

The idea is that 'teacher trainers' attend the courses, study the packs, become familiar with the software, discuss some of the educational issues involved, and then return to their base and pass the education on. The courses have all been oversubscribed. The NPP has been filling a vital need.

The output from the team has been prolific. They have covered, or will cover, most of the major applications of micros in education. At a time when teachers have been crying out for a range of flexible software, the NPP has been producing masses of it, and more to the point, distributing it free, (although there may be one program within a pack which is copyright). As many of us have discovered though, software alone is not enough. We all have, occasionally, a bright idea about how a program can be used, but what we would really like is a whole range of suggestions for educational activities relating to, or incorporating, software. This has now been provided for us.

A copy of each of the packs has been sent to every LEA. Copies have also been sent to Colleges of Education. Every MEP Regional Information Centre also has a copy of each pack.

The following packs have been produced and distributed prior to this term:

Mary Rose – some classroom applications;
Primary Maths and Micros;
Micros in Project Work;
Language Development in the Primary School;
Posing and Solving Problems with LOGO;
Infant and First Schools – the role of the micro.

Two more packs will be distributed before March 1986:

Posing and Solving Problems using Control Technology;
Posing and Solving Problems with the Microcomputer.

If you are not aware of the contents of the packs, or indeed of the existence of the packs, contact your LEA. The software should be distributed in association with a certain amount of in-service based work. The educational value of the packs will be diluted if they are treated merely as collections of free software.

We need in-service education. The NPP has provided LEAs with ready-made materials for courses. It will take some time for these materials to be processed but there is a vast amount of support ready for those who need it.

* * *

If you are working in an independent school then contact: ISMEC, Westminster College, North Hinksey, Oxford.

If you have no easy access to the packs, it is possible to purchase a copy; the price is approximately £45 per pack. For further details write to MEP Resources Centre, Dunelm House, Newcastle Polytechnic, Coach Lane Campus, Newcastle upon Tyne NE7 7XA.

The *Mary Rose* pack is sold out.

Letters

Elementary Informatics in Australia

I want to tell the world what we've been doing down here in South Australia.

I believe we are leading the way in the area of Informatics in just one school – mine.

Since 1981–2 I have been developing a course in Information Handling and Problem Solving and have set up an Informatics Centre at an Elementary School to trial this course. I believe it will be the kids coming through an Informatics Course like ours, who will solve the massive problems of accession and indexing now looming on the horizon.

We have produced a booklet and various newsleaf sheets to sell to visitors to our centre, but being 'ahead of our time' so to speak, in relation to the rest of our education department, things move slowly in other places.

I have given various addresses to groups of educationalists and all have been excited by the concepts I have uncovered.

We begin by demystifying electronic gadgetry, not with courses in computing at a formal level, but by allowing very young children to pull damaged electronic equipment to pieces – with screwdrivers etc. – until those once solid, unitary looking 'NoNo's are broken down into the many building bits of IC chips, wheels, motors and screws etc. Then we establish a concrete understanding of storage retrieval and publishing skills, (filing cabinets and index cards etc.), using the child's personal heritage as an information base. This then leads on to creation of their own databases and the various applications of computers in the manipulation of their information.

I hope you will make the right connections for me to let the world know what we are doing down here! Computers are only the tool – Informatics is the Discipline of the Total Information Society. The elementary school should be the place where grounding is established.

Kevin Nicholas
Strathmont Informatics Centre
Strathmont Primary School
Cowra Avenue
Gilles Plains
South Australia 5086

More on 'Evidence from MAPE'

Thank goodness for the letters from Brian Richardson and Mike Matson *[Letters, MICRO-SCOPE 15]*. As a parent of three primary school aged children, a member of MAPE and a producer of commercial software, I was beginning to feel totally irrelevant. Why, oh why, do educationalists seem to forget their job is preparation for life outside school. How can the MAPE Executive say 'the commercial software market is not appropriate'? For most people life is the commercial market. Surely, though teaching is a skill not available to all, the materials that a good teacher uses must come from everywhere.

I found the MAPE Executive's attitude particularly disappointing in that, as a non-teacher, I have found the MAPE conferences and the teachers I have met there, an inspiration, and have been reassured that the education we are offering our children is something worth having.

I would also like to endorse what Mike Matson said about the needs of the handicapped child. Might I add that in the contacts I have had with MEP and its regional centres it is those concerned with special education that have been most efficient and effective. It is a good thing this is the case as these are the children that stand to gain most from the micro revolution. MAPE look to your laurels. The integration of the handicapped seems to have been overlooked in your philosophy and practice.

Mary Wain
4 Church Road
Felixstowe
Suffolk IP11 9NF

Viewpoint
Computers in the ideal school

Patrick Drewett

Given unlimited resources, one 16 bit micro per child would be my ideal. This would require a network linked to a Winchester Drive with sufficient storage for the necessary range of software. Sixteen bit micros would give the flexibility required by top quality programs, and sufficient memory for user-space. But even this sophisticated network would need several stand-alone units so that pupils could familiarize themselves with 5¼″ disc drive technology. To support this network, each machine would need a printer to produce a hard copy of children's work. Printers are an essential part of today's primary school computer systems and are also a requirement of the ideal system.

Would computers be used by children all day? In the main, yes. Since the majority of lessons can be taught better by computer with a well-designed program, it is logical to transfer the teacher to the role of overseer of the overall teaching process. Where the program fails, the teacher can step in with human expertise.

Are the majority of lessons taught better by computer? Not with today's programs; but theoretically they could be. For example, how many children go out to their teacher's desk only to find that they have a whole page of sums wrong with the same mistake in each? A good CAI (Computer Aided Instruction) program would be much more time-efficient. How many children waste time repeating the same already-mastered skills because the rest of the class needs more practice? The brighter children are held back, just as some of the slower children are given work which is too advanced for them. Individualised learning is much more efficient.

What maths do we teach children anyway? Long multiplication, long division, multiplication of fractions? Let LOGO come into its own and children will want to learn mathematics, real mathematics where children pose their own problems and effectively solve them. It is doubtful whether 50 per cent of children educated on teacher-maths are capable of solving problems. Using LOGO, the children do nothing else, but at a level that they are capable of because they set that level themselves. LOGO covers the maths syllabus: addition, subtraction (where you can see things getting bigger or smaller), multiplication, division, decimal fractions, algebra, geometry, Cartesian values, negative number and many more concepts are encountered.

The LOGO user must understand these concepts in order to use them (or use them to understand them). LOGO would make maths lessons enjoyable for the teacher too! Goodbye exercise books; hello hard copy. Wot! No marking?

Word processing is shown to be an effective system of communication in offices and businesses and, in an ideal situation, children's work would be written with the aid of a wordprocessing chip. The effect of this might be to transfer the attention of the teacher from presentation to content. With the inclusion of a spelling check, emphasis would be shifted from spelling correction to syntax and vocabulary improvement. An on-screen Thesaurus might also be a possibility.

The ability to communicate with other stations is an important skill to learn in the computer age, and an extension of this would be a system similar to the Times Network which would give access to mainframe databases. This system would be the key to unlock the door to the vast amounts of knowledge contained in these databases and allow almost unlimited project work and research.

Interfacing the computer to various other devices will be a requirement from time to time, and the necessary equipment should be available. The control of Turtles, Buggies, Robotix models and simple electrical circuits all need an interface, and the development of work in the control technology area has major implications for the scientific and mathematics curriculum. The investigation of control technology lends itself to the problem-solving approach, which is an important approach in child-centred learning. And a collaborating influence in control technology must be LOGO.

With so much individualised learning, one must not forget about socialisation! This may be encouraged by community singing with music played by an electronic synthesizer. Video cameras, to help English, Drama, Environmental Studies etc., also encourage group work.

However, there are many barriers in the way of such a lavish educational provision, not the least of which is the oscillation of the pendulum of change. This may be summarized thus:

```
TO INNOVATION
FORWARD  100
BACK      95
END
```

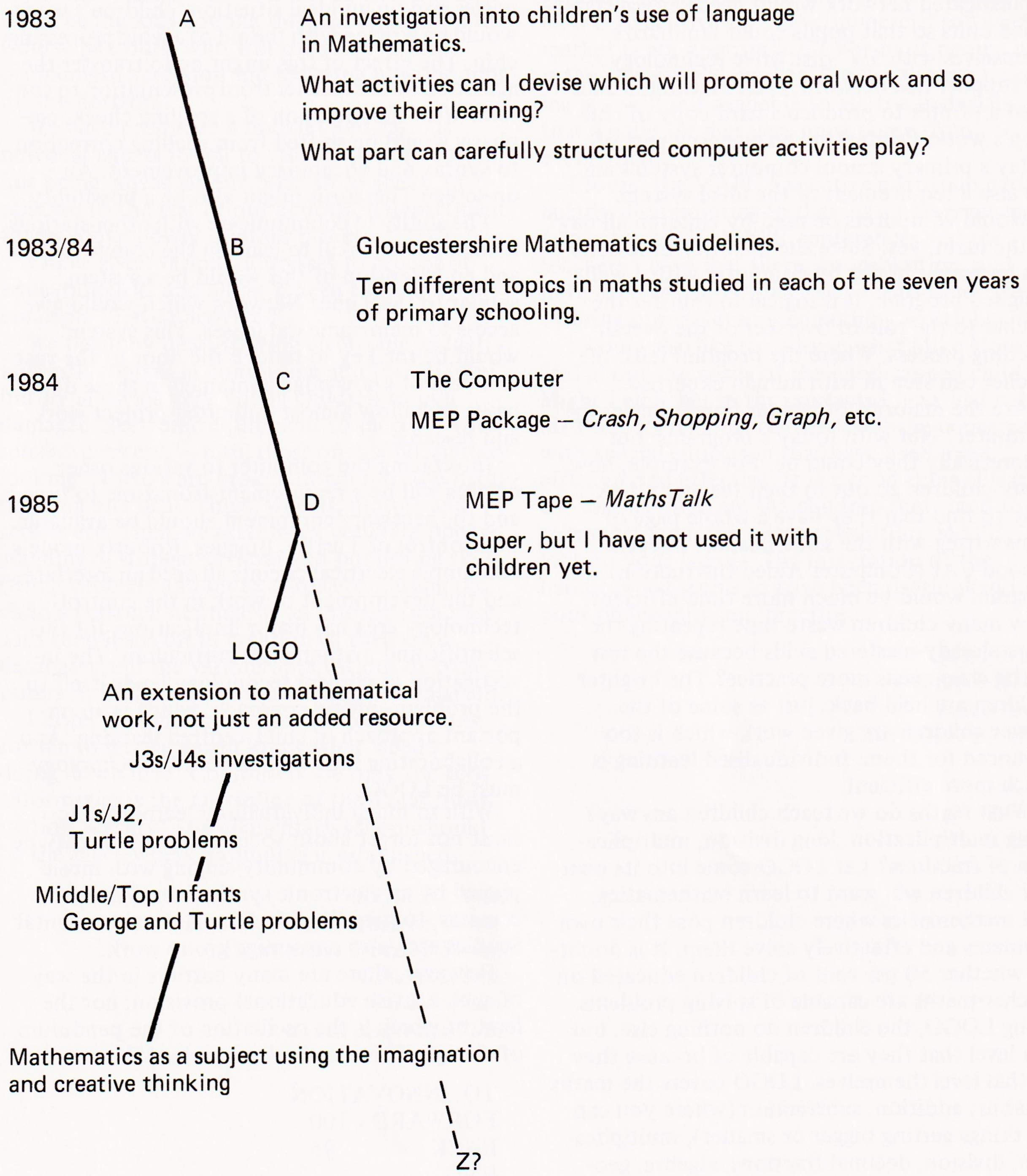
Children's use of language in mathematics

Dee Barnes
St Andrew's School, Chedworth, Glos.

Where am I now?

The quick answer is I don't know, but I am learning so much about children's acquisition of language and the development of mathematical thinking that I don't think it matters. This is an open-ended investigation.

I know where I started.



1983 A An investigation into children's use of language in Mathematics.

What activities can I devise which will promote oral work and so improve their learning?

What part can carefully structured computer activities play?

1983/84 B Gloucestershire Mathematics Guidelines.

Ten different topics in maths studied in each of the seven years of primary schooling.

1984 C The Computer

MEP Package – *Crash, Shopping, Graph,* etc.

1985 D MEP Tape – *MathsTalk*

Super, but I have not used it with children yet.

LOGO

An extension to mathematical work, not just an added resource.

J3s/J4s investigations

J1s/J2, Turtle problems

Middle/Top Infants George and Turtle problems

Mathematics as a subject using the imagination and creative thinking

I have clearly been side-tracked and I am not sure whether I shall ever get to Z – but then I do not know where it was, or even that it existed. Perhaps it was the ultimate solution (42?).

A Primary Scheme for LOGO

I worked out a scheme for LOGO in my school, but I worked backwards as materials became available. I began with the J3 and J4s because they can work with the least supervision; teaching heads tend to disappear to hear a first-year read or to answer the phone at inconvenient moments. Initially this group carried out simple teacher-directed projects but gradually they began to ask 'What happens if . . . ?' and they were away. I gave them some picture drawing, which I think was too much too soon, and they have moved now to modifying a program. They are investigating spirals: each group has its own basic shape to work on and they are observing the changes that occur when one element of the program is changed. A printer is invaluable here as they can keep a record of everything they do. The language work is not just about shape but also about number. One group included negative numbers on their own initiative and I suggested to two others that they include decimal fractions to make the changes very small. These nine- to eleven-year-olds are interested and absorbed in experimenting with number and shape.

The younger juniors are a more stolid group, less imaginative, unwilling to devise their own questions. They prefer a simple adventure game to LOGO any day. However, the acquisition of a Valiant Turtle improved this group tremendously. The screen turtle was too quick for them; they prefer the slower speed of the floor model, and they relate to it more easily. The quality of their language work is entirely due to this more leisurely speed, because now there is time to talk. Younger children take much longer to put their ideas into words; they are reaching for new vocabulary and new forms of expression. As they work with the Valiant I can hear them formulating hypotheses and testing them, predicting outcomes of computer commands and observing the results. Real learning is taking place and should provide an excellent basis for future work.

