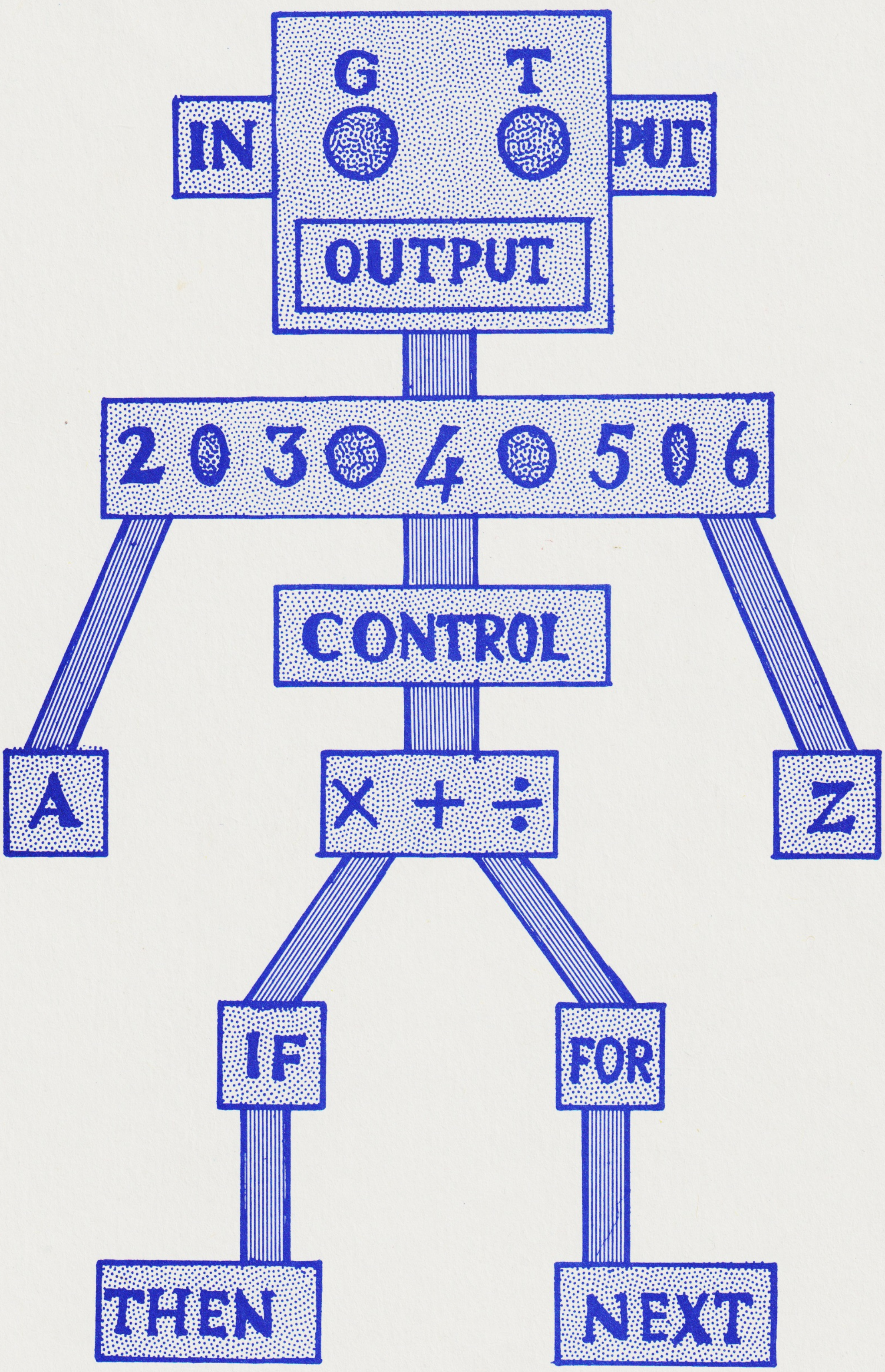
MICRO-SCOPE



'MICRO-SCOPE: BEGINNINGS'

A selection from issues 1 to 4, 1980/1

CONTENTS

Classroom Uses of the Microcomputer

With the advent of low-cost microcomputers we must begin to assess the potential of this technology in the field of primary education. This article looks at areas of possible application and will hopefully serve as a stimulus to the production of many more ideas.

I must emphasise at the outset that the micromputer is not going to dominate or replace the teacher. We must learn to use it as an aid. Different teachers will have their own approaches and will exploit different characteristics of the machine. Each will aim to identify those characteristics that will add to his own existing resources. It is equally important to recognise that inappropriate use of the machine might cause a lack of thinking or attention, and so stifle the learning process.

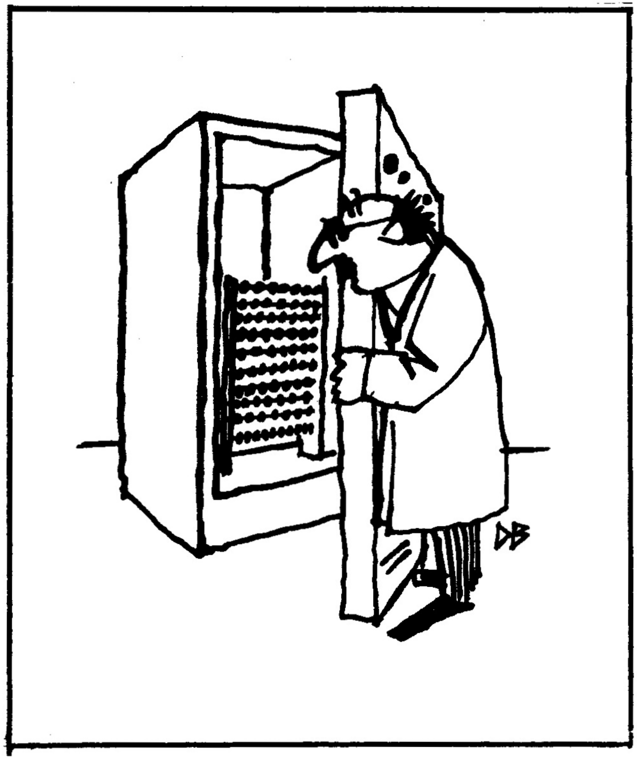
A few promising applications are given below.

1. AS AN ELECTRONIC BLACKBOARD.

The graphics facility can be in colour or black and white. It can draw graphs, diagrams and maps more quickly and accurately than a teacher at the blackboard. Williams (1) cites two disadvantages of the blackboard:

"The difficulties usually fall into one of two categories: a) an accurate graph or diagram is needed which I cannot produce quickly enough, with the accuracy required, without losing the pace of the lesson; b) the situation has a dynamic aspect to it which I cannot cope with fast enough."

We can investigate changes caused by varying the data and produce a revisec graph for comparison almost immediately. The momentum of the lesson is maintained and it allows investigations to be deeper and more extensive than may have been possible previously. With animated graphics we can move a cursor around the screen to indicate a particular flow or route, or move a shape around the screen by translation, rotation, reflection or enlargement.



2. AS A TEACHING AID.

It can be used with a whole class, a group or an individual. In particular the machine can provide a tailor-made teaching environment for children requiring remedial help, while it is equally capable of challenging the brightest pupils. In fact it partly answers the criticism made in "Primary Education in England" (2)

"... for the children who showed most marked mathematical ability the work was often too easy and it is a matter for concern that these children's abilities were not fully extended in their work in this subject."

The machine has certain advantages over the human teacher. The microcomputer is the epitome of patience and it provides instant feedback - personalised where appropriate. It also has an amazing capacity to motivate. This leads in turn to the third point.

3. TO GIVE PRACTICE IN SKILLS.

This could be simple drill work in maths or language. The micro can, for example, generate simple arithmetic problems: but envelope this work within a games situation and add a graphics display and you have a recipe for total pupil involvement that even the best motivators would be hard pressed to match.

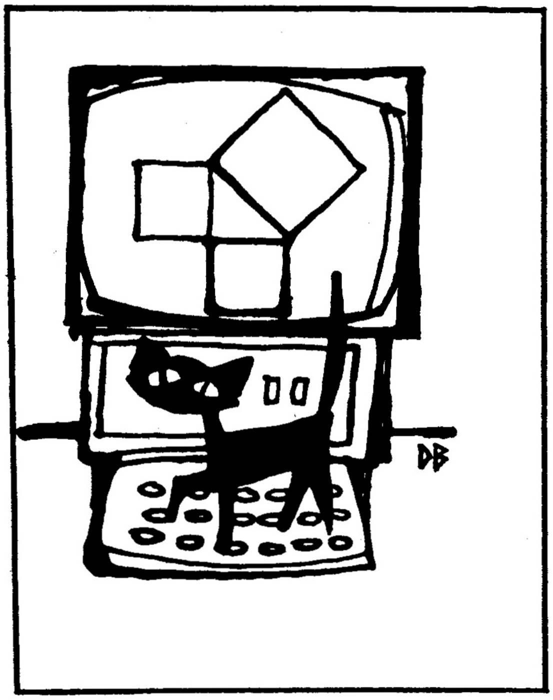
"Primary Education in England" states :

"In describing what their schools set out to achieve in mathematics, heads' comments indicated clearly that they attached considerable importance to children achieving competence in the basic skills of arithmetic and understanding mathematical processes..... Many heads also referred to the importance of children gaining confidence, enjoyment and satisfaction from their work in mathematics."

Here the microcomputer can provide the skill practice and certainly provide enjoyment, satisfaction and a growth in confidence.

4. FOR TESTING AND DIAGNOSIS.

The more information we have about each pupil, the greater our accuracy when assessing strengths and weaknesses. The micro can be used to test individuals across the curriculum for certain concept formations, skill acquisitions and understanding and then to store the results for future reference.



The teacher can then interrogate the file at any time and find out, at varying levels of detail, the performance of each pupil. The micro could then be used to generate work cards or question sheets, for any individual child, using the information it has stored to assess the standard. This is potentially a very powerful application but requires the very best software to make it effective. To quote "Primary Education in England" again:-

In about a third of the classes, at all ages, children are spending too much time undertaking somewhat repetitive practice of processes which they had already mastered. In these circumstances there was often a failure to make increasing demands on the children's speed or accuracy, or to introduce new and more demanding work."

The microcomputer could go a considerable way to answering this criticism.

5. AS A COMPUTER!!

In the previous discussion I have looked at applications involving the micro as a teaching aid, in situations where it can add a dimension to the teaching environment. Yet naturally we can also use it as a very fast and powerful calculator. It is ideal for handling problems of chance and strategy - that is, by the use of the (pseudo) random number generator for fast simulations - dice throwing in mathematics, population growth in biology, or location problems in geography.

In outlining these possibilities, which are very much interrelated, it seems evident that the future is bright and exciting but is very much dependent on the rapid development of suitable software. Primary teachers cannot all be expected to acquire technical expertise or learn to write their own programs, although no doubt some will and hence provide a valuable contribution. Other teachers will only be required to use 'packages', and these need not be long and complex. Simple, short programs illustrating a particular point or concept are likely to be equally successful.

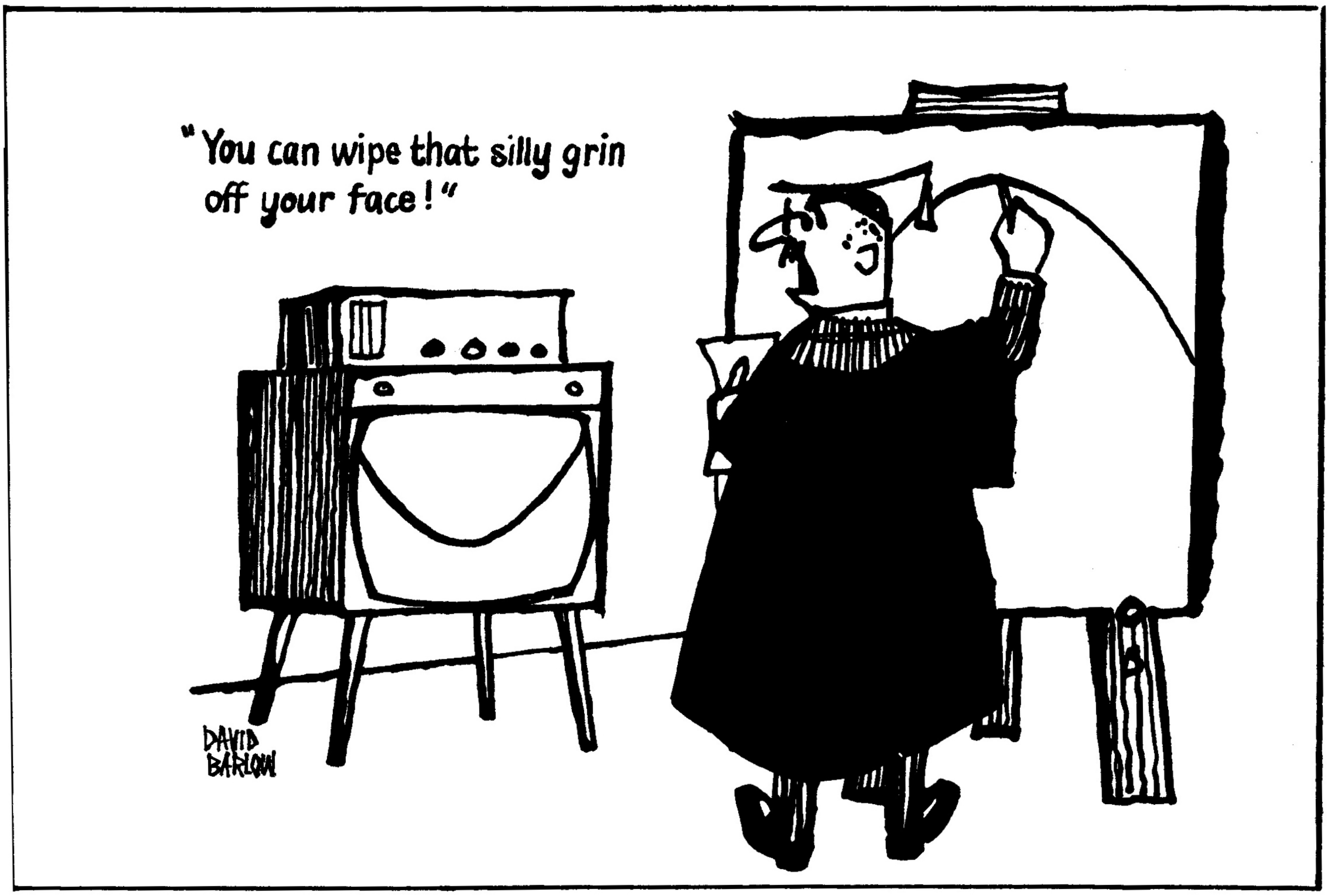
Computer awareness itself is an important bonus. Young people will need to learn to accept and control the technology with which they will grow up. Classroom trials have shown they are far more adaptable to the new technology than are their parents or teachers.

Perhaps we are only at the beginning, but expansion will be swift. The microcomputer will become an aid to learning and understanding but will not replace it. Teachers must become aware of the possibilities so that they can guide progress in the direction they think will be most useful. No doubt there will be many false trails but hopefully many real advances as well. These advances must be developed, tested and evaluated by primary teachers and not dictated by computer boffins.

R.K.

1) H. Williams, "The Computer as a Visual Aid", <u>Computers in Schools,</u> September 1979, page 16.

2) "Primary Education in England," H.M.S.O, 1978, Chapter 5.



A PRIMARY B.A.S.I.C.

Programming for teachers of young children.

Many teachers in primary schools will be aware of the coming social and industrial revolution made possible by computers. Some resist it; many fear it; others will wish to mould this giant and channel its strength into educational benefit. Some will feel the need to understand, modify or write programs.

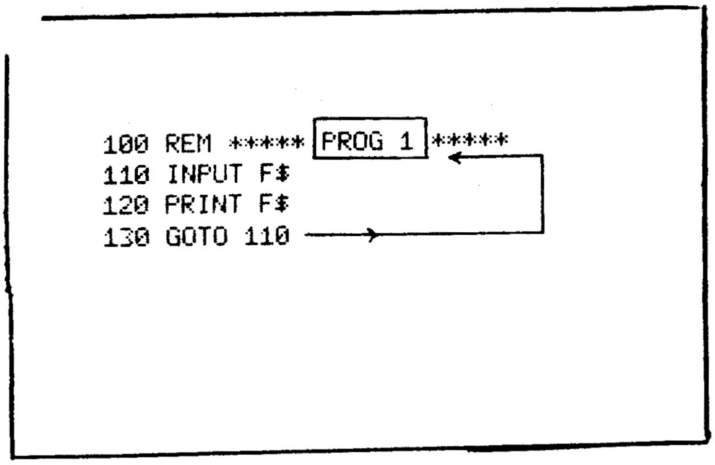
Of course, programs will be available ready-made, but good teachers do not like to rely blindly on textbooks or teaching packages. Adaptations are often desirable. Also, things do go wrong, and a little expertise can often, in seconds, save a whole lesson.

This is the first in a series of articles designed to introduce a simple computer language - the well-known Beginners All-purpose Symbolic Instruction Code. There are more sophisticated alternatives, but B.A.S.I.C. is the language used now in this educational context.

As an illustration, I shall discuss how to make the computer print out verses of the action song "One finger keep moving". There are many ways to write this program. My first approach is trivial, but later refinements will introduce useful principles of wide application, and indicate the power of the computer to handle repetition economically.

STATEMENTS.

Look at Program 1. There are four lines of program, each with a number: each line will be read and obeyed in sequence.



```
100 REM ***** PROG 1 *****
110 INPUT F$
120 PRINT F$
130 GOTO 110
```

Line 100 is a REMARK, used here as a label only - the computer reads any REM statement but takes no action. REM statements are helpful to the programmer or user in reading a program.

Line 110 is an INPUT Statement which stores whatever is typed on the keyboard in the computer memory. Here the words "One finger keep moving" can be typed in and stored under the reference code F$ (say it as " F dollar or " F string"). F$ is a string variable - a series of symbols denoted by the variable name.

Line 120 is an OUTPUT Statement which PRINTS the string stored as F$ (A Printer or a V.D.U. screen may be used.)

Line 130 is an Unconditional Jump Statement which returns the control to line 110 again.

This program merely prints out one line of the song at a time, as it is typed in. As yet it is neither faster nor more economical than an ordinary typewriter! But we have now met some useful principles:-

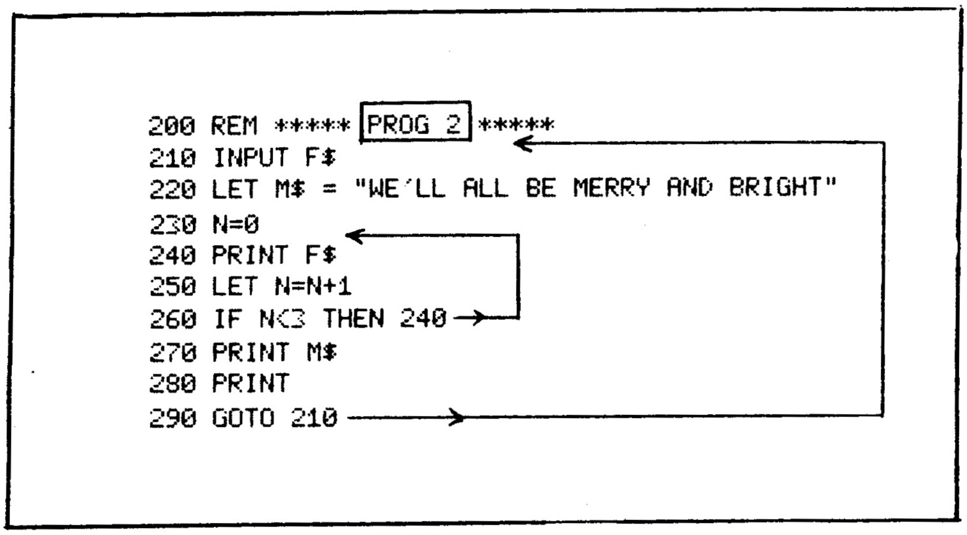
1. BASIC programs consist of a sequence of instructions called Statements.

2. Each statement has a line number. Statements are read and obeyed in numerical order.

3. We have met (a) Input/Output Statements (INPUT -, PRINT -);
(b) Unconditional Jump Statements. (GO TO -);
(c) REM Statements.

LOOPS.

Now we begin to refine the program, and note that in each verse a line is repeated. We want to construct a "loop" to print the line three times exactly (when input once only!). For this we can use a Conditional Jump. Look at Program 2.



```
200 REM ***** PROG 2 *****
210 INPUT F$
220 LET M$ = "WE'LL ALL BE MERRY AND BRIGHT"
230 N=0
240 PRINT F$
250 LET N=N+1
260 IF N<3 THEN 240
270 PRINT M$
280 PRINT
290 GOTO 210
```

Lines 230 to 260 give us a Loop using the "IF - THEN -" Conditional Jump. The control will twice jump back to 240 and so read line 240 three times. The next time it reaches line 260, the condition N < 3 will not be satisfied, so the program continues to line 270. The "IF - THEN -" Conditional Jump gives us a way of making decisions and "branching". In the program above we use it for counting in a loop.

Program 2 also illustrates some new points. Lines 220, 230 and 250 are all examples of Assignment Statements. In line 220, M$ denotes the string variable which locates the storage of the sentence "We'll all be merry and bright".

In line 230, N is a numerical variable (No $ - it is not a string variable). The number 0 (zero) is assigned to a store with reference code N.

In line 250, the variable denoted by N is increased by one and re-stored, still under the label N. (Remember that line 250 is an assignment statement - don't try to read it as algebra!). The ordinary operations of arithmetic: +, -, * (multiply), / (divide by), ↑ (raise to the power of) may be used with numerical variables.

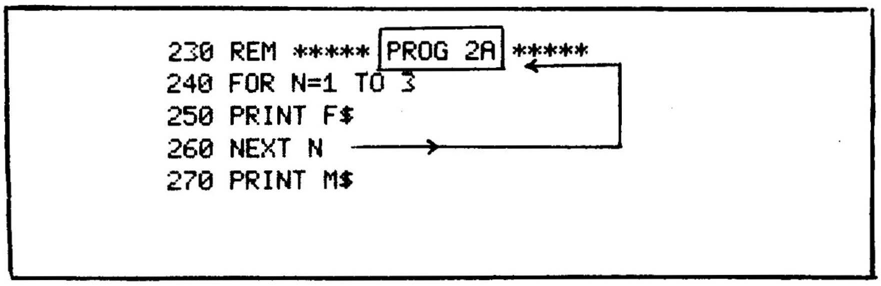
In assignment statements the word LET is optional (see lines 220, 230, Any sequence of line numbers will do. I have started at 200, to avoid confusion with Program 1, and have left gaps of ten between line numbers. Later I can modify the program by inserting a new instruction into any gap (as say 225 or 242). I just type the addition at the end - sorting into numerical order is automatic!

Line 280 prints a blank line, for double spacing. I can also change any line by overwriting it. If I type 220 INPUT M$ at the end, I replace the old line, which was an assignment statement, by a new input statement. (The program would still work!)

COUNTING.

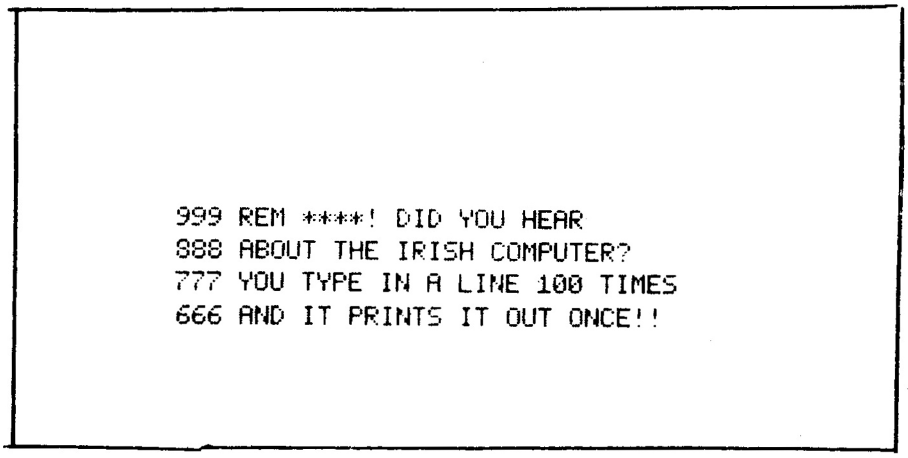
The "IF - THEN -" conditional jump has many applications. When, as here, it is used for counting only, an alternative formulation may be simpler.

Try replacing lines 230 to 270 by :-



```
230 REM ***** PROG 2A *****
240 FOR N=1 TO 3
250 PRINT F$
260 NEXT N
270 PRINT M$
```

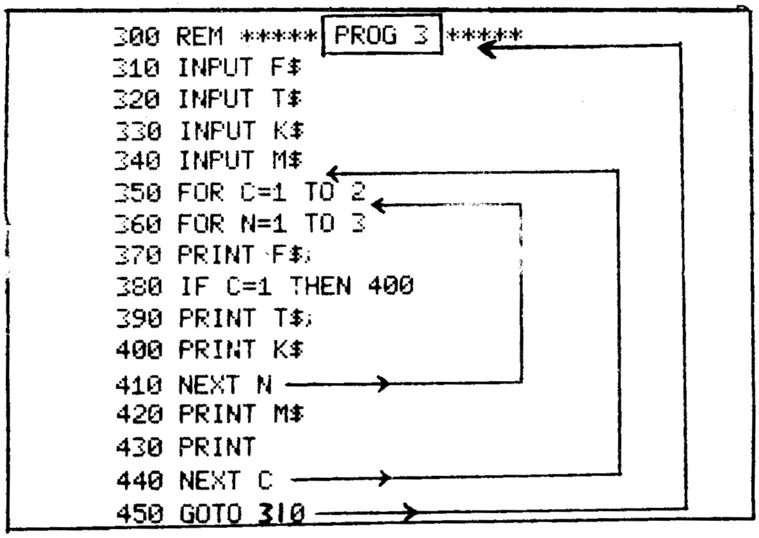
The control keeps returning from line 260 to line 240 until the counter N has moved through its range of values (1 to 3). This gives three readings of line 250. This is called a "FOR - NEXT" LOOP. (Note that the instructions replaced have been over-written: we could just type the line number 230 and leave it blank.)



```
999 REM ****! DID YOU HEAR
888 ABOUT THE IRISH COMPUTER?
777 YOU TYPE IN A LINE 100 TIMES
666 AND IT PRINTS IT OUT ONCE!!
```

NESTING.

So far we have used a single loop to print repetitions of a line. Program 3 will print two verses, and shows how any phrase can be recalled repeatedly:-



```
300 REM ***** PROG 3 *****
310 INPUT F$
320 INPUT T$
330 INPUT K$
340 INPUT M$
350 FOR C=1 TO 2
360 FOR N=1 TO 3
370 PRINT F$;
380 IF C=1 THEN 400
390 PRINT T$;
400 PRINT K$
410 NEXT N
420 PRINT M$
430 PRINT
440 NEXT C
450 GOTO 310
```

The strings F$, T$, K$, M$ are input respectively as "One finger" "One thumb" "Keep moving" and "We'll all be merry and bright".

The C loop gives the verses (1 to 2). In verse 1, C = 1 at line 380, so line 390 is omitted: in verse 2 it is included.

Note that the C loop completely encloses the N loop, which repeats a line three times within a verse. Loops must be properly "nested" and must not cross each other.

We want the strings located by F$, T$, and K$ on the same print line and M$ on a new line. Semi-colons in lines 370 and 390 keep the print line open, but line 400 closes it.

To summarise further principles we have met:-

4. BASIC programs contain assignment statements for string variables (A$) and for numerical variables (A). The latter use the usual rules of arithmetic.
5. Jump Statements allow us to loop, to branch or to count:-
   (a) GO TO - is an Unconditional Jump;
   (b) IF - THEN - and FOR - NEXT are Conditional Jumps.

As an exercise you could extend this program to further verses. Also, try "Polly put the kettle on", or "She'll be coming round the mountain."

In later articles we shall develop further the ideas of loops, strings and print format.

J.F.

Programs in Current Use

A full listing of "TRAINS", and notes on eleven more programs.