Their reaction to a wrong instruction is very interesting – using the screen turtle the group will turn on the child at the keyboard:

'Oh you've done it wrong!
You should have . . .'

but with the floor turtle the mistake is seen as the turtle's and the children address him instead. Of course they realise that fresh instructions must be given and they immediately work out a modified program but no-one loses face. The turtle absorbs the blame.

These seven- and eight-year-olds are building instructions into simple programs and they are learning to repeat programs and make sequences and patterns. They enjoy using the pen, and talk about the results on the paper. Mathematically they have a real understanding of an angle as an amount of turn, and use a 360° protractor with ease: I left two on a table near the computer and they found it helped them to draw more accurately. The language here is all of shape and angle, and inside and outside angles, though not always using the correct terminology.

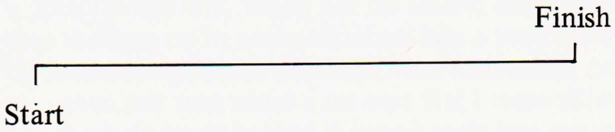
Still working backwards, I progressed to the top infants – not because they are less able at computer work but because I am less confident with them; my mind works on junior tracks and I do not believe in giving six year-olds watered-down junior teaching.

I started with the screen turtle because it was all I had, and I wanted the children to learn the basic commands. They found it difficult when large numbers caused the turtle to disappear in one place and reappear in another, due to screen wrap-around. The arrival of the Valiant Turtle was a delight for them because here was something less abstract: it moved in cm steps and rotated through a clearly defined angle of turn. Also, it was fun. Although they enjoy using it with a pen I think I introduced this too soon and I should have stayed longer with paths of movement activities – turning through gateways, knocking down boxes, etc. Ink patterns on the floor gave me a visible end-product for the lesson, (like sums in an exercise book), but the mathematical thinking was constricted by the need to concentrate on movement plus pen-line. Given a simpler objective (movement only) the children relaxed and talked around the subject, they discussed alternatives, and my favourite phrase 'What happens if . . . ?' reappeared. This group also worked better with a tangible object – they stood beside the turtle and worked out whether he needed to move left or right; they speculated on whether the finishing line was 20 or 40 cm away.

The middle infants are now joining in some of these activities, but as they are still on a 'settling in' programme I cannot comment on their learning. The reception children have 'had a go' with the Valiant, but not much more.

My group of eight 6/7 year-olds have settled down to useful work with the Turtle but they cannot yet write a complete program. When guiding it around a course they give the commands one at a time. I have worked with them to build these into a program but they are not ready to progress to this stage; they are still using a single step approach. They prefer working out more complicated paths to forming a simple one into a single entity. They have shown me there is another step in this development by their excitement over their latest discovery:–

Having worked out one path of movement, for example:



which sets the problem of estimating the long stretch in the middle, they consolidate several steps into one. They may estimate the central stretch as

FD 50
FD 30
FD 10

but having reached the FINISH gate they have found they can come back again much more quickly by using the instruction

FD 90

The next stage must be to write the complete program – but that they wish to discover for themselves!

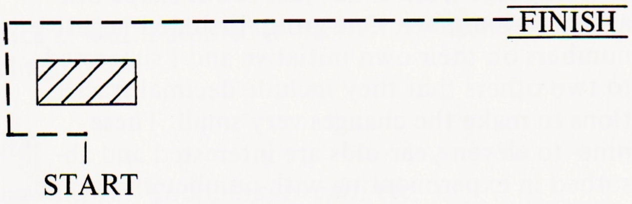
The Valiant Turtle moves in centimetres and these six- and seven-year-olds have a much more precise understanding of measurement as a result. I watch their estimates grow daily more accurate. At first they used FORWARD 10 (a good safe figure, they use it for everything), and FORWARD 100 (which to them just means 'a big number'). Tentatively they began to use 20 and 30, then with growing conviction, falling back on 99 when they wanted to express 'a number too big to imagine'. Now they can predict 40 cm and even 60 cm with some accuracy. All this is achieved with much talk and discussion and many wrong guesses, but that is how children learn. This group has positive experience of measurement and recently completed a measuring page in their textbook with little difficulty. They then progressed to the next page, on perimeter, and finished it with the same ease. Does this mean I shall have a set of children moving up through the school who actually *understand* the difference between perimeter and area?

Our newest purchase is 'George'[1], a computerised robot. Here I began with the top infants because it was convenient to the time-table. George is very noisy and there are times when I hold this against him, but after the first disastrous lesson which was all squeals and rushing around, I realised his enormous potential as a teaching tool. He provides the instant reaction to commands that very young children need. Most important of all, he stores the program in his memory and runs through the whole thing each time he is activated, so for the first time my six- and seven-year-olds can appreciate the meaning of a program.

George teaches programming.

Press the buttons ↑ 4	and he will move forward for 4 seconds
Press the buttons ←2	and he will move forward for 4 seconds then turn left for 2 seconds
Press the buttons ↑ 3	and he will move forward for 4 seconds then turn left for 2 seconds and move forward for 3 seconds
Press the buttons →2	and he will move forward for 4 seconds then turn left for 2 seconds then move forward for 3 seconds and turn right for 2 seconds

Given a course:



the children will work out the instructions one stage at a time, but they will have to return George to the START line every time they add a new instruction as he always repeats the entire program, adding on the new line. There is, of course, a cancel button.

Working in a group the children tend to take turns so the running commentary goes:

‘ “That's Andrew's Forward 4.”
“That's Caroline's Left 2.”
“That's Gareth's Forward 3.”
“That's Henry's Right 2 only it should have been 1.” ’

Each time the program runs through they comment upon it and suggest alternatives.

I think they will gain an excellent understanding of what a program is from this work, and will make better progress as juniors than this year's J1s (only of course I shall give the J1s George as well).

At this point in my study I have attained one of my preliminary objectives: I have a scheme of work for LOGO suited to my own primary school. There is a logical progression of activities.

Primary LOGO

Activities with George:
Paths of movement;
Simple programs;
Complex programs;
Defining own objectives.

Activities with Valiant Turtle:
Paths of movement;
Writing a program;
Sequence of programs;
Using the pen;
Pattern making.

The screen Turtle:
Completing given tasks;
Investigations.

Where next?

'I know it is all great fun' said the Vicar at my last Governors' meeting, 'but are the Valiant and George really teaching them anything about maths?' This, coming from my staunchest ally, made me realise what the rest were thinking, so it is clearly time to organise my own thoughts and prepare to explain myself next September.

I believe computer work should be an integral part of the curriculum, not something tacked on as an afterthought or a Friday afternoon activity. To that end I attempted in my last school to 'tie in' each new program to a specific chapter in the maths textbook so that children would meet the appropriate program at the right moment. I saw commercial programs as extension and reinforcement resource material, and one which added a new dimension to classroom teaching. It was a useful task in that situation as the school was much larger and it was not reasonable to expect each teacher to know the content of every program. I divided the material into two categories – those giving intensive practice to a child with a particular problem, and those giving opportunity for group oral work.

LOGO presents a new challenge: it teaches concepts, it does not reinforce them, and it does not follow accepted primary sequencing of topics.

I am using LOGO as an infants teacher uses sand and water. Through it the children are gaining experience in mathematical thinking which they will draw on when they meet particular topics in their textbooks. There are already chapters we skip over when the children meet them. Graphs are an obvious example because I teach graph construction from real situations in topic work. Apart from a few exercises from the end of the chapter I see no need to go through it all again with fictional children drinking fictional bottles of milk. The primary maths scheme always did contain a large practical element in the measuring sections, but now I think the children will have the capability for greater accuracy much earlier. I shall make no attempt to tie LOGO to the textbook, but I shall note where work in the textbook is superfluous.

In all directions

Learning to use LOGO is work for a year – but what about next year when the children are already competent in the language, and the year after that?

My aim in education is to create an environment where learning can happen. I can arrange funds for the hardware and buy published material as it becomes available. I can create space on the timetable and welcome specialists in computing into my school. I can encourage mums to come in and help children with computer work rather than just hear them read for a busy teacher. I have to accept the limitations of being a non-specialist in this field myself and compensate for this by ensuring continuity across the curriculum which is an important benefit in small schools. What else can I do?

Teaching consists of letting children learn, but teaching also consists of breaking down new knowledge into many parts, each developed from the next, and then allowing children to progress through those parts, each at his own pace. I hope my primary scheme provides this opportunity.

LOGO is a language, a logical language. What will the children do with it?

They have yet to explore other aspects of LOGO. Last week I showed them they could use it to solve sums; this is a whole new area to investigate.

As a written language tool they have not progressed further than covering the screen with 'Ian is the greatest'. There is scope for development here.

Is it really Maths?

I have been considering this question since I attended the meeting of the Association of Teachers of Mathematics in Warwick last February. I know the answer is yes but I cannot yet provide a full and convincing answer.

Ideally, mathematics is a practical subject. The child learns from carefully structured activities or investigations; exercises in the text book are a check for the teacher that the work has been understood. Teaching direct from the text book is usually a failure.

'I can't do this page.'
'What does it tell you to do?'
'I don't know.'
'Have you read it?'
'No.'

LOGO affords us one more way of giving the necessary experiences.

Secondary science teachers ask us to teach primary children fewer facts and give them more experience of air, water, structures, capacity, the environment, etc. They can build their own programmes much more effectively on this kind of teaching. Secondary mathematics teachers will appreciate children with a background of real understanding rather than rote learning.

What are they learning?

They are learning to tackle problems logically, to break them down into a number of very small steps, each of which can be tackled separately.

They are learning about variables.

They are learning some applications of number.

They are learning a useful computer language.

They are learning that mathematics is not a subject with right and wrong answers to which the teacher responds with a tick or a cross. Imagination is as important here as it is in other parts of the curriculum; creative thinking is necessary in investigative work or it soon grows stale.

Teachers are trying to define LOGO to justify it as part of the mathematics syllabus. Perhaps we should redefine mathematics so that it can include LOGO. The calculator/computer is freeing us from much mechanical teaching, just as the adoption of decimal currency meant the end of pages of money conversion sums. HMIs want to see the 9–13-year-olds solving problems and carrying out investigations. LOGO is an ideal tool in this new climate of opinion. Can we develop a new definition of mathematics that takes account of creativity? For years we have encouraged parents to look at their children and decide on their probable specialism.

'Oh, she's very imaginative; she'll go on the Arts side.'

'Oh, she's very good at number; she'll go for sciences later.'

Should we not aim to produce imaginative mathematicians who show great skill in problem-solving and can use modern technology to carry out the mechanical side?

Reference

[1] *George* is distributed by CGL, CGL House, Goldings Hill, Loughton, Essex, and costs approximately £23.95. Consult your Argos catalogue for a cheaper price.

Networks? Maybe. Computer rooms? No.

Mike Matson
4Mation Educational Resources

In *MICRO-SCOPE 13* I wrote about how I thought computer specialists were not necessarily the right people to lead the micros-in-primary-education revolution.

In *MICRO-SCOPE 14* the MAPE Executive stated that, *'Networking of computers will be the most manageable and economically realistic way of extending schools' computing resources.'*

I do not agree with the Executive's view but I would like to add a note of caution because, if we are not careful, the proliferation of networking may well put the next stage of the 'revolution' back in the hands of the specialists. 'What evidence has he got?' you may well be asking. Read on and I'll tell you.

* * *

On a recent trip to Australia I visited a dozen schools (in Melbourne and Sydney) including primary, secondary and all-age schools. In each of these schools there was a network. In only one school was there *not* a computer room. In only one school (the same one) were the micros actually used in the normal classrooms. I'll relate my experiences in just two of these schools.

A primary school

I was introduced to a teacher who was going to be using the computer room for an adventure session with a class of 11-year-olds. I waited outside the computer room while he went off to collect his class. The children filed into the room and sat down in groups at the dozen or so BBCs. A woman followed them into the room and I assumed that she was either a parent who had come to help or a teacher who wanted to gain some first-hand experience about using micros in education. When she sat down with a pile of exercise books and started looking through them I guessed that my assumption hadn't been entirely accurate, but I still couldn't work out why she was there. The teacher ensured that the required program was available to each group and, after a short discussion about what had been accomplished during last week's session (more for my benefit than the children's, I felt), the groups made a start at solving the mysteries of the adventure. The teacher kept an eye on the proceedings and, at fairly regular intervals, made all the children stop what they were doing in order to ask them what they had discovered. By pooling their knowledge they were able to proceed through the adventure at quite a rate, (which wasn't quite what the author of the material had in mind when he wrote it). Towards the end of the session I approached the woman who was still poring through her heap of books. I discovered that she was the class teacher. Who, then, was the man taking the lesson? Yes, you've guessed it: he was the school's computer specialist. I asked him what he did with his time and was told that he did 'this sort of thing' all day, every day. Back to the class teacher. No, she was too terrified of all this equipment to want to use it herself. Yes, it was a good time to catch up with some marking. Subversive question from a fellow visitor: 'But he hasn't actually done anything which you couldn't have done, has he?' Slowly the truth seemed to dawn on this woman and when it was suggested that the excitement and enthusiasm of the children might be taken back to the classroom to inspire some away-from-the-computer work it was like a revelation to her. Whether or not she did anything I know not, but she did, at least, take the adventure's manuals away with her. The specialist did little more than grunt when informed that she seemed to be taking an interest.

An all-age girls' school

The specialist in this establishment was female and was responsible for using the micros with all the girls from five up to whatever age they completed their education. During my hour's stay the room was unused except by a single girl who was in need of 'extra' attention, and was getting it by using an adventure on her own. I managed to stop myself thinking that if the machines were actually in classrooms at this time they would probably be in use. The specialist told me that because she was responsible for computing throughout the school she had designed a heavily-structured programme for 'computer-awareness and use'. She introducted different sorts of computer applications at different stages in the girls' development. It occurred to me that her programme probably looked wonderful on paper and would impress everyone who didn't have much idea about how computers could be used effectively in education (and that meant everyone else connected with the school because *she* was the expert and they left it all to her). I suggested (in a bold moment) that she must have had great difficulty deciding at what age to introduce the use of word-processing because while a 16-year-old's language skills were probably more advanced than a five-year-old's, children of both ages were, in fact, able to use the facilities to create something worthwhile. This, she admitted, had been a bit of a problem, but it had to go somewhere on her programme, didn't it? Feeling a little more confident that I wasn't upsetting her too much I then (acted out the pretence of having a brainwave and) suggested that the same thing was probably true of LOGO and databases. There was a hesitant affirming nod from the good lady. I pressed on and said that as far as I was concerned one of the benefits of having word-processing facilities available on all these machines was that quite large numbers of children could come in and produce some written work connected with normal classroom activities. That sort of thing had yet to happen, however, because 'the other teachers are a little apprehensive about using the equipment'. I said that that didn't really matter though, because I was sure that instead of getting the girls to copy out irrelevant passages of text she inspired their creativity first – perhaps by letting them go outside and stretch out on the grass to dream. I think that was the final straw. 'What on earth would *they* say if they saw me doing that!' I'm not sure who *they* were but I reckon they were probably all those people who had been led to believe that computing is a deadly serious business and computer specialists must spend all their time with computers.

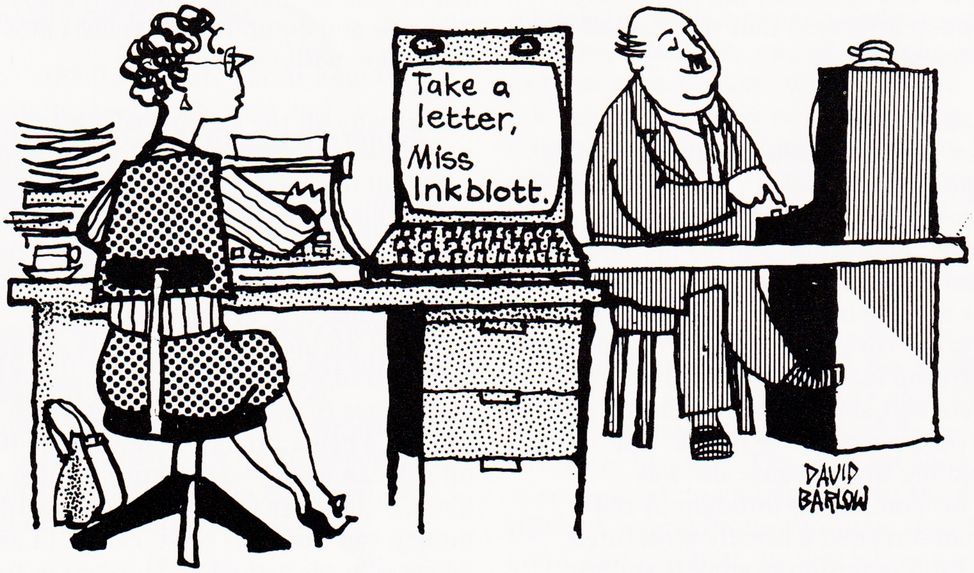
In virtually all the schools which I visited the picture was the same. It was only in the school which didn't have a computer room (but was networked) that the computers were being used by all the staff. Almost without exception the specialist was the single individual making 'educational' use of the micros.

I kept asking myself why this situation existed and decided that it was possibly an abundance of available funding which was the cause. (When a primary school with 400 children on roll can have two deputies earning something like £24,000 per year and there are 14 BBCs, money can't be that tight, can it?) I assume that where schools had decided to 'go in for computers' they had been advised by commercial interests that a network would give them the best value for money and it would be much cheaper to install it in one room. Besides, if the school is going to have an additional member of staff to be responsible for the use of micros it would seem illogical to do anything other than install all the micros in one room. I really don't think it occurred to anyone to do things any differently.

The result of the computers being centralised is that (a) teachers are reluctant to make use of them, and (b) their use is isolated from normal educational activities. From a teacher's point of view if a school requires a specialist for computers then it is obvious that the use of computers requires a specialist and if I am not the specialist then I am not sufficiently qualified. Additionally the existence of a number of machines clustered together in one place is far more intimidating to the average teacher than a single machine in the classroom (or at home for the weekend). One teacher did say that if she could have a micro in her classroom she would be much more interested in trying to make it do something, but there was no chance of her making a journey to the computer room where there are '****** computers everywhere'. From the specialist's point of view: it is not in my interests to encourage the other teachers to use the equipment because, after all, *I* am the expert and if they all get in on the act I won't be. Several did admit to me that the idea of computers in the classroom was quite attractive but not if it was to be at the expense of the computer room. In only one school did I find a specialist who thought that his role was to 'do himself out of a job' by encouraging the rest of the staff to make full use of the facilities available. It was possibly not a coincidence that he was intent upon finding employment elsewhere.

When the expert is the sole person making use of the micros and when the computers are in a separate room it is inevitable that there is going to be a minimum of integration into the curriculum. It is a chronic misuse of these powerful resources when they are used purely for their own sakes. I believe that we are fortunate in the UK in that we were not given the opportunity to go in for micros in a big way. The introduction of computers may appear to have been sudden but in terms of computer power per pupil the micro advance has been at a comfortable rate. I am quite sure that if British teachers had suddenly found that computers had taken over their schools (as the introduction of a computer room would have appeared to be to many), then we would not have the numbers of computer-orientated teachers which we now have. And is it not true that there is a higher proportion of primary than secondary teachers making use of micros – presumably because the secondary schools have experts? So yes, let's have networks if they really are going to be an advantage, but let's not be tempted either to stick all the machines in the same room or to employ someone to look after the network and its users all the time. I am in full agreement with the MAPE Executive's view that there should be, '*the minimum of restriction or of technical expertise on the part of the children and teachers*'.

I have a feeling that questions about the educational (and economic) value of networks in the primary school are going to become more commonplace and I would certainly be interested in hearing of other people's opinions and/or experiences.



Where the grass is greener

Andrea Tapsfield
Newman College

'Why did you call it *Grass*?', is the first question that people ask about the latest software to come from Newman, rapidly followed by, 'why Information Retrieval, aren't there lots of programs available for that?'

Yes, there are lots of information retrieval and database programs available such as *Factfile*, *Quest*, *Scan*, *Inform*, or*Dataprobe*, to name but a few of those most commonly used in primary education. But does that really mean that everyone is catered for by the choice available? If they are, why are not more primary teachers actively using information retrieval in their classrooms? After all, *Factfile* was given to all schools in the *MicroPrimer* scheme and many LEAs have agreed licence arrangements for *Quest* or *Scan* or *Inform*, which give schools cheap access to this software. Moreover, data handling programs are 'content free' and thus provide some of the most flexible software that can be incorporated into a wide range of topics taught at primary level.

Do you use a database program in your school, with your class? If not, why not? Do you not consider this to be a good use for the school micro, since it will surely help to provide our children with the 'information skills' they will need for the future?

We frequently use information retrieval packages at Newman. We introduce all our teacher training students to *Factfile* and *Picfile* and run courses for in-service teachers on *Quest* etc. We use *Quest*, too, for our software library. What has this contact with database packages taught us?

We have found that adults find database programs difficult to use. There has not been any rigorous academic research at Newman into the topic, but observation of a variety of teachers and prospective teachers has led us to the definite feeling that many teachers and teacher training students probably lack the information skills that everyone says they should be teaching to their pupils. This is not very surprising, since there has been very little in-service training to introduce these skills – the courses run by LEAs and MEP have really only scratched the surface.

Why do teachers avoid information retrieval? The first reason must be that the educational purpose of using a database program is not always so immediately obvious as, for example, using a simulation or word processing package. It is easy to identify knowledge or facts to be searched from a database, with no real thought behind the educational validity of making that enquiry, or the educational reason for the process. Those of us who consider ourselves as possessing information skills and as having some prowess with a database package, are all guilty of demonstrating the most puerile activities with complex information retrieval programs – the 'Look, I can find all the dinosaurs which are herbivores and have four legs, even if I can't pronounce their names', syndrome. Information retrieval will only 'take off' in the primary school when we harness our information skills to begin to ask *meaningful* questions, that are related to our current curriculum.

The big handicap to convincing people that information handling is worthwhile in the classroom, is the shortage of available databases. The historians have made the greatest contribution here, but local history databases do not travel well, and they take a long time to compile with sufficient records to give a real insight into the social history of 1851. What is lacking is a series of databases that would form a core of topics commonly covered in junior school classrooms. *Dino* was a start here, but why are there so few others? Such datafiles would provide an additional resource for topic work and be seen by children and teachers alike to be a worthwhile activity – introducing and re-inforcing information skills in the manner in which we might expect the normal information search to be done by the year 2000. With such a bank of datafiles available, teachers could begin seriously to consider techniques of interrogation.

The advantage of starting with an existing datafile on a subject that suits a topic you are following in class, is that the purpose of the use of the database is clear. Its value is demonstrated by a genuine search for information. You obviate the problem of creating a file for a package with which you are not familiar and, therefore, do not know what questions it will allow you to ask.

Educationally, it is considered more worthwhile for the class to collect their own information, enter it into the database and then use their own data for analysis. In practice, the unwary teacher can create for herself multivariate problems unless she is completely knowledgeable about the package and its facilities. It is all too easy to create a datafile and then find that, because of poor choice of fieldnames, or the omission of a key field such as that containing the name of each record, it is impossible to search on the questions you want to ask. This has been done many times – and the frustration on the part of the teacher and the children might explain why databases are not used as often as we might hope. If you have no in-service support, or a helpful primary co-ordinator to turn to, my advice is to use a created database first, and when you feel you can handle the package, embark on your own class file.

If you begin with an information handling package and a ready made datafile, and even with some questions to start the interrogation, you soon reach the second hurdle – the jargon. In an attempt to be accurate, information handling has a vocabulary all of its own – fields, records, archive, and, or, format, strings, sub-strings etc. – which can send any teacher back to *Granny's Garden* with a rush. But you persevere, only to find two more problems lurking in wait for you: loading in the datafile and remembering all the commands.

The file loading is no problem if you have a disc drive. If you are one of the schools who have never got beyond the standard DTI 'Micros in Schools' package, and you are still using a cassette recorder, you are setting yourself a difficult task and might be better advised to leave database programs until you have access to a disc drive.

So then you get to the commands. We are told by the computer scientists (who still have an uncanny way of 'knowing' all about primary classrooms), that all flexible packages must have command structures. But does this approach make them 'user friendly' for the primary school pupils who want to find out some information for their topic? Commands are fine when you know what the possibilities are, because they then open up all avenues of enquiry – but what of the beginner?

Hence the emergence of *Grass* from the Newman stable. We had watched, at length, the problems student teachers have with the existing packages. We had talked to practising teachers about their needs, and we had experience of our own data handling package, *PQuery*, written some years ago for the RML 480Z and now available on *MAPE Tape 2*. We felt that it was time for a new package to solve some of the problems we had identified, and one that would suit the needs of the primary school child.

Grass is a menu-driven program, so the busy teacher, or child, does not have to remember a set of commands. We have also tried to keep keyboard input down to a minimum so that you make your choices with the cursor keys, space bar and return, until you get to the specific element of your question. We wanted to run the program with a mouse to make it even easier to handle – but although it is likely that a mouse version of *Grass* will be developed, few primary schools have the facility yet.

Grass gives children every possible assistance when they make their enquiries and it helps them remember what they have asked. There is a 'question box' always on the screen, even when displaying graphs or listing records.

Conditions are chosen from a menu, which offers the child possible alternatives, so that they can't make a mistake such as searching for a word in a numeric field. Above all, it is fast and flexible so that children will not lose interest while the computer is searching, sorting or displaying.

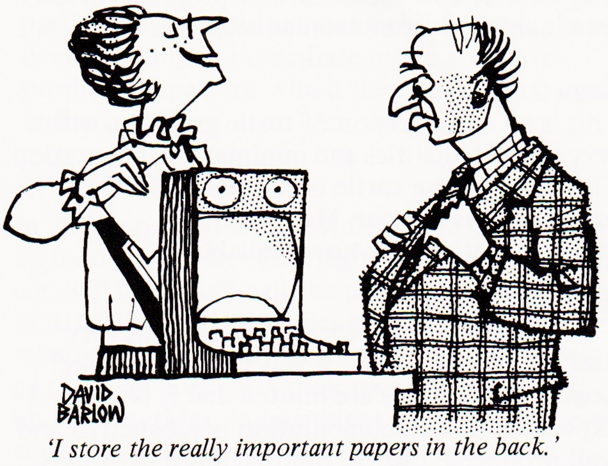
Grass makes information retrieval appealing to the busy teacher because it is simple to use. The program is only on disc (for both the BBC and RML), and at the start of an interrogation the files on your disc are displayed on the screen for you to choose from. At any time you can set up the printer, or change the datafile, or look at some graphic displays, without leaving the program or changing the disc. It is impossible for children to leave the editor program without saving their datafiles. The program has been written to be friendly, fast, flexible and forgiving. We hope it will encourage many teachers to give information handling its rightful place in their curriculum.

So, finally, why did we call it *Grass*? There is no prize for guessing the correct answer, although Newman College has received a number of suggestions! *Grass* stands for GRAphics, Searching and Sorting. As well as all the usual searching and sorting functions, *Grass* will display pie charts, count graphs, scatter graphs etc. This enables children to ask a question and have the results displayed in diagrammatic as well as list form – and both can provide printed copy. We wanted a database program that would do this within the package, without the need for changing programs, because the educational purpose of information retrieval is to search for, and find answers to, questions through using a variety of data formats. Graphs and sorted lists are the means to that end and should be integrated within the database package.

At Newman we would like to hope that this *Grass* package does make a brighter (if not greener!) outlook for information retrieval in primary schools. We hope that this is a package that will encourage more teachers to use information retrieval. We have used our experience to try to develop a package to fulfil the needs of the eight to twelve year-old who is facing the first steps towards being a member of the 'information generation'. What we need now are datafiles. As I said earlier, there are advantages in approaching information retrieval via an existing database. So this article is really a plea to MAPE members for help. If you have any suitable datafiles that you have developed for *any* database, or alternatively any ideas for suitable topics, please let me know.

Reference

Grass is available for RML 480Z and BBC micros, in disc and network versions. It is available on LEA licence and individual schools wanting a copy should in the first instance contact their LEA.



'I store the really important papers in the back.'

Turtle Graphics packages

Maureen Blake, Howard Gillings and **David Griffiths**
Headteachers and members of the Berkshire Computer Support Team

For some teachers, turtle graphics packages provide a unique opportunity for children to develop skills of spatial perception and language. For others, turtle graphics is the acceptable face of programming in the primary school. Whatever the interests of individual teachers, it is likely that most primary schools will have to choose a single turtle graphics package which:

1. satisfies both the needs of infants, who are still coming to terms with concepts of direction and estimation, and
2. satisfies the needs of juniors, who are developing problem-solving skills and an understanding of the sequencing of instructions.