When space permits we will list whole programs with full documentation. I have selected "TRAINS" for the first issue. It has proved very popular, and it motivates children to practise arithmetic better than many teachers can.

It is also a straightforward program to adapt for different makes of microcomputer, and REM statements have been added for this purpose. The awkward part to re-write is the graphics routine to draw the trains. There are eight trains per line, starting in the top left corner. The program converts a parameter C (1 to 56) to a co-ordinate system (A, B) to position the train.

User Documentation.

This is a program designed to provide practice in addition, subtraction or multiplication. It is for use by individual pupils, or by small groups.

To use the program type RUN, and return.

The first decision is whether you are going to practice addition, subtraction or multiplication, in response to which you type the appropriate key (+,-,or *).

For each question that follows you have to specify the magnitude of the problem by choosing the number of digits in the two operands: a maximum of 4 in each is allowed. The resultant question is randomly generated. For example:

```
"How many digits in the problem?"
"Left hand side" 3
"Right hand side"2
263+47=310 ↲
```

In the example the user input is underlined, and the arrow indicates that the input must be followed by the return key. The answer to the problem must be provided as quickly as possible. In fact the program incorporates single key input and therefore the answer to each problem is the only number that needs to be followed by the return key.

If the answer is correct then you win 5 trains (the 3+2) or lose 5 if the answer is wrong. In the case of winning, the 5 trains are drawn in white on the screen, while in the case of losing they are changed to grey (so that you can still see how many have been deleted) only to be overwritten in white again if followed by a correct answer to a subsequent problem. This procedure is then repeated but may include either of two additional facilities. In response to "Left hand side" you can type S. This generates automatically a problem of the same magnitude as the previous one. Alternatively, if a particular answer is wrong you can type C. This generates the same problem for a second time and allows you one opportunity to correct it.

Targets may be set in two ways:-

i) to fill the screen with white trains (56 in all) in the minimum number of moves. The number of moves taken is printed out on the screen when 56 trains have been achieved. This should encourage pupils to try for the harder problems (ie. 4,4)

ii) Against the clock. The total time taken to answer all the questions is printed out. This should encourage pupils to try for speed and accuracy.

These two factors allow the teacher to build up a number of competitive situations in the classroom in order to encourage pupils to beat their own records. If at any time you wish to end the program then type F in response to "Left hand side."

One further facility is the ability to change the number of trains it needs to complete the game. For example if you wish to reduce the 56 trains to 40 trains then change line 820 in the program to:

```
820 IF C=40 THEN 910
```

This was not incorporated as a program option, so it remains outside the control of the person operating the program.

TRAINS

```
10 REM PROGRAM NAME-TRAINS
20 CLEAR100:RANDOMIZE
30 REM SET UP THE GRAPHICS AREA.
40 GRAPH 1
50 REM CHR$(12) CLEARS THE TEXT ON THE SCREEN
60 PRINT CHR$(12)
70 DIM A(56):TT=0
80 REM READ IN DIAGRAM DATA FOR DRAWING TRAIN.
90 FOR L=1 TO 10
100 READ A(L)
110 DATA 40,32,42,3,0,47,15,15,31,0
120 NEXT L
130 CM=0:SI=0
140 LV=1.TM=1
150 PRINT "DO YOU WISH TO ADD(+),MULTIPLY(*) OR    SUBTRACT(-)   ";
160 REM WAITS INDEFINITELY FOR A KEYBOARD CHARACTER
170 UU=GET(1):TV=GET()
180 IF TV=0 THEN 1260
190 REM OUTPUT CHARACTER TO SCREEN.
200 PRINT CHR$(TV)
210 IF TV=45 OR TV=43 OR TV=42 THEN 240
220 PRINT "SORRY,YOU CAN ONLY ADD,SUBTRACT OR MULTIPLY"
230 GOTO 150
240 A$=CHR$(TV)
250 PRINT"HOW MANY DIGITS IN THE PROBLEM"
260 PRINT"LEFT HAND SIDE      ";:X=GET()
270 IF X=70 THEN 1180
280 GH=0
290 REM SAME DIMENSION PROBLEM
300 IF X=83 THEN X=PV:Y=PW:PRINT:GOTO 430
310 IF X=67 THEN 380
320 PRINT X-48
330 PRINT"RIGHT HAND SIDE     ";:Y=GET():PRINT Y-48
340 X=X-48:Y=Y-48
350 IF X>0 AND X<5 AND Y>0 AND Y<5 THEN 430
360 PRINT "SORRY, THE NUMBER OF DIGITS MUST BE IN THE RANGE 1-4"
370 GOTO 250
380 GH=1:PRINT CHR$(67):X=PV:Y=PW:TM=TM*-1
390 IF TM=1 THEN 430
400 PRINT CHR$(12)
410 PRINT"SORRY,YOU ARE ONLY ALLOWED 1 CORRECTION"
420 TM=1:GOTO 250
430 CM=CM+1:B$="":CT=0:G=X+Y:PV=X:PW=Y
440 IF GH=1 THEN 550
450 REM W=RANDOM NO. IN RANGE 0-10^X
460 W=INT(RND(1)*10^X)
470 REM U=RANDOM NO. IN RANGE 0-10^Y
480 U=INT(RND(1)*10^Y)
490 IF A$ = "+" THEN V=U+W:GOTO 550
500 IF A$ = "*" THEN V=U*W:GOTO 550
510 IF W<U THEN 530
520 GOTO 540
530 Q=W:W=U:U=Q
540 IF A$="-" THEN V=W-U
550 ND=1:AN=0
560 PRINT W;A$;U;"=";
570 REM WAIT FOR 320 SECONDS FOR A CHARACTER FROM KEYBOARD.
580 XZ=GET(32000)
590 IF XZ=0 THEN TT=TT+32000:GOTO 580
600 IF XZ=127 THEN 1210
610 IF XZ=13 THEN710
620 IF XZ>=48 AND XZ<=57 THEN 650
630 PRINT CHR$(7);:GOTO 580
640 REM GET(-2)=TIME REMAINING FROM 32000.
```

```
650 TT=TT+32000-GET(-2)
660 PRINT CHR$(XZ);
670 REM +=CONCATENATION 0PERATOR.
680 B$=B$+CHR$(XZ)
690 GOTO 580
700 PRINT
710 AN=VAL(B$):PRINT
720 IF AN=V THEN 790
730 D=128:TM=-1
740 FOR C=LV-1 TO LV-G STEP -1
750 IF C<1 THEN 880
760 GOSUB 950
770 NEXT C
780 GOTO 860
790 D=192:TM=1
800 FOR C=LV TO LV+G-1
810 GOSUB 950
820 IF C=56 THEN 910
830 NEXT C
840 LV=LV+G
850 GOTO 870
860 LV=LV-G
870 GOTO 240
880 PRINT:PRINT"YOU ARE NOW BACK TO ZERO"
890 LV=1
900 GOTO 240
910 GRAPH0:PRINT CHR$(12)
920 PRINT "YOU HAVE FINISHED THE GAME IN";CM;"MOVES"
930 PRINT "YOU HAVE TAKEN";TT/100;"SECONDS":?:?:?:?
940 GOTO 1260
950 IF C<=8 THEN 1020
960 IF C<=16 THEN 1030
970 IF C<=24 THEN 1040
980 IF C<=32 THEN 1050
990 IF C<=40 THEN 1060
1000 IF C<=48 THEN 1080
1010 A=10*(C-49):B=4:GOTO 1100
1020 A=10*(C-1):B=59:GOTO 1100
1030 A=10*(C-9):B=50:GOTO 1100
1040 A=10*(C-17):B=41:GOTO 1100
1050 A=10*(C-25):B=32:GOTO 1100
1060 A=10*(C-33):B=22:GOTO 1100
1070 GOTO 1100
1080 A=10*(C-41):B=13
1090 REM DRAW TRAIN WITH (A,B-3) AS BOTTOM LEFT HAND CORNER.
1100 L=1
1110 FOR J=B TO B-3 STEP -3
1120 FOR K=A TO A+8 STEP 2
1130 PLOT K,J,A(L)+D
1140 L=L+1
1150 NEXT K
1160 NEXT J
1170 RETURN
1180 PRINT CHR$(12):PRINT"SO FAR YOU HAVE WON ";LV-1;" TRAINS IN ";CM;" MOVES"
1190 PRINT"IT HAS TAKEN YOU";TT/100;"SECONDS"
1200 GRAPH 0:GOTO 1260
1210 PT=LEN(B$)
1220 IF PT=0 THEN 580
1230 B$=MID$(B$,1,PT-1)
1240 ?CHR$(127);
1250 GOTO 580
1260 GRAPH 0
1270 END
```

"NECOPS" Documentation Booklets

"TRAINS" is one of a series of programs described in the first edition of "Newman College Primary Software Development". This booklet contains the documentation for six programs written for primary schools.

1) GRID - a program associated with regions and addresses, as in Fletcher Maths, level II, book 1.

2) DIAGRAM - a program designed to draw pictures by directing a cursor using either compass directions or rotations in multiples of 45^{o}.

3) COIN - an introduction to simple probability by spinning a coin n times. It also includes a study of the sequences generated.

4) TRAINS - see full notes above.

5) FSNAP - fraction snap presented as a game. The fractions can either be in numeric or diagrammatic form.

6) MULTEST - a program which generates individual 'tables' tests based upon information from previous tests held upon a disc file.

The second edition will contain six more programs. They are:

7) DICE - a follow up to COIN. The user designs a die by specifying 4, 6 or 12 sides and numbering the faces as he likes. The program displays the build-up of a bar graph as the die is thrown n times.

8) GRIDREF - a geography program to help teach the basic ideas of 4 and 6 figure grid references.

9) SPELL - a program for early steps in reading. The pupil supplies the missing letter of a word. The output on the screen is in enlarged 'Ladybird' script.

10) SPELL 2 - a similar program in which the pupil has to retype a word after it has appeared on the screen for 5 seconds
(N.B. both SPELL and SPELL 2 can cater for any vocabulary - in <u>any</u> language. French teachers take note!)

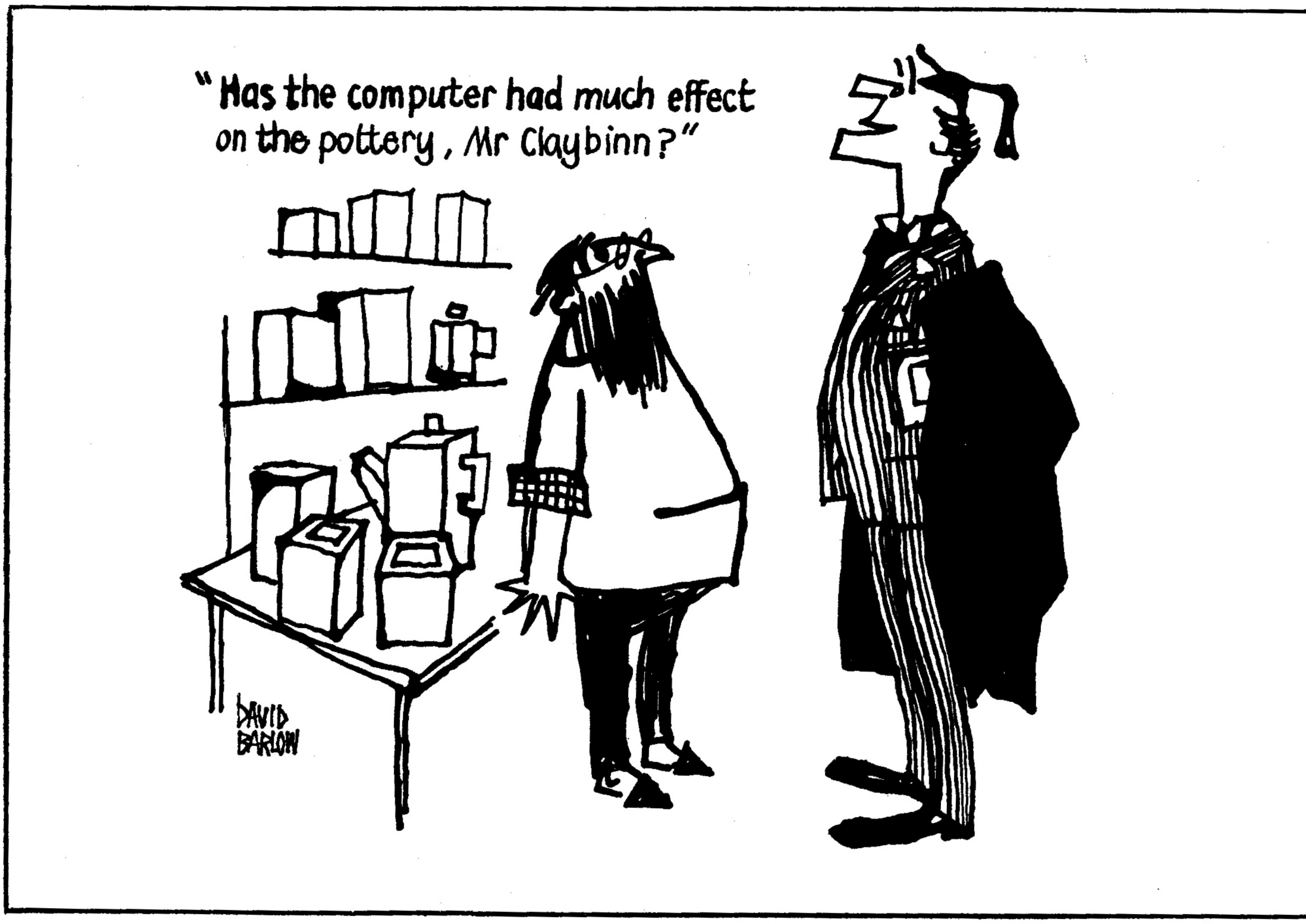
11) PHRASES - a jumbled sentence appears on the screen which the pupil has to re-arrange. It is the phrases (not the individual words) that are scrambled.

12) FTRAIN - a program similar to TRAINS, but providing practice in addition, subtraction and multiplication of fractions.

All these programs have been developed on Research Machines 380Z; FSNAP and SPELL require the high resolution graphics board and MULTEST requires a disc system.

The programs themselves can be obtained free by sending us a 380Z disc or cassette. Remember to let us know which version you need - DBAS9, BASICSG or the new cassette BASICV5.

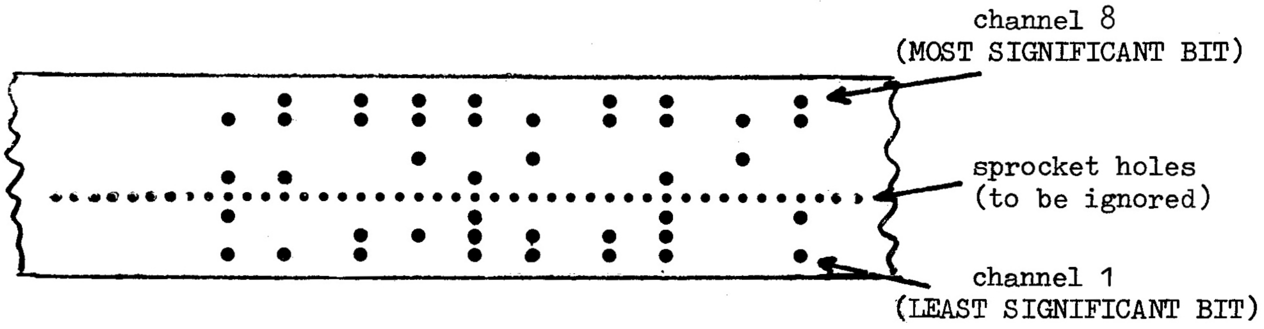
The two booklets containing full documentation can be obtained by sending us a postal order for 75p (50p for just one booklet). Write to R. Keeling, Newman College.


"Has the computer had much effect on the pottery, Mr Claybinn?"
DAVID BARLOW

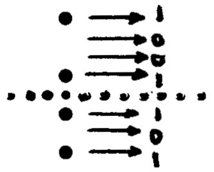
PUZZLE PAGE - a coding diversion

Why do we teach multi-base arithmetic? Not mainly for its practical applications, but because it reinforces the concepts of our everyday number system. Yet base 2 (binary) does have a practical use. All computer arithmetic is performed in this base. Input is converted, internally and automatically, into a series of on/off pulses (1's and 0's) and then manipulated internally in this form. Only at the output stage is conversion back to denary (decimal) achieved.

One device which illustrates this translation process is paper tape. Before the advent of microcomputers, the school at which I taught had a teletype terminal. Any characters typed on the keys would also appear as a coded message on paper tape; that is, a narrow strip of paper with a series of small holes punched out. Once a week the tapes were sent for processing to a mainframe computer with a paper tape reader. Thankfully, microcomputers mean the computing power is now on the premises and input can be direct. However, decoding paper tape 'by hand' can be fun. For example, consider this section of 8 channel paper tape.



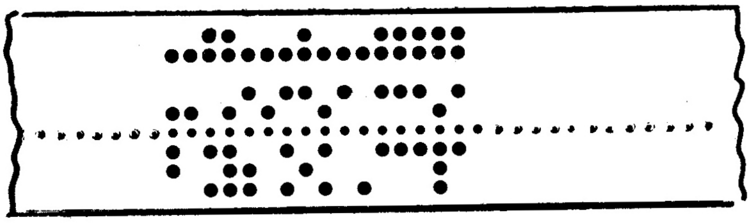
We can ignore the eighth channel which acts as a check against faulty data transmission. The standard paper tape code represents an A by the number 65, B = 66, .. Z = 90. A 'space' is represented by the number 32. Therefore to translate paper tape, write down each character as a binary number. A hole represents a 1 (because it allows a beam of light or a metal tooth to pass through to complete a circuit), and a 'filled' position represents a zero. Here is the first character on the left from the above example. Reading downwards and ignoring the eighth channel, this gives 1001101.



$1.2^6+0.2^5+0.2^4+1.2^3+1.2^2+0.2^1+1.2^0 = 64+8+4+1 = 77 = M.$

Repeat this process for every character on the tape, and you should end up with MICROSCOPE.

Problem: This is an anagram of a well known football team. Which?



R.K.

The following article comes from the Headmistress of a tiny rural two-teacher school with 36 children. It is encouraging to see such a realistic approach to long-term development coupled with obvious enthusiasm and vision. Clearly this "Teacher's Pet" will remain a favourite for a long time!

A Computer Project at Yealand C.E. Primary School

During the Summer Term of 1979, Mrs J. Tunnicliffe-Wilson, a parent and School Governor of Yealand C.E. Primary School offered to the school a PET Computer for unlimited use, an offer which was promptly accepted. On several occasions Mrs Tunnicliffe-Wilson instructed the children and myself as to some of the simple functions of the computer and then left us with the parting comment, "You may do anything with the computer; you won't damage it".

This comment set the tone of future work with the computer. Obviously the situation at Yealand, with only twenty children in a class, lends itself to such a project, but our PET sits happily in the classroom on three days each week and all the children in the school, from 4+ to 11 year olds, have access to it at sometime during the school day, computer work now becoming an integral part of the curriculum. The model used is a PET 8K and, as yet, we have found it entirely satisfactory for our needs. It is very compact and quite easily movable, an advantage over the computer with added disc drive or printer interface.

Computer work in the Primary School classroom has various aspects. Many teachers would wish simply to use the computer as a teaching aid or to give extra practice in the basic skills of Mathematics and Language and the computer at Yealand is used partly in this way. However, we also feel that children should develop an awareness of the computer as an exciting and tremendously useful tool in a technological age, that they should begin to understand the various functions of a computer and that they should have some knowledge of elementary programming and it is in this direction which the children of Yealand are encouraged to progress.

Computer work cards have been written guiding the children through the many aspects of the computer, beginning with the basic manipulation of the machine, the keyboard functions, the methods of calculating, the 'string' processes, the ways and means of editing and leading on to the writing of simple programs. All this work includes many programs which give extra practice in Mathematics and Language skills but which, at the same time, are training the children in one or more of the above aspects.

We have worked on the computer project now for well over one year and the value of such work is quickly becoming apparent. Children of this age do not show nervous apprehension but only fascination for hardware of this kind and quickly learn to handle the machine competently. An attempt at programming the computer needs much logical thinking and the work is encouraging and developing this as well as adding another dimension to normal classroom work. Children who often find concentration difficult will work at the computer with much longer spans of concentration. The computer work is emphasising more accuracy on the part of the child.

Our first instincts were to write Mathematics programs such as sequencing, fractions, 'shopping' and multiplication: then to progress to English programs based on spelling, punctuation, phonics and adjectives. Now we are moving into other fields with programs identifying plants found in the nearby pond and with a Histogram program analysing rainfall data. The opportunities seem endless and we at Yealand C.E. Primary School are convinced that the computer work, which is both stimulating and worthwhile, has a very exciting future.

Mrs B.E. Hartley
(Yealand C.E. School,Carnforth,Lancs.)

Program Listing - "PHRASES"

The first issue contained a description of TRAINS, a program designed to encourage the practice of arithmetic. We now turn our attention to language work and to a program that could be used by individuals or small groups of pupils. N.B. The programs as listed will only run under Basicsg on a RML 380Z machine, and will need adapting to work on different microcomputers. We would be very pleased to hear from anyone who has adapted TRAINS to run on a different microcomputer (PET, Tandy etc).

I must first emphasise that the object of providing a reasonably detailed explanation of a particular program is to provide the reader with an insight into the term 'educational software'.

Now for the program. It presents in jumbled order a set of phrases which make up a meaningful sentence. For example, the screen display might appear as:

```
_to_play_   they_go   with_the_others   into_the_water
    1          2             3                 4
```

The pupil has to determine a sensible order - e.g. 2413.

In response to RUN the first problem is displayed on the screen. If the pupil has no ideas, he can type "R" (for re-shuffle) and the phrases are presented on the screen in a new order - this may now trigger some response. When the pupil is ready, he types in what he thinks is the correct order as a single number, like 2413. (This stage will generate a great deal of lively and purposeful discussion if a small group is gathered round the keyboard - a valuable offshoot of the work.)

The phrases are now displayed, in the order the pupil has typed, as a complete "sentence". Marking is instantaneous, on the screen! Even if right, the pupil is encouraged to search for other valid possibilities. When he is ready, he can type "N" (for Next) to call up any sentences he has missed, and then advance to the next problem set of phrases. A total of three errors similarly produces corrections followed by a new problem.

The program ends when all the problems stored have been presented, or when "S" (for Stop) is typed. An individual score then shows the number of problems attempted and the total of correct rearrangements.

In the listing that follows we have inserted trivial data as examples on lines 1390 onwards. The teacher writes his own selection of data into these lines. This gives great flexibility, a typical feature of language programs. It can cater for any age or ability level, and embody any subject matter or even a foreign language.

This program can be set to store any number of problems, and to present a given number of these at random, without repetition. For each problem, the machine must recognise <u>all</u> the right answers. In the example given above, data would be provided in this form:

```
 1390 DATA they go, into the water, to play, with
the others, 0, 2134, 3412,1
```

<u>Commas</u> separate the phrases, <u>0</u> terminates the sentence, and <u>1</u> terminates the list of acceptable rearrangements. (There could still be awkward decisions - is 1342 valid?) The last data line must be ZZ.

One headmaster listed three advantages of this program over the traditional approach:-

1) the facility to reshuffle the phrases, in the hope of providing the pupil with a clue, is something that no textbook or workcard can offer;
2) teachers tend to mark this work as right or wrong whereas the program encourages the pupil to look for all the correct possibilities;
3) this program enables pupils to work through many more examples than would be possible if they had to write the answer down each time. Perhaps we ask children to write far too much - this program concentrates on constructive thinking.

The program has also proved a useful stimulus in suggesting other applications in the language area.