In assessing turtle graphics packages, then, it is necessary to look for simplicity and ease of use, but also for a language which provides a good grounding for more advanced work, possibly using a full implementation of the *LOGO* language.

Any implementation of turtle graphics should include functions which make good use of the computer's capabilities. Perhaps the most obvious of these is the use of colour. In the BBC machine use of certain modes drastically reduces program memory space. However, even with this drawback, quite reasonable use of colour can be made whilst preserving graphic quality. Coupled with the use of colour there is the possibility of enclosing an area of the screen with a predetermined colour.

A facility to include sound is another advantage. Again the capabilities of the machine can be utilised, and definitions of pitch and duration can be included.

For younger children, in particular, it is important to provide a transition from the concrete activities of 'playing turtle' or working with a toy like BigTrak, to the abstractness of the computer screen. Floor turtles provide the ideal intermediate step and, as their price continues to fall, more and more schools will be able to afford them. It will, therefore, become increasingly important to have good turtle graphics packages which provide turtle control.

Turtle graphics packages for the BBC computer

There is a large number of tur+le graphics packages available, some of wnich are very new and others which were designed two or more years ago. Initially we considered all the packages which were available in February 1985:

Acornsoft Turtle Graphics
Dart
Delta
Honeylogo
Logo 2
Logo Challenge
Nectarine Logo
OK Logo (supplied with the Jessop turtle)
S-Logo
Turtle (Open University Awareness pack)
Valiant Logo (supplied with the Valiant turtle)

It soon became clear, however, that we could disregard some packages and programs, either because they had been superseded or because they offered few facilities. The programs below were excluded from detailed analysis.

Logo 2
This is an early version of turtle graphics, with very limited facilities and minimal documentation. The shape of the turtle is confusing, as its orientation is unclear. Many better implementations are now available.

Logo Challenge
This is another early version, which was very useful at the time that it was published. Like *Logo 2* its facilities are limited and it is very expensive. The documentation is substantial and well produced, but the approach is particularly didactic, and not at all in sympathy with Papert's child-centred exploratory approach.

Nectarine Logo
This is an extended implementation of turtle graphics, with REPEAT . . . WHILE loops and the ability to draw lines in colour. There is also a good range of logical tests and there is a decidedly mathematical approach throughout. In practice the language was difficult to use and the documentation was confusing and inadequate. It was felt that this program was not appropriate for the primary school.

Turtle
This is part of an Open University computer awareness pack and is not, therefore, intended to be a full implementation of turtle graphics. The documentation is adequate, but the facilities offered are severely limited. Unfortunately, whilst the abbreviations used are standard to the BBC, RML, Apple and Spectrum versions of Turtle in the OU Awareness pack, they are non-standard as far as most other Turtle graphics packages are concerned. FORWARD, BACKWARD, LEFT and RIGHT are replaced by M (move) and T (turn).

N.B. This program should not be confused with the Open University Logo package, which is a full implementation of *LOGO*.

OK Logo and *Valiant Logo*
These are small, simple programs provided with the Jessop and Valiant turtles. Each is adequate for simple control of a turtle, but a fuller version of turtle graphics is needed for children to benefit from the power of the computer.

A detailed analysis

Our final shortlist of packages for consideration thus consisted of the following:

Acornsoft Turtle Graphics (hereafter referred to as *Acornsoft*)
Dart
Delta
Honeylogo
S-Logo

For each of these packages we asked the following questions:

- is the program easy to use?
- does the program provide a good range of facilities?
- how good are the editing facilities?
- is it possible to use colour?
- is it possible to use sound?
- can a floor turtle or buggy be connected?
- how good is the documentation?
- is the package expensive?

Of course these questions are not exhaustive, and there are no packages that scored full marks on all counts. However, the questions provided a starting point for analysis and our findings are described in the sections that follow.

Starting the program

Most of the programs are loaded from disc by auto-booting, using Shift/Break, or can be loaded using a command such as CHAIN "TURTLE" or *LOGO.

Delta uses both a chip and a disc; the program can be auto-booted in the usual way, or run by typing *DELTA.

Screen display

Most versions give a fairly clear and straightforward screen display. In the usual 'split screen' mode, *Acornsoft* commands are displayed on the bottom four lines of the screen, with the rest of the screen being available for drawing.

Dart and *Delta* have similar screen formats to each other, with a roughly square drawing area to the left of an information panel, and two lines available at the bottom of the screen for commands.

Honeylogo is a little confusing for beginners as the initial command is entered at the top of the screen, with no turtle visible. Only when the first command has been entered does the turtle appear, and the input prompt moves to the bottom of the screen. At this stage four lines are available for instructions, with the rest of the screen for graphics. We understand however that future versions of the software will be modified to begin with the split screen.

S-Logo is rather different as, initially, the user may choose a number of options – the drawing mode, the colours, the scale of drawing and the design of any sprites required. Once selected these cannot be changed from within the main program. In the main program a text area of four lines is given at the bottom of the screen, with the rest of the screen for graphics.

Giving commands

Almost all versions allow commands to be input in full (e.g. FORWARD) or in an abbreviated form (e.g. FD or FO). Unfortunately abbreviations are not standardised, and there is little similarity between any of the implementations. *S-Logo* requires any abbreviation to be followed by a full stop. Most versions require a space before any following value – thus FORWARD50 would usually be ignored or rejected.

Delta uses a completely different approach. Only the first two letters of any command are required, and the program then automatically completes the keyword on the screen. In effect, two-letter commands are compulsory, but these are changed into full commands by the machine. A following space is provided automatically and erroneous characters (e.g. typing FORWARD +) are trapped and a warning beep given.

In immediate mode in *Acornsoft* (i.e. when not defining a procedure), REPEAT loops are not allowed.

Honeylogo places a 15-line maximum on procedures, but as multi-statement commands are allowed on each line, this is not a significant problem.

S-Logo does not allow loops to be nested. This can, however, be overcome by defining one loop as a procedure and then including that procedure in another loop.

Creating and editing procedures

Almost all the versions move to a full non-graphics screen to allow procedures to be defined. The exceptions are *Dart*, which uses a narrow panel at the right of the screen, and *Honeylogo* and *S-Logo* also allow instructions to be copied and altered without having to go through the tedious process of deleting and retyping.

In *Acornsoft*, lines for editing must be referenced using line numbers which are displayed at the editing stage. Whilst this is quite satisfactory for short procedures, it is not appropriate when a procedure extends beyond one screen. This approach is especially difficult when more than one line is edited, as the line numbers of the instructions then change.

Procedures for editing in *Honeylogo* have to be prefaced with a " – for example EDIT "SQUARE. As the rest of this program can cope with references to procedures without the ", this seems an unfortunate quirk of syntax.

Delta has a facility which enables procedures to be created retrospectively. The command KEEP, followed by a procedure name, causes the last 20 or so instructions to be displayed. Any number of these instructions can then be turned into the new procedure. This facility is particularly useful for young children who have been working in immediate mode and have produced a pattern which they wish to use again in the future.

Using procedures

In all the programs except *Delta*, procedures can be used simply by referring to them by name, e.g. SQUARE will call the procedure named SQUARE. This is a particularly useful feature when enabling children to build 'microworlds', as they can construct their own sets of words and instructions, independent of the language of the turtle graphics program being used. One consequence of this approach, however, is that some potential procedure names cannot be used as they would conflict with standard instructions – for example, a program which allowed PU (for Pen Up) could not allow the procedure names PUDDLE or PUPIL. The error-trapping in *Delta* means that it is not possible to call procedures simply by giving their names. Instead, the command DRAW (or, synonymously, CALL) is used with the procedure name.

In all versions except *S-Logo* parameters can be passed to procedures. Thus it is possible to define a procedure to draw a square of any size, and then use an instruction like SQUARE 100 to draw a square measuring 100 by 100.

Use of colour

All versions except *Dart* allow colour to be used, but the extent of the colour and its ease of use vary considerably.

In *Acornsoft* the number of colours available depends on the mode chosen. The disc-based program will work only in mode 4 or mode 5 unless a 6502 Second Processor is being used. This greatly affects the number of colours available at one time, and the fineness of the drawing. The program starts up in two-colour mode with a black background and a white foreground. Any of the eight available colours can then be selected to replace these, using a VDU command from BBC BASIC.

Selecting mode 5 allows up to four on-screen colours, but gives a much 'chunkier' display of text and graphics. Colours are then selected using the command COLOUR with a number.

Delta allows up to four colours (including the background colour) to be used at a time, from a palette of eight colours. The current pen colour is then chosen using the command COLOUR with either the name of the colour or a number. The pen colour and current colours in the palette are displayed on the screen. Colours in the palette can be exchanged for other colours using the command CHANGE. Thus CHANGE

RED TO BLUE would change any red lines or areas into blue ones. A colour can be changed into the background colour and vice-versa, enabling lines to disappear and reappear.

Honeylogo allows two colours on the screen at a time – the pen colour and the background colour. These colours can be chosen at will from eight colours using the commands SETPC and SETBG, each followed by a number.

With *S-Logo* the mode, and hence the number of colours available, have to be selected before loading the main program. All eight colours are available, but only two can be used in mode 4 and four in mode 5. The command PEN followed by a number selects the current pen colour.

Acornsoft and *Delta* allow closed areas to be filled by moving the turtle into the area with the pen up and using the command FILL. This option is particularly popular with children and can greatly enhance a completed design. *Delta* also allows filled areas to be refilled in a different colour.

S-Logo has a similar fill command, but the turtle must be moved to the very top of the area to be filled, and then pointed down the screen. Simple shapes can then be filled, but any irregularities are left unfilled.

Use of sound

All the versions provide at least a single-note toot, and all except *Dart* allow children to choose the pitch and duration of notes.

Acornsoft provides a full range of sound facilities, although these are available only by using the BBC sound and envelope commands. Thus a sound command has to include four parameters (numbers) and an envelope command 14.

Delta allows a predetermined BEEP, and also a BEEP followed by two parameters, specifying pitch and duration. The pitch is given as the name of the note (A, B etc.), with C′ and C, being used to denote higher and lower octaves. The keys ‘ and b are used to denote sharps and flats, which appear on the screen in their usual musical form. Duration is given in tenths of a second.

Honeylogo also provides a predetermined PIP and a flexible PLAY command. PLAY requires 3 parameters – for note, octave (there is a default value) and duration. As in *Delta* notes are referred to by letter, and the octave is specified using a number such as 3, 4, 5 etc. In the absence of a number *Honeylogo* dafaults to 4. The duration is specified by using SHIFT with the numbers 1 to 9, to give the characters ! " ' etc. A note having duration 3 is three times as long as a note having duration 1. A useful feature is that sequences of notes can be put in a single PLAY command, for example PLAY C6;A4;B.

In *S-Logo* the instruction PLAY, followed by two numbers for pitch and duration, produces a simple note. Apart from a reference to BBC BASIC, the documentation gives no indication as to what the numbers represent.

Documentation

It seems clear that most publishers of the turtle graphics packages under consideration appreciate the importance of good documentation. With the exception of *S-Logo*, all the programs came with documentation that was at least adequate.

Acornsoft provide a 70-page book which is well produced and includes colour photographs, diagrams and listings of sample procedures. The book is well written and friendly, and provides an excellent introduction to turtle graphics.

Dart has a 26-page booklet, giving a practical introduction to the language. The documentation is by no means extensive, but it provides a friendly introduction, with examples and diagrams.

Delta comes with a ‘Walk-through Guide’ which provides an easy introduction to the use of turtle graphics, There is also a reference card which summarises the features of the program, and a Delta Program Guide which provides a statement-by-statement description of the language.

Honeylogo has extensive supporting materials, which include a 90-page book for pupils, with an annotated version for teachers. The book is supplemented with 21 workcards for pupils, and a *Guide for Parents and Teachers* which has over 250 pages. Whilst there are undoubtedly more exciting books available on turtle graphics, Honeyfold Software have produced a substantial and useful learning resource and in September 1985 this will be supplemented with an extension of *Honeylogo* into Primary Science.

S-Logo is supported with a booklet of 18 A4 pages. The documentation is very sketchy and rather confusing. Although there are references to the *S-Logo Users' Guide*, this was not available at the time of this review.

Using a floor turtle or buggy

As the price of floor turtles continues to fall, many teachers, especially those of young children, will undoubtedly consider it very important that a Turtle graphics program provides control of a floor turtle. Of the programs in our shortlist only *Dart* does so currently, although it is understood that both *Delta* and *Honeylogo* are to be enhanced to provide this facility.

Robustness

In addition to performing as expected when the correct instructions are given, it is important that programs intended for young children are sufficiently robust to cope with:

1. programming errors;
2. the use of unexpected keys at unexpected times.

We tried giving a number of incorrect instructions and pressing a variety of keys, in a number of random combinations.

Acornsoft, Dart and *Honeylogo* produced error messages which we felt were, in general, useful. Some of the messages in *S-Logo* however, were not particularly helpful; a message such as STACK FULL would have little meaning to the average user of a turtle graphics program.

Pressing the CTRL key with some other keys caused *Acornsoft* and *S-Logo* to crash irretrievably.

The version of *Honeylogo* which we tested was a pre-production one which did not function on any computer having an Econet or DNFS chip fitted. This problem has been overcome in the versions currently available and the publishers are now working on Network versions.

Other features

Users of turtle graphics programs are, of course, particularly interested in the turtle graphics features that are provided, and this report has concerned itself with these features, and with their ease of use. However, all the programs considered have additional features that are worthy of mention.

All the programs have facilities for performing arithmetic and for storing numeric values in variables. *Delta* uses integer arithmetic, so that a calculation such as 9/4 would give the result 2. All other programs allow decimal values to be calculated and manipulated.

All the programs except *Dart* have *some* facilities for printing messages and other text on the screen.

All the programs except *Dart* have IF structures, which allow certain instructions to be obeyed only if a certain condition is true. *Honeylogo* and *S-Logo* allow simple IF constructs, with conditional statements limited to one line. *Acornsoft* and *Delta* allow a number of lines of instructions to be placed in an IF block.

Acornsoft also allows loops to be terminated by using WHILE and UNTIL commands, which can be very useful in more advanced programming. *Honeylogo* provides a similar feature using REPEAT . . . UNTIL.

All the programs except *Acornsoft* allow graphics to be dumped to an Epson printer.

Dart and *S-Logo* allow the red function keys to be used to represent instructions, thereby reducing the amount of keying required by pupils.

Delta and *S-Logo* have a SCALE command which causes turtle movements to be multiplied by the specified value. This is useful for allowing infants to give instructions such as FORWARD 2, whilst producing a substantial movement on the screen. *S-Logo* allows a scale to be set up before the program is run, but this cannot be changed while the program is being used.

S-Logo supports 'sprites' – these are characters defined before the main program runs, which can then be displayed at the turtle position using the DISPLAY command.

Finally, *Honeylogo* allows the use of logical conditions (True and False) and provides extensive list-handling facilities, well beyond the scope of other turtle graphics programs.

Speed

In our evaluation of the programs we did not pay specific attention to speed of operation as it is our view that the ease of use of a turtle graphics program is much more important than its speed. Our own experiences have shown that children need to be quite advanced in their use of turtle graphics before speed becomes important; even then, they seem quite happy to wait whilst a program performs a complicated procedure, or even to go away and return a few minutes later.

Despite these views we carried out two 'benchmark' programs, to enable the speed of the various packages to be compared. Benchmark 1 drew a pattern of 10 squares, each separated by a rotation of 36 degees. Benchmark 2 drew a circle, formed by moving forward and turning one degree, and repeating these operations 360 times.

The results are summarised below, with the times being given in seconds.

	Benchmark 1	Benchmark 2
Acornsoft	1	4
Dart	13	97
Delta	15	87
Honeylogo	37	270
S-Logo	28	195
Arrow (see below)	42	243

Conclusions

The choice of a turtle graphics program depends on many factors, some of which will be peculiar to a given school. For this reason it is not appropriate to suggest a 'best buy'.

It is our view, however, that the packages mentioned below are worth considering particularly seriously.

Acornsoft

This program provides a reliable implementation of turtle graphics, with some use of colour and a FILL command. The screen display when defining procedures is clear and the documentation is good. The speed of drawing can be varied, and can be very fast.

The program is not entirely robust – pressing CTRL with some other keys can cause it to crash. The editor is cumbersome, particularly with its reference to line numbers.

Delta

This program provides a particularly easy-to-use implementation of turtle graphics, with good error-trapping. It is the only program to support four simultaneous colours with reasonable graphics, and it offers a FILL command. Notes can be produced easily and there is a useful facility for retrospectively defining procedures. The screen display when defining procedures is clear, and the editor is easy to use, if a little cumbersome. The graphics area can be dumped to an Epson printer.

A drawback is that the program requires a computer fitted both with a *Delta* chip and a disc interface, as the use of a disc drive is essential. The program executes procedures more slowly than *Acornsoft*. Procedures are called using the command DRAW, rather than being able to refer to procedures by name only. *Delta* can only handle whole numbers and so calculations such as 360/7 do not produce wholly accurate results. Whilst arithmetic can be carried out, only two numbers can be handled at a time; thus adding three numbers requires two separate additions to be performed.

Honeylogo

This program provides the greatest number of facilities, and is the only one to support list-processing. The quality of the screen drawings is adequate, and notes can be produced easily. The editor is particularly convenient to use.

The program can only support two colours at a time, and its speed of execution is relatively slow. The initial screen display, with no turtle visible, is somewhat disconcerting. Difficulties in using the current version of the program with some computers also give cause for concern.

Dart

Dart has far fewer facilities but it is available very cheaply in many LEAs. It is one of the few programs to provide control of a floor turtle and it is quite fast, as well as being robust and easy to use.

A note for RML users – *Arrow*

Users of the RML 480Z computer are spared the problem of choosing between rival versions of turtle graphics programs. For them the program *Arrow* is generally available and provides an acceptable implementation. We used version 2.0 of *Arrow* for this review. In this version there is no facility for making sounds.

Arrow is in some ways similar to *Dart*, having many of the same commands and allowing the option of using the function keys to represent the most common commands.

Arrow is loaded from tape or disc and run in the usual way. The title page asks the user to type an instruction and the option is given to obtain a list of all acceptable instructions. Four lines are allowed at the foot of the screen for instructions. Like *Delta*, error messages are given at the foot of the screen (*Dart* gives messages at the side of the screen).

One interesting difference from *Dart* is that, as soon as a REPEAT loop is started, *Arrow* immediately moves to a full instruction screen. This has the advantage that the set of instructions can be viewed as a whole, but has the disadvantage that the effect of the instructions cannot be seen immediately. When the REPEAT loop is complete, *Arrow* returns to the turtle screen – unlike the BBC turtle graphics programs, any previous drawing remains on the screen and is not lost.

The speed of response, when instructions are entered, is quite slow, although it is possible to 'type ahead', thereby partially overcoming the problem. Error messages are adequate, although not always as helpful as they might be.

Procedures can be edited in a similar way to *Dart*, and the full screen is used to list the procedure. In *Arrow*, instructions may be abbreviated to their first three letters; usually *Arrow* takes these abbreviations and turns them into full instructions at the same time as the syntax of the instruction is checked. Procedures are called simply by using their names. However, the use of three-letter abbreviations (FOR, CLE, COL, etc.) means that many procedure names are not acceptable.

One very helpful facility is that a procedure can be renamed. When this takes place, all references to that procedure name are changed.

Hence it is not necessary to edit other procedures which refer to the changed procedure, as this happens automatically.

A significant advantage is that three colours, plus a black background, can be used to produce high quality drawings. The way of changing colours is a little unusual, however – the three colours available at any given time are represented by INK 1, INK 2 and INK 3, and any one of these colours can be changed for another by using the COLOUR command. Hence INK 2 followed by COLOUR RED would change INK 2 to RED. A total of eight colours is available in the palette. There is no fill instruction for colouring enclosed areas.

Arrow appeared to be extremely robust. We did not succeed in producing any untoward effects by pressing keys at random. The program supports the use of a floor turtle, and we were pleased to see that *Arrow* tests for the presence of a turtle when the TURTLE ON command is input, giving an appropriate message if a turtle is not connected.

Other useful features include an IF . . . THEN . . . ELSE facility, which enables instructions to be obeyed only if a certain condition is true. However, multi-line instructions are not allowed within the conditions. Simple arithmetic can be carried out using the MAKE instruction.

There is also a SCALE instruction, which multiplies all movement by the scale factor and can be applied differentially in the horizontal and vertical directions.

The last line drawn can be erased by means of the RUBOUT command, which can be applied up to 20 times in succession. Finally, the contents of the screen can be output by a printer using the PRINT command.

In conclusion, *Arrow* provides an implementation of turtle graphics which is comparable to most of the BBC versions. Its particular strengths are its support of colour and its ease of use. Unfortunately it is very slow, both in checking the correctness of instructions and in drawing, and this can make it a little frustrating to use. In addition, it lacks a sound instruction and the ability to fill an enclosed area. Despite these drawbacks, RML users can feel happy that they have got a turtle graphics program that is adequate for most introductory needs, and provides a suitable introduction to further work using LOGO.

Appendix – summary of instructions

All versions allow:

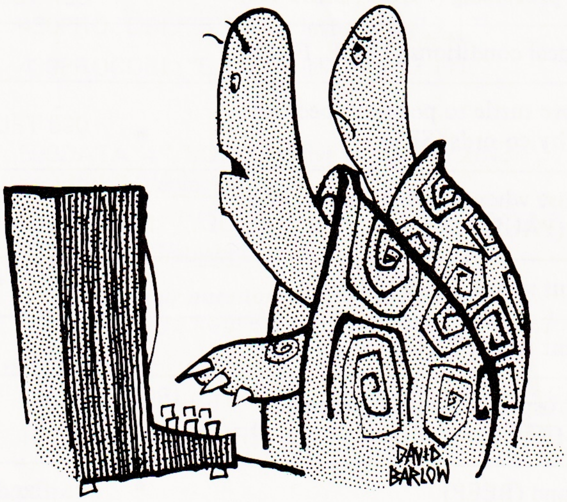
- movement forward and backward;
- turns left and right;
- creation and editing of procedures;
- repeat loops;
- clearing the screen and returning the turtle to home;
- saving procedures to disc/tape;
- hiding the turtle;
- pen up and pen down.

In addition, the features shown on the table overleaf were also provided (the name in brackets indicates the typical instruction name).

Notes

1. *Delta* is produced by Berkshire. Further details from Colin Monson, Computer Adviser, Department of Education, Shire Hall, Shinfield Park, Reading RG2 9XE.

2. This report originally appeared in the MEP Primary Project LOGO Pack. At that time the Walsall Turtle chip was included. As this chip is no longer available, references to it have been deleted.



'Beats me! It says it's something with four legs, a tail, and a shell!'

Summary of instructions

	Acornsoft	*Dart*	*Delta*	*Honeylogo*	*S-Logo*	*Arrow*
alter procedure name (ALTER)			*			*
arithmetic (MAKE)	*	*	integer	*	*	*
change scale (SCALE)	normally		*		normally	*
colours available (plus background) (COLOUR)	1 from 8		3 from 8	1 from 8	1 from 8	3 from 8
colour fill (FILL)	*		*			
conditional statements (IF . . . THEN)	*		*	*	*	*
(REPEAT . . . UNTIL)	*			*		
erase last line (UNDO)			*			*
floor turtle		*				*
give current turtle position (WHERE)	*		*	*	*	
input a number (READ)			*	*	*	
list available instructions (HELP)		*	*			*
list current values (VALUES)		*	*		*	*
list processing (FIRST, LAST etc)				*		
logical conditions (TRUE, FALSE)				*		
move turtle to position specified by co-ords (SET)	*		*	*	*	
pause when obeying instructions (PAUSE)	*		*	*	*	
print screen image (DUMP)		*	*	*	*	*
print text (PRINT)	*		*	*	*	
retrospectively make procedure (KEEP)			*			
sound (BEEP)	*	fixed sound	*	*	*	
sprites					*	
	Acornsoft	*Dart*	*Delta*	*Honeylogo*	*S-Logo*	*Arrow*

'Ani thing' you wish

Reg Eyre
College of St. Paul and St. Mary, Cheltenham

The program *Animal*, which is in the Micro Primer Pack, can be made to fit into virtually any project where the teacher decides that its use is relevant.

Most topic and/or project work includes elements of data collection, sorting and presentation. These areas are ably catered for by programs such as *Factfile, Picfile, Inform* and *Quest*. We use variations of *Animal* to see if children can discriminate between items of information by asking relevant questions based on knowledge gained from the data collection process.

Ideal variations are for example: *Vehicle* for a transport project; *Anibird* for a bird project; *Anitree* for a tree project; *Flower* for a project on flowers. Other variations have included buildings, jobs, shapes, chemicals and food.

Some care has to be taken when planning this type of work. While doing a project on food, we left a version called *The Food Game* available for the children to work on unattended. The program started with the items 'potato' and 'pear', and we expected that the children would continue the list with other fruit and vegetable produce. No way! When we looked at the list at the end of the day, we saw chips, crisps, ravioli, spaghetti, etc., with a lot of interesting questions to distinguish between the foods. In this case, perhaps we should have provided a list of foods and asked the children to tick off from the list those foods they had entered.

What follows are the lines of the program which need changing to convert the original *Animal* program (on BBC), to anything you wish. I have shown the original line and the changed line, with the changes underlined.

```
LOAD"ANIMAL"
```

[From tape or disc]

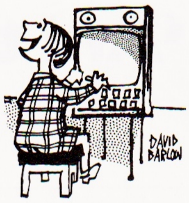
```
LIST 130
  130PROCcentre(CHR$C%+"The",9) :PROCcentre
  (CHR$C%+"Animal Game",11) :PROCspace(21)
  130PROCcentre(CHR$C%+"The",9) :PROCcentre
  (CHR$C%+"Bird Game",11) :PROCspace(21)

LIST 360
  360PROCcentre(CHR$C%+"ANIMAL",1)
  360PROCcentre(CHR$C%+"BIRD",1)

LIST 390
  390PROCDBL("1. List the animals",7,5)
  390PROCDBL("1. List the birds",7,5)

LIST 400
  400PROCDBL("2. Let me guess your animal",7,8)
  400PROCDBL("2. Let me guess your bird",7,8)

LIST 410
  410PROCDBL("3. Save the animals on tape",7,11)
  410PROCDBL("3. Save the birds on tape",7,11)

LIST 430
  430PROCDBL("5. Delete last animal",7,17)
  430PROCDBL("5. Delete last bird",7,17)

LIST 590
  590DATA"4","/QDoes it live in water/Y2/N3/","/
  agoldfish","/ablackbird"
  590DATA"4","/QDoes it have a red breast/Y2/N3/
  ","/arobin","/ablackbird"
```

[Put in your question with no extra spaces; remember to keep exactly to the same format.]

```
LIST 1280
  1280PROCDBL(CHR$F%+"Think of an animal",9,8)
  1280PROCDBL(CHR$F%+"Think of a bird",9,8)

LIST 1420
  1420PROCDBL("Please tell me your animal.",3,6)
  1420PROCDBL("Please tell me your bird.",3,6)

LIST 1950
  1950PROCcentre(CHR$C%+"Animals I know",1)
  1950PROCcentre(CHR$C%+"Birds I know",1)

SAVE"ANIBIRD"
```

[Remember to save your changed version with a suitable name.]