R.K

'May I see your program, Sir?'

```
100 REM ****** PHRASES PROGRAM ******
110 CLEAR 1000
120 ON BREAK GOTO 1320
130 DIM A$(20),B$(10),S$(20),A(20),S(20),B(10):RANDOMIZE:?CHR$(12):RT=0
140 QQ$="123456789":SP$="                                        "
150 Z$="--------------------------------------"
160 INPUT "HOW MANY PROBLEMS";N
170 IF N<1 OR N>20 OR N<>INT(N) THEN ?"EH?":GOTO 160
180 FOR BL=1 TO N:WT=0
190 FOR U=1 TO 20:A(U)=0:A$(U)="":NEXT U:FOR U=1 TO 10:B$(U)="":B(U)=0:NEXT U
200 T=INT(RND(1)*4)+1:CC=0
210 NN=0
220 NN=NN+1:READ A$(NN)
230 IF A$(NN)="0" THEN 260
240 IF A$(NN)="ZZ" THEN RESTORE:GOTO 210
250 GOTO 220
260 NS=1:B$(1)=LEFT$(QQ$,NN)
270 NS=NS+1:READ B$(NS):IF B$(NS)="1" THEN 290
280 GOTO 270
290 CC=CC+1
300 IF CC<T THEN 210
310 NN=NN-1:NS=NS-1:FOR L=1 TO NN:A(L)=L:NEXT L
320 ?CHR$(12):GRAPH 1:PLOT 28,24,"SENTENCE"+STR$(BL)
330 UU=GET(50):GOSUB 470:?CHR$(12)
340 GRAPH 1:TL=2:PP=55
350 FOR NO=1 TO NN:ZZ$=A$(NO):GOSUB 610:NEXT NO
360 G$="":TL=2:PP=25:?"INSTRUCTION? ";
370 PLOT 0,30,1:LINE 79,30
380 UU=GET(0):I$=GET$()
390 IF ASC(I$)=13 THEN 680
400 IF ASC(I$)=27 AND G$<>"" THEN ?CHR$(12):GOSUB 1330:GOTO 360
410 IF LEN(G$)=NN THEN 380
420 IF I$="R" THEN ?CHR$(12);"RE-SHUFFLING ";:GOTO 330
430 IF I$="N" THEN ?CHR$(12);"NEXT QUESTION";:UU=GET(50):GOTO 890
440 IF I$="S" THEN BL=N:GOTO 1230
450 IF ASC(I$)<49 OR ASC(I$)>48+NN THEN 380
460 G$=G$+I$:?I$;:NO=VAL(I$):ZZ$=A$(NO):GOSUB 610:GOTO 380
470 REM ********* RANDOMIZER *********
480 L=A(1)
490 FOR U=1 TO 10
500 R1=INT(RND(1)*NN)+1:R2=INT(RND(1)*NN)+1
510 T$=A$(R1):T=A(R1):A$(R1)=A$(R2):A(R1)=A(R2):A$(R2)=T$:A(R2)=T
520 NEXT U
530 IF A(1)=L THEN 490
540 REM ********* RESET!?
550 FOR U=1 TO NN:S(A(U))=U:NEXT U:FOR U=1 TO NN:A(U)=U:NEXT U
560 FOR L=1 TO NS:T$=""
570 FOR U=1 TO NN:UU=VAL(MID$(B$(L),U,1)):T$=T$+CHR$(S(UU)+48):NEXT U
580 B$(L)=T$
590 NEXT L
600 RETURN
610 REM ********* PLOTTER ************
620 ZL=LEN(ZZ$)
630 IF TL+ZL+2>38 THEN TL=2:PP=PP-9:IF PP<0 THEN PP=PP+9
640 PZ=INT(ZL/2+0.5)
650 U$=MID$(Z$,1,PZ-1)+CHR$(48+NO)+MID$(Z$,1,ZL-PZ)
660 PLOT 2*TL,PP,ZZ$:PLOT 2*TL,PP-3,U$:TL=TL+ZL+2
670 RETURN
680 REM ******** GUESSER ************
690 ?CHR$(12)
700 IF VAL(G$)=0 THEN ?"A NUMBER  @E.G. ";LEFT$(QQ$,NN);" ";:GOTO 1110
710 IF LEN(G$)<NN THEN ?"TOO FEW NUMBERS. ";:GOTO 1110
720 R=0
730 FOR L=1 TO NN-1
740 FOR U=L+1 TO NN:IF MID$(G$,U,1)=MID$(G$,L,1) THEN R=1
750 NEXT U
760 NEXT L
770 IF R=1 THEN ?"DON'T REPEAT ANY PHRASES ! ";:GOTO 1110
780 CT=0:NX=NZ+1
790 FOR L=1 TO NS
800 IF G$=B$(L) AND B(L)=1 THEN CT=2
810 IF G$=B$(L) AND B(L)=0 THEN CT=1:B(L)=1
820 NEXT L
830 IF CT=2 THEN ?"YOU'VE ALREADY GIVEN THIS SOLUTION !! ";:GOTO 1110
840 IF CT=1 THEN 1120
850 REM ****** WRONG ROUTINE *******
860 ?CHR$(12):WT=WT+1
870 PLOT 0,30,63:LINE 78,30:LINE 78,58:LINE 0,58:LINE 0,0:LINE 78,0:LINE 78,30
880 IF WT<3 THEN 1020
890 UU=GET(100):GRAPH 1:FF=0:?CHR$(12)
900 IF WT=3 THEN PLOT 17,24,"HERE, LET ME TELL YOU.":UU=GET(200)
910 GRAPH 1
920 FOR U=1 TO NS
930 IF B(U)=1 THEN 990
940 PP=40:TL=2
950 IF FF>0 THEN PLOT 35,24,"ALSO...":UU=GET(200):GRAPH 1
960 FOR NO=1 TO NN:ZZ$=A$(VAL(MID$(B$(U),NO,1))):GOSUB 610:NEXT NO
970 FOR L=1 TO 6:PLOT 32,0,"      ":UU=GET(20):PLOT 32,0,"ANSWER":UU=GET(20):NEXT L
980 FF=FF+1:UU=GET(0):?"OK?";:UU=GET():?CHR$(12):GRAPH 1:UU=GET(80)
990 NEXT U
1000 IF FF=0 THEN PLOT 10,24,"THERE WERE NO MORE ANSWERS !!":UU=GET(200)
1010 GOTO 1230
1020 ?CHR$(12);"SORRY, THIS IS NOT A PROPER SENTENCE.":UU=GET(400):?CHR$(12)
1030 GRAPH 1:PLOT 15,24,"HERE'S YOUR ANSWER AGAIN.":UU=GET(100)
1040 PLOT 20,20,"HAVE ANOTHER TRY.":UU=GET(200)
1050 FOR U=1 TO NN
1060 J=VAL(MID$(G$,U,1)):S$(U)=A$(J):S(U)=A(J)
1070 S$(U)=A$(J):S(U)=A(J)
1080 NEXT U
1090 FOR U=1 TO NN:A$(U)=S$(U):A(U)=S(U):NEXT U
1100 GOSUB 540:GOTO 340
1110 UU=GET(300):GOSUB 1330:?CHR$(12):GOTO 360
1120 REM ****** RIGHT ROUTINE *******
1130 ?CHR$(12)
1140 FOR U=1 TO 11
1150 L=(U/2-INT(U/2))*12+192
1160 PLOT 0,30,L:LINE 78,30:LINE 78,58:LINE 0,58:LINE 0,1:LINE 78,1:LINE 78,30
1170 NEXT U
1180 FOR U=1 TO 5:UU=GET(20):GRAPH 2:UU=GET(20):GRAPH 3:NEXT U:UU=GET(300)
1190 GRAPH 1:PLOT 15,24,"YES, THAT IS ONE ANSWER !":UU=GET(300)
1200 GRAPH 1:PLOT 8,24,"CAN YOU SEE ANY OTHER ANSWERS ?"
1210 RT=RT+1:UU=GET(400):GOTO 340
1220 GOTO 340
1230 NZ=NX
1240 NEXT BL
1250 REM *****************************
1260 ?CHR$(12):GRAPH 1
1270 IF NZ<>1 THEN PLOT 12,30,"YOU ATTEMPTED"+STR$(NZ)+" SENTENCES"
1280 IF NZ=1 THEN PLOT 12,30,"YOU ATTEMPTED 1 SENTENCE"
1290 IF RT<>1 THEN PLOT 15,24,"AND GOT"+STR$(RT)+" RIGHT ANSWERS."
1300 IF RT=1 THEN PLOT 15,24,"AND GOT 1 ANSWER RIGHT."
1310 UU=GET(500)
1320 TEXT:?CHR$(12):END
1330 REM ****** CLEAR ROUTINE **********
1340 FOR U=25 TO 1 STEP -3
1350 PLOT 1,U,SP$
1360 NEXT U
1370 RETURN
1380 REM *********** DATA **************
1390 DATA IT IS GOING,TO BE,VERY HOT,THIS AFTERNOON,0,4123,1
1400 DATA AFTER SOME TIME,THEY ALL CAME OUT,OF THE WATER,ONTO THE SAND
1410 DATA TO HAVE SOME TEA,0,1
1420 DATA I CAN GO,INTO THE WATER,IF I GET,TOO HOT,0,3412,1
1430 DATA THE CHILDREN,LIKE TO BE,ON THE SANDS,IN THE SUN,0,1
1440 DATA PETER FINDS,A FRIEND,WHO LIVES,NEXT DOOR,TO HIS UNCLE,0,1
1450 DATA THEY GO,INTO THE WATER,TO PLAY,WITH THE OTHERS,0,2134,3412,1
1460 DATA JOHN TAKES,HIS DOG,FOR A WALK,IN THE PARK,AFTER SCHOOL
1470 DATA 0,51234,1
1480 DATA JANE HELPS HER MOTHER,TO MAKE,CAKES FOR TEA,IN THE KITCHEN,0,4123,1
1490 DATA ZZ
```

MICRO-SCOPE 2,April 1981

A PRIMARY B.A.S.I.C.-Part 2

In this second article we intend to build upon the ideas introduced in the first issue of MICRO-SCOPE. We shall develop new ways of handling "strings" to show how the computer can recognise words typed in by pupils.

REVISION Here is a reminder of the principles we met in Part 1.

BASIC programs consist of a sequence of instructions called statements. Each statement has a line number. Statements are read and obeyed in numerical order. BASIC statements include:-

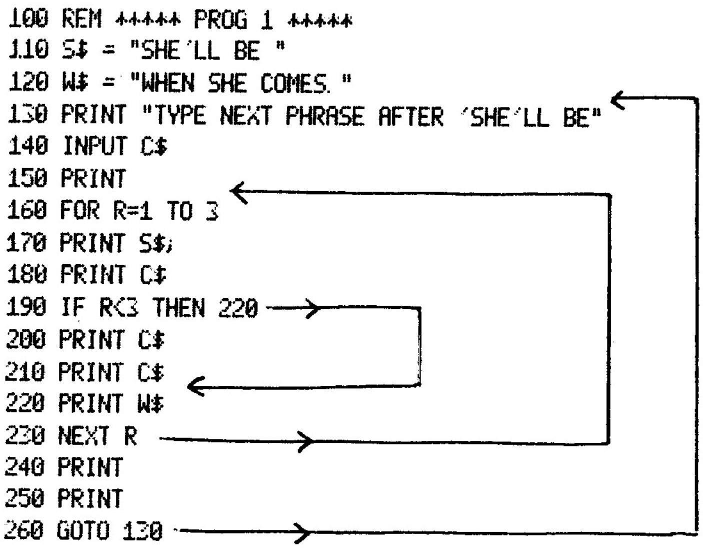
1.Assignment statements for 'string' variables (A$) and for 'numerical' variables (A). The latter use the usual rules of arithmetic;

2.Jump statements a) Unconditional GOTO -
b) Conditional IF - THEN -
c) Conditional FOR - NEXT (for counting in loops);

3.Input/Output statements
a) INPUT -
b) PRINT - which closes the print line
c) PRINT -; which keeps the print line open;

4.REM statements. Remarks are ignored by the computer.

I am sure that the more adventurous of you did try your hand at programming. In most cases I guess you did not have a computer to hand. Here is one possible solution to one of the extension exercises in part 1 - "She'll be coming round the mountain". Think of the pattern of repetitions of the verse. We will use the first letters, i.e. S = She'll be, C = Coming round the mountain, W = When she comes. The pattern is SCW, SCW, SCCCW. Now look at program 1.



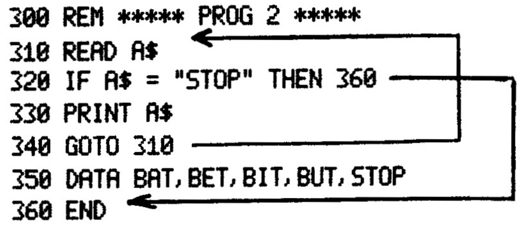
```
100 REM ***** PROG 1 *****
110 S$ = "SHE'LL BE "
120 W$ = "WHEN SHE COMES. "
130 PRINT "TYPE NEXT PHRASE AFTER 'SHE'LL BE"
140 INPUT C$
150 PRINT
160 FOR R=1 TO 3
170 PRINT S$;
180 PRINT C$
190 IF R<3 THEN 220
200 PRINT C$
210 PRINT C$
220 PRINT W$
230 NEXT R
240 PRINT
250 PRINT
260 GOTO 130
```

The R loop (lines 160-230) gives the three SCW repeats which are output at lines 170, 180 and 220. The conditional jump at line 190 causes the extra lines 200 and 210 to be omitted except on the third occasion. Lines 240, 250 give line spacing between verses by printing nothing and line 260 returns us to line 130 for another verse.

DATA SEARCHES So far we have considered computer applications which involve repetition and there are many uses of computers under this heading. But now we are going to consider another distinct type of application which involves matching. One of the powers of the computer is to be able to compare any two sets of symbols. For example, a list of INPUT words can be compared with a set of stored DATA. The educational implications of such comparison is very wide. In the example which follows we are tyring to help children think of and spell words beginning with B and ending with T. This exercise could be useful for spelling, vocabulary or for word recognition in early reading. However, this program would be just as suitable, with slight modification, for a test on last week's vocabulary (in any language!) or a revision of a child's personal dictionary.

Program 2 illustrates a new way of storing and printing a list of words. In Program 3, our use of "string handling" extends to the recognition of words. Each word typed in is compared with the words in a stored list. The programmer provides a framework, but the teacher can select the contents of the list anew each time he uses the program.

READ + DATA Statements We shall require a few more BASIC ideas, so let us first examine program 2.

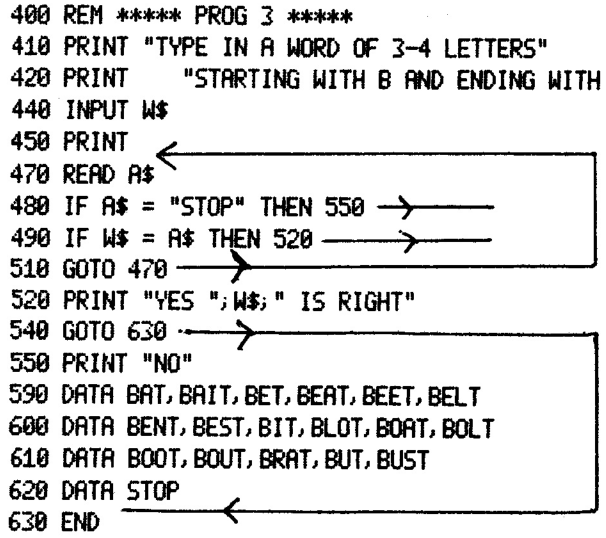


```
300 REM ***** PROG 2 *****
310 READ A$
320 IF A$ = "STOP" THEN 360
330 PRINT A$
340 GOTO 310
350 DATA BAT,BET,BIT,BUT,STOP
360 END
```

A list of words is stored at line 350. This is a DATA statement in the program, the different items of DATA being separated by a comma. The words can be copied one at a time by the READ statement at line 310 into the computer store located by A$. Each time the READ statement is used in its loop, a reading pointer inside the computer moves to the next item of DATA. This in turn replaces the previous item at A$. This method of assigning the contents of A$ (or indeed A if we were using 'numerical' instead of 'string' variables) is often preferred to an INPUT statement. Here it allows us to process a whole set of data without interrupting a fast-running program by keyboard entries. Note that DATA can only be read by READ statements and not by INPUT statements.

Line 320 is a conditional jump designed to check when all the DATA have been read. If the word "STOP" is recognised here by the matching process, the condition is fulfilled and a jump follows. Line 360 is an END statement which finishes the program run. A column of four words is printed from line 330. How would you use punctuation to have them printed on one line?

Now look at program 3.



```
400 REM ***** PROG 3 *****
410 PRINT "TYPE IN A WORD OF 3-4 LETTERS"
420 PRINT    "STARTING WITH B AND ENDING WITH T."
440 INPUT W$
450 PRINT
470 READ A$
480 IF A$ = "STOP" THEN 550
490 IF W$ = A$ THEN 520
510 GOTO 470
520 PRINT "YES ";W$;" IS RIGHT"
540 GOTO 630
550 PRINT "NO"
590 DATA BAT,BAIT,BET,BEAT,BEET,BELT
600 DATA BENT,BEST,BIT,BLOT,BOAT,BOLT
610 DATA BOOT,BOUT,BRAT,BUT,BUST
620 DATA STOP
630 END
```

Lines 410, 420 print out relevant instructions to the pupil. Line 440 locates the pupil input as W$. Lines 470 to 510 form a loop in which each item of DATA is reviewed in turn and compared with the pupil input. Line 470 locates one word at a time from DATA as A$: line 490 recognises when there is a match. When DATA list is exhausted, without a match, the condition in line 480 will be fulfilled. In every case, a jump out of the loop follows, and the result of the search of DATA is printed. Note that, after line 550, the program reaches line 630 END by passing over the DATA statements, which can only be accessed by READ statements.

RESTORE If we now modify the program by overwriting with the lines in program 3A, the above process can be repeated for a succession of pupil inputs.

```
430 REM ***** PROG 3A *****
440 INPUT W$
450 PRINT
460 RESTORE
470 READ A$
480 IF A$ = "STOP" THEN 550
490 IF W$ = A$ THEN 520
500 IF W$ = "STOP" THEN 630
510 GOTO 470
520 PRINT "YES ";W$;" IS RIGHT"
530 PRINT "TRY ANOTHER OR TYPE 'STOP' TO FINISH"
540 GOTO 440
550 PRINT "NO ";W$;" IS NOT IN MY DICTIONARY"
560 PRINT "PLEASE TRY AGAIN"
570 PRINT "OR ASK FOR THE CORRECT SPELLING"
580 GOTO 440
590 DATA BAT,BAIT,BET,BEAT,BEET,BELT
600 DATA BENT,BEST,BIT,BLOT,BOAT,BOLT
610 DATA BOOT,BOUT,BRAT,BUT,BUST
620 DATA STOP
630 END
```

As mentioned earlier the DATA pointer moves along the items of DATA one at a time. If it attempts to go beyond the last item of DATA the program would stop, probably with some error signal. The statement RESTORE, as in line 460, resets the pointer to the beginning of the DATA list. This has the effect of making all the items of DATA available repeatedly for review.

Observe carefully the punctuation of lines 520, 550. The semicolon (;) keeps the print line open so that the output is all on one line.

Try to follow the program through for successive keyboard entries by a pupil (e.g: BIT, BEIT, BAIT, or even STOP!) Obey each instruction in order making jumps where necessary (e.g. if a condition is fulfilled). Note at each stage the correct position of the DATA pointer (it starts as BAT) and the current contents of locations WØ, AØ.

SUMMARY Now in addition to learning different ways of jumping and looping for searching through a word list, we have met the following BASIC statements.

1. READ and DATA Statements
   These always go together: DATA can only be read by means of READ statements.
2. RESTORE resets the DATA pointer to the first item of DATA.
3. END Statements stop a program run.

EXERCISE How would you modify program 1 using READ + DATA statements (without any INPUT) so that several verses are printed out without a break?

* * * * * * * *

Please write to us if you have some simple idea for a program which involves the concept of matching. Try to break it down into very elementary operations (using, for example, the BASIC statements we have met so far). We will have a go at completing the program!

J. Fair.

A SCRAP-BOX COMPUTER

Children can learn a great deal about how computers work without waiting for expensive hardware. This article, the first of a series, describes a role-playing use of a simple model. It emphasises the function of the control unit, and shows how even a simple program needs many steps.

Miss Jones has just finished a discussion with her class on their latest topic work: Space Travel.

Miss Jones: What do you remember most?

Children: (in chorus): The computers!!

Miss J: Why is that?

Peter: They're so clever, much cleverer than us. They have lots of flashing lights and they can speak.

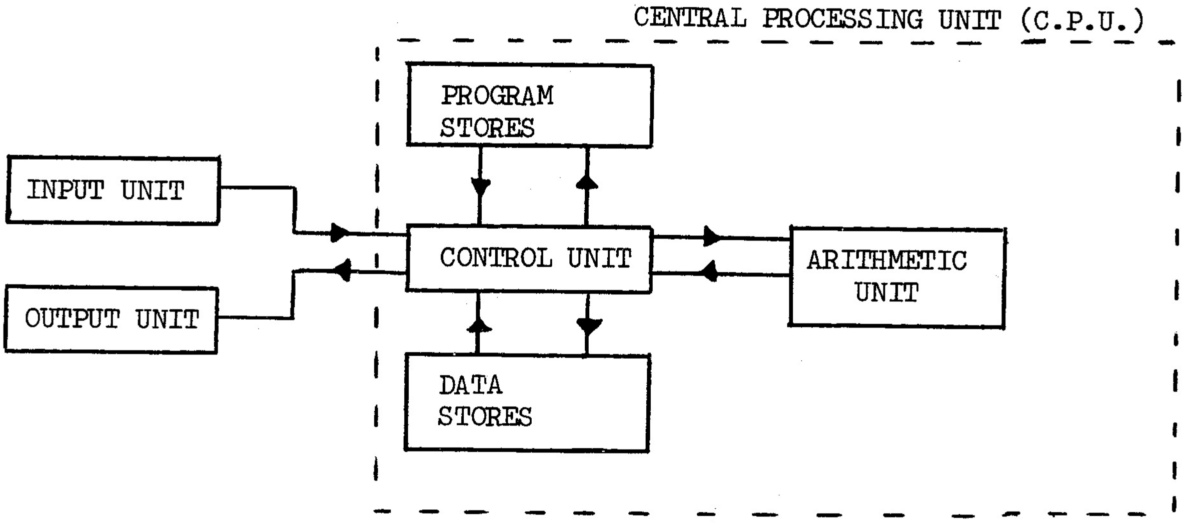
Miss J: Computers aren't clever at all. They can only do what people tell them to do, but they can do these things very quickly.

Dawn: (her eyes alight with **excitement**):Please tell us how a computer works, Miss.

Miss J: Computers are very complicated machines which can carry out many simple tasks very quickly; much faster and more accurately than we humans can. That is why we think they are so wonderful.
The best way I can help you to understand how a computer works is to let some of you imitate the main parts of a computer in a game. We will do this after the weekend.

(Next Monday)

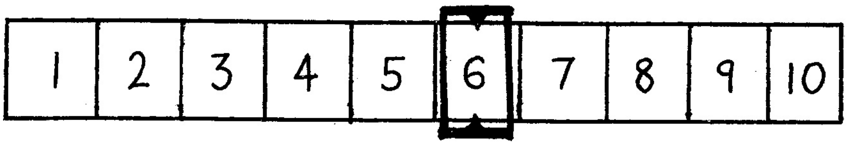
Miss J: A computer is made up of several parts and I have drawn this diagram showing how they are connected. The arrows indicate in which direction the information can pass. (She shows the children a large sheet containing the diagram below).



Computers have to carry out a set of instructions, called a PROGRAM. You will learn best if you act out how the different parts are used. Everyone taking part will need a magic slate and a pencil. Andrew can be the Arithmetic Unit and he can use a calculator. The rest of you will need these items I have here.

She allocates the items as follows:

(a) Colin, who is playing the part of the Control Unit, is given a position marker. This has a sliding window to indicate the program instruction being executed.



COLIN'S POSITION MARKER

(b) Pam, who is in charge of the Program Stores, is given a set of storage compartments which can be made from eggboxes, matchboxes or other suitable material. They are coloured blue to match the Program Instruction Cards, which will be stored in them. The individual stores are given numbers to identify them.

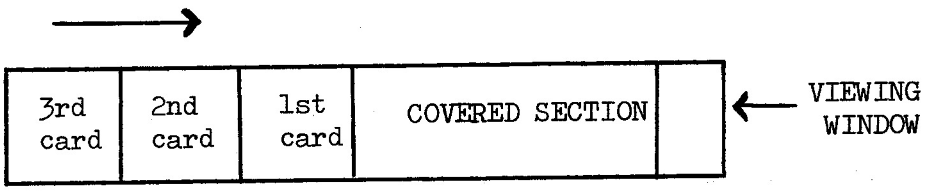
1	2	3	4	5
10	9	8	7	6
11	12	13	14	15
20	19	18	17	16

PAM'S STORES

(c) The Data Stores, being looked after by Dawn, resemble the Program Stores but are painted yellow to match the Data Cards. They also have letters with which to identify individual compartments.

(d) The Output Unit, operated by Oliver, does not need any special equipment but Miss Jones felt that writing the output on a piece of paper might help to distinguish output for humans from the internal communications, which are carried out by means of the magic slates.

(e) Ian, playing the part of the Input Unit , is given a track along which he can slide (from Left to Right) the program or data cards. These are then read, in correct sequence, at the viewing window after passing under a short covered section.



Cards are entered from the left

IAN'S INPUT TRACK

The children sit down in their places and link the pieces of apparatus together with wool to represent lines of communication. Black is used to show the movement of information into Colin's Control Unit and white wool is used for the movements of information from his Control Unit.

The rest of the class gather round to watch.

Miss J: I have already written a short program, in the BASIC programming language, which I am sure you will easily understand. The program will make a computer find the average of three numbers.
I have put the Program Instruction on blue cards and the information to be processed, called Data, on yellow cards. That way we can tell them apart.

INSTRUCTION 1	INSTRUCTION 2	INSTRUCTION 3	INSTRUCTION 4
INPUT A,B,C	LET D=A+B+C	LET E = $\frac{D}{3}$	PRINT E

INSTRUCTION 5	DATA	DATA	DATA
END	7	8	3

(The cards are made from thin plastic, or card which has been covered with a plastic film. Each card can then be written on with a felt pen containing washable ink, so the cards are re-usable.)

Miss J: The instruction numbers tell the computer the order in which to obey the instructions.

The program input sequence is very simple. The input unit (that's you Ian!) passes on the instruction cards, in sequence, to Colin as he requests them. Colin, you set your position marker to 1 when you are given the order to start and then you request the first card from Ian and pass it on to Pam, who will store it in her program store, 1. When you've done this, Pam, you must tell Colin that the task is completed by saying: 'Done'. Colin will then advance his marker to position no. 2 and ask Ian for the next instruction. This order of events is followed until Ian tells Colin that there are no more cards left in his unit.

The children find the repetition of these small and obvious steps rather slow - perhaps computers are already beginning to lose some of their mystique.

Miss J: Good, the execution sequence is more interesting and this is where the computer actually calculates the average and prints the result. I want each of you to remember to say 'Done' at the end of each operation that Colin asks you to carry out. He is the Controller and should not move on to the next operation until he has been informed that the current one has been completed. Also, I want you all to note that Pam does not give Colin the cards with the instructions written on them, she only gives him a copy on her slate. This makes sure that the instructions remain available for use with new information over and over again, until we have finished with the program. It can then be saved, in a special box, for future use. Let us go through the instructions slowly so that we will not (hopefully) make mistakes.

When I say RUN, Colin sets his position marker to 1 and asks Pam for the first Program Instruction. She lets him have a copy on her slate. Colin sees that it is "INPUT A,B,C"and he asks Ian for his first piece of data (7). Ian, who will have already placed his data cards in order in his input track, pushes them to the right and takes the first card to appear in the viewing window. He gives this to Colin who passes it on to Dawn to place in store A. When Dawn says 'Done', Colin asks Ian for the next item of data (8). He passes this to Dawn for placing in store B. This sequence is repeated until data stores A, B and C have been filled and Dawn reports 'Done' for the last time. Colin, saying "Instruction Completed" then sets his position marker to 2 and asks Pam for her second Program Instruction.

Miss Jones then discusses Program Instruction 2 and how this involves Andrew with his calculator. Because Andrew's calculator (like all calculators) only adds numbers together in pairs it is clear that Colin will send the first two numbers (those in Data stores A and B) to Andrew to be added; then he will send the contents of C to be added to the previous total. The actual sequence of operations, for program instruction no. 2, is as follows:

Colin (after obtaining the instruction "LET D = A + B + C" from Pam on her slate) asks Dawn for the number in store A; this he passes on to Andrew with a request to enter it into the calculator. When Andrew replies 'Done', Colin asks Dawn for the contents of B and passes these to Andrew with a request to add them to the previous number in the calculator's display. This is then repeated for the contents of C. Finally, Colin asks Andrew for the total (18) which he then writes on a yellow card and passes this on to Dawn to be placed in store D.

Program Instructions 3, 4 and 5 are executed in a similar manner. It is noted that no. 4 involves sending information to the output unit where it is printed in a form readable by humans. Instruction no. 5 tells Colin to stop the execution of the program.

Miss J: Well, what do you think of that?

Colin: It involves so much work for a simple calculation, but it did get the right answer.

Oliver: I didn't have much to do, it was boring.

Dawn: I noticed that Colin asked me several times to give him back data which he had just given me to store. Couldn't he have just kept it, knowing that he would need it soon?

Miss J: No, we must remember that Colin represents part of a machine which doesn't know what the next instruction is until it receives it. A computer cannot think, no matter how much it may *appear* to do so.

Pam: Can we do some more programs, Miss?

Miss J: Of course.

Peter: (Who has been watching all the time): It's all very well, what you've done, but *I* wouldn't find the average of numbers like that anyway. I would add the numbers together as they were entered. Why can't the computer do it my way, mine seems easier?

Miss J: The computer can find the average your way and I will write a program for this.

She writes, on blue cards, the following program:

```
 1   INPUT  B
 2   INPUT C
 3   INPUT A
 4   LET B = B + A
 5   LET C = C + 1
 6   IF C = 3 THEN GO TO 8
 7   GO TO 3
 8   LET E = B/3
 9   PRINT E
10   END.
```

Miss Jones makes the following points about the new program:

(a) Store A is used repeatedly for the input of data numbers (7,8,3) one at a time. Store B accumulates from A, keeping a running total. Store C counts how many items have been entered. Initially, B and C are set at zero, in lines 1 and 2.

(b) This gives the machine a means of deciding when all the data have been entered. The decision is made by a Conditional Jump at line 6, as follows:

If C = 3 when line 6 is reached then Colin's position marker will be set to position 8 and he will then call for the instruction in program store 8. If C is not 3 when line 6 is reached then Colin will read this line but take no action, set his position marker to the next number, which is 7, and call for the instruction in Program Store 7. This instruction causes Colin to set his position marker to 3 and so he then calls for the instruction in Program Store no. 3. Line 7 is an example of an Unconditional Jump (i.e. there is no choice).

(c) Line 4 causes the value of Data Store B to be sent to Andrew, who adds on the contents of store A and then the new total is sent back to Data Store B to replace the previous contents.

(d) In line 5 Andrew creates 1 and adds it to the current value of C. Real computers can create, or generate, numbers, like Andrew's calculator.

The children enter the new program, noting how the instructions of the previous program are overwritten (computer jargon for 'replaced') by the new instructions. When Pam and Dawn are given new instructions, or data, they effect the overwriting of a store's previous contents by removing these contents and handing them to Miss Jones. Any store which does not receive new instructions, or data, retains its previous contents.

The program is RUN using three new items of data and the children seem satisfied by the whole operation, except that Dawn thinks it silly for Colin to ask her for data from B and C almost immediately after he has given her the same data. Miss Jones agrees with Dawn but points out, yet again, that the computer is only a machine. It cannot think and can only act upon a precise set of instructions.