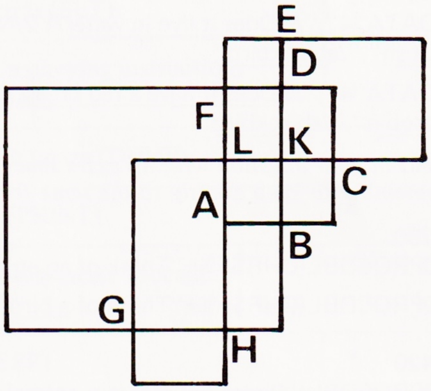
Children's competition



This is the first of a series of regular competitions related to commonly available software, for this issue three maths programs in the Micro-Primer pack: *Watchperson, Eureka* and *Ergo.* Entries should be from individuals, aged 12 or under, and accompanied by a teacher's signature to confirm that the work is solely that of the pupil. Entries must be sent to the Editor (Comp), *MICRO-SCOPE*, Newman College, Genners Lane, Bartley Green, Birminghham B32 3NT by first post on Friday 13th December. The first all-correct entry selected will receive £50 of free software for his/her school from any titles produced by Newman College plus a £10 prize for the individual pupil. The runner up will receive £25 worth of free software and £5 for the individual. Note now that the next competition will be based on programs in the MEP Primary Project Maths Pack, namely *Boiled Eggs, Bounce* and *Blocks.*

Right, now for the questions:

Q1. Based upon *Watchperson.*
Write down ALL the solutions to this network given the starting point is A. The solutions will be in the form of a series of vertices visited and not compass directions. (e.g. A B C K . . .)



NB. This route plan is not one of the three in the program.

Q2. Based upon *Eureka.*
Sketch a graph to illustrate the story:

A man put the plug in the bath and turned on the tap. Suddenly the phone rang and he dashed off to answer it, forgetting all about the bath. This bath didn't have an overflow pipe so that when he returned there was water all over the floor. He turned off the tap and let some water out of the bath. Then he put the plug back in and immediately leapt into the bath. He hadn't let enough water out and so it overflowed again. Nevertheless, he had a quick bath and took the plug out. When the bath was half empty he got out of the bath.

Q3. Based upon *Ergo.*
Complete these 2 number squares:

	15			30
16				52

	16			
				50
		42		

Television review

The Learning Machine
BBC 1

The Learning Machine has ostensibly been a series about educational computing, but the real lesson to be learned from these programmes has not been about the microcomputer, but about another insidious and powerful 'learning machine', the media.

The main thesis of the series, described as representing 'the personal view' of Tim O'Shea, series presenter, has been that educational computing is a shambles, that schools are using microcomputers in a way that not only fails to capitalise on the best that technology has to offer, but is actually detrimental to education, and that the government is wasting our money on underpowered and underused machines.

In one respect Mr O'Shea is right. There is a great deal wrong with a system which allows the Department of Industry to make schools an offer they could not refuse, a half-price computer, with only a belated and inadequately funded response from the DES. Clearly the Ministers' extravagant claims for British superiority in this sphere needed to be shot down.

However, the 'truth' represented by some of the episodes in this series has been so severely distorted and the views expressed so poorly supported by argument, that there is a grave danger of it doing real harm, undoing much of the progress which has been made in educational computing, and providing fuel for the fires of the ill-informed Luddites. It will moreover do little for the flagging morale of teachers, who yet again are accused of not knowing their job.

As an example, take Episode One, much of which was filmed at a parents' evening at Courtwood Primary School in the London Borough of Croydon. This programme must have been a source of great confusion to many of those watching. On the one hand, Mr O'Shea was complaining about how badly computers were being used in schools. On the other, viewers were treated to pictures of children making powerful use of adventure games, word processors, and turtle graphics; parents enthusing over the learning potential of software, which was giving *them* plenty to think about; and an experienced primary teacher explaining his carefully thought-out strategy for the use of the micro throughout the school. The teacher, Ron Gandalfo, explained that at Courtwood, the micro is not 'god' but just another resource for learning, along with the blackboard and the television set. It is not being used to change the curriculum in any drastic way, merely to enhance the good practice well established in the school.

This has been picked up by reviewers of the programmes as a condemnation of the use of the micro, but it was certainly not intended to be so, and it is hard to see how this view can fairly be criticised. Education must move with the times, but it is right that teachers should be a little on the conservative side, conserving what is good and proven in teaching and learning, whilst being prepared to explore new possibilities and opportunities.

Where the series has really presented a gross distortion of the truth has been over the issue of the involvement of girls in work with the computer. At Courtwood School, both Ron Gandalfo and I explained that great care is being taken in Croydon schools to avoid sex stereotyping. Indeed, in answer to a *primed* parent's question, Mr Gandalfo explained that he had deliberately involved equal numbers of girls and boys in the parents' evening, where children were acting as demonstrators at the dozen or so micros, which had been borrowed for the occasion. Yet the camera dwelt upon groups of boys – film shot of girls was simply not broadcast.

This did not go unnoticed by a reviewer who argued that it showed that the school was failing to live up to its own ideals. *It did not.* It showed that the camera can, and does, easily lie. As Episode Three of the series should have explained, many teachers have been very well aware, from the first introduction of RML 380Z micros into schools five years ago, of the danger that computers would be viewed as mainly for boys. To make sure that this would not happen in Croydon, the appointment was made of a full-time female teacher, whose role has been to collect data, to visit schools encouraging girls to follow computing courses, and to arrange visits to industry to expand the horizons of pupils of both sexes. An Information Technology course is now followed by all pupils in the lower years of secondary school and each school has an IT department with its own staff, who are quite separate from maths departments.

This has not been done as Celia Hoyles implied, because of a belief that girls are no good at maths. On the contrary, a clear policy of positive discrimination has been adopted in this area. We are, however, aware of the well documented loss of interest in mathematics by many girls in their teens and are making strenuous efforts to stimulate this interest, while ensuring that the girls' prejudice against maths is not transferred to educational computing by default.

Episode Two, called 'Why is too much educational software so lousy?' sounded promising.

There *are* a lot of poor programs around, written to emulate the curriculum of the 50's, concentrating on basic drills and skills with fancy 'rewards' of jingles and smiling faces. Far too many of these programs have been written with no teacher input and no school trials and encourage a backward-looking approach to education. But the opportunity to criticise such programs was missed! Instead Mr O'Shea chose to examine two pieces of software which have been well received by many teachers and are among the best 'useful little programs' around, namely *Podd* and *Granny's Garden*.

And who was asked to review these programs? A group of experienced infant teachers? HMI? No, the software was shown to a group of teenagers and their teacher, none of whom has had any experience of work at the lower end of the primary school and who claimed that the programs 'failed to correct spellings' (*Podd*) or were not truly interactive because they would not allow negative responses (*Granny's Garden*). The first of these criticisms is irrelevant; that is not what *Podd* is about. The second demonstrates a complete failure of these older children to appreciate what might be amusing to younger pupils. *Podd* and *Granny's Garden* may be no great shakes as examples of the computer scientist's art, but then neither is a sand-tray. Those of us who have seen the rich educational experiences arising from the use of these programs will need a much more powerful argument to persuade us that they are 'lousy'.

Mr O'Shea should stick to what he knows well. Programming languages were discussed in a much more even-handed way in Episode Five, with proponents of BASIC being offered the opportunity to defend their corner against criticisms, and the demonstration of LOGO, leaving us in no doubt that even using this language, programming is really quite hard!

The final episode showed that true intelligent machines in the classroom are still a long way off. In discussion with industrialists and fellow academics, Mr O'Shea examined what is possible now and likely to be possible in the near future. Aaron Sloman of Sussex University argued that, given the current state of the art, it is better to regard the computer as a tool for the teacher rather than a teacher substitute.

This is a view with which most teachers would heartily agree. *Podd, Granny's Garden* and similar programs are, to use Mr Sloman's word, 'toys' with which children may learn. The teacher's role is to direct and structure their 'play' in order to achieve maximum benefit.

The main conclusion about this series must be that it was an opportunity missed. Teachers and parents still have a considerable thirst for knowledge about educational computing. They need positive advice about what is good, not the carping criticism offered by this series. Of course, there are problems, of course mistakes have been made, but many schools are now using micros in refreshing and exciting ways to improve the quality of teaching and learning. If only we could have seen more of this!

The Learning Machine was broadcast in May 1985 on BBC1 as follows:–

Programme One	*Promises, Promises . . .*
Programme Two	*Why is so much educational computing so lousy?*
Programme Three	*The Gender Gap*
Programme Four	*Which Way to a Job?*
Programme Five	*Having a Bash at BASIC*
Programme Six	*Intelligent Pets*

The programs will be repeated on BBC2 on Fridays, 12.35–1.00 p.m., beginning on January 10th 1986.

Heather Govier
Avery Hill College

Book reviews

Title: **Play LOGO. An Invitation to Computing for Parents and Children**
Author: John Cunliffe
Publisher: Andre Deutsch
Price: £5.95 (128pp spiral bound)

Play LOGO is written for the young absolute beginner and the interested adult. It aims to help you choose a computer for LOGO, to show you how to write your own programs, and suggests some projects and puzzles. Being spiral bound is a great advantage for any computer companion book, but I found the very spread-out format, with a page to head each chapter complete with the requisite number of turtles, and another page at the end of some chapters for notes, as well as the half-page columns of text throughout most of the book, rather wasteful of space in a book with only 128 pages in it anyway. The cover reports that 'LOGO is already widely and successfully used in the USA'. Is it? However the author wends his way chattily through 17 somewhat bitty chapters. Enough criticism for what essentially is a carefully thought-out scheme for an introductory LOGO distance learning package. Some would argue about the validity of writing such a prescriptive book where the child doesn't discover the relevance of RT 90 for herself, (I've four daughters!); others might feel that the introduction of variables half-way through such a book is totally inappropriate. It is hard to please the many emergent LOGOphile experts.

John Cunliffe sets out to help the child at home to discover LOGO, possibly on her own. *Play LOGO* is written for the Atari with notes at the end of each chapter on Commodore LOGO and an appendix comparing commands and other primitives for the Atari, Commodore 64, Acorn LOGO and the Spectrum. The 380Z rates thirty-one words which is more than it deserves in a book concerned with the home market. However it would be a pity if we all felt satisfied with either LCSI/Terrapin LOGOs, or content with the limitations of 8-bit micros. Yes, I do believe that 16-bit micros will be cheap enough for serious home users. Anyway, back to the book.

After a sound introduction, a chapter on 'Why LOGO?', and another on choosing a computer, the book progresses patiently, taking the child through the use of FD, BK, LT, RT, CS and HOME. Next comes PU, PD, PE, HT, ST REPEAT and delete. Putting it in the memory and writing procedures using TO and END is carefully explained in chapter five. There are chapters on managing workspace and saving programs. Use of variables and sub-procedures results in a forest with beasts in it. RANDOM produces a herd. It must be hard for an author to find the balance between suggesting too much, giving the child too much to read, providing too many examples *and* not giving enough to get her teeth into. Maybe extensions to each section at the end of the book could be a solution. Chapter twelve mentions sprites. Next comes sound and colour. Chapter fourteen is interestingly entitled 'Can you do this?' and you have to write the procedures that could produce twelve LOGO turtle drawings which are given in the book. One set of possible answers is given at the end of the chapter.

The book comes to a close with chapters entitled 'Where next?', details of other LOGO books and BLUG, an 'Appendix for parents', 'How some other LOGOs do things differently', a Glossary and an Index. You will either like or hate this book!

Henry Liebling
The Beacon School, Amersham

Title: **Handbook of Primary Education and Computing**
Author: D.W.W. Ellingham
Publisher: Castle House Publications
Price: £7.50 (156 pp. paperback)

In the MEP video that comes with the Language in-service pack, Chris Schenk picks up the comment that one of the most important contributions that the micro can offer in the primary school is 'to make thinking respectable'. That is a conviction that David Ellingham has also held for some time. Indeed, one of the central sections of this book deals specifically with 'Thinking skills and the computer' because of the problem-solving aspect of many of the programs described in the following chapters. Based closely on the work of Edward de Bono, Ellingham shows how the micro can play a central role in encouraging

> *the introduction and practising of decision-making and thinking skills.* [p. 43]

However, as he also points out in the preface, we find ourselves on a course heading into the educationally unknown and it is too early for anyone to offer definitive solutions, particularly when you are reminded that in 1981 only 25 primary schools in the whole of England had a computer! Then with the DTI scheme thousands of teachers were effectively obliged to consider carefully the potential of this new learning/teaching machine. Hence the writing of this book, based on his wide practical experience, which aims to offer a variety of starting points for general consideration and discussion.

Beginning from the viewpoint of 'Why bother with a micro in the classroom?', Ellingham looks at ways in which the computer can contribute to, and can change, the primary curriculum. In many instances the computer is no different from any other educational resource, in that the teacher still has to analyse its role in terms of aims, methods and evaluation. What is new is the need for new management skills in the classroom as well as the ability to assess software as to whether the computer would be the best medium, the most appropriate or the most effective, to meet the educational needs of the children. As you might expect from the co-author of the MEP's document 'Evaluation of Microcomputer Programs for Primary Schools', the book's final chapter is a usefully condensed treatment of the thorny topic of software assessment.

Management skills are considered very closely in Chapter 2; these range from acquiring and cataloguing software to organising the classroom and how (and for how long) children might use the micro. Chapter 3 looks in general terms at five main categories of educational software: information handling, simulations, games, 'teaching the computer' as in LOGO, and reinforcement programs. After the chapter on the thinking skills, the next three chapters describe programs in use for upper juniors, lower juniors and infants. Each of the fifteen programs is looked at in some depth under such headings as aims, source, description, materials, pre-knowledge required, method of use and further comments. The software itself ranges from the MicroPrimer *Watchperson* and *Yourfacts* to others that are commercially available such as *Granny's Garden, LOGO2* and *Pressword(2)* which uses a concept keyboard. Even if you have used these programs already, the chapters are still worth reading because of the thought that has gone into using the programs and into their integration with other aspects of the curriculum.

The emphasis throughout the book is on the educational aspects. If you are looking for another book on BBC computing hardware this is certainly not it. Discs and ROMs are scarcely mentioned, nor even what to plug in where. In some ways, though, due to the nature of the world of printing as well as the world of computing, the book already appears dated – word-processing, control, Prestel and 16-bits are increasingly part of the current educational scene. Yet because of its insight and scope, the book is little the worse for that and well worth a place in any staff library.

One point continually underlined is the use 'thinking' teachers make of the computer as an educational resource. Furthermore, precisely because the micro is so very powerful, the role and consequently the responsibility of the teacher becomes even more important. As Ellingham states, we are only at the very beginning of a long, long road and there are bound to be mishaps and mistakes, but the dangers will be minimised

> *'if we always put education first and computers second.'* [p. 148]

This is a book which does just that.