The equipment can be easily assembled from scrap materials as described. One model set is available for loan to teachers who wish to carry out for themselves the ideas of this article. Please contact Mr A. James at Newman College, who will also be pleased to answer questions or give further advice. Interested readers are invited to run the programs given below. Perhaps pupils should try to guess the purpose of each program before running it!

```
PROGRAM 1

1 INPUT A,N,B
2 PRINT A
3 LET A = A + B
4 LET B = B + 1
5 PRINT A
6 LET N = N + 1
7 IF N < 10 THEN GO TO 3
8 END
```

```
PROGRAM 2

1 INPUT A,B
2 INPUT D
3 LET C = A - B
4 IF C < 0 THEN GO TO 8
5 LET A = C
6 D = D + 1
7 GO TO 3
8 LET E = B + C
9 PRINT D
10 PRINT E
11 END
```

```
   IN PROGRAM ONE LET A=1,
N=1 AND B=2.
   IN PROGRAM TWO D IS INITIALLY
SET AT ZERO. I SUGGEST YOU TRY THIS
WITH A=37 AND B=5, AND THEN WITH A=35
AND B=7.
```

EDITORIAL

Momentum has increased dramatically. In this issue we report on the Microelectronics Education Programme, on BBC plans and on the Exeter conference. A new and significant move is the creation of a national organisation of teachers using microcomputers in primary schools. We support this development, and hope that our reports (see p. 40) will help to publicise it. All of this offers a genuine prospect of a powerful professional influence on progress. Commercial pressures will be enormous: we need countervailing educational principles. This applies to software also. Should teachers sell each other programs? Let us make our views clear. Commercial software has a rôle, but we should examine it critically. It is quite legitimate for teachers to write for commercial companies, or accept consultancies. But we see the co-operation of teachers freely exchanging software as an important principle of development. There are crucial yet difficult issues here, including questions of copyright, on which we invite comment from readers. An efficient network of communications, as planned by the MEP, should help to provide solutions.

* * * * * * *

Now that the Department of Industry has plans to enable every secondary school to possess a micro, an extension to primary schools would be welcome. But note the irony, if schools making sacrifices to buy and get involved now miss out on the handouts later! If the Government is serious about support, it should urgently review the provision of in-service training. Part-time courses of many kinds are needed, and BBC initiatives will help too. But substantial full-time courses, backed by generous secondment, offer the only prospect of providing expertise in the schools on an adequate scale. Initial training courses too must soon reflect rapidly changing needs. We shall expand on this theme later.

* * * * * * *

Micro enthusiasts will do their cause a disservice if, as the bandwagon rolls, they fail to listen to more sceptical colleagues. Children do like punching buttons and summoning flashing lights! Many current programs seek to harness (or pander to) this extrinsic motivation. What is the child then learning about learning? Some programs by-pass the class teacher and the education process completely. It is too early for anyone to be dogmatic about the strengths and weaknesses of the micro in the classroom. We shall again look to the MEP for guidelines in due course.

Meanwhile Newman College has planned a programme for monitoring the use of micros in six local primary schools over a year (see p. 39). We shall report on this and other exercises, as MICRO-SCOPE in its second year moves on from raw enthusiasm to more sober efforts at evaluation.

Computers and Primary Science

It may seem to many teachers rather premature to suggest a use for computers in primary science when in many schools no science is taught. Nationwide there is a concerted effort to encourage the incorporation of science into the primary curriculum.

The phrase "computers and primary science" may suggest that primary science has need of sophisticated technology. The exact opposite is, in fact, the truth. Primary science, whether observational or experimental, requires only the most readily available equipment, much of which would normally find its way into the wastebin. Of course, specifically scientific equipment is not excluded from science: it is merely not essential.

The microprocessor in the primary school has found its immediate and most obvious use in the fields of mathematics and language, the emphasis in each case being on computer-aided learning, interaction between pupil and microprocessor. So far as science is concerned the micro should be used to assist, complement and extend practical investigations without in any way trying to supplant them. We will, however, at the end of this article suggest a possible future development in primary science which will mean that the computer can play a role as equipment in first-hand, practical investigations. As well as enriching scientific investigation the micro can help with science by simulating events and situations which are impossible in school.

Because science is essentially the observation and manipulation of materials, direct contact with the physical and biological world, wherever possible, is a necessary pre-requisite for the use of a relevant computer program.

By way of illustration and example six areas of program development are described below, some of which are currently being developed by the author.

DATA HANDLING

The computer is probably most "at home" when employed as a tool for the rapid analysis of data. Data collection and sorting are important components of graph construction in the junior school (e.g. foot length, eye colour, height etc.) Programs to extend this area of education need to be sufficiently flexible to enable almost any kind and quantity of data to be input.

Such programs make it possible for the results of analysis, rather than the method of analysis, to be given the focus of attention. Large numbers of measurements can be stored, sorted and output in tabular or graphical form. Thus, for example, the analysis of sizes of children in virtually a whole school, rather than just a class, becomes a real possibility. Such sorted data would be invaluable in the context of a topic such as Growth. There are obvious applications of such programs in other areas such as environmental studies.

SIMULATIONS

The science of sound is a commonly chosen area for investigation in both infant and junior classes. Observations result in the description and classification of sounds with regard to their pitch, loudness and quality; investigations include trying to change pitch, discover resonance and achieve amplification. The speed of sound through air is measured by the echo method - measuring the time taken for a sound (two bricks banged together) to travel across a playground and echo back to the observer off a high wall. Such measurements are essentially practical exercises but may be extended and made more accurate by computer simulations of travelling sound waves. The program ECHO, by the author, written with graphics for the 380Z, enables the pupil to find the time taken for sound to travel through chosen distances in air, water and metal and so eventually enables the pupil to calculate for himself the speed of sound in these different materials.

INTERACTION

Certain areas of primary science lend themselves naturally to very interactive programs. One example is astronomy, with obvious use of graphics. Another is the topic of magnetism and electricity. Concepts and information derived from experiments with magnets and circuits can be thoroughly tested, verified and assessed by suitable question/answer programs. Circuit diagrams in particular, because of their simple 'rectangular' nature, readily lend themselves to a modular graphics display. Jones (1980) mentions such a program, called "CIRCUITS 8-12" for Apple II.

KEYS

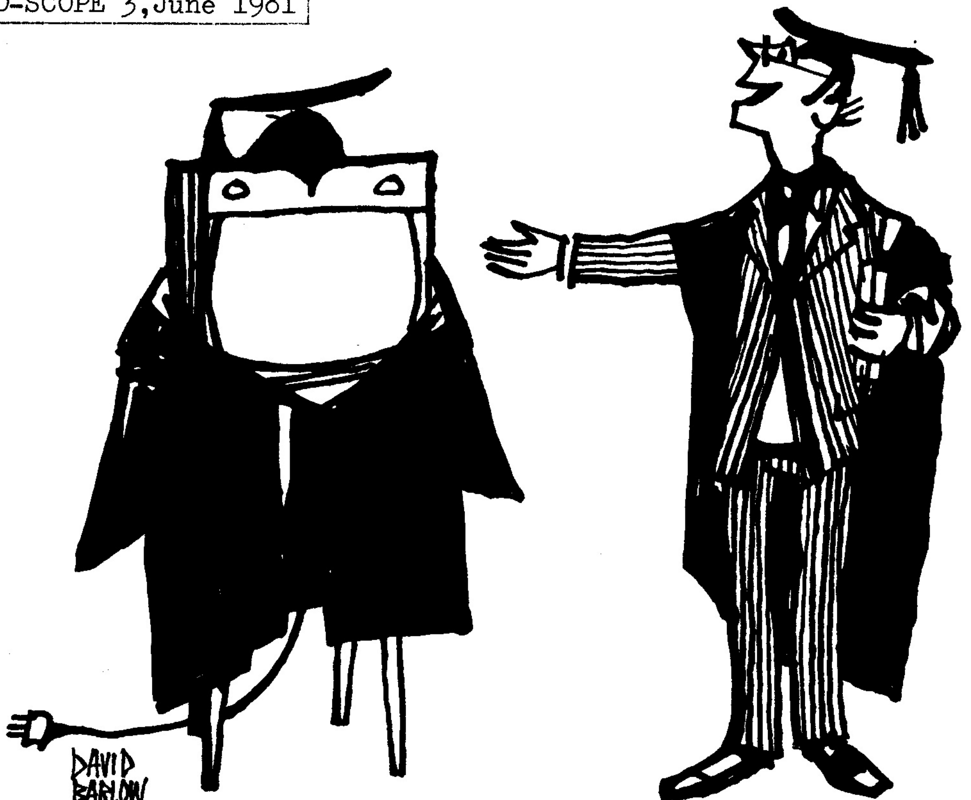
Classification is an important aspect of primary mathematics. It is also of cardinal importance in the interpretation of scientific information as well as in the construction and use of keys for the identification of unknown plants and animals. ITMA (The College of St Mark and St John, Plymouth) has recently produced an ingenious package of programs developed from the well-known ANIMAL key program. Two master programs THINK and SEEK make it possible to use the files SLUGS, TWIG, POWDER and ANIMALS. These offer interactive binary tree systems for the identification of slugs, twigs, powders and animals. The files grow with use; mistakes, which inevitably will be added to the files by children, can be replaced by the teacher. New files can be created by a program called INTREE. Programs of this kind do have a few problems. The teacher must regularly monitor the files for mistakes; also, the micro needs a 48 K RAM.

CONCEPT DEVELOPMENT

The author has a particular interest in the correct development of such concepts as mass, force, weight and pressure in primary science together with an understanding of the standard units involved. The computer can offer considerable help in the development of such concepts by means of programs which combine graphical illustrations with interactive question/answer components. One such program, called MASS (for 380Z) is currently being developed by the author. Another program, to be called CHAIN, will assist the development of concepts associated with ecological food chains.

COMPUTER GAMES

Some computer games can be useful from a scientific point of view. Programs like LUNAR LANDER or ROCKET (Digital Computer Games, 1975) can easily be adapted for 380Z with graphics. These programs together with others that involve missiles, bombs and so on require the pupil to estimate speeds, distances, forces etc. The programs have obvious educational uses: missile programs



"I'd like you to meet our new member of Staff, ... Mr Chips."

teach 'intuitively' the resultant of vectors; ROCKET, the landing of an Apollo capsule on the moon, teaches the relationships between fuel consumption, thrust and the controlled rate of descent against lunar gravity.

* * * * * * * * * * * *

The use of computers in primary schools is in its infancy. The use of computers in the context of primary science has hardly begun. Undoubtedly many mistakes will be made, but mistakes are part of an ongoing learning process. Primary science programs will become increasingly available as computers become the normal possession of primary schools.

To conclude, mention should perhaps be made of a relatively new development in primary science. This is primary technology or applied primary science, the design and construction of equipment to perform a specific task or make a desired measurement. Those interested should read Evans (1977, 1980). At the same time we can expect that microelectronics will gradually be incorporated into the electricity sector of primary and middle school science. The 'black box' aspect of integrated circuits probably makes them more suitable for this age range than microelectronics. It is quite possible to suppose that once primary schools have computers cheap interfaces will become more available (one is already being sold by Oundle School). Then the computer could be used to analyse inputs from peripheral equipment (made in the primary school) or even to act as a simple control system.

Michael Negus,Newman College

References

Digital Equipment Corporation (1975):101 Computer Games. Maynard Massachusetts.
Evans, P. (1977): Technology in the Primary School. ASE booklet
Evans,P. (1980): Science; pure and applied? Education 3-13:8 (1),16-23
Jones,R. (1980): Microcomputers: their uses in Primary Schools.Council for Educational Technology.

MICRO-SCOPE 3,June 1981

MICROCOMPUTERS IN ENGLISH TEACHING

The development of the Microcomputer has reached the stage where English teachers can no longer regard its use as the sole monopoly of the Science departments. Language teachers must now respond to the implications of this new technology.

For many years teachers have been using a number of technological aids to enrich the pupils' learning process and they have had to adapt their styles of teaching in order to use these aids successfully. For example, television is now established in most schools; teachers regard it as a powerful means of conveying information to their pupils and particularly for its capability to put life into a subject which is remote from the personal experiences of the pupils.

That whole schools have adapted themselves in this way is striking evidence of the commitment of teachers to a technology which they believe is of great value to their pupils' learning. The really important influence underlying this commitment is that the broadcasters have developed educational programmes (i.e. the software) of a high professional standard and supported by follow-up material of excellent quality.

The skills which the teacher of English tries to develop can be broadly divided into four categories:

Reading   Writing   Talking   Listening

Underlying all of these is the need to learn conventions and rules of grammar, vocabulary and spelling; and to broaden experience in comprehension, expression and communication.

The microcomputer can complement existing teaching aids in a number of ways provided that the software clearly meets the teacher's requirements. One of the major criticisms of the text-book is that it allows the teacher little freedom in the structuring of lessons. The computer's big advantage over other teaching aids is that it can permit the teacher to control the software. In the writing of any program, therefore, one of the most important criteria to bear in mind is that the teacher using it will require as much flexibility as possible. This point can be illustrated by reference to the methods and practice employed in one approach to the teaching of spelling.

A requirement of any spelling program is visual repetition of the word and its components. The teacher wants the class to see a word as frequently as possible, both in its entirety and broken down into syllables, because this greatly assists the pupil to understand and mentally retain the construction of the word.

A simple spelling program could be structured as follows:

A word is displayed on the screen, e.g.

```
TRANSPARENT
```

Using simple key commands, the teacher then shows the class how the word breaks down into syllables:

```
TRANS - PAR - ENT
```

The word is then illustrated in the context of a sentence:

```
The glass was transparent.
I could see through it.
```

The teacher will now wish to supply the class with further uses of the word and perhaps to input a sentence or alternative words, similar in meaning, suggested by the class. This repeated emphasis on the visual use of words, and particularly on those suggested by the children themselves, is important because it motivates pupils to comprehend the meaning and hence encourages them to use new words.

Having gone through this series of displays for the word TRANSPARENT the teacher will proceed to the next word in the program and repeat the exercise. At the end of the program the teacher should have a number of options available such as:

- Display the words as a list.
- Display the words complete and broken down into syllables.
- Display the syllables only.
- Display a selection of sentences including some of those which the teacher or pupils have added during the running of the program.
- Have the class copy the words into their own exercise books.
- Simply memorise the words direct from the display in preparation for a test.
- After a test (not necessarily held on the same day) recall the words for display so that pupils can mark and correct their own work.

The more flexible microcomputers are in terms of teacher-machine interaction, the more the teacher will use them and gain the substantial benefits which they can provide. Ideally, the teacher would like to be able to stop the program at any point for questions or discussion, recall at will and bypass sections of program. In the type of spelling program illustrated here, the teacher will not want a string of sentences coming up on the screen if he is only dealing with syllables. Most important is the capability of breaking into a program to insert information or descriptions which arise during the lesson and which need to be repeated - as in spelling tests - on later occasions.

Teachers do not have time to write programs any more than they have the time to write their own text-books. What they need, therefore, is program modules which they can readily assemble into lesson texts prior to the lesson, and a simple way of interacting with the program during the teaching session. Even this will require the acquisition of new skills and this raises the crucial question of - who will teach the teachers? For some years to come it may be the teachers themselves; but they will need the backing of their educational authorities to build up the necessary pool of experience and software and to provide in-service courses for teachers of all subjects.

David Hawthorne,Butler's Court Middle School,Beaconsfield.

The above extracts are taken from an article in "COMPUTER AGE", August 1980.

MAP READING -

Description and listing of a geography program

GRIDREF is a program that has been used with top junior and lower secondary pupils. It aims to give practice in a basic map reading skill, the use of four and six figure grid references.

On typing RUN the program will give a choice of four alternatives. The first two give practice in four figure grid references, the third in six figure references. Once an alternative is selected, a grid is displayed on the screen, numbered along the bottom and left margins as with a map.

For alternative one, the simplest, a complete grid square is shaded and the pupil asked to give the four figure grid reference, while the second alternative requires the answer when only a point is indicated within the grid square. This point is flashed on and off the screen to draw attention to the location. Alternative three is similar, but calls for a response in the form of a six figure grid reference.

In each case ten locations are given before the alternative is completed. The pupil responds by typing in his attempt at the answer. This response is checked for the right number of digits and tested to see if it is correct. The "easting" is tested first and if incorrect a message is given for the pupil to "try again". If the easting element of the answer is right, the northing is then tested in the same way. A correct answer in both elements brings the response, "you are correct", and the next location is displayed on the screen.

This part of the program has provided a useful revision exercise for pupils in the practice of grid references. It can be undertaken by a small group, without the teacher present. Certainly after a pupil has achieved the correct answer to the requisite ten points or squares, he has grasped the order of "easting first, followed by northing". It uses these terms, which will require prior introduction by the teacher, because this reinforces the relationship of the grid on the screen to the national grid in the United Kingdom and is part of the reality of the exercise.

One of the advantages of this program which classroom use has shown is the accuracy that it requires of the pupils, especially in the alternative using six figure points. In conventional pen and paper exercises responses are often inexact, but the computer demands accuracy in the answers and this seems to motivate pupils. Both third and fourth **year** juniors find it well within their capabilities and respond to such a level of accuracy.

In alternative four the user is asked to search for the hidden point (or treasure!) on the "map". This seems very difficult to adults, yet pupils need very little encouragement to make a start! After an initial six figure grid reference is entered a clue is given in terms of the eight compass points to show the direction in which you have to move to find the hidden location. The aim of this alternative was to give further practice in the use of six figure grid references, while making pupils think of the reference required to move in a particular direction. It reinforces the points of the compass and encourages children to set about a logical search strategy.

```
10 REM GRIDREF ***** BASICS5 VERSION *****  MAY 1981
20 CLEAR200
30 ?CHR$(12):GRAPH 1:GRAPH 0
40 ?"NEWMAN COLLEGE GEOGRAPHY DEPT. "
50 ?"BARTLEY GREEN, BIRMINGHAM B32":?:?:?
60 ?"GRID REFERENCES PROGRAM":?:?:?:?
70 RANDOMIZE
80 GOTO 100
90 TEXT:?CHR$(12);
100 ?"CHOOSE YOUR ALTERNATIVE PROGRAM":?
110 PRINT "1 FOUR FIGURE SQUARES"
120 PRINT "2 FOUR FIGURE POINTS"
130 PRINT "3 SIX FIGURE POINTS"
140 PRINT "4 SIX FIGURE 'FIND THE`SQUARE'"
150 ?"5 END PROGRAM"
160 ON BREAK GOTO 90
170 ?:INPUT LINE "WHICH ALTERNATIVE";K$
180 ?CHR$(12)
190 IF LEN(K$)<>1 THEN 220
200 C=VAL(K$)
210 IF C>0 AND C<6 THEN 230
220 ?"NUMBER 1-5, PLEASE !":?:GOTO 100
230 IF C=5 THEN ?"END OF PROGRAM": END
240 IF C<=2  THEN C$="FOUR":G=4 ELSE C$="SIX":G=6
250 REM***** DRAW GRID*****
260 GRAPH1
270 FOR Y=10 TO 50 STEP 10
280 PLOT 10,Y,2: LINE 70,Y
290 NEXT Y
300 FOR X=10 TO 70 STEP10
310 PLOT X,10:LINE X,50
320 NEXT X
330 REM***** NUMBER GRID*****
340 Z=91
350 FOR X=6 TO 66 STEP10
360 PLOT X,8,STR$(Z)
370 Z=Z+1
380 NEXT X
390 Z=61
400 FOR Y=10 TO 50STEP 10
410 PLOT 4,Y,STR$(Z)
420 Z=Z+1
430 NEXT Y
440 ON C GOSUB 470,620,730,810
450 GOTO 90
460 REM***** ALTERNATIVE ONE*****
470 FOR S=1 TO 10
480 A=(INT(RND(1)*6)+1)*10
490 B=(INT(RND(1)*4)+1)*10
500 REM***** SHADE SQUARES*****
510 FOR Y=B+1 TO B+9
520 PLOT A+1,Y,2:LINE A+9,Y
530 NEXT Y
540 GOSUB1350
550 REM***** CLEAR SHADING*****
560 FOR Y=B+1 TO B+9
570 PLOT A+1,Y,0:LINE A+9,Y
580 NEXT Y
590 NEXT S
600 RETURN
610 REM***** ALTERNATIVE TWO*****
620 FOR S=1 TO 10
630 GOSUB 1600
640 GOSUB 1630
650 J=A:K=B
660 A=INT(A/10)*10
670 B=INT(B/10)*10
680 GOSUB1350
690 PLOT J,K,2-D
700 NEXT S
710 RETURN
720 REM***** ALTERNATIVE THREE*****
730 FOR S=1 TO 10
740 GOSUB 1600
750 GOSUB 1630
760 GOSUB1350
770 PLOT A,B,2-D
780 NEXT S
790 RETURN
800 REM *****ALTERNATIVE FOUR ****
810 GOSUB 1600
820 PRINT "GUESS THE SIX FIGURE GRID REFERENCE OF  THE HIDDEN POINT"
830 FOR S=1 TO 10
840 INPUT G$
850 IF LEN(G$)<>6 THEN ?"SIX FIGURE REFERENCE, PLEASE":GOTO 840
860 A$=MID$(G$,1,3)
870 B$=MID$(G$,4,3)
880 AH=VAL(A$)-900
890 BV=VAL(B$)-600
900 IF AH<10 OR BV<10 OR AH>70 OR BV>50 THEN ?"ON THE GRID PLEASE !":GOTO 840
910 PLOT AH,BV,2
920 IF AH/10=INT(AH/10) THEN PLOT AH,BV,0
930 IF BV/10=INT(BV/10) THEN PLOT AH,BV,0
940 UU=0
950 IF AH=A THEN 1150
960 IF BV=B THEN 1180
970 IF AH>A THEN UU=1
980 TA=(AH-A)/(BV-B)
990 IF ABS(TA)=1 THEN 1210
1000 AL=ATN(TA)/3.1415927*180
1010 IF AL<0 THEN AL=AL+180
1020 IF UU=0 THEN AL=AL+180
1030 FOR XI=0.5 TO 8.5
1040 IF AL-XI*45<0 THEN XX=XI:XI=8.50
1050 NEXT XI
1060 ON XX+0.5 GOTO 1110,1120,1130,1140,1070,1080,1090,1100,1110
1070 ?"MOVE NORTH":GOTO 1250
1080 ?"MOVE NORTH EAST":GOTO 1250
1090 ?"MOVE EAST":GOTO 1250
1100 ?"MOVE SOUTH EAST":GOTO 1250
1110 ?"MOVE SOUTH":GOTO 1250
1120 ?"MOVE SOUTH WEST":GOTO 1250
1130 ?"MOVE WEST":GOTO 1250
1140 ?"MOVE NORTH WEST":GOTO 1250
1150 IF BV>B THEN ?"YOUR EASTING IS CORRECT, ";:GOTO 1110
1160 IF BV<B THEN ?"YOUR EASTING IS CORRECT, ";:GOTO 1070
1170 GOTO 1310
1180 IF AH<A THEN ?"YOUR NORTHING IS CORRECT, ";:GOTO 1090
1190 ?"YOUR NORTHING IS CORRECT, ";
1200 GOTO 1130
1210 IF TA=1 AND UU=1 THEN 1120
1220 IF TA=1 AND UU=0 THEN 1080
1230 IF TA=-1 AND UU=1 THEN 1140
1240 GOTO 1100
1250 ?"TRY AGAIN"
1260 NEXT S
1270 PRINT "DO YOU WANT TO SEE THE CORRECT LOCATION (YES) OR TRY AGAIN (NO)";
1280 INPUT X$
1290 IF X$="YES" THEN GOSUB 1630:GOTO 1330
1300 ?CHR$(12);"SIX FIGURE REFERENCE";:GOTO 830
1310 PRINT "YOU ARE CORRECT - WELL DONE"
1320 FOR L=1 TO 3000:NEXT L
1330 RETURN
1340 REM***** TESTING GRID REFERENCE INPUTI*****
1350 PRINT "GIVE THE "C$" FIGURE GRID REFERENCE"
1360 INPUT G$
1370 IF LEN(G$)<>G THEN ? C$;" FIGURE REFERENCE, PLEASE":GOTO 1360
1380 IF C>2 THEN 1440
1390 A$=MID$(G$,1,2)
1400 B$=MID$(G$,3,2)
1410 AH=(VAL(A$)-90)*10
1420 BV=(VAL(B$)-60)*10
1430 GOTO 1480
1440 A$=MID$(G$,1,3)
1450 B$=MID$(G$,4,3)
1460 AH=VAL(A$)-900
1470 BV=VAL(B$)-600
1480 IF AH=A THEN 1510
1490 PRINT "YOUR EASTING IS INCORRECT - TRY AGAIN"
1500 GOTO 1350
1510 IF BV=B THEN 1540
1520 PRINT "YOUR NORTHING IS INCORRECT - TRY AGAIN"
1530 GOTO 1350
1540 PRINT "YOU ARE CORRECT"
1550 REM DELAY
1560 FOR H=1 TO 200: NEXT H
1570 ?:?
1580 RETURN
1590 REM *****RANDOM NUMBERS FOR GRID REFERENCES*****
1600 A=INT(RND(1)*60)+10
1610 B=INT(RND(1)*40)+10
1620 RETURN
1630 REM***** TO SHOW SITE (FLASHING)*****
1640 D=2
1650 FOR I=1 TO 10
1660 PLOTA,B,2
1670 FOR J=1 TO 200:NEXT J
1680 PLOT A,B,0
1690 FOR J=1 TO 200:NEXT J
1700 NEXT I
1710 IF A/10=INT(A/10) THEN D=0
1720 IF B/10=INT(B/10) THEN D=0
1730 PLOT A,B,D
1740 RETURN
```

The program gives ten chances of finding the correct location. Most pupils find the hidden point within this limit (although not all teachers do!). If they have failed they are given an opportunity to see the right answer, or they can continue to search for it.

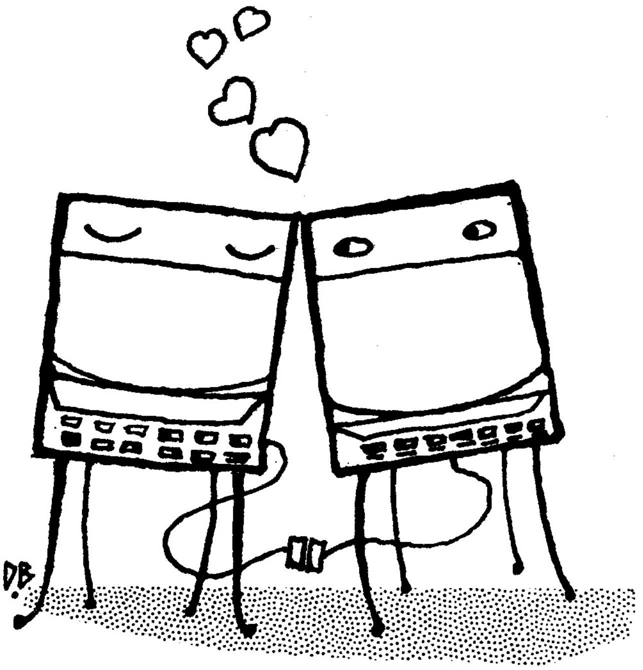
Point locations are generated randomly - unless the teacher wishes to know the sequence of hidden points and show his/her powers of second sight, by removing the"RANDOMIZE"statement on line 70. The program can be interrupted by Control Z - the "menu" of alternatives will be redisplayed. Alternative 5 ends the program and clears the screen.

The program is written to run under BASICS on a RML 380 Z and makes use of low resolution graphics. This makes it difficult to adapt, although a Coventry teacher has adapted the program to run on a Tandy TRS80. We are most interested to hear from anyone who has a version of GRIDREF for a different microcomputer.

Though written mainly with junior pupils in mind, this program has been readily accepted and used in geography lessons in some local secondary schools. One teacher has commented:-

> "Slow learners benefited from repetition of the task in what was regarded as a one-to-one challenge situation. Faster learners enjoyed the challenge of a fast (and accurate!) response. I don't think it has replaced Space Invaders in their (first formers')affections, but its benefits are undoubted - and the long term recall is sound, it is not merely flash -in-the-pan attainment ".

Andrea Tapsfield ,Newman College



'I can't wait to hear the patter of tinny feet.'

CALCULATORS IN THE PRIMARY SCHOOL

Many teachers are reluctant to have calculators in their classrooms, fearing that, through their use, the children will lose all powers of mental calculation. We doubt if this is the case in general: and certainly teachers should be able to guard against its happening whilst utilising to the full the benefits which this branch of microelectronics has to offer.

Even a simple calculator with the 'Four Rules' and a memory has considerable potential as a classroom aid. This article will open up some possibilities going far beyond mere computation. In a later issue, these ideas will be extended through the greater power of a <u>programmable</u> machine. We shall assume, in all cases, that the instruments chosen handle arithmetic operations as they are normally written, from left to right ("algebraic logic").

The calculator is useful in two broad areas: firstly, as a direct help in 'basic' number work; and secondly, when providing a useful tool for those investigations which involve difficult computation, beyond the numerical capability of many children otherwise quite able to understand the mathematical principles involved.

An aid for 'basic' number work

In the early stages of number work a calculator can be used, along with more conventional aids such as Cuisenaire rods and counters, to illustrate the underlying concepts of arithmetic. Instead of using buttons to produce an answer swiftly and magically we can deliberately slow down the process and produce intermediate steps for greater understanding. Areas of application could include:

(a) <u>Discovering by trial the 'story' of a given number</u>

e.g. $8 = 0 + 8 = 1 + 7 = 2 + 6 = \ldots\ldots$

A measure of the child's memory could be the time taken to reconstruct a given 'story' at intervals of a few days.

(b) <u>Verifying the associative and commutative laws for addition and multiplication</u>

(and demonstrating that they do not hold for the other two rules). Later on the distributive law of multiplication (or division) over addition (or subtraction) will prove easy to illustrate :-

$$\begin{aligned} 27 \times 16 &= 27 \times (9+7) = \underline{27 \times 9} + \underline{27 \times 7}, \\ &= 27 \times (10+6) = \underline{27 \times 10} + \underline{27 \times 6}, \ldots\ldots \end{aligned}$$

The first product in the RHS of each example would be saved in the memory and then added to the other product. The LHS would be computed in the usual way to demonstrate the equality.
Note: To store in a memory requires pressing a single button of the type $\boxed{\overset{\uparrow}{M}}$ for most single memory machines. For calculators with up to ten memory locations, pressing two buttons (such as $\boxed{STO}$ then $\boxed{3}$) will cause the

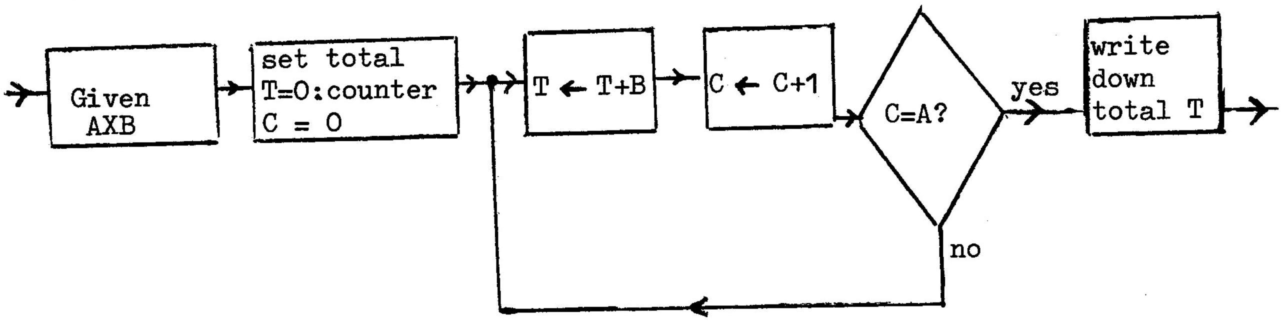
number visible in the display to be stored in memory number 3.

(c) An understanding of partial sums in long multiplication.

In finding the partial sums, children would be reminded that, when multiplying by 17, the multipliers are really 10 and 7. This is readily extended to three digit multipliers so that 216 x 123 is emphasised as (216 x 100) + (216 x 20) + (216 x 3). Repeatedly pressing the zero button acts as a considerable reinforcement.

(d) Multiplication as repeated addition, which is demonstrated with the aid of a counter to record the number of additions.

e.g. $11 \times 7 = \underbrace{11 + 11 + \dots + 11}_{\leftarrow 7\ \text{times} \rightarrow}$. The counter is necessary to determine when no more elevens are to be added. The general process is shown in the following flowchart

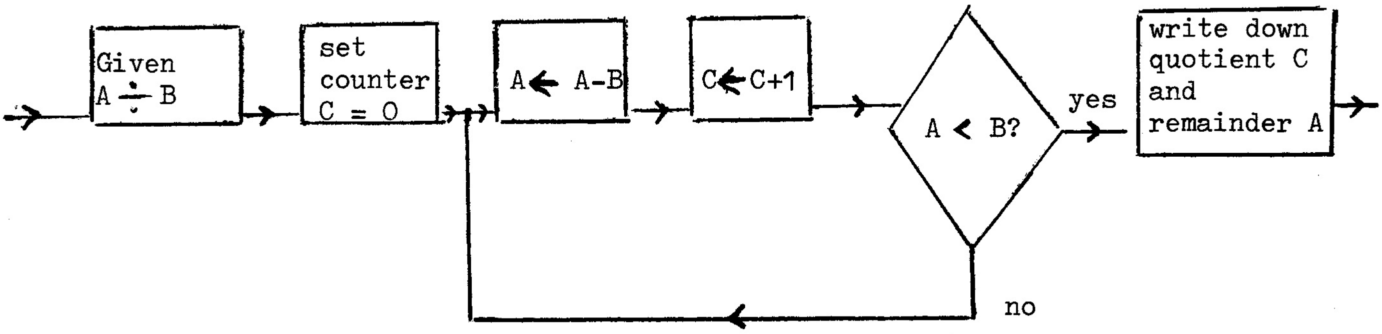


With a two-memory machine there is no need to write down any intermediate results but children may find it easier to use a pencil and paper along with the calculator. A neat alternative is to combine the counter and the accumulating total as a single artificial number in display, with zeros to separate the parts. For instance, to multiply 4 x 732 we could enter the number $\boxed{1}\boxed{0}\boxed{0}\boxed{7}\boxed{3}\boxed{2}$ into the memory and repeatedly add to it to the display, yielding 201464, 302196, 402928 successively. The last number shows that 4 x 732 = 2928.

(e) Division as repeated subtraction.

This is a similar process to multiplication except that (i) the dividend is successively reduced by subtracting the divisor until it is less than the divisor and (ii) the counter, recording the number of subtractions, becomes the required quotient.

e.g. 29 ÷ 7 ⟶ 29-7 = 22
22-7 = 15
15-7 = 8
8-7 = 1 ⟶ stop subtracting. The quotient is 4 and the remainder is 1. The general process is shown in the flowchart below.



The tracing of such flowcharts can be fun, will certainly deepen the child's understanding of the arithmetic process and will prove a valuable asset later on for the planning of computer programs.

(f) Decimal fractions

An understanding of these is enhanced by the calculator's reliance upon decimal notation:-

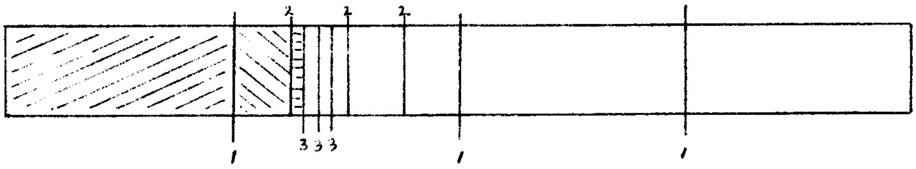
e.g. $\frac{3}{10} + \frac{4}{100} = 0.34$ (key this in as $3 \div 10 + 4 \div 100 = \ldots\ldots$).

Calculators as a tool in numerical investigations

The calculator is a great boon where the main purpose is not to practise numerical competence but to use the results of computations for interpretation. Examples of its use include:

(g) Fractions

Divide a rectangular strip of paper into thirds as accurately as possible (or perhaps a cake fairly amongst three children!) It is easy to divide the rectangle into quarters by folding. Three of these quarters can be allocated, one to each 'third', whilst the fourth quarter is then used for further division into quarters (now sixteenths of the original strip). The process is now repeated until the strip of paper left is too small for further division. At this point most children will agree that the strip is almost in thirds and that continuing the process indefinitely would give an exact division into thirds. An alternative way of carrying out the folding is illustrated below, where a single piece is obtained for the fractional part.



The sum of the shaded regions rapidly approaches a third of the strip. These examples give rise to the representation of 1/3 as a never-ending series thus:-

$$1/3 = 1/4 + (1/4 \times 1/4) + (1/4 \times 1/4 \times 1/4) + \ldots.$$

Summing this series to a given number of terms is easy with a calculator! The first few partial sum are given in the table below to show how rapidly they approach 1/3.

No. of terms	Value of last term	Partial sum
1	0.25	0.25
2	0.0625	0.3125
3	0.015625	0.3281
4	0.003906	0.3320
5	0.000977	0.3330

Children will enjoy proving, for themselves, that

$$1/7 = 1/8 + (1/8 \times 1/8) + (1/8 \times 1/8 \times 1/8) + \ldots$$

and

$$1/9 = 1/10 + (1/10 \times 1/10) + (1/10 \times 1/10 \times 1/10) + \ldots$$

Many children will be capable of generalising the work to obtain

$$1/(n + 1) = 1/n + (1/n \times 1/n) + (1/n \times 1/n \times 1/n) + \ldots$$

(Though probably not in an algebraic form!!)

(h) Finding square roots (without, of course, using the $\sqrt{\ }$ button!!)

Two basic strategies can be practised,both of which illustrate very useful mathematical principles. Each method makes use of basic definitions for squares and square roots. However, one is clearly superior and this provides children with a useful discussion point.

First Method If A is the square root of N then A multiplied by itself will equal N,i.e.

$$\text{If } A = \sqrt{N} \text{ then } A^2 = N.$$

The relationship is used to find $\sqrt{19}$ in the following example. It is clear that $\sqrt{19} \times \sqrt{19} = 19$; therefore the child is looking for a number which gives 19 when it is multiplied by itself. The child starts with a first guess, 5 say. If 5 is the sought-after root then 5 x 5 should give 19, but 5 x 5 = 25 which is greater than 19 and it is concluded that 5 is too big. Now a second guess is tried which is less than 5, say 4. The test leads to 4 x 4 = 16 which is less than 19, and it is concluded that 4 is too small. These two results imply that $\sqrt{19}$ lies between 4 and 5. Most children will choose, as their next guess, the average of 4 and 5 i.e. 4.5. This is tested as before and a fourth guess is made by averaging the two guesses which give squares just above and below 19. This procedure is repeated until sufficient accuracy has been obtained. The first few steps are shown in the table below:

Guess number	Guess	Square	decision
1	5	25	Too large
2	4	16	Too small
3	4.5	20.25	Too large
4	4.25	18.06	Too small
5	4.375	19.14	Too large
6	4.3125	18.60	Too small
7	4.3438	18.87	Too small
8	4.3594	19.004	Too large (but nearly there)

This method is very slow and only after eight guesses are the squares near 19.

Second Method. Here we use the principle that $\frac{N}{A} = A$ if A is the square root of N. This leads to the following procedure which will be illustrated for finding $\sqrt{19}$. Let the first guess be 5, as before. Now $\frac{19}{5} = 3.8$ and since $3.8 \neq 5$ the square root has not been found. It is not difficult to accept the reasoning that $\sqrt{19}$ lies between 5 and 3.8, so a good choice for a second guess would be $\frac{5+3.8}{2}$ i.e. 4.4. Now $\frac{19}{4.4} = 4.318$ and, by similar reasoning to previously, a third guess is taken to be $\frac{4.4+4.318}{2}$ i.e. 4.359. This process is repeated until the divisor and quotient are equal when working to a specified accuracy. The first steps are traced in the table below:

Guess number	Guess	Quotient	Decision
1	5	3.8	Not equal - take average
2	4.4	4.318	Not equal - take average
3	4.359	4.359	Correct answer to 3 dec.pl.

It will be readily noted that the first method requires two arbitrary guesses whereas only one arbitrary guess is required for the faster second method.

(i) Calculation of maximum and minimum values.

As an example we shall find the maximum area of a rectangle which must have a perimeter of 27 metres. Firstly it is established that length + breadth = 13.5m. Now it is possible to tabulate the area obtained for chosen breadths (or lengths)

breadth (m)	length (m)	area (m^2)
1	12.5	12.5
2	11.5	23.0
3	10.5	31.5
4	9.5	38.0
5	8.5	42.5
6	7.5	45.0
7	6.5	45.5
8	5.5	44.0
9	4.5	40.5

The last two calculations show that the maximum value for the area has been passed. After discussion the children will probably suggest looking for higher area values for breadths between 6m and 7m, thus obtaining

breadth (m)	length (m)	area (m^2)
6.5	7	45.5
6.75	6.75	45.5625
6.875	6.625	45.5469

Other values close to 6.75m may be tried but the children should soon reach the conclusion that a length and breadth of 6.75m gives the maximum area for the rectangle. The rectangle is now a square and children may wonder if this will always be the case.

It is hoped that readers will be stimulated to try the examples given and to devise others. Please let us know your reactions to the opportunities opened up by the use of calculators in primary schools. In particular we would like to hear of other applications, including games.

A. James
Newman College

USERS' GROUPS AND PROJECTS

These are preliminary notes towards a fuller directory of useful contacts.

Dave Futcher
Beaconsfield First and Middle School, Southall,Middx. (01 574 3506) has a TRS-80 Educational Users' Group and is compiling directories of users and of software. He is also involved in conversion of programs from other machines - he has adopted PHRASES from MICRO-SCOPE 2! Send a cassette for a copy.

Derrick Daines
(Carsic Primary School, Sutton in Ashfield, Notts) has also adapted PHRASES for his SWTP 6800.

Brian Richardson
(18 The Lawns, Benfleet, Essex) sends a varied list of PET Software and comments that he has a program for arithmetic practice very similar to our TRAINS.

P.J. Wayth
has written a thoughtful report on the Computer Project (PET) at St Andrew's Junior School, Willesden, concluding: "The key... is software.... Perhaps there is some way that teachers and experienced programmers can meet to produce programs to the teachers' specifications. Teachers are the only people who can describe their exact requirements."

Harry Dolphin
Writes from 23 Bruche Drive, Padgate, Warrington, WA1 3JX, about introducing pupils to programming. He uses 'families' of programs based on LET, FOR, INPUT and READ, DATA.

Eric Deeson
Highgate School, Birmingham B12 9DS, will send the Educational ZX80/1 Users' Group Newsletter on receipt of a s.a.e. He believes the Sinclair range is often undervalued just because it is cheap and insists it is not a toy.

Don Walton
is director of the Houghton School C.A.L.Project, at Huntingdon, Cambs. 0480 63398.

Barry Holmes
St Helen's Co. Primary School, Bluntisham, Huntingdon has programs "The Spanish Main" and "Hunt the Thimble" for 16K Apple or PET.

Newman College Primary Software documentation booklets are still available from R. Keeling at Newman College for 50p + 25p p&p.

BEHAVIOURAL OBJECTIVES

- A Report from Coventry

The article on the M.A.C.E. project and the extract from M.E. Fowler's letter (p.2.5) stimulated me to write this article with a view to informing your readers of a few aspects of educational computing in the primary field in Coventry.
I gave a short demonstration and talk at the local Teacher's Centre and hope to make a video-tape illustrating the applications of the computer in our school. I will be showing this to parents and interested colleagues in the hope that it will encourage the Local Authority to provide our school computing efforts with some financial support.
I teach in the Junior department of an all-age E.S.N.(M) school, so my aims may differ slightly from those of your other readers, but still I find your magazine the most useful for my needs.

Yours sincerely,

Mrs Myra Duffy

I teach in a primary department of an E.S.N. (M) school in Coventry and I only recently became acquainted with computers in education. My interest was initially aroused by a need to reduce the amount of paperwork associated with teaching and recording behavioural objectives. My colleagues and I have been busy for some time writing behavioural objectives. Each basic subject is task-analysed and written as sequential target objectives. It is intended that each child should have a copy as part of his individual programme of work. As you can imagine, the amount of paper-work and recording is enormous.

It seemed that a computer might be the answer to this side of our problem and I set out to discover if this was so. One of the experts I approached was Dr John Karron at the Lanchester Polytechnic, Coventry. He assured me that a computer certainly could help and personally assisted by arranging the loan of an 8K Commodore computer.

The Polytechnic has supplied, on loan, three 8K Commodore computers to primary schools - two in Coventry, one in Rugby. The other two teachers and myself who are responsible for the computers attend a 'workshop' every week to share ideas, programs and learn programming techniques.

My own interest lies in the investigation of using computers to teach maths to slow-learners. I anticipate that I will also have to explore the application of peripheral devices as an interface to the computer - especially for the younger slow-learner. To this end I have begun to write a series of programs. This is a typical example:

Program 1 Recognise and match sets with a cardinality of two, four, three, one and five.

Demonstration:

1 block appears. 1 star appears to match it.
2 blocks appear. 2 stars appear to match them.
3 blocks appear. 3 stars appear to match them.

Practice: A set of blocks (1-5) appears in random order. The child has to press a '*' for each block and then press RETURN.

If correct: ✓ 'Well done'
If incorrect: 'Sorry, try again'

When 10 consecutively correct answers are given, test appears. (a different criterion can be substituted)

Test: A set of blocks (1-5) in random formation and random order. The child has to press a '*' for each block and then press RETURN.

If correct: When all 5 attempts are correct, 'Well done' 'Goodbye'.
If incorrect: Child is taken back to practice section.

Next I hope to write a series of diagnostic tests for each behavioural objective. The child's name, the date and his results will be stored on a 'results' file. If the tests are written with sufficiently fine objectives, the area of a child's lack of understanding will be specifically identified. Thus future teaching strategies can be inferred from these results - and the amount of paper-work cut to a minimum. Already my experience with a computer in the classroom and the evaluation of my programs has shown me where I can considerably improve them and has altered many of my original ideas.

The Adviser in Coventry who has responsibility for Computers in Education has called together a group of teachers interested in computers. One of these is the City's representative to M.A.C.E. (Micro-electronics And Computers in Education - West Midlands). This group is to investigate ways in which program ideas can be distributed to members of the group, and others, who will act as program writers. It is intended that the software will then be brought back into the special schools for evaluation. One idea is to concentrate on the writing of programs based on behavioural objectives which will be suitable for all children - but with remedial loops for slow-learners. There are problems, of course. One is that there are, at the moment, three principal makes of computers in schools in Coventry and so the software will not be compatible. The group is also seeking ways to make the software more easily adaptable from one machine to the other.

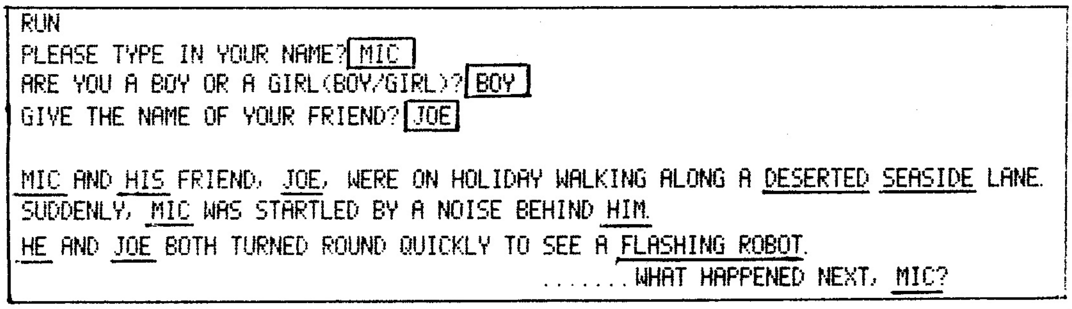
It is still very early days and a great deal of work in writing and evaluating programs has to be done - as well as attempting to make them more consistent. However, there is much interest, enthusiasm and co-operation in the field so I am confident that computers will be used more extensively in Coventry as more and better educational software becomes available.

Myra Duffy,
Deedmore E.S.N.(M) School,Bell Green,Coventry

A PRIMARY B.A.S.I.C. - Part 3

In this article we shall develop two important programming techniques which greatly extend our range in handling passages of text. One is the incorporation of personalised information, fed in by the user, into a prepared text. The other is the random selection of one item from a whole set of stored data. The classroom application chosen shows how both techniques combine to provide a starting point for creative writing.

This PRINTOUT shows the first program in operation:-



```
RUN
PLEASE TYPE IN YOUR NAME?MIC
ARE YOU A BOY OR A GIRL(BOY/GIRL)?BOY
GIVE THE NAME OF YOUR FRIEND?JOE

MIC AND HIS FRIEND, JOE, WERE ON HOLIDAY WALKING ALONG A DESERTED SEASIDE LANE.
SUDDENLY, MIC WAS STARTLED BY A NOISE BEHIND HIM.
HE AND JOE BOTH TURNED ROUND QUICKLY TO SEE A FLASHING ROBOT.
                                          .......WHAT HAPPENED NEXT, MIC?
```

After RUN is typed, the child is asked to supply names and data for the story (shown in boxes above). Then the computer gives the opening lines of a story. The words underlined are provided in these three different ways:-

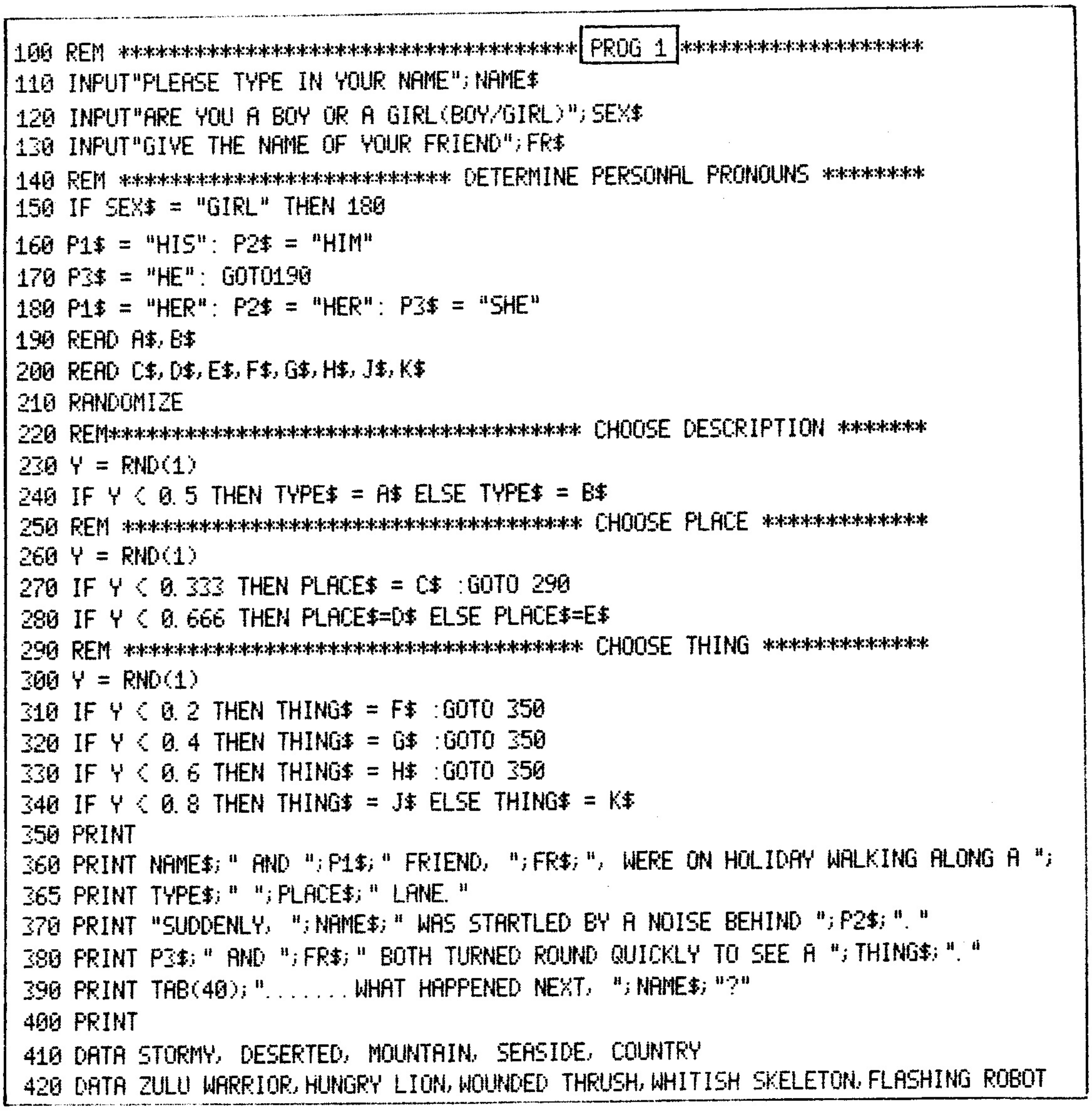
1. Pupil input is returned as appropriate (Mic, Joe)
2. Pronouns and possessives dependent on pupil input (boy) are selected (his, him, he)
3. Basic elements for the story are chosen randomly from a set previously stored by the teacher (deserted, seaside, flashing robot).

Now for the programming details. Look at the listing of Program 1 (obtained by typing LIST). The pupil provides the INPUT information at lines 110-130. Line 110 is a contraction into a single line of the formulation we have used in previous articles, namely

```
110 PRINT "Please type in your name";
111 INPUT NAME$
```

A number of contractions are given in this article. The objective is space saving. Often a computer runs out of storage space which could have been saved by efficient programming.

Note that NAME$ is the so called 'address' of a computer store. In previous issues we have used 'one letter addresses' like M$ and W$. The computer only reads the first two letters of an address (NA$ in this case) but the whole word is used to help the human reader!



```
100 REM ***********************************  PROG 1  ********************
110 INPUT"PLEASE TYPE IN YOUR NAME";NAME$
120 INPUT"ARE YOU A BOY OR A GIRL(BOY/GIRL)";SEX$
130 INPUT"GIVE THE NAME OF YOUR FRIEND";FR$
140 REM ************************* DETERMINE PERSONAL PRONOUNS ********
150 IF SEX$ = "GIRL" THEN 180
160 P1$ = "HIS": P2$ = "HIM"
170 P3$ = "HE": GOTO190
180 P1$ = "HER": P2$ = "HER": P3$ = "SHE"
190 READ A$,B$
200 READ C$,D$,E$,F$,G$,H$,J$,K$
210 RANDOMIZE
220 REM********************************** CHOOSE DESCRIPTION *******
230 Y = RND(1)
240 IF Y < 0.5 THEN TYPE$ = A$ ELSE TYPE$ = B$
250 REM ********************************* CHOOSE PLACE *************
260 Y = RND(1)
270 IF Y < 0.333 THEN PLACE$ = C$ :GOTO 290
280 IF Y < 0.666 THEN PLACE$=D$ ELSE PLACE$=E$
290 REM ********************************* CHOOSE THING *************
300 Y = RND(1)
310 IF Y < 0.2 THEN THING$ = F$ :GOTO 350
320 IF Y < 0.4 THEN THING$ = G$ :GOTO 350
330 IF Y < 0.6 THEN THING$ = H$ :GOTO 350
340 IF Y < 0.8 THEN THING$ = J$ ELSE THING$ = K$
350 PRINT
360 PRINT NAME$;" AND ";P1$;" FRIEND, ";FR$;", WERE ON HOLIDAY WALKING ALONG A ";
365 PRINT TYPE$;" ";PLACE$;" LANE."
370 PRINT "SUDDENLY, ";NAME$;" WAS STARTLED BY A NOISE BEHIND ";P2$;"."
380 PRINT P3$;" AND ";FR$;" BOTH TURNED ROUND QUICKLY TO SEE A ";THING$;"."
390 PRINT TAB(40);".......WHAT HAPPENED NEXT, ";NAME$;"?"
400 PRINT
410 DATA STORMY, DESERTED, MOUNTAIN, SEASIDE, COUNTRY
420 DATA ZULU WARRIOR,HUNGRY LION,WOUNDED THRUSH,WHITISH SKELETON,FLASHING ROBOT
```

The personal pronouns in the output text are determined by the use of the matching idea developed in the previous article. We use the contents of SEXØ provided at line 120. If SEXØ = "Boy" at line 150 then his/him/he are located at addresses P1Ø, P2Ø, P3Ø by lines 160 - 170 (again P1Ø is a two "letter" address), but if SEXØ = "GIRL" then her/her/she are located by line 180 at the same three addresses. (Notice that we have contracted two lines at line 160 by using a semicolon.)

Now the story has been "personalised". In the next step it is given variety by the random selection of some basic elements. The program DATA at lines 410 - 420 are READ at lines 190 - 200.(Line 190 is a contraction). Thus all ten items of data are read in order into appropriate stores.