Barry Wake
Primary Support Team
Birmingham Educational Computing Centre

Software reviews

Title: **Arithmetic Plus**
Publisher: Fernleaf Educational Software,
Fernleaf House, 31 Old Road West, Gravesend,
Kent DA11 0LH
Machine: Acorn BBC B
Price: Suite of 4 programs: £35.95 + £1 p&p
(disc or cassette). Individual programs:
£12.95 disc, £9.95 cassette
Reviewed on: Acorn BBC B (disc)

The four programs are intended for use by small groups of children aged 7 to 12. They all provide practice in basic skills together with an element of problem solving, formulating strategies, decision making and making deductions from the information supplied. They do not set out to teach arithmetic but rather to give practice to those who have a grasp of the techniques.

Find the Question is a race track game where the players can move forward 1, 2 or 3 places each time. Along the track are hidden bonus squares. Landing on one produces a question worth one or two extra moves. Finding the square gets you two points, correct answers are rewarded with the appropriate bonus move, whilst incorrect responses incur missing a turn.

The game is played by the two groups and the first home gets an additional bonus. The strategy, and I think the fun, comes in trying to get the bonus squares and amassing points. Some of my 7–9 year-olds were a little upset at first when they found that the winner was not always the first past the post!

The screen layout is simple but effective. The little men representing each team seemed rather small to me but the children found them satisfactory. The animation of characters climbing from one part of the track to another, caused much interest. So too did the footprint markers left on the track.

There is a rules section which is clear and readable. The type and degree of problem can be easily altered and with a choice of addition, subtraction and multiplication and variations in the size of the answer, there is something to suit a wide range of ability.

Bridges is the second program in the suite. With this program the children either have to build bridges to span a six-square grid before their opponents or, alternatively, amass more points than their opponents. Points are gained by building bridges and spanning the grid. To build a bridge they have to answer either an addition or subtraction question and then decide whether that answer is represented on their grid. If it is, and it is next to a square they have already built on, they may build on it if they enter the correct co-ordinates. Failure at any point loses them their turn. Their opponents may also be able to make use of the discarded square.

It sounds very complicated but my 8–9 year-old pupils soon took to it, although the rules could perhaps have been a little clearer. We had an anxious moment when the screen went blank whilst the grid was being drawn, naturally assuming some sort of operator error. In fairness to the program, a message did tell us to wait but we were taken unawares. The children soon forgave it this idiosyncrasy.

The screen display is clear and unfussy. The bridges are shown in red or green. Some of the 'guinea-pigs' expressed the opinion that they didn't look much like bridges.

The suggestion that this program is aimed at the 8–10 age range seems sensible. It is not designed to teach addition and subtraction but to offer practice to those who are reasonably proficient. It is helpful if they are also proficient readers.

Like the previous program, the main benefit is the development of strategies and the ability to communicate the reasoning behind a particular move. Because of the two ways of playing the game, the choice of addition or subtraction, the variable limit which can be set on the size of the answer and the random element, each program run will provide different situations. Thus it allows a lot of pupils to use it many times without losing its usefulness.

Treasure is more suitable for older or more able pupils. The program aims to offer practice in multiplication through finding factors for chosen products.

It has been designed to be used by two groups, each one trying to find the submerged treasure. They have two choices, either to drop a marker buoy which will give them a colour clue as to the nearness of their treasure, or to enter the co-ordinates of the find. However if they are wrong they miss a turn and are given no clues. If they wish to place a buoy the co-ordinates have to be entered as the factors of a product. For example 5,5 is taken to be 55 so 5 x 11 has to be entered. Not all the squares on the grid can be entered in this way so some deduction has to be done.

A worksheet is necessary and the authors have provided one which may be copied. Total scores are shown at the end of a game and the teacher can access the results of up to six games if required.

The children wanted to succeed at this game but found the method of placing the buoys frustrating as they were not as competent as they would have liked. This program would provide the stimulus for more learning.

I found that sometimes perfectly good factors were rejected because the numbers were too big. It didn't like the use of 2 and a number over 20. Also it was difficult to enter very small factors like 2 x 0 or 1 x 1.

Product patterns is the last of the suite and perhaps the most difficult program. The children are given three or four rows of ten squares and asked to correctly place randomly chosen products. Two number clues are shown to help them decide which 'table' is required in each row. Sometimes these are not very helpful especially if they appear in several tables. However as the game progresses a pattern should become more apparent. A worksheet, on which to record the hypotheses, is essential.

The rules pages are carefully laid out with appropriate examples. They should provide little problem to a proficient 10 year-old reader.

The children did not like having to type the letter for the row, then the column (a reversal of co-ordinates) in order to indicate where to put the products and would have preferred some sort of cursor control.

In general these programs should find a lot of use in primary schools. Their unattractive title may be offputting and disguise their value. The teacher's booklet is well written and covers what you want to know, unlike some I can think of. It even has clear worksheet grids that can be copied legitimately, but its presentation lacks sparkle.

Finally, an important consideration these days, the cost of the programs. At first, £37 seems rather a lot to pay out, but if you consider their potential for use by a large range of pupils, you may think them worth it.

Mike Billings
Cookley Sebright First School, Worcestershire

Title: **Micro Smile 1**
Publisher: Centre for Learning Resources,
275 Kennington Lane, London SE11 5QZ
Machine: Acorn BBC B
Price: £15 (£10 ILEA and MEP Capital Region)
For details of RML 480Z versions contact ILEA,
ILECC, John Ruskin Street, London SE5 0PQ

It is quite obvious that this collection of thirty problems and investigations have been written by teachers. They are well designed and packaged, robust and, in many cases, very attractive and motivating.

Smile produces learning resources for all abilities in the eleven to sixteen age range, but most of the programs would also lend themselves to fruitful use in the primary range, especially where different levels are offered. The only clumsy input routine is in *Locate* which requires a comma between two co-ordinates.

Although the use of sound is not always purposeful, the graphics are generally clear (although *Boat* was a bit primitive) and would be good enough for use with a large group if necessary, although they have been designed with small groups (two or three) in mind. They have also been designed for the children to use independently. Many offer a demonstration routine which is useful both for individuals and for introducing the program to a larger group.

The documentation is clear and concise, with just the right amount of information for the busy teacher. However the order of the materials is alphabetical, even though names hang on the whims of authors. It would have been better if the program documentation had been grouped in clusters relating to underlying concepts. For example, it would have been useful to have had *Elephant* and *Rhino* together as they both involve similar strategies. *Reverse* and *Frog* also relate to each other.

Although many of the programs have appeared elsewhere, (MEP Maths Pack, various Anita Straker packages and ATM software), this is an ideal purchase for a confident staff wanting to branch out into this approach to mathematics, especially if their access to other sources of similar material is limited. The package provides a wide range of activities and occasionally gives suggestions for extensions and follow-up work away from the keyboard, although there could have been more. *Smile* seems to credit the teacher with a level of competence which is heartening, but might be a little optimistic in some cases when experience of this kind of working is limited.

Sarah Wells
Manor CP School, Uckfield, Sussex

Title: **ABC**
Publisher: Acornsoft Ltd., Betjeman House,
104 Hills Rd, Cambridge, CB2 1LQ.
Machine: Acorn BBC B
Price: £11.50 disc £9.95 cassette.

The program is packaged in a fairly robust box and includes a disc or cassette, a function key strip and a 29 page documentation booklet.

The program is described by the publishers as 'a writing tool, designed to meet the needs of young writers (aged seven years and upwards)', i.e. a wordprocessing package.

It is pointless to list all the facilities that this wordprocessing package offers as most packages of this type offer roughly the same range of facilities. It is more important to consider this package in relation to the claim of the publisher and to consider its implementation in the classroom and its ease of use by the children.

The program, as it is available on cassette/disc, is machine portable and this means that it is possible to use all the computers in the school as processors if required. The only problems with loading were encountered on a computer with a Watford DFS 1.2.

The program offers three types of screen display. Write, Read and Write and Slate.

The opening screen is the Write screen so the children are able to begin typing immediately, with no complicated menus to move through – ideal for the younger child. The Write screen offers all the conventional facilities of a wordprocessor; however there are four special features that bear examination. Firstly, there is a facility to switch on/off the auto repeat key, this is useful with younger children who may tend to hold down a key. Secondly, there is a facility that will automatically generate a capital letter, after a full stop, question mark or exclamation mark. A dubious facility this, as it could be argued that children should be made to generate capital letters, when necessary, themselves; although it could be used as a useful teaching aid. Thirdly, there is a facility to determine the way in which text can be deleted. It is possible to delete at the cursor, or to the right, or to the left of the cursor. It would be impossible here to describe fully the way in which this works, however it is explained in the documentation booklet. If the children are accustomed to delete working backwards this can be a little confusing. However, teachers may see other benefits from being able to encourage the children to delete from the right, or at the cursor, after considering the way the eye normally travels when reading text. The final option is the facility to display the current line number and the number of lines left. This is of far more use to children than displaying the number of words or characters left. All of these facilities can be set up by the teacher prior to the children using the package.

Editing text gives all the conventional facilities, search and replace, centring text, moving text, deleting text, etc. However, there are a couple of innovations which are worth consideration. Tug, Paint and Tidy are three facilities available through the function keys. Tug and Paint are for moving a line or block of text. The text to be moved is highlighted and the children can move the text and position it in the correct place using the arrow keys. The children are in charge of the process and can see the movement of text in response to their inputs. Tidy, as its name suggests, just tidies up the text after editing.

The second screen option is the Read and Write screen in which part of the screen is dedicated to writing with all the normal editing facilities, whilst the rest of the screen is used as a read screen, allowing eleven lines of text to be viewed. The text may be scrolled and may be copied to the writing area. This is a useful facility and would encourage older children to consider previously written text before continuing.

The final screen is the Slate screen, which is a notebook. Notes may be jotted in the top part of the screen without affecting the main text. The bottom part of the screen is a writing area. Notes from the slate may be copied into the writing area. Again this is a useful facility for encouraging children to make notes to be used in later text.

Text can be printed out, saved and loaded. All of the available facilities are fully covered in the documentation. The booklet is concise and readable. It adopts a tutor style for certain options whilst others are just explained. It is a comprehensive booklet and explains the package well.

ABC would be a useful tool for young juniors and older children as it allows instant access to a wordprocessor without the need to follow complex menus or techniques. As the children become more adept on the package more options are accessible in an easily understandable format. *ABC* offers all that older children need, the only limiting factor being the length of the document, at roughly 170 lines of text.

Finally, it is worth mentioning that there is an option for converting *ABC* files into *View* files. *View* is a more sophisticated package, but perhaps less practical and less usable with younger children.

Bob Butcher
Rounds Green School
Oldbury, Sandwell

Title: **The Last Adventure**
Developed by the Microelectronics Education Program.
Publisher: Learning and Training Systems, Haydon House, Alcester Rd, Studley, Warwickshire. B80 7AP.
Machine: BBC Acorn B (disc only).
Price: £23.00 for complete package + £1 post and packing, £14.95 for *Last Adventure* and *Wiz* + £1 post and packing, £6.95 for *Spr* + £1 for post and packing

The Last Adventure is an authoring or generating package which enables the user to create adventures, simulations and other application packages. The complete pack contains three discs:
1) *The Last Adventure* program.
2) *Wiz* – A ready made fantasy/adventure which has been generated by the main program.
3) *Spr* – An example of a different educational application of the main program which is based on an Environmental Studies trail around the Springfield EVS Centre, Birmingham.

The pack also contains a large folder of documentation for the main program and transcripts/information to support the two other programs.

The Last Adventure has facilities for creating up to 16 scenes or locations in the Create mode. Graphics can be added to these locations and there is space for descriptive prose.

Each location is assigned a function key and can be entered by pressing the relevant key or by pressing Shift and the key together, (giving 15 screens). Entry to each scene can be restricted by the author until various problems have been solved or questions answered. In this way the adventure author can control the sequence of visits, and set tasks or challenges at each location.

The Last Adventure has been designed with the concept keyboard in mind, and there are blank overlays and detailed instructions to facilitate its use. Because of the program's suitability for use with a concept keyboard it may be particularly useful for children with reading difficulties or, within the special education sector, for children with manipulative problems.

The complete package includes 72 A4 size pages of instructions about creating your own adventure; 80 pages of documentation for the programs *Wiz* and *Spr*; 3 discs; sample concept keyboard overlays and several function key strips. Altogether *The Last Adventure* is a formidable package for the busy primary or special needs teacher to come to grips with.

The documentation, though extensive, is very difficult to follow and the sequence of activities needed to create an adventure is confusing and complex. A trial adventure of only four screens was devised to try out the main program and it took several hours of intensive keyboard thumping, not to mention documentation thumbing before a recognisable adventure game emerged. This was without adding even rudimentary pictures to the scenes of the adventure.

Most teachers would take one look at *The Last Adventure* package and turn sharply towards other adventure generators such as *Your Adventure, Lost Frog, Quill* or even *Tracks.* Teachers with determination could overcome the off-putting documentation and create their own adventure, but might question whether the achievement of a reasonable result justified the amount of teacher effort required. This is unlikely to be so, which is rather a shame as the *Spr* program demonstrates how *The Last Adventure* can be used imaginatively to produce creative and educationally valid, supportive software.

LTS would be well advised to repackage *The Last Adventure* and make it less intimidating by producing easy-to-follow documentation. *The Last Adventure* could have valuable educational applications but it will probably be quietly ignored.

P.J. Mate
Camm's Endowed C of E Primary, Derbyshire

Educational Software Support Scheme

Roger Keeling
Newman College

After a long awaited Government announcement, it is now clear that the MEP will cease to exist as from April 1986. It will be replaced by two initiatives, one from the DES and one from DTI. The former will oversee the establishment of a Microelectronics Support Unit, probably with a budget of £2.2 million in the first year and a possible life span of up to five years. Further details are still being worked out.

However, on October 4th at Loughborough University, at a one-day conference organised by EdIT, the details of the DTI scheme were announced to a large audience of advisers, trainers and commercial publishers. Peter Waller from the DTI presented the figures. In effect half a million pounds will be made available to LEAs before the end of March 1986, with £2 million available in 1986/7 and £1 million in 1987/8. LEAs must match the expenditure from 1986/8; hence creating a scheme valued at £6.5 million. It will be divided proportionately among all LEAs (there will be no need to bid for the money). It must be spent on the purchase of software from commercial sources, and the software must be for classroom use. An average metropolitan borough could therefore expect to receive £10,000–£25,000 while the larger county authorities could expect to receive £40,000–£70,000 on average. Many authorities will probably try to negotiate licence arrangements in order to maximise the effect of the money.