We now make use of the computer's RANDOM number generator which produces a sequence of arbitrary decimals between 0 (zero) and 1 (one). Each six digit decimal (d) could be zero but not one - more precisely 0 ⩽ d ⩽ 0.999999. The statement RANDOMIZE (line 210) is necessary to start the sequence at a different place each time. The assignment statement (LET) Y = RND(1) (lines 230,260.300) assigns the next decimal in the random sequence to the location Y. Y is a 'dummy' variable used to sort out the stored data. We use this temporary location three times in this program. In line 240, one of the descriptions 'stormy ' or 'deserted' (located respectively at AØ, BØ) is copied to the new location TYPEØ. If, for example, Y = 0.21 (i.e. Y < 0.5) then 'stormy' is selected and copied to the address TYPE$, but if Y = 0.73 (i.e. Y ⩾ 0.5) then 'deserted' would be selected. The two choices are equally likely.

IF - THEN - ELSE - in line 240 is a contraction for the lines

```
240 IF Y < 0.5 THEN TYPE$ = A$
241 IF Y < 0.5 THEN 250
242  TYPE$ = B$
```

There are two conditionals here. The first is new: line 240 is a conditional assignment statement . The second is the conditional jump we have met previously (line 241 above). This conditional jump is implied in the original line 240 where the word 'ELSE' allows for an unfulfilled condition.

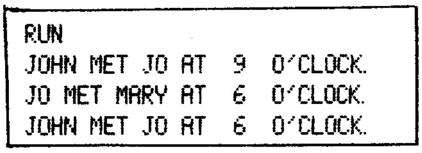
Lines 260 - 280 choose one "place" out of three located at C$, D$ and E$. In one third of the cases (Y < 0.333) the word 'mountain' is chosen and copied to PLACE$; in another third, 'seaside' (0.333 ⩽ Y < 0.666); and in the rest, 'country' (0.666 ⩽ Y). Similarly the "thing" is chosen by lines 300 - 340.

The output text - the introduction to the pupils' story - is printed by lines 360 - 390. The only new statement is the PRINT TAB(40); "..." in line 390 which as can be seen from the printout acts as a (40 space) tabulator.

Apart from the child's input there are 30 variations now possible in the output (2 types, 3 places, 5 things - 2 x 3 x 5 = 30). In this way this program can be used many times to provide different stories. The story introduction provided is, of course, quite arbitrary and limited only by the imagination of the teacher. Anyone who carefully compares the printout with the BASIC listing (lines 360 -390) can create a different one. Here are the steps:-

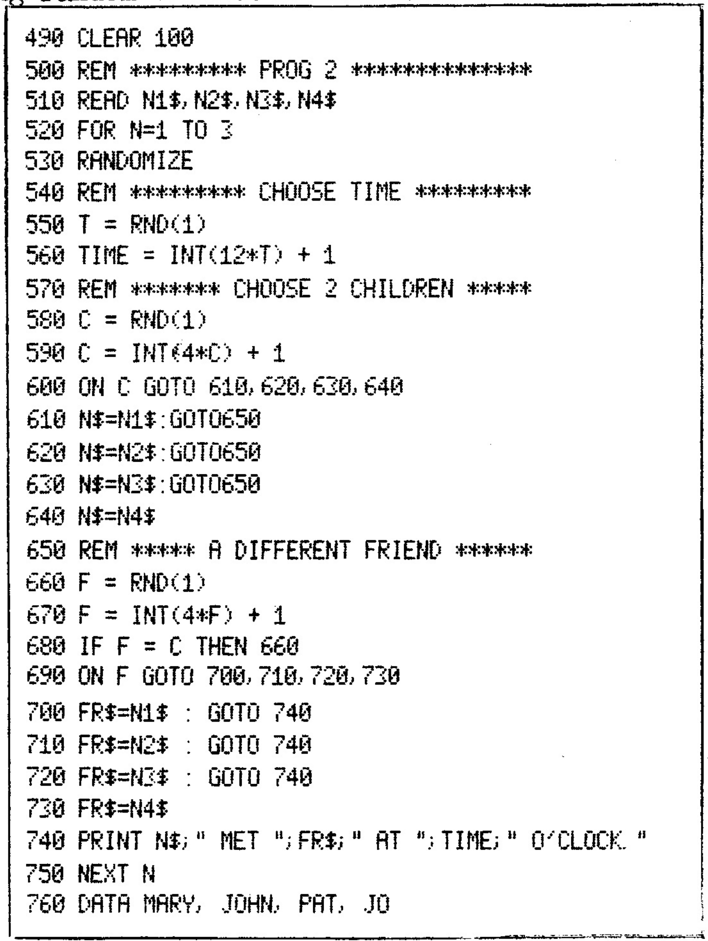
1. Write out your own 'introduction' containing as many names, dependent pronouns and parts of speech as you wish to vary - underlining each.
2. Place the whole text in quotation marks.
3. Replace each chosen variable, in 1 above, by a variable address (e.g. NAME$, N$, NZ$ for words or NA, N, NZ for numbers). This 'address' must be separated from the rest of the text by semicolons and further quotation marks. It may be necessary, if two addresses occur consecutively, to specify a space within "text quotation marks" as in line 360 between TYPE$ and PLACE$. Otherwise "desertedseaside" would be printed out.
4. Carefully examine the spaces and punctuation on either side of a 'variable address' and compare with the printout.
5. Write down as many alternatives as required for each variable and put these on DATA lines (e.g. 410 - 420).
6. For each 'variable address' in the text output, prepare a random selection routine.
7. Check carefully, for each variable, that you have the same number of items of DATA as of READ statements and of random choices. You have been warned!

Program 2 illustrates an appointment between two classmates at an arbitrary time:-



```
RUN
JOHN MET JO AT  9  O'CLOCK.
JO MET MARY AT  6  O'CLOCK.
JOHN MET JO AT  6  O'CLOCK.
```

Multibranching techniques are introduced to demonstrate an alternative method of programming random choices of numbers or of words.



```
490 CLEAR 100
500 REM ********* PROG 2 *************
510 READ N1$,N2$,N3$,N4$
520 FOR N=1 TO 3
530 RANDOMIZE
540 REM ********* CHOOSE TIME *********
550 T = RND(1)
560 TIME = INT(12*T) + 1
570 REM ******* CHOOSE 2 CHILDREN *****
580 C = RND(1)
590 C = INT(4*C) + 1
600 ON C GOTO 610,620,630,640
610 N$=N1$:GOTO650
620 N$=N2$:GOTO650
630 N$=N3$:GOTO650
640 N$=N4$
650 REM ***** A DIFFERENT FRIEND ******
660 F = RND(1)
670 F = INT(4*F) + 1
680 IF F = C THEN 660
690 ON F GOTO 700,710,720,730
700 FR$=N1$ : GOTO 740
710 FR$=N2$ : GOTO 740
720 FR$=N3$ : GOTO 740
730 FR$=N4$
740 PRINT N$;" MET ";FR$;" AT ";TIME;" O'CLOCK. "
750 NEXT N
760 DATA MARY, JOHN, PAT, JO
```

Notice that we have used different dummy variables in lines 550,580,660. This is only important in line 680 which ensures, as we shall see, that two different (!) classmates are chosen. The statement in line 560 TIME = INT(12 * T) + 1 needs some explanation. Now by line 550 T lies between 0 and 1 so 12 * T (12 times T) is a decimal between 0 and 12. The INT ( ) function gives the 'integer value' of 12 * T - that is, the whole number which is less than 12 * T (e.g. INT 2.71 = 2). Thus INT(12 * T) is a whole number in the range 0 to 11 (but not 12 since RND(1) is never 1) and this number is then stored in location TIME.

Similarly line 590 makes C = 1,2,3 or 4. Line 600 is a multibranching conditional jump which is used here as an alternative to the method in program 1. It is a contraction of four lines like "600 IF C=1 THEN 610".

Only one of the lines 610 - 640 is read, only one of the names in line 760 (stored by line 510) is chosen at lines 610 to 640 and the friend is chosen similarly at lines 700 to 730. The friend's name is different because the conditional jump in line 680 rejects and replaces any random choice giving F = C.

Clearly this multibranching technique has many applications. Some of these were explored in older programmed learning techniques. I am sure that we should use the best of these 'old' methods, for example, in consolidations and revision but let us at all costs not be bound by them. It would be a pity if,through lack of imagination in this new field, the majority of programs turned out to be of the 'conditioned response' type.

One side effect of this article may be useful. The next time you receive a mail order circular with your name (and address) apparently specially typed within the text of a letter promising you personally numbered prizes etc etc - perhaps you will pause, consider how it was constructed and file it in the dustbin.

Let us summarise the new ideas covered in this issue:

1. BASIC commands RUN, LIST

2. Several statements may be contracted onto one line - separated by a colon (:). Address names may have two (significant) letters or one letter plus one digit (both for strings (NA$) and for numbers (NA))

3. Input/Output Statements (I/O) - mainly contractions.

   i) INPUT "printed message"; NAME$

   ii) READ A$, B$,C$

   iii) PRINT TAB(5); "Indent this paragraph"

4. Conditionals can use statements as well as line numbers after THEN

   i) IF (statement) THEN (statement or line number)

   ii) IF (statement) THEN (-) ELSE (-)

   ELSE allows for unfulfilled conditions.

   iii) ON (whole number 1,2,3) GO TO(line 1, line 2, line 3 respectively.)

5. Functions

   i) RANDOMIZE - variable start

   A = RND (1) - locates a random number less than 1 at A

   ii) I = INT (X) - locates the integer part of X at I.

J. Fair

NEWS FROM M.A.P.E.
-The New National Association

Readers who attended 'Exeter' will, no doubt, be wondering what has happend to the primary microcomputer association which the conference decided to form. Others may have heard rumours of the new organisation. Well, I am able to reveal that the association is to be officially launched on the 1st January 1982, by which time the final 'bugs' should be under control.

Conference members will remember the concern expressed that the emphasis of the association should be on computers and education rather than on expertise in computing. Although experts will undoubtedly emerge, the steering committee has attempted to reflect this view of the conference both in the name and the aim of the association. The name, Micros And Primary Education (M.A.P.E.), was chosen and it was agreed that the principal aim should be: - "To promote and develop the awareness and effective use of microelectronics as an integral part of the philosophy and practice of Primary Education."

The association will start by issuing to all members an information pack, which has been developed with the aid of M.E.P. (the government-sponsored Microelectronics Education Programme) and is being published by C.E.T.

The pack will include: -

1. Case Studies There are five case studies, by teachers, on how they have used specific programs in their classrooms. They cover upper and lower Juniors and Infants, and various aspects of the curriculum.

2. Micros across the Curriculum. A series of booklets edited by Roy Garland, correlating approaches in various areas of the primary/middle schools curriculum. Starter titles in the series are: Micros in English, Micros in Science, Micros in Mathematics, Micros in Humanities.

3. Classroom Management of the Micro for the Beginner . This is a collection of classroom experiences detailing how teachers actually manage micros within the classroom. Edited by David Ellingham, it is concerned not with program content but with organisation, time allocation, where to locate the micro, etc. This should provide many useful hints for the class teacher faced with the problem of sharing a micro within a class of 30 children.

4. A 'before-you-buy' Guide. This guide will outline the major points to be considered before a microcomputer is purchased in order to ensure that the purchaser is at least aware of the many pitfalls.

5. Micro Resources Pamphlet. This will include a list of useful addresses of organisations, sources of help, magazines, video tapes etc.

Unfortunately, some of the titles may not be ready for our deadline but members will receive copies of late publications hot from the press. The Annual Conference is already well into advanced stages of planning, organised by Roy Garland, with the assistance of committee members from the South West. Again it will be held in April, though please do not start making Roy's life a misery by requesting details yet - these will be published in January. Confidentially, if you wish to keep the 2nd, 3rd and 4th clear, it should be a most interesting weekend.

Another important feature of the association will be its magazine. We are very fortunate to have Roger Keeling, from Newman College, Birmingham, on the committee and through Roger 'MICRO-SCOPE' has agreed to act on behalf of M.A.P.E. This will be sent free to members, and currently appears three times a year. Contributions from M.A.P.E. members will certainly have an important part to play in the future development of the magazine. So let's have your ideas and articles so that we can reflect the views of the members.

We also hope to be involved with the development of software for the primary school. Quite how this will be done is still being discussed but we will be ready for January. In the meantime, any aspiring authors please contact me and I will detail the type of proposal which we are considering. This is almost certain to be linked with a commercial software organisation who,it is envisaged, will produce a catalogue of M.A.P.E. programs. Again we are looking to members to support this project - you will be the ones to benefit.

Members will also be kept in touch with developments through newsletters and their regional representatives, who will have a contact list for their area and will help members to contact others with specific interests. It is hoped that groups will be formed in each area to exchange ideas, information etc. on a local basis.

For those that require it the association is arranging insurance cover for machines and accessories at preferential terms. Details will be available later.

Finally, money. The subscription for membership to the association has been set at £7.50p. per annum, running from January to January each year. This was decided to save the secretary and treasurer suffering nervous breakdowns, we still have to work.

When establishing any new organisation there is always the initial period when unforeseen 'bugs' become apparent. I hope you will bear with us. We are anticipating an exciting first year. Why not join us? Membership forms will be printed soon. Send a S.A.E. so that I can post you yours as soon as they are available. Please bear with me if you do not receive yours by return of post, I shall try and reply as quickly as possible.

Barry Holmes (Secretary)
St Helen's C.P. School,Bluntisham,Cambs.

Microelectronics And Computers in Education (M.A.C.E.) —West Midlands Regional Centre.

As you are a reader of 'MICRO-SCOPE' then I can assume that at least you have already heard of the Microelectronics Education Programme (MEP), even if you are not quite sure how it aims to help the classroom teacher. Well, the Government has made available £9 million in the period 1980-84 to help schools to prepare children for life in a 'microtechnological' society. The strategy adopted by the MEP has been to divide the country into 14 regions, each with a Regional Centre from which the Programme's activities will be coordinated.

The Programme itself will support activities in the areas of curriculum development, teacher training, and resource organisation, and will work closely with LEA's to support schools and individual teachers.

The West Midlands Regional Centre serves teachers in the following authorities: - Birmingham, Coventry, Dudley, Hereford and Worcester, Sandwell, Shropshire, Solihull, Staffordshire, Walsall, Warwickshire, and Wolverhampton - and is based at this address:

MACE Regional Centre
Four Dwellings School
Dwellings Lane
Quinton
Birmingham B32 1RJ
Telephone: 021 421 6361

We hope to provide a comprehensive information and advice service to all teachers interested in the use of microelectronics and computers in education, and you are invited to telephone or write to Ian Glen, the Regional Director, contact our Information Officer, Christopher Pedley, or come up and see us between 9.00 and 5.00 on any schoolday.

For further information contact the MACE office or look out for our Regional magazine, to be published in December (send A4 SAE for your personal copy).

Ian Glen
Regional Director.

BBC SUPPORT FOR MICROELECTRONICS EDUCATION

This article was written for 'MICRO-SCOPE' by Hazel Sumner, the Teacher Education Officer of the School Broadcasting Council. It confirms the importance attached to the Primary field by the BBC, whose contribution is likely to be increasingly influential.

Earlier this year the School Broadcasting Council completed an investigation into the current relationship of microelectronics to education in schools. There were three main dimensions to the enquiry -

(i) a postal survey of all LEA's;

(ii) discussions with project personnel and other leaders of curriculum development involving microelectronics;

(iii) visits to schools, including some primary schools, by the SBC's Education Officers.

A major problem was the rapid development taking place in this aspect of schools' work. The situation shifted, even as it was being investigated! Now, only four months later, the findings have an aura of 'historical evidence' about them. Nevertheless, the investigation has provided a data basis for short term policy decisions. The strength of the trend towards the incorporation of microelectronic technology into education, which the enquiry confirmed, has reinforced the intentions of educational broadcasters to make a long term commitment to the provision of support for this aspect of education.

Replies were received from 81 LEA's, giving a 65% response rate. It will come as no surprise to readers of this journal to learn that.very, very few primary schools have any hardware - about 1% according to the LEA returns, though this is probably an under-estimate. In any case the situation may well be transformed in the not too distant future by the probable extension of the DOI 'micro in every school' scheme to the Primary sector.

The survey also indicated that Computer Assisted Learning (C.A.L.) was rated by LEA's generally as the most important application of computers in primary schools, though this was displaced for middle schools by computer appreciation/awareness as a curriculum priority.

The consultations and visits left the investigators with no clear picture as to the kind of support which broadcasts could provide for primary schools, except that there was an obvious need for increasing primary school teachers' level of awareness as to the potential of microelectronics to contribute to primary education. However, as needs clarify, the provision of microelectronics-related broadcasts and series for primary pupils is sure to follow.

The current output of the BBC's educational broadcasting departments includes several series which are of interest to teachers in primary schools, though there is at present only one set of programmes for primary pupils on the microelectronics revolution. Called Today and Tomorrow, this series of four 20-minute television programmes will be broadcast again on Monday afternoons (repeats on Wednesday mornings) during the Summer Term 1982. It is designed for 11-year-olds and aims to provide a springboard for ideas about developments in technology which may significantly affect children's lives in the future. One of the programmes shows how microprocessors are likely to revolutionise our lives and it aims to appeal to pupils' imagination by projecting these developments forward into the next century.

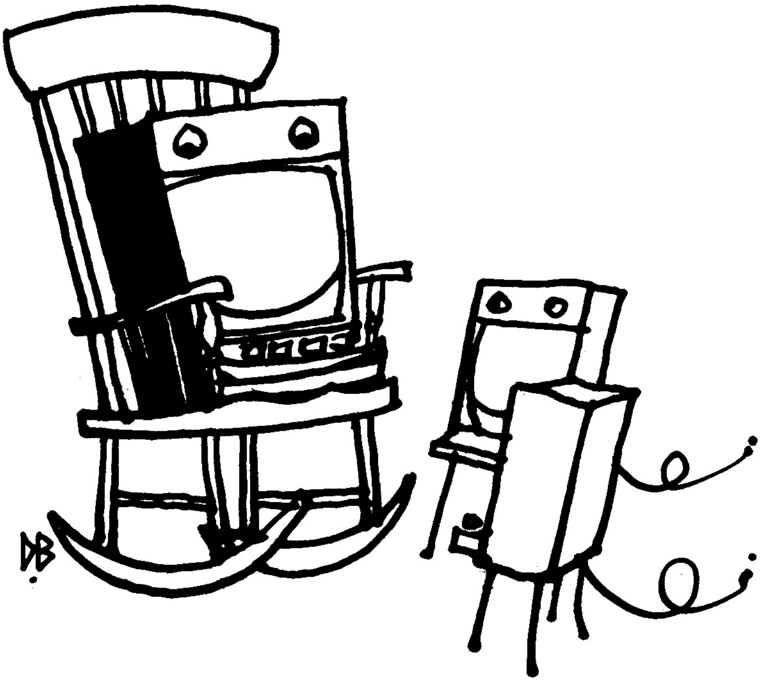
We are all learners when it comes to the understanding of revolutionary technological developments and several series for older pupils can be of use in helping teachers to understand the scientific background and the social implications of the micro revolution. Among them is "The Silicon Factor", a unit of three television programmes in the General Studies series for sixth formers. These are being broadcast again during the Autumn Term 1981 and, used in recorded form, the programmes could form an ideal focus for school-based inservice meetings.

For teachers who want to gain some insight into the scientific background to microelectronic developments there is Electronics and Microelectronics, a ten-programme radio series for 14-16 year olds. Those of us in the Primary sector who feel less than confident about science and technology will find the **five radiovision** programmes in this series to be particularly helpful.

Though made primarily for pupils , all these programmes can be of help to teachers also. However, the 1981-82 schedules contain two series made specifically for teachers. These are -

(i) Micros in the Classroom - two programmes showing how microcomputers are being used at the moment in a selection of schools in various parts of the country. Programme 1 focuses on secondary schools, but programme 2 is about micros in primary schools. The programmes are being supported by a training package which is being developed by MEP. This is being distributed to all LEA's. Further information and copies for inspection are available at each of the MEP's Regional Centres.

(ii) Learning About the Chip - This series of six radio programmes will be broadcast on Monday evenings during the second half of the Autumn term. The aim is to examine key issues facing educationists in recognising and meeting the demands implied by the new technology. The series is likely to be invaluable as background to the making of



'Mummy, can you really remember when there were valves ?'

curriculum policy decisions. Recorded use, in group meetings, is strongly recommended.

The other major broadcast contribution during 1981-82 is what has come to be called The BBC Computer Literacy Project. Starting in January 1982, this project centres around ten television programmes and a specially developed microcomputer which are likely to be of seminal importance in their impact on the general public's insight into what a microcomputer can do. For further details readers are referred to a previous article in'MICRO-SCOPE 3'(June 1981), 'Microcomputers and the BBC'.

So much for the present - but what of future broadcasts in this area? Several possibilities are worthy of note. Two of these will be considered by the Schools Broadcasting Council when it meets during November. (The School Broadcasting Council is the body which decides educational broadcasting policy for the BBC.) One proposal is that there should be a short radio and radiovision series for 14-16 year olds. As it stands at present, the first programme would give an over-view of the impact of computers on everyday life. This programme would provide background for the teachers. The remaining four programmes would be for pupils. They would be radiovision programmes illustrating how computers affect selected aspects of daily life.

The second proposal is for a television series for 13-16 year olds. This would provide broadly based resource material on the uses and wider implications of computerisation.

For teachers themselves, a follow-up to the television series Micros in the Classroom is being considered. This might run to another three programmes on micros in education. In addition, for general audiences - teachers included - there is the possibility of a further ten-part television series, designed as a sequel to the first series of broadcasts in the BBC Computer Literacy Project.

It must be emphasised that these are possibilities, not hard plans, though of one thing readers can be certain - there will be broadcasts to support teachers as they seek to ease their pupils' entry into a world made different by the advent of the micro. Some of these broadcasts will appear as identifiable series. Others will appear incidentally in established series. In the long run these incidental references to the uses of microtechnology may well be the more significant for they will reinforce for pupils the fact that the micro revolution has become an integral part of our culture.

In conclusion, mention must be made of the exciting possibilities inherent in the Telesoftware and Education Project currently in progress at Brighton Polytechnic. This enables BBC's CEEFAX and ITV's ORACLE to transmit programs to schools. Special television sets which are capable of capturing this telesoftware have been developed by Mullard . The Project is in the early stages at present, but it opens up the possibility of a whole new dimension to educational broadcasting as we know it now.

Hazel Sumner

* * * * * *

Telesoftware - Program Exchange Made Easy!

- We know only too well that the dissemination of programs by cassette, disc or listings is time-consuming, and not always reliable. The future lies with new technology which can capture software directly off-air or by telephone. Brighton Polytechnic's Telesoftware and Education Project (referred to above by Hazel Sumner) is exciting and important.
- We recently met Leslie Mapp, the Project's Research Fellow, for an exchange of ideas and information, and will keep readers informed of developments. Here are some extracts from the project's first newsletter (Summer 1981):-

"The BBC and ITV TELETEXT services will transmit pages of programs and the project staff will work with the nine participating schools to study their use and help develop suitable materials.

"A telesoftware set consists of a 22" colour teletext television with a full keyboard attached. While not exactly portable, it is nevertheless easily moved once trolley-mounted, and the equipment design reflects its multiple roles. In a classroom as a colour television; in a library as a teletext information resource; in a lecture room as a visual demonstration screen or in a laboratory or classroom as a micro-computer, a telesoftware receiver could satisfy tasks presently requiring many separate pieces of equipment. For education, telesoftware therefore offers a number of benefits. The integrated nature of telesoftware sets provides sophisticated equipment capable of many uses; its flexibility enables several teaching areas to use a single unit.

"By making telesfotware simple to use, expert computing knowledge will be less necessary and microcomputers can move more easily out of their present maths/science bias. One of the express aims of the research project is to investigate software for non-specialist use which, in practice, means arts and humanities teachers.

"Test programs are currently being broadcast on pages 700-702 (CEEFAX, BBC 1) and 175 and 184 on ORACLE (ITV). These are available visually on any teletext receiver but, of course, cannot be captured except on telesfotware equipment. Broadcast software will include programs representing the range of those currently available. The project team is also developing special software sub-routines to capture information broadcast as part of normal teletext pages (e.g. FT index, weather statistics), to provide continuously revised data for use in other programs.

"The TELESOFTWARE AND EDUCATION PROJECT will provide insights into the educational potential of telesoftware, testing the feasibility of providing cheaply distributed and centrally produced educational software over a wide spectrum of subjects. It is an ambitious venture in that the equipment and techniques being placed in the hands of schools are very much state-of-the-art, yet the project's management has ensured that it will be the educational needs which will lead the technology not vice-versa."

● Other developments in this field are noted in C.E.T. Information Sheet No.3 (address on P. 40)

CET has just set up a two-year trial scheme for the distribution and reception of computer programs via telephone lines, using PRESTEL. Up to 25 institutions will be supported with a library growing to a limit of about 50 well tried and tested programs. Other institutions can use the service at their own expense: contact the Telesoftware Project Manager, Burleigh Teachers Centre, Wellfield Road, Hatfield AL10 OBZ. Currently, access is available for an RML 380Z with a modem and Prestel jack socket from British Telecom.

The BBC will initiate experiments in telesoftware, based on its own new Microcomputer and using CEEFAX and ORACLE. A low-cost teletext decoder with a Prestel facility will be an optional peripheral. Alternatively, program listings can be copied by hand from teletext sets!

Running costs via Prestel compare favourably with postal charges for discs. The Teletext distribution service is free. "It is not yet clear how program providers will charge for their programs, where a normal selling price is involved."

John Lane.

TWO EVALUATION PROJECTS

It is time, now thatthe first wave of enthusiasm has settled, to take a long hard look at the prospects for computers in primary schools. We are all working in the dark. Progress is diffuse, practice is haphazard, successes and failures too often go unrecorded. 'MICRO-SCOPE' intends to provide a forum for informed debate based on an exchange of experiences and opinions.

In particular, we wish to report on any organised efforts to summarise and appraise current practice, or to fill the software gap. In this issue we have initial outlines of two projects just getting under way: future issues will follow their progress. Our pages are open to other initiatives in this field from around the country.

We also invite discussion of the issues raised, and additional reactions from readers' own experiences and observations. Co-operation in this crucial area is a key to sound progress.

The Birmingham Project 1981/82

Introduction

MICRO-SCOPE 2, April 1981, has already reported on the Microelectronics And Computers in Education (M.A.C.E.) project, which was run for the West Midlands during the year 1980/81. It gave us the opportunity to 'dabble' with microcomputers in Primary Schools. We put two Tandy TRS 80 machines into two classrooms for the year in order to see what would happen when the children in the classrooms had got over the initial excitement and had accepted the micros as just another piece of equipment.

We learned a lot during the year, but the biggest lesson was that the software available was sadly lacking in both professionalism and educational content. Many games were available, and proved to be popular(what a surprise!), but when it came to educational software there was very little from which to choose.

It also became apparent that few people had given much thought to the management of the micro in the school. It is easy to say what should happen,from outside the school, but it is another matter when you are a teacher within the school being asked to 'evaluate' the machine.

We decided that if we were to progress from this start, we would need a group of teachers who had some experience of using micros, and who could specify what software they wanted so that programmers could write to these specifications. It was not good enough to leave the programmer to work alone as the software produced often lacked educational viability. Thus we have developed the idea into the Birmingham Project for 1981/82!

The Aim

We hope to produce a group of Primary schools who have had experience in operating microcomputers and who can become the mainstay of software generation for Birmingham Primary Schools. We hope to keep in very close contact with these schools during the coming year, as well as providing the opportunity for them to meet each other and discuss their progress. We will provide the programming expertise so that ideas generated within these schools can be programmed and evaluated in a reasonable time. Software developed in this way will then be available to other Birmingham schools.

The Equipment

Birmingham has standardised on the Research Machines 380Z for use in secondary schools, and it was thought desirable that the machine used for the Primary Project should be compatable. Therefore it was decided that we should use the Research Machines 480Z for the Primary Project. This, of course, is a completely new machine, but it provides us with a clean start . We do not have to 'evaluate' a lot of software which is already available, because there isn't any! We can develop our own ideas right from the start.

The Schools

During the year 1980/81, we ran a Primary Computing Group every Monday evening. In that time the total number of schools who became involved must have been close to 30, with a 'hard core' of about a dozen. It was agreed that this group of twelve should be the start of the Project. When we had sorted out the finances for the year, we had five schools who were sufficiently interested to put some of their hard-won money into microcomputers, and with some help from the Birmingham Educational Computing Centre, orders were sent off for the machines. Before they arrive, we hope to be able to visit all the schools individually and offer an in-school session so that all the members of staff can see a computer, and a human from the Computer Centre, as well as discuss their thoughts and apprehensions about the Project. At least one member of staff in each school will have been to the Computer Centre on the Monday night sessions, but we would like the whole staff to feel involved in the Project. We have started work on a program specification system for teachers, and we hope that during the coming year we will be able to make use of this system and generate software for the 480Z which will overcome the inadequacies found in current software.

Colin Watkins

Newman College Schools Project 1981/82

As announced in our last issue, this project is designed to monitor and evaluate the use of micros in six local schools. Financial support for the hardware, from Research Machines Ltd and the DOI, has already been acknowledged gratefully. Now we can thank MEP for providing funds for a 4-term Teacher Fellowship in support of the project.

A final report is expected in Summer 1983. Meanwhile, 'MICRO-SCOPE' will provide some commentary on developments. The following objectives have been defined:-

a) to monitor and record the applications;

b) to comment on the effectiveness of existing software;

c) to provide new developments in software (in so far as this is directly relevant to the needs of the Project);

d) to seek to identify promising areas for positive development, with relation to subject content, classroom organisation or teachers' needs;

e) to note inappropriate uses and areas of difficulty;

f) to propose outlines for further research.

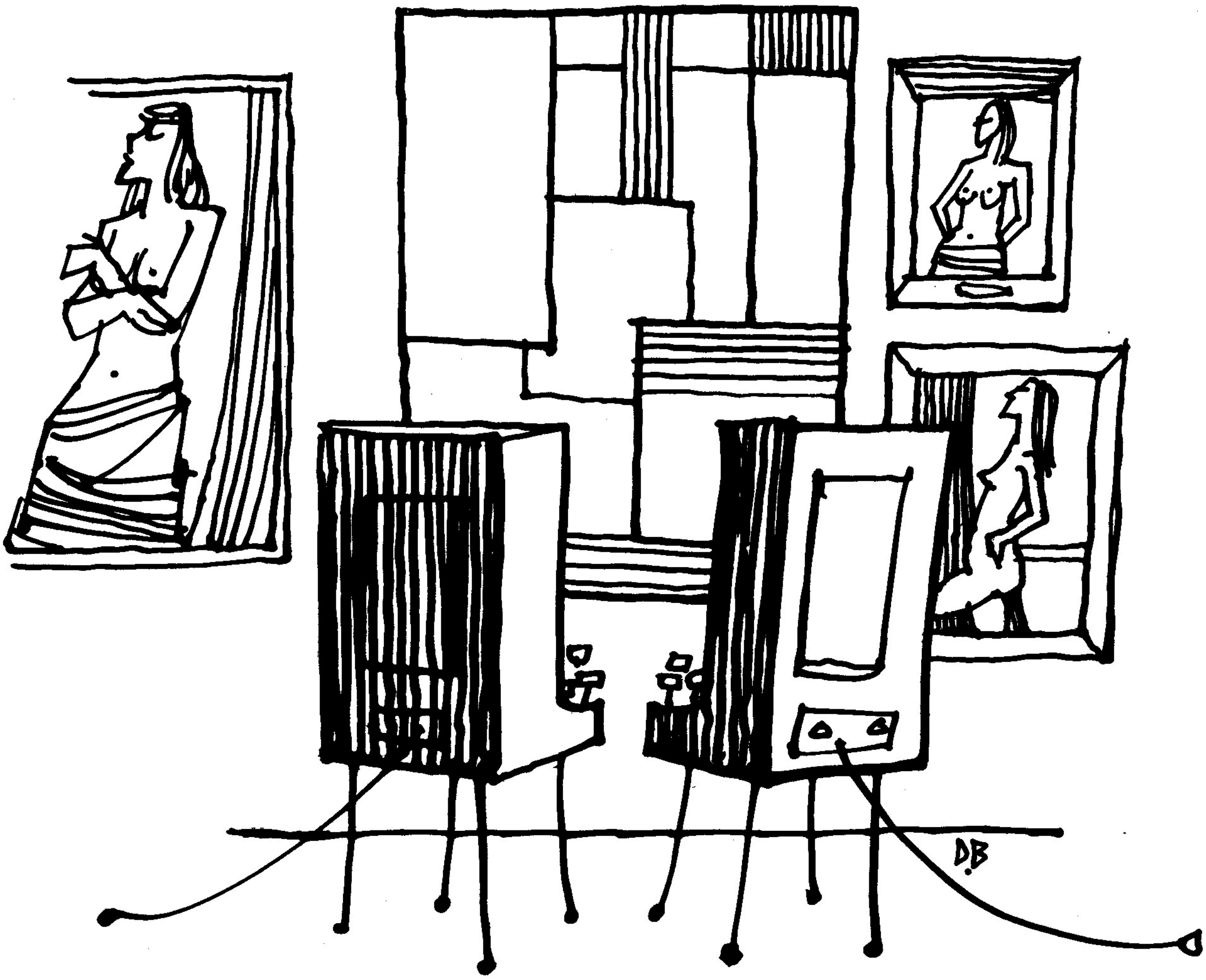
A great deal of valuable spade-work has been done in the participating schools already, before the computers arrive in November! In the following article, one of the teachers records the process of preparation.

Getting Ready for the Micro

On learning of the possible acquisition of a micro my first aim was to justify its use in school, not only from theoretical educational aspects, but from the most important aspects of all. Would it work in my school? And would my colleagues be prepared to use it and thus stop it from becoming one of the fabled "nine-day wonders/dust collectors"?

At some unspecified day in the future there would appear on the school doorstep a box marked " dn ʎɐʍ sᴉɥʇ", containing the "mighty micro". Great - then what do you do? Is there someone available who can program the beast? I satisfied myself that by hook, or by crook, programs would be available and would be capable of being developed to further stages. Then I started the process of convincing my colleagues that - Yes ! - there was a place for a micro in our School.

This consisted of finding any micros that were available and having them in classrooms as much as possible. The micros that were available were a PET and a RML 380Z, the first loaned by



'I don't know much about Art, but I know what I like.'

Sandwell and the 380Z by Newman College. The PET created its own problems and the first practical test ...... there were no programs! So one weekend and one bottle of Optrex later and no small thanks to Newman's Basic Programming Course I emerged with a suite - be it ever so small - of working programs. The PET was then tried around school in a variety of age and ability ranges. The remarks that came back from the colleagues concerned sounded quite favourable. The next major task was yet to come - how to get the micro classroom-tested by someone who initially was very wary of "That machine or any other machine for that matter!" This was overcome by a Fifth Column; I think the quote was "Dave, can I borrow that machine of yours because little Johnny in my class has just complained to me that his friend in the next class has had the computer in his room for two half mornings and we haven't had it at all!" Little Johnny has no idea how much I am in his debt!

The PET proved to be quite a handy little machine and is still being very useful in the interim period between the decision to have a micro and the actual delivery date. It is helping me to develop a Flying Squad of third and fourth years who are capable of loading programs wherever the micro is situated, thus freeing the class teacher of the labours of loading each time the machine is moved to a new location or needs a new program.

The 380Z was loaned to us by Newman with a suite of programs already on disc! Yes, you guessed it! My Flying Squad had just been grounded - they were used to cassettes. As the machine we were hoping to buy was cassette-based and the 380Z was on limited loan, the Flying Squad was reduced to one. The programs developed for the 380Z by Newman proved to be very successful with the classes that had the opportunity to use them in the time available. Towards the end of the loan and as interest amongst my colleagues was increasing, Roger Keeling visited the school to give a talk on "the Role of a Micro in Primary School". Another notable quote from the staff before he arrived - "He will speak in English - I hope! Not all this peripheral this and that!" Forewarned, Roger pitched his talk at the appropriate level and increased our interest still further.

At this stage it became clear to us that we would soon want to have our own micro. After consultation with Sandwell's adviser, Nigel Power, we were invited to become one of the six schools in Newman's evaluation project.

We wanted the micro to fit us rather than us fit around it. As we have nineteen classes containing approximately 600 children the first decision was whether it was to be a plaything for all or an educational tool for some. We decided to specialise in two areas, rather than try to cover the whole curriculum with a variety of topics and areas with no real continuity. Having had the experience and problems of moving micros around school, we decided to locate the micro in a room and move the necessary groups of children to it. We chose Compensatory Language as one of our two specialist curriculum areas, based on individual or small-group learning. In order to gain a different experience we selected a Geography-biased area from our Social Studies scheme to develop larger group work.

Our planning starts with a group of staff with expertise in each of these curriculum fields who meet and decide the educational needs of the programs to be developed, rather than a group of programmers writing a program and then looking around to find children that will fit it. Our programmers will come on the scene after the style of the program needed has been decided. If we think it can be programmed then we progress from there - if not, back to square one. Notice I say programmers - the interest is growing steadily from one teacher on a Basic Programming Course last year to five this year at various Colleges around the West Midlands and one on Advanced Basic at Newman.

The only thing we are all waiting for now is a little box labelled "THIS WAY UP!".

David Breedon, Head of Maths,
Parkside Junior School, Sandwell.

WINNING TEACHERS OVER

In 'MICRO-SCOPE 1'(Jan 1981) I wrote about my experiences with a borrowed 380Z. In March this year our school purchased a Tandy TRS-80 Model 1 microcomputer. Half of the cost was borne by Walsall L.E.A. in line with their plan to standardise on the TRS-80.

Our headteacher is very enthusiastic about the new technology and has given his full support to its development within the school. The initial reaction of the great majority of the staff was, however, one of suspicion and scepticism. The feeling was that the money could have been much better spent. It was my belief that this antipathy would disappear once the staff had experience of using the micro and realised its potential. A major snag became apparent almost immediately - lack of software. The L.E.A. had supplied two programs with the micro; a mathematics test and a maze game; hardly sufficient to justify the expense in terms of educational usefulness! I set to work to produce a wider range of programs and by the end of the summer term we had about a dozen programs covering basic number work and language skills.

With such a limited range of software available, the staff were still having to plan their work to fit in with the available software rather than being able to use the micro to supplement and enhance work in progress. They felt, quite justifiably, that they should be able to say, "This is what I am going to do, and I would like a suite of programs to supplement the work."

Obviously we could not meet this demand ourselves. I got in touch with Dave Futcher who runs the National TRS-80 Educational Users' Group. I sent him a tape of the programs I had written and was delighted to receive, by return of post, a cassette containing 30 educational programs free of charge. The software was of the highest quality, written and tested by teachers and covering all basic subject areas.

A couple of days after receipt of this package, a member of staff approached me to ask for a program to help with visual memory related to spelling. I was able to use one of the programs we had received and she was pleased with the results. That was the beginning of a quiet revolution in staff attitudes towards the micro. Most of the staff have now attended programming courses run by the L.E.A. and are becoming more and more confident in handling the computer. They suggest new programs and modifications or improvements to existing software. The computer is fully timetabled each week and has come to be regarded as an asset rather than a very expensive liability. Our software library now contains over 90 programs and is growing weekly. We have purchased no commercial software. Everything is either home-produced or from the Users' Group Software Library.

I feel that there is a very important lesson here. It is that a school with a computer needs someone who can provide software to meet the specific needs of the staff. The curriculum of the school must shape the software and not vice versa.

Finally, a thought for those with the power to influence L.E.A. policy. A microcomputer needs programs and programming is a very time-consuming business. Either provision should be made for exta staffing for in-school programming, or support on the lines of the Walsall E.D.C. Microcomputer team must be provided. Otherwise there is a risk that microcomputers in primary schools will end up gathering dust alongside the teaching machines of ten years ago, and a great opportunity will have been lost.

Steve Moss, Deputy Head.
Hundred Acre Wood First School
Valentine Close, Streetly, Sutton Coldfield.

*** Any TRS-80 users in the West Midlands area who are interested in joining the Educational Users' Group should contact me at the above address. (Tel. 021-353 4792)***

A Computer Aptitude Test (for pupils)

This is a short test to see whether or not you may be suited to working with computers. You have three minutes to complete the test. Please read the questions through before you start writing.

Begin when your teacher tells you.

1. Put a cross in this box: □

2. Write your Christian name backwards. ____________________

3. What is 12 + 7 ? ____________

4. How many days are there in 2 weeks? ____________

5. Stand up and sit down again.

6. ¿uoıʇsǝnb sıɥʇ pɐǝɹ noʎ uɐƆ YES/NO

7. How many zeros in one million? ____________

8. Recite out aloud 'Humpty Dumpty'.

9. Write your teacher's name here: ____________________

10. Wave to your teacher.

11. How long is a piece of string? ____________________

12. Take your shoes off and put them on your hands.

13. Now, with your pencil in your mouth, write down the numbers 1 to 10: ____________________

Now that you have read all the questions, as you were asked to do at the beginning, do not answer them. Just sit quietly and fold your arms till the time is up. You have finished!

R.K.

OUR SECONDARY NEIGHBOURS

As more teachers in primary schools become interested in microcomputers, the pressure on courses increases. One under-used resource is the nearby Secondary School. This account shows an interesting initiative across the Great Divide.

During recent visits from our Primary intake schools, the staff and pupils spent the last quarter of the day looking at our Tandy TRS-80 computers. The children showed considerable enthusiasm and knowledge, but most of the staff admitted they knew little about computing and were not aware of the computer's potential for them.

With this in mind, we decided to put on a short course at the end of the Summer Term on "Computer Assisted Learning", with special emphasis on "Uses across the Curriculum". The response was quite good, and twenty-two people from primary schools, including seven headteachers, visited our school for the afternoon.

The idea of the course was to show examples of software suitable for primary schools in as many subject areas as possible. We selected thirty-five educational programs, which covered English, Mathematics, Geography, History, French, Science, Music and Art. All the software had been written by my wife, who is a computer professional. With this in mind, she was also invited to attend, so that expert advice would be on hand. She was also interested in listening to people's opinions of her work.

We set up five 16K level II TRS-80's in a laboratory. I gave a short introduction, showing the difference between identical programs, one with no documentation and one with instructions, in order that the importance of documentation in a program could be made. The group was given about one and a half hours to browse from machine to machine to see as many programs as possible, and then invited to write down any comments or criticisms, and also any ideas for programs which they might consider suitable for the primary school.

The initial difficulty of ensuring five programs were always loaded, on a cassette based system, was soon overcome and an interesting and helpful afternoon followed. Some people were loath to leave the machines once they had the idea. We had computing material available in the form of "Micro-scope", "Educational Computing" and an excellent paper entitled "Computer Assisted Teaching", by Philip Crookall, a Cheshire County Adviser.

There seems to be no reason, providing suitable software is available, why other schools in the secondary age range should not provide a similar service for their primary neighbours. Perhaps I should mention some of the pitfalls which one should try to avoid.

Firstly, one should not attempt too much in one session, especially as some of the group knew nothing about computing and even less about its classroom possibilities. In fact one or two expressed doubt as to its use at all, which seems a little shortsighted. It is therefore suggested that any one attending such a course should have some background knowledge.

Secondly, even though I always had three other members of staff available, we found it difficult to keep programs loaded. Staff also need to be well briefed on the programs being shown in order to answer questions. This obviously takes time, which may not be available.

Thirdly, we felt that the use of pupils in the demonstrations would have been helpful. Some of the teachers felt, despite the fact that most of the programs had been tested in schools, that some were a bit trivial, because they were not prepared to think at the children's level.

Finally, people were prepared to make criticisms, but not to write them down, neither were they prepared to follow instructions given in program documentation!

One thing was blatantly obvious. Teachers in Primary Schools keenly want to know about computing. Supply does not appear to be meeting demand, despite many courses being run by our local authority, which are quickly filled up.

Peter N. Dowdle (Mathematics Department)
The Heath Comprehensive School,
Clifton Road, Runcorn, Cheshire.

The Micro Does the Admin!

As a head, I now teach half-time, with the result that a few visits from anxious parents soon puts an enormous strain on the admin. The problems I have to face have led me to develop programmes for primary school administration. With cuts in secretarial hours, efficiency is essential. This letter is stored on disc till this afternoon, when the day's outgoing mail is typed by our electronic typewriter. Our school meals administration is dealt with in a matter of seconds. We produce up-to-date lists of children in any chosen order and matrices of the form 7 kind, showing age and catchment area distribution throughout the school. Our school accounts have an electronic audit each day, so that errors are pointed out more or less as we make them.

No doubt, I have lost much time (my own time!) in programming activities but we are catching up fast. The big pay off is the reduction of worry. I believe computers have much to offer in the field of personal relations. Our PET system gives me more time for people. The equipment is paid for by commercial work carried out on my own premises (the computer trolley goes home each night) and by hire fees. The money-raising part of the computer activities is registered as "Nutfield Computers Educational."

Peter Matts
Nutfield Church First and Middle School
Redhill, Surrey .

A PRIMARY B.A.S.I.C. - Part 4

In this series of articles we give examples of ideas that can be programmed and some elementary tips on how to do it. Software will only be educationally sound if teachers contribute to its planning. We are avoiding sophisticated techniques since we believe that most readers are, in fact, beginners. This article deals with Program Control, an important feature of Computer Based Learning (C.B.L.).

Choices within a program may be controlled internally by the program instructions and externally by the pupil or teacher during a program run. The separate parts of a program which cater for the different possibilities may be represented as SUBROUTINES. For example, a school may have its school list which may need sorting in various ways (e.g. alphabetically, by age/sex or form). A program could contain routines for these and also for (i) reading in the information and (ii) printing out the sorted lists. It is necessary to control such a program and to be able to jump from one part of it to another as desired. Some kind of initial option list is necessary so that the teacher/operator can choose the right route for his current purpose. Such a MENU of OPTIONS is illustrated in the article on CLOZE.

Many programs on multiplication tables are already available. Our purpose in producing yet another one here is just to illustrate some principles of control. Look at the FLOW CHART (Fig.1).

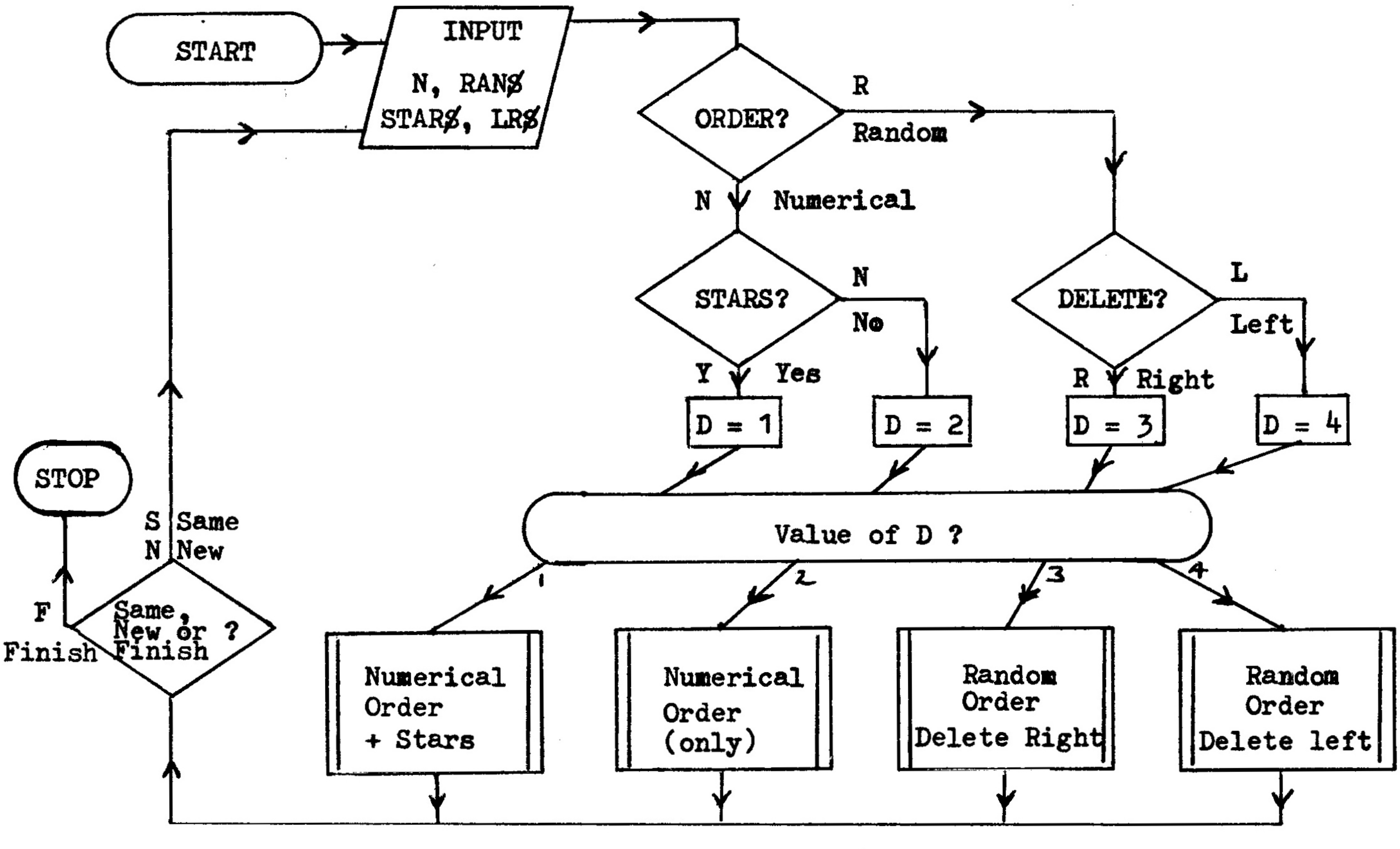


Fig. 1

First we select a multiplication table by choosing a value for N. Then there are four possible routes, each with its own subroutine. The first option asks for the answers to the products 1xN,2xN,... to be typed in. Alongside, it displays the appropriate number of rows of N stars. In the second option the task is the same, but the stars are omitted. The third presents the lines of the table in random order, as a test. Fourthly, in a similar test, factors are to be filled in rather than products.

The computer's questions and the teacher/pupil responses are shown in Fig. 2 with the responses <u>underlined.</u> This is the external control. The four values of D are determined by the logic box,

```
Numerical or Random Order (N/R)? N
Shall we include stars (Y/N)? N

Numerical or Random Order (N/R)? R
Delete on Left or Right (L/R)? R
```

fig.2

```
Which table? 7

Numerical or Random Order (N/R)? N
Shall we include stars (Y/N)? Y

Count the stars

******* 1 x 7 = ? 7
******* 2 x 7 = ? 14
******* 3 x 7 = ? 18
No, 18 is incorrect
The correct answer is 21
******* 4 x 7 = ? 28
******* 5 x 7 = ? 30
No, 30 is incorrect
The correct answer is 35
```

```
Same / New table, or Finish (S/N/F)? N

Which table? 3

Numerical or Random Order (N/R)? R
Delete on Left or Right (L/R)? L

? x 3 = 3 ? 1
? x 3 = 21 ? 6
No, 6 is incorrect
The correct answer is 7
? x 3 = 24 ? 8
? x 3 = 18 ? 6
? x 3 = 24 ? 8

Same / New table, or Finish (S/N/F)? N
```

fig.3

and these values control <u>internally</u> which subroutine is 'called' i.e. how the table appears on the screen. We can see the effect of the external control decisions in the print-outs in Fig.3. For this article we have underlined the INPUT from the keyboard. Some programming techniques dealt with in the first three issues of MICRO-SCOPE are here revised and seen in use. The loops are specially indented here to make the reading easier. We also introduce two new ideas. The first is the LOGIC BOX. We have used the relational operators before. In MICRO-SCOPE 1 we used:

```
        260 IF N < 3  THEN  240

  and   380 IF C = 1  THEN  400
```

=, <, > are logical operators. We may have two conditions to consider simultaneously. In this case we may have either "both-AND" or "either-OR". These are logical connectives and you may well recognise them as the INTERSECTION (AND) and the UNION (OR) of the two SETS. In the program which follows we use the connective AND to indicate that both "numerical order" and "print stars" are chosen (in line 300 of fig. 4). In BASIC a subroutine is 'called' by a GOSUB statement (e.g. GOSUB 1200) and the subroutine ends with a RETURN statement. Such a program would look like:-

```
1000  GOSUB  1200
   **********
   **********
1200 REM ** Subroutine
   **********
   **********
1300 RETURN
```

with the subroutine in lines 1200-1300. The GOSUB acts in the same manner as a GOTO statement except that its position is noted and the RETURN statement will take us back to the statement immediately following the GOSUB. In the program the GOSUB is contained within an ON - GOSUB statement which we illustrated in MICRO-SCOPE 3. Look at line 340. D can take any one of the values 1,2,3 or 4: the GOSUB jumps to lines 430,590,700 or 820 respectively, with the RETURN statements at lines 550,660,780 and 900. The starred REMARK lines help to identify the four subroutines.

Let us now examine the program in detail. Line 140 starts the random number generator. The pupil input at line 160 is required, by lines 170-190, to lie between 1 and 12. (See how easily this range could be amended by changing lines 170/180).

Next follows the first OPTION question at line 210. The program branches here to select one further OPTION question - the response "R" leads to line 230, while "N" leads to line 250. Lines 300-330 form the LOGIC BOX in which the previous two answers are considered. Since there were two possible answers to each of two questions there will logically be four possibilities. Each combination considered assigns a particular value (taken from a counting sequence) to a variable D. Line 340 selects the appropriate subroutine according to the value of D. Now look at the first subroutine (430-540). The inner loop (460-480) prints N stars all on one line (N is the multiplicand). Line 490 prints the question M x N = , with the "?" printed on the same line by line 500, waiting for the pupil response, "P". In the loop, M takes the values 1 to 10 successively and is the multiplier. If the response P is the correct product M x N, the next value of M is chosen at line 540. If not, the error routine (520-530) gives the answer. The second subroutine is identical, except the stars are missing.

In the third subroutine R (not M) is the multiplier as in line 710. This is a random number in the range 1 to 10, as illustrated in 'MICRO-SCOPE 3'. Ten values of this random multiplier are chosen by the loop at 700-770. The fourth subroutine only differs from the third in line 840, where the "multiplier" is missing, and lines 860/880, where the answer is considered.

Each of these subroutines returns to line 350 (i.e. the line following the GOSUB routine in line 340). Control now passes back to the user, to repeat the same subroutine, choose a different subroutine, change the table or finish (360-390).

There are many ways you could invent of improving this program. An initial demonstration could be incorporated or the level of difficulty for a second run could be altered, depending on the correctness of the initial answer . Possibly number patterns could be explored and, of course, we have not considered ways of making the program foolproof. What we have examined are ways of jumping about internally within a program and externally modifying the output by prescribing conditions. There is ample scope here for your own inventiveness.