How does this affect the average *MICRO-SCOPE* reader; the teacher in the primary classroom? The scheme is in operation now (from October 1st). Therefore let your computing adviser know what software you would like purchased. Form self-help groups to apply pressure. If you don't ask you may find the money largely goes on providing software support for the secondary sector. Stand up and be counted.

* * *

Roger Keeling attended the conference on behalf of MAPE as a guest of John Barker and EdIT.

MAPE Conference 1986

A commercial viewpoint

Brian Richardson,
Cambridgeshire Software House

As I am sure you will have seen from recent publications, commercially produced software is attracting indiscriminate criticism. Recent issues of *MICRO-SCOPE* have included suggestions that MAPE, as an organisation, may not want to see a commercial involvement in the primary software field and, indeed, such authorities as the BBC have made it quite clear that in its opinion much commercial software is of dubious educational value. So where do we go from here?

Assuming that I am invited to exhibit our programs at the next Conference there are one or two suggestions I would like to put forward to the organisers for consideration.

MAPE Conferences tend to follow a format of main lecture, workshop, talk in bar etc. The main lectures always seem to be interesting enough, but I cannot help thinking that now the time would perhaps be better spent in allowing delegates more time to 'do their own thing'. Speaking for a company whose product was the subject of a main lecture three years ago, I appreciate the value this slot has in publicising a piece of software. However, taking Cheltenham as a guide, the delegates spend much more time working through software that for once, (mail order being what it is), they could actually touch and use. Our postbag since MAPE '85 has contained many letters 'We saw you at MAPE for a few minutes but can you please tell us . . .'. It is a pity that we were not able to deal with those questions at Conference. How many more delegates went away with questions in mind but have not written?

Another suggestion might be to invite some of us to run 'hands-on' workshops, with some of the teachers who have used the software in classrooms available, to, once again, give teachers a chance to have a good and detailed examination of the software.

Why not also put a list of exhibitors and their programs in the Conference folder so that the delegates are made aware of what is on show, as despite all this 'close scutiny' one or two of us do have a few good programs in our lists.

If the end result of the MAPE Conference is such that much of the software is not seen by the delegates then surely that is wrong. We receive hundreds of letters asking for 'review', 'sample' or 'test' copies, yet here, at the MAPE Conference, a golden opportunity for 200+ people to go back with all the news seems to be missed.

The Conference itself always appears well organised and it is good to meet up with some of our customers and hear their opinions. However, maybe, given more time, those customers might well be better informed, if only they *were able* to evaluate, discuss and comment on more of the (available) software on show.

I am sure others will have similar and perhaps more constructive comments to make, but I hope the organisers of MAPE '86 will take note of some of these suggestions.

Conference thoughts

Mike Matson
4MATION Educational Resources

Being the 'creative' (and least business-like) half of a publishing company probably doesn't qualify me to offer a very realistic commercial view of the annual MAPE conference but as someone concerned with the use of micros in education I do have one or two comments to make.

Personally, I feel that even contributions by the world's best speakers and exhibits by the most illustrious of software companies, do not negate the fact that the main function of such a conference is the provision of an opportunity for people to meet and talk. I regard the workshops, lectures and commercial exhibition as an *excuse* for people to get together, so I'd be quite happy to attend a conference with no guest speakers and no exhibition, but a bar which is open all night. I'm convinced that the most important and valuable discussions take place when people are relaxed in an informal setting. This, of course, does not mean that the 'important' people and the commercial fold shouldn't be there as well. To be a little more practical, what I'm saying is that I would welcome more opportunities for delegates (and those with commercial interests) to meet informally, without feeling guilty and without having to find excuses for missing so and so's lecture.

Practical suggestions

How about:
1. Giving exhibitors the opportunity to despatch the appropriate number of hand-outs before the conference? These could then be collected at registration. Yes, I know it would mean extra work for someone.
2. Providing sensible times when there are no formal activities taking place so that delegates can meet informally and/or visit the exhibition?
3. Offering the 'practitioners' more opportunities to demonstrate their craft? I'm sure fellow teachers would appreciate contributions from people who've 'really done it'.
4. Placing a suggestion box at a convenient location? Many ideas occur to delegates at the time but are not passed on to the organisers.
5. Providing opportunities for representatives from commercial bodies (software, magazines etc.) to talk for a few minutes about their companies? I'm sure I'm not the only one who would love to know if it's true that the Cambridgeshire Software House are so heavily into archaeology because they lay pipelines in their spare time.
6. Allowing more time for discussions, preferably in smallish groups as lots of people are intimidated by the presence of hundreds of delegates?

As a final thought I'd like to make it quite clear to the consumers that publishers in the relatively minute educational software arena are not making a fortune and really cannot afford to hand out freebies willy nilly to all and sundry. I'm not sure how other publishers handle the steady stream of begging letters but it is now our policy not to offer free copies of packages in response to requests, even when the 'evaluator' resorts to blackmail with remarks like 'if you are unable to provide me with a complimentary copy, I will not be in a position to recommend your products'. In certain cases, we may agree to provide a loan copy and I suggest that other publishers adopt this approach.

Computers and Liaison Group (CALG)

Ken Atkin
CALG Chairman

A Joint Working Party of CEG, MAPE and MUSE has been set up in order to consider and report on the difficulties experienced by Primary and Secondary Schools at the age of transfer in the area of Computers in Education. It has also now been agreed that there shall be representation from BCS. This group shall be known as CALG.

Two meetings have been held to date, in June and September. It is hoped that there will be two meetings of the group each term together with a week-end seminar next February, which will be limited to an absolute maximum of fifty active working members. A final report, to be presented to each organisation, is planned for the end of the current academic year. It is hoped that this report will be made available to members of each association through the normal magazine either as an article or a separate document, depending upon its length. A questionnaire is to be sent to each member of the three original associations in order to obtain data for the group and should be included with this magazine. Please complete it and return to *the address printed on the form* as soon as possible.

The problem being considered appears to be currently one of the major concerns in the field of computing in schools and it is hoped that the report will provide ideas for reducing, if not solving the problem, to the advantage of our pupils.

Individual members of the associations are invited to submit written evidence to be considered by the group. Examples of case studies, working papers, minutes or individual views could be most helpful. The names and addresses of those colleagues interested in attending the seminar week-end should also be forwarded as soon as possible.

17 Alderson Road, Worksop,
Notts S80 1XB.

MAPE news

Mersyside and Cheshire

A day conference (Language Development and the Micro) was held on Saturday 22nd June, organised by MAPE (M & C) in conjunction with Chester College, which attracted approximately 100 teachers.

As well as main talks by Mike McEvoy (Developing Tray) and Steve Hughes (Mikefax) the delegates were treated to a variety of demonstrations by local teachers showing what they had achieved with their children.

These ranged from Bill Mowbray's roomful of children's work inspired by *Flowers of Crystal,* Mike Greatorex demonstrating other 'adventure' possibilities, Malcolm Glover with *Spacex* and some excellent locally developed language programs, Tony Evans and Sue Watkins with *Mallory* adaptations, Mike Barratt's *Excursion into the Unknown,* Jackie Hambling challenging our minds with *Podd,* Jim Fawcett with home produced programs for infants and top juniors, Steve Webb exploring data bases and languages, Paul Collins with his own infant language programs, Jeff Hughes explaining the educational possibilities of Prestel, Robert Davies with *Edword,* and George Derby with a host of peripherals for special needs, including alternative input devices.

During the day there was a buzz of interest and excitement from the teachers attending, who were seeing examples of good practice and were being given a chance to talk to teachers who have given real thought to what they were doing with micros in their classrooms.

I would like to publicly express my grateful thanks to all the above teachers who gave their time, and to Chester College for offering us excellent facilities, and especially to David Hughes and Jackie Hambling for all their organizational work, without which the day would not have been so successful.

J. Fawcett

MAPE National Committee Members 1985

Chairman
Roger Keeling, Newman College, Genners Lane, Bartley Green, Birmingham B32 3NT.
Tel. 021 476 1181

Treasurer
Keith Whiting, 149 Sherbourne Avenue, Nuneaton, Warwickshire CV10 9JN.
Tel. 0203 396132

Secretary
Ann Liddle, Pentland Primary School, Pentland Avenue, Billingham, Cleveland TS23 2RG.
Tel. 0642 552848

MICRO-SCOPE Editor
Senga Whiteman, Newman College, Genners Lane, Bartley Green, Birmingham B32 3NT.
Tel. 021 476 1181

Publicity Officer
Heather Govier, 219 Osward, Courtwood Lane, Croydon CR0 9HG.
Tel. 01 657 0423

MAPE Administration
Mrs G Jones (MAPE), 76 Sudbrooke Holme Drive, Sudbrooke, Lincoln LN2 2SF.
Tel. 0522 754408

Regional Representatives

CAPITAL REGION
Charles Bake and Di Wailing, IT Unit,
Davidson Centre, Davidson Road,
Croydon CR06 6DD
Tel. Office: 01 654 3769
LEAs
Bexley, Bromley, Croydon, Ealing, Hounslow, ILEA, Kingston, Merton, Richmond, Sutton
Code 01

CHILTERN REGION
Bill Bailey, 'Kingsley', Scatterdells
Lane, Chipperfield, Herts WD4 9EX
Tel. Kings Langley 63532
LEAs
Barnet, Bedfordshire, Brent, Buckinghamshire, Cambridgeshire, Enfield, Haringey, Harrow, Hertfordshire, Hillingdon, Oxfordshire
Code 12

EASTERN REGION
Newton Coen, Ridgeway, The Row,
Hartest, Bury St Edmunds
Tel. 028 483 535
LEAs
Barking, Essex, Havering, Newham, Norfolk, Redbridge, Suffolk, Waltham Forest
Code 03

EAST MIDLANDS
Stan Norman, 70 Mount Pleasant,
Keyworth, Notts. Tel. 06077 5540
LEAs
Derbyshire, Leicestershire, Lincolnshire, Northamptonshire, North Derbyshire, Nottinghamshire
Code 10

GREATER MANCHESTER
Dave Whitehead, 550 Whitworth Road,
Rochdale, Lancs.
Tel. School: 0706 49507
LEAs
Bolton, Bury, Lancashire, Manchester, Oldham, Rochdale, Salford, Stockport, Tameside, Trafford, Wigan
Code 05

MERSEYSIDE & CHESHIRE
Steve Hughes, New Brighton Primary
School, Vaughan Road, Wallasey,
Merseyside. Tel. 051 639 3869
LEAs
Cheshire, Knowsley, Liverpool, Sefton, St Helens, Wirral
Code 09

NORTHERN REGION
Marilyn Nellist, Woodham Burn
Junior School, Humphrey Close,
Newton Aycliffe, Co. Durham
Tel. 0325 312505
LEAs
Cleveland, Cumbria, Durham, Newcastle on Tyne, North Tyneside, Northumberland, South Tyneside, Sunderland, Gateshead
Code 07

SCOTLAND REGION
To be appointed.

Code 20

SOUTHERN COUNTIES
David Marshall, Rocks Park Junior
School, Uckfield, East Sussex
LEAs
Berkshire, East Sussex, Hampshire, Isle of Wight, Kent, Surrey, West Sussex
Code 11

SOUTH WEST
Reg Eyre, Dept. of Maths, Science
and Computing, College of St Paul
and St Mary, The Park, Cheltenham,
Gloucestershire
Tel. 0242 513836
LEAs
Devon, Cornwall, Somerset, Avon, Wiltshire, Gloucs, Dorset
Code 04

SOUTH YORKSHIRE & HUMBERSIDE
David Smith, 55 Dobcroft Road,
Millhouses, Sheffield S7 2LQ
LEAs
Barnsley, Doncaster, Humberside, Rotherham, Sheffield
Code 08

WALES REGION
Patrick Drewett, Parkdale, 16 Fields
Park Road, Newport, Gwent
NP9 5DA
Tel. 0633 52382
LEAs
Clwyd, Dyfed, Mid Glamorgan, Powys, South Glamorgan, West Glamorgan
Code 13

WEST MIDLANDS
Barry Wake, Birmingham Educational
Computing Centre, Bordesley Centre,
Stratford Rd, Camp Hill, Birmingham
Tel. 021 772 6534
LEAs
Birmingham, Coventry, Dudley, Hereford/Worcester, Sandwell, Shropshire, Solihull, Staffordshire, Walsall, Warwickshire, Wolverhampton
Code 02

WEST & NORTH YORKSHIRE
Miss Marjorie Briggs, Flat 3,
67 Leylands Lane, Bradford
BD9 5QT
LEAs
Bradford, Calderdale, Kirklees, Leeds, North Yorkshire, Wakefield
Code 06

NORTHERN IRELAND
Margaret Murphy, Computer Centre,
Stranmills College, Belfast B79 5DY
Tel. 0232 664271 Ext. 389
Code 14

CO-OPTED MEMBER
Ron Jones, 76 Sudbrooke Holme
Drive, Sudbrooke, Lincs LN2 2SF
Tel. 0522 754408

Software at discount prices for MAPE members

Reg Eyre

The following software publishers are prepared to allow a 10% discount off the published price of their software for MAPE members. By publishing such a list, MAPE is in no way endorsing these products, and members are still advised to preview software before purchasing, at MEP regional centres, LEA centres, through reviews, etc.

ASK
London House
68, Upper Richmond Road
London
SW15 2RP

HILDITCH SOFTWARE
4, Church Road
Felixstowe
Suffolk
IP11 9NF

LTS Ltd.
Haydon House
Alcester Road
Studley
Warks.
B80 7AP

CAMBRIDGESHIRE
SOFTWARE HOUSE Ltd.
The Town Hall
St. Ives
Huntingdon
Cambs.
PE17 4AL

COLLINS/HILL MACGIBBON
8, Grafton Street
London
W1X 3LA

To obtain your discount for any software, please send the address label from the packing of your MICROSCOPE as proof of membership. This will be returned to you with your purchase.

In accord with our policy of not carrying advertisements, the discounts offered are general, and should you require detailed information about programs available, we suggest you either write to the publishers for brochures, or see advertisements in the commercial educational computing magazines.

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

£1.75