J. Fair.

'Next!'

```
LIST

100 REM
110 REM **************** TABLES ****************
120 REM ************** Set Option **************
130 REM
140 RANDOMIZE
150 PRINT
160 INPUT "Which table";N
170 IF N>0 AND N<13 THEN 200
180 PRINT"Between 1 and 12 please"
190 GOTO 160
200 PRINT
210 INPUT "Numerical or Random Order (N/R)";RAN$
220 IF RAN$="N" THEN 250
230 INPUT "Delete on Left or Right (L/R)";LR$
240 GOTO 260
250 INPUT "Shall we include stars (Y/N)";STAR$
260 PRINT
270 REM
280 REM **** Jump to appropriate subroutine ****
290 REM
300 IF RAN$="N" AND STAR$="Y" THEN D=1
310 IF RAN$="N" AND STAR$="N" THEN D=2
320 IF RAN$="R" AND LR$="R" THEN D=3
330 IF RAN$="R" AND LR$="L" THEN D=4
340 ON D GOSUB 430,590,700,820
350 PRINT
360 PRINT "Same / New table, or ";
370 INPUT "Finish (S/N/F)";M$
380 IF M$="S" THEN 200
390 IF M$="N" THEN 150 ELSE 910
400 REM
410 REM ********* D=1 (Stars,numerical) ********
420 REM
430 PRINT "Count the stars"
440 PRINT
450 FOR M=1 TO 10
460   FOR L=1 TO N
470     PRINT "*";
480   NEXT L
490   PRINT M;"x";N;"= ";
500   INPUT P
510   IF P=N*M THEN 540
520   PRINT "No,";P;"is incorrect"
530   PRINT "The correct answer is";N*M
540 NEXT M
550 RETURN
560 REM
570 REM ******* D=2 (No stars,numerical) *******
580 REM
590 FOR M=1 TO 10
600   PRINT M;"x";N;"= ";
610   INPUT P
620   IF P=N*M THEN 650
630   PRINT "No,";P;"is incorrect"
640   PRINT "The correct answer is";N*M
650 NEXT M
660 RETURN
670 REM
680 REM ** D=3 (Random order,right deletions) **
690 REM
700 FOR M=1 TO 10
710   R=INT(RND(1)*10+1)
720   PRINT R;"x";N;"= ";
730   INPUT P
740   IF P=N*R THEN 770
750   PRINT "No,";P;"is incorrect"
760   PRINT "The correct answer is";N*R
770 NEXT M
780 RETURN
790 REM
800 REM ** D=4 (Random order, left deletions) **
810 REM
820 FOR M=1 TO 10
830   R=INT(RND(1)*10+1)
840   PRINT "? x";N;"=";R*N;
850   INPUT P
860   IF P=R THEN 890
870   PRINT"No,";P;"is incorrect"
880   PRINT "The correct answer is";R
890 NEXT M
900 RETURN
910 END
```

```
RUN

Which table? 2

Numerical or Random Order (N/R)? N
Shall we include stars (Y/N)? Y

Count the stars

** 1 x 2 = ? 2
** 2 x 2 = ? 4
** 3 x 2 = ? 6
** 4 x 2 = ? 7
No, 7 is incorrect
The correct answer is 8
** 5 x 2 = ? 10
** 6 x 2 = ? 12
** 7 x 2 = ? 13
No, 13 is incorrect
The correct answer is 14
** 8 x 2 = ? 16
** 9 x 2 = ? 1 8
** 10 x 2 = ? 20

Same / New table, or Finish (S/N/F)? S

Numerical or Random Order (N/R)? N
Shall we include stars (Y/N)? N

 1 x 2 = ? 2
 2 x 2 = ? 4
 3 x 2 = ? 6
 4 x 2 = ? 8
 5 x 2 = ? 10
 6 x 2 = ? 12
 7 x 2 = ? 14
 8 x 2 = ? 15
No, 15 is incorrect
The correct answer is 16
 9 x 2 = ? 18
 10 x 2 = ? 20

Same / New table, or Finish (S/N/F)? N

Which table? 7

Numerical or Random Order (N/R)? R
Delete on Left or Right (L/R)? R

 10 x 7 = ? 70
 6 x 7 = ? 42
 1 x 7 = ? 7
 1 x 7 = ? 7
 5 x 7 = ? 34
No, 34 is incorrect
The correct answer is 35
 7 x 7 = ? 49
 5 x 7 = ? 31
No, 31 is incorrect
The correct answer is 35
 5 x 7 = ? 35
 1 x 7 = ? 7
 2 x 7 = ? 14

Same / New table, or Finish (S/N/F)? S

Numerical or Random Order (N/R)? R
Delete on Left or Right (L/R)? L

? x 7 = 21 ? 3
? x 7 = 42 ? 6
? x 7 = 49 ? 7
? x 7 = 7 ? 2
No, 2 is incorrect
The correct answer is 1
? x 7 = 70 ? 10
? x 7 = 49 ? 7
? x 7 = 14 ? 2
? x 7 = 70 ? 10
? x 7 = 7 ? 0
No, 0 is incorrect
The correct answer is 1
? x 7 = 7 ? 1

Same / New table, or Finish (S/N/F)? F

Ready:
```

Fig. 4

THE CLOZE PROCEDURE

In this issue we take a look at a flexible approach to the teaching of the Cloze Procedure. The Bullock Report describes this as "the use of a piece of writing in which certain words have been deleted, and the pupil has to make the maximum possible use of the context cues available in predicting the missing words". In the example given here the first gap could vary from simple words such as "all" or "the" to more descriptive text such as "bleak", "Canada's" or "scarcely populated".

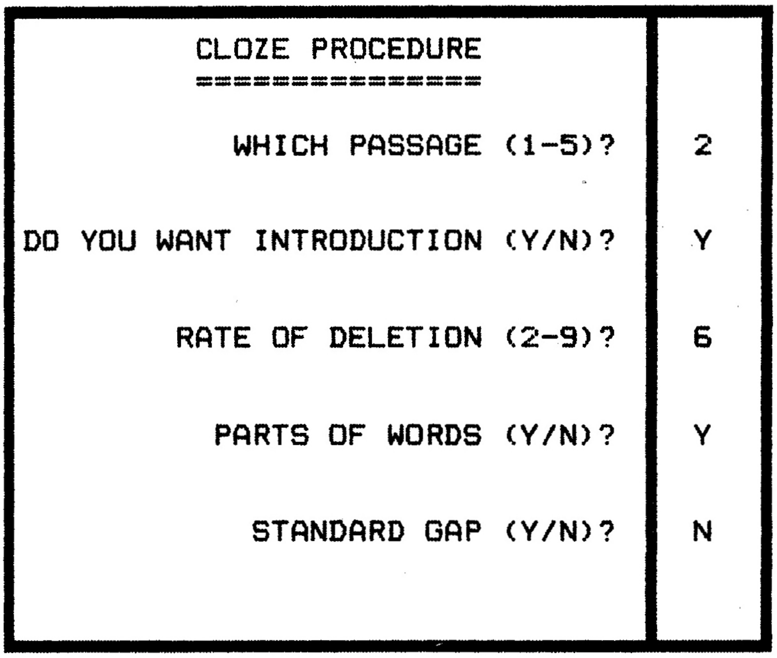
```
   In ----- northern lands the -----
is short and ----- winter is long
----- cold . Life ----- a continual
battle ----- the grim powers -----
nature ; against ----- cold and the
----- , the snow ----- ice of winter
----- bitter winds the ----- rocks
where no ----- thing will grow -----
against the terrors ----- dark
mountains and ----- haunted ravines .
```

It is more than a mere gap-filling exercise, because the discussion which follows the attempt of an individual or group to complete the passage involves a critical examination of language, structure, context, meaning and style. The child is in a genuine searching situation in which there is no single right answer. The teacher only requires justification of a choice of words on the grounds of contextual aptness. Obviously therefore there may be several correct answers, and no computer program can take account of all the possibilities - just think of all the available alternatives if the missing word is a colour.

All the program does is to suggest a possible answer for each deletion. This can be done one word at a time in response to each input, or the complete set of deleted words can be displayed at the end of the passage. This is determined by the teacher. To achieve the former mode simply remove line 900.

'I wonder what it said to the Caretaker's dog'.

The program is intended for use in a group situation, as interaction with other pupils is crucial in advancing vocabulary and improving verbal skills and understanding. The program simply provides a stimulus, but a very flexible one. The 'menu' shown below demonstrates this flexibility:



```
        CLOZE PROCEDURE
        ===============

          WHICH PASSAGE (1-5)?     2

DO YOU WANT INTRODUCTION (Y/N)?    Y

      RATE OF DELETION (2-9)?      6

         PARTS OF WORDS (Y/N)?     Y

           STANDARD GAP (Y/N)?     N
```

```
WORD PARTS....

A) FIRST LETTER LEFT
B) LAST LETTER LEFT
C) FIRST HALF LEFT
D) SECOND HALF LEFT

WHICH OPTION (A-D)?
```

At present five passages are stored, but (as you can see from the listing) the teacher can easily extend the number of passages or alter the level of difficulty of the existing ones. The first line of the menu then enables you to call up whichever passage you want. The remaining questions determine the presentation. The introductory lines can be left intact without any deletions, to set the context and style. If every second word is to be deleted the rate is set at 2, and so on up to a deletion rate of nine, always from a random starting point. If "standard gap" is selected, five dashes will be used to indicate a deletion, regardless of the length of the missing word - the alternative is to represent each letter in the missing word by a dash. There is an extension to the latter possibility indicated by the expression 'parts of words'. Instead of deleting a whole word it is possible just to omit a selected part, as follows:-

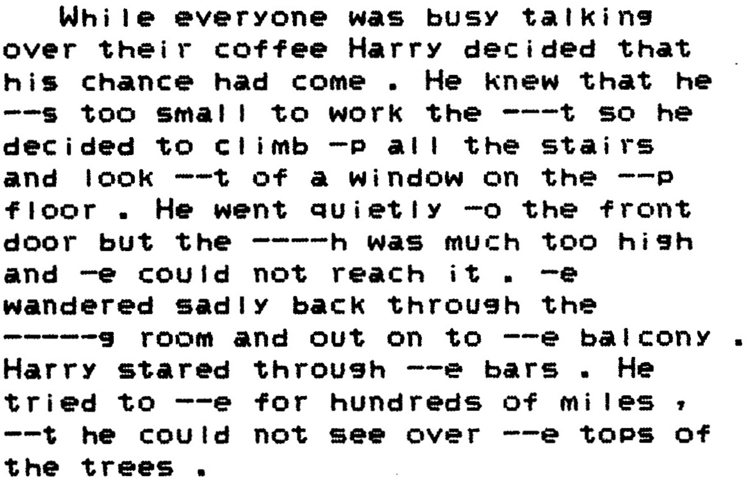
a) all but the first letter,

b) all but the last letter,

c) the second half of the word,

d) the first half of the word.

These choices allow endless possible presentations depending upon age and ability level. The menu shown on the previous page, with option B selected, generates the following passage.



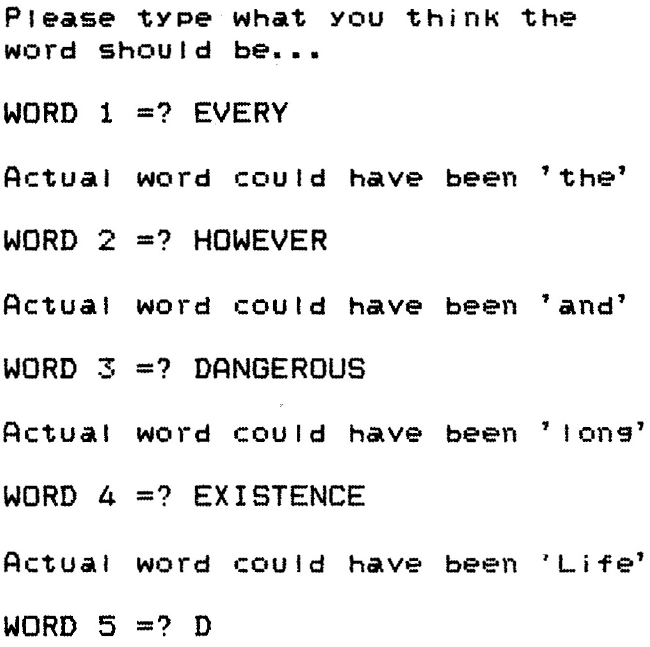
```
  While everyone was busy talking
over their coffee Harry decided that
his chance had come . He knew that he
--s too small to work the ---t so he
decided to climb -p all the stairs
and look --t of a window on the --p
floor . He went quietly -o the front
door but the ----h was much too high
and -e could not reach it . -e
wandered sadly back through the
-----g room and out on to --e balcony .
Harry stared through --e bars . He
tried to --e for hundreds of miles ,
--t he could not see over --e tops of
the trees .
```

Perhaps you can now guess the sequence of inputs to the menu that produced the passage at the beginning of the article.

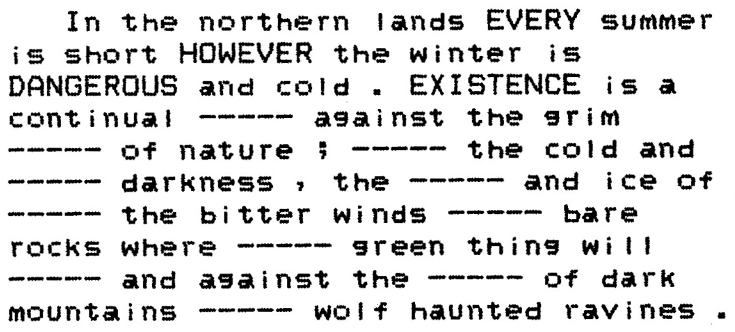
A few points of programming interest:

1) The passage is displayed on the graphics area and not in text mode. This avoids scrolling and the possibility of losing the beginning of the text off the top of the screen.

2) Each time a word is input, a pupil can type D and see an immediate display of the passage with the word inserted into its correct position. An example of this is illustrated below:



```
Please type what you think the
word should be...

WORD 1 =? EVERY

Actual word could have been 'the'

WORD 2 =? HOWEVER

Actual word could have been 'and'

WORD 3 =? DANGEROUS

Actual word could have been 'long'

WORD 4 =? EXISTENCE

Actual word could have been 'Life'

WORD 5 =? D
```



```
   In the northern lands EVERY summer
is short HOWEVER the winter is
DANGEROUS and cold . EXISTENCE is a
continual ----- against the grim
----- of nature ; ----- the cold and
----- darkness , the ----- and ice of
----- the bitter winds ----- bare
rocks where ----- green thing will
----- and against the ----- of dark
mountains ----- wolf haunted ravines .
```

3) The menu is written so that any false inputs are ignored. For example, it will ignore a rate of deletion of 12 or a request for passage number 8.

4) The ability to demonstrate possible solutions after each input or at the end of the passage could be <u>incorporated into the menu</u>, as compared to altering one line of the program as at present.

5) The program could be extended in its scope. In particular it could be arranged for the pupils to input all the words they consider feasible for any one specific gap, and for those words to be stored for later recall by the class teacher, or to be output to a printer. A further extension, now being written, is the facility, after completing the exercise, to go back and change certain words if a group can suggest improvements to the completed passage.

By the use of the Cloze procedure the teacher can gain real insight into a child's reading strategies by listening carefully to the discussion and by studying the words that the child chooses. He can take the opportunity of pointing out cues that the child has missed, and strategies that the child has failed to use. These strategies develop naturally in many readers, but where they do not it is the teacher's responsibility to teach them. The computer can do no more than suggest possible answers and provide the stimulus to trigger off discussion,but with a variability in the passage to suit all age and ability levels.

Roger Keeling.

```
10 REM ******** CLOZE 32K **********
20 REM ******** AUTHOR  Newman College
30 CLEAR 1000:GRAPH 1:PUT12
35 CD=1
40 DEF FNX(U)=50.5*U+141.5
50 ON BREAK GOTO 950
60 RANDOMIZE
70 DIM A$(300),X(300),C$(100)
80 PLOT 0,0,2:LINE 79,0:LINE 79,59:LINE 0,59:LINE 0,0
90 PLOT 67,0:LINE 67,59
100 FOR Y=45 TO 9 STEP -9
110 READ A$:PLOT 65-2*LEN(A$),Y,A$
120 NEXT Y
130 PLOT 20,54,"CLOZE PROCEDURE"
140 PLOT 20,51,"==============="
150 AR=1
160 FOR Y=45 TO 9 STEP -9
170 IF Y=9 AND I(2)=89 THEN I=78:T=1:GOTO 270
180 PLOT 71,Y,FNX(AR)
190 I=GET(10):IF I=0 THEN AR=-AR:GOTO 180
200 PLOT 71,Y,192
210 IF I=89 OR I=78 OR I=121 OR I=110 THEN T=1:GOTO 240 ELSE T=0
220 IF Y=45 THEN U=53:L=49 ELSE U=57:L=50
230 IF I<L OR I>U THEN 180
240 IF T=0 AND Y<>45 AND Y<>27 THEN 180
250 IF T=1 AND (Y=45 OR Y=27) THEN 180
260 IF T=1 AND I>90 THEN I=I-32
270 PLOT 73,Y,I
280 IF T=0 THEN I(Y/9)=I-48 ELSE I(Y/9)=I
290 NEXT Y
300 P=I(5):I$=CHR$(I(4)):R=I(3):P$=CHR$(I(2)):SG$=CHR$(I(1))
310 UU=GET(200)
320 TEXT:PUT12
330 IF P$="N" THEN 430
340 ?"WORD PARTS...."
350 ?:?:?"a) FIRST LETTER LEFT":?"b) LAST LETTER LEFT"
360 ?"c) FIRST HALF LEFT":?"d) SECOND HALF LEFT":?:?
370 ?"WHICH OPTION (A-D)? ";
380 OP$=GET$()
390 O=ASC(OP$):IF O>70 THEN O=O-32
400 IF O<65 OR O>68 THEN 380
410 ?OP$:UU=GET(50)
420 O=O-64:PUT12
430 REM ******* PASSAGE BIT *********
440 IF P=1 THEN RESTORE 1220
450 IF P=2 THEN RESTORE 1250
460 IF P=3 THEN RESTORE 1330
470 IF P=4 THEN RESTORE 1380
480 IF P=5 THEN RESTORE 1420
490 K=1
500 READ A$(K)
510 IF A$(K)="*****" THEN 530
520 K=K+1:GOTO 500
530 N=K-1
540 B=INT(RND(1)*6)+1
550 IF I$="Y" THEN B=B+20
560 U$="````````````````````":K=0
570 FOR L=B TO N STEP R
580 IF L>N THEN L=N:GOTO 720
590 A$=A$(L):LN=LEN(A$):H=INT(LN/2)
600 IF SG$="Y" THEN LN=5
610 IF H=0 THEN L=L+1:GOTO 580
620 IF ASC(A$)<65 THEN L=L+1:GOTO 580
630 ON O+1 GOTO 640,650,660,670,680
640 A$=LEFT$(U$,LN):GOTO 690
650 A$=LEFT$(A$,1)+LEFT$(U$,LN-1):GOTO 690
660 A$=LEFT$(U$,LN-1)+RIGHT$(A$,1):GOTO 690
670 A$=LEFT$(A$,H)+LEFT$(U$,LN-H):GOTO 690
680 A$=LEFT$(U$,H)+MID$(A$,H+1):GOTO 690
690 REM ***?!
700 K=K+1:X(K)=L:C$(K)=A$(L)
710 A$(L)=A$
720 NEXT L
730 GOSUB 1080
740 PUT12
750 ?"Please type what you think the"
760 ?"word should be...":?
770 FOR L=1 TO K
780 ?"WORD";L;"=";
790 INPUT LINE G$:F=0
795 IFCD=1AND(G$="D" OR G$="d")THENGOSUB1080:GOTO780
800 IF G$="" THEN 780
810 FOR Q=1 TO LEN(G$)
820 V=ASC(MID$(G$,Q,1)):IF V>90 THEN V=V-32
830 IF V<65 OR V>90 THEN F=1
840 NEXT Q
850 IF F=1 THEN ?:?"EH??  -LETTERS ONLY !!":?:GOTO 780
860 PUT12
870 A$(X(L))=G$:IFCD=0THENGOSUB 1080
```

```
880 REM +++++++ ANSWERS MODIFICATION +++
890 REM +++ M=1 Continuous, M=2 Batched+
900 M=2
910 REM +++++++++++++++++++++++++++++++
920 IF M=2 THEN 940
930 ?:?"Actual word could have been '";C$(L);"'":?
940 NEXT L
945 IFCD=1THENGOSUB1080
950 ?CHR$(12):UU=GET(300)
960 IF M=1 THEN 1060
970 PLOT 0,0,"These are some possible answers..."
980 FOR L=1 TO K STEP 2
990 ?CHR$(12);"Word";L;"= '";C$(L);"'";
1000 IF L=K THEN ?:GOTO 1020
1010 ?TAB(18);":Word";L+1;"= '";C$(L+1);"'"
1020 ?:?"Press any key";:UU=GET()
1030 NEXT L
1040 PLOT 0,0,"                                   "
1050 ?CHR$(12):INPUT"Do you want to return to menu";Q$:IFLEFT$(Q$,1)="Y"THENRUN
1060 TEXT:PUT12:END
1070 REM ******** PLOTTER *************
1080 X=6:Y=54:GRAPH 1
1090 FOR LL=1 TO N
1100 LN=LEN(A$(LL))*2+2
1110 IF LN=4 THEN 1140
1120 IF X+LN>76 THEN X=0:Y=Y-3
1130 IF Y<3 THEN 1170
1140 PLOT X,Y,A$(LL)
1150 X=X+LN
1160 NEXT LL
1170 RETURN
1180 REM*******DATA**********
1190 DATA WHICH PASSAGE (1-5)?,DO YOU WANT INTRODUCTION (Y/N)?,RATE OF DELETION (2-9)?
1200 DATA PARTS OF WORDS (Y/N)?,STANDARD GAP (Y/N)?
1210 REM **************************
1220 DATA My,house,has,a,red,roof,and,a,brown,chimney,.,The,door,and,the,windows
1230 DATA are,blue,.,Will,you,come,and,play,with,me,?,We,can,have,cakes,and,buns,to,eat
1240 DATA We,will,run,and,skip,and,jump,in,the,garden,.,Then,we,will,sit,down,and,rest,.,****
1250 DATA While,everyone,was,busy,talking,over,their,coffee,Harry,decided
1260 DATA that,his,chance,had,come,.,He,knew,that,he,was,too,small,to,work,the,lift
1270 DATA so,he,decided,to,climb,up,all,the,stairs,and,look,out,of,a,window,on,the
1280 DATA top,floor,.,He,went,quietly,to,the,front,door,but,the,catch,was,much
1290 DATAtoo,high,and,he,could,not,reach,it,.,He,wandered,sadly,back,through,the
1300 DATA living,room,and,out,on,to,the,balcony,.,Harry,stared,through,the,bars
1310 DATA .,He,tried,to,see,for,hundreds,of,miles,",",but,he,could,not,see,over,the
1320 DATA tops,of,the,trees,.,****
1330 DATA In,the,northern,lands,the,summer,is,short,and,the,winter,is,long,and
1340 DATA cold,.,Life,is,a,continual,battle,against,the,grim,powers,of,nature,;
1350 DATAagainst,the,cold,and,the,darkness,",",the,snow,and,ice,of,winter,the,bitter
1360 DATAwinds,the,bare,rocks,where,no,green,thing,will,grow,and,against,the
1370 DATA terrors,of,dark,mountains,and,wolf,haunted,ravines,.,****
1380 DATA I,have,got,lots,of,toys,.,They,are,very,old,but,I,like,them,.,I,have
1390 DATAsome,pets,.,My,cat,is,black,and,my,dog,is,white,.,My,goldfish,lives,in,a
1400 DATAlittle,tank,and,my,rabbit,lives,in,a,big,cage,.,I,never,forget,to,feed
1410 DATAthem,before,I,go,to,school,and,when,I,get,home,****
1420 DATA One,day,John,and,Susan,were,by,the,lake,.,Their,dog,Patch,was,there
1430 DATAas,well,.,Patch,was,a,black,dog,with,a,white,patch,over,one,eye,.,This
1440 DATAgave,him,a,wicked,look,.,He,looked,like,a,pirate,.,That,day,by,the,lake
1450 DATA",",they,all,looked,like,pirates,.,John,was,called,Black,John,.,He,had,a
1460 DATApirate's,hat,and,a,patch,over,one,eye,.,****
```

'I said we shouldn't have
put him in goal!'

Have You Written Any Good Programs Lately?

If you have just removed the last bug, or so you think, from your latest programming masterpiece - what next? Do you distribute it free to as many schools as possible in the hope of receiving constructive criticism or look for a means of distribution that will give you some financial return? In advocating the former policy, we have expressed the following reservations about the sale of software:-

a) It restricts generous exchange of programs, essential for testing and feedback.

b) There may be a substantial difference between one's expectation of a program and what it actually does. It is one of the very few products we buy 'blind', and it could well be that novices who think their money has been wasted will be prevented from voicing genuine criticism.

c) Most programs are written (or should be) as the result of discussion between groups of teachers and a programmer. Is the latter any more entitled to the revenue than the former?

d) If a program is rewritten, how much of any revenue should belong to the original author? Who is entitled to the revenue if the program is developed in school time on an LEA machine?

Now there are two sides to every argument and I am grateful to Charles Sweeten, the MUSE software librarian, for replying to the above points.I quote from his reply.

> "Selling v. free: My experience was that free programs were not accessed from the Library and very poor quality stuff was submitted for distribution. Everyone with quality software hung on to it himself in the hope of making some money out of it. The flood of software being offered now is,I am convinced, a result of being able to offer something to the authors.
>
> "Testing and feedback: I am getting very little feedback on programs in the MUSE Library. I suspect that you are getting considerably more back but I also suspect that feedback comes as a result of a positive search for feedback. The difficulty comes particularly once one has got over about a dozen programs. Trying to cope with positively finding reactions to 70 programs cannot be done in one's spare time. Several of us are beginning to think that the only way of dealing with this is to have a professional service and do the job properly.

> "Buying blind: This is a real difficulty of course. If we send material out on appro, if it is being sold, then obviously it is going to get copied. On the other hand, we must have a mechanism whereby people can view what they are thinking of buying. I think the answer is to have software workshops and demonstrations on a very wide basis throughout the country which means that teacher centres and training colleges have got to take a very active part in buying available software, testing it for suitability and giving honest opinions to feeder schools about what would be suitable. I would like to see much stronger review sections for software in places like 'Computers in Schools' and other educational magazines.
>
> "Groups of teachers and the programmer: It is always very difficult to assess the commercial worth of an idea as against the number of hours spent on development. The same problem applies to books of course. If a teacher talks to a large number of colleagues and gets his ideas sorted out, then writes a book, it is only the author of the book that gets the return. I am not entirely happy with this argument and I think in practice that each situation is going to have to be viewed in its own light and will depend on the sort of teacher discussions that have taken place and what sorts of institutions are involved. I would wish to be extremely flexible on this point.
>
> "Rewriting a program: I am sure that any revenue obtained should be shared between the original author and the rewriter but how this is shared should depend on the amount of work done by each person. The case has arisen recently where a program was submitted originally and has since been modified by two other people in a major way. In this particular case the original author is being given the whole amount."

Obviously we can see that the whole area is fraught with problems, to which there are no absolute answers.

The MUSE software library is a genuine attempt to validate programs and to distribute them at a modest charge, whilst at the same time it can offer some reward to the author. At the other extreme, it is the 'private enterprise' individual we are particularly wary of. Anyone can put together a poor program, make exaggerated claims for it, sell it for £5 and leave a dissatisfied customer no come back when he realises the poor quality. This could quickly disillusion many teachers who innocently part with their annual capitation in the expectation of receiving quality software. Perhaps we may one day hear of the distraught teacher explaining to the judge that she was driven to software piracy after having been 'conned' by the false claims of programmers.

There are many problems still to be solved and we invite comments from interested readers. In particular I hope that the MEP will provide guidance on the subject before commercial organisations step in and dictate the ground rules - are you listening up there in Newcastle?

Roger Keeling

Charles Sweeten has the last word:

> "Commercial organisations are stepping in already in large numbers. Computer manufacturers already have a poor reputation and now everybody, it seems, is trying to get in on the act. MUSE is offering a professional assessment and review service to these organisations to try and improve the situation, and MUSE also operates a development service. It may also be of interest that MUSE is about to produce standards for Assessment. I share Roger Keeling's concern about standards, about buying blind, and about feedback, but I do believe that a good service (provision of quality software) has to be paid for in the end, or it is likely to fold up when one key individual leaves."

Footnote: MUSE Software Library is at MUSE, FREEPOST,
Bromsgrove, Worcs. B16 OJT.

CALCULATORS IN THE PRIMARY SCHOOL-2

Developing guessing strategies with a calculator

Many teachers use numerical investigations to help their pupils to develop a logical outlook. Progress is often severely hindered by long and complex calculations or even by simple calculations which have to be carried out a great many times.

It is quite possible to play all three games described in this article, and to develop the best strategies, without recourse to a calculator. It is hoped however that using a calculator will enable strategies to be developed more quickly and clearly. Its use should also allow some children who lack confidence in numerical manipulation, but who nevertheless may comprehend general relations, to extract maximum benefit from investigations of this kind.

* * * * * * *

Game A - Guessing a number chosen from a pre-determined range.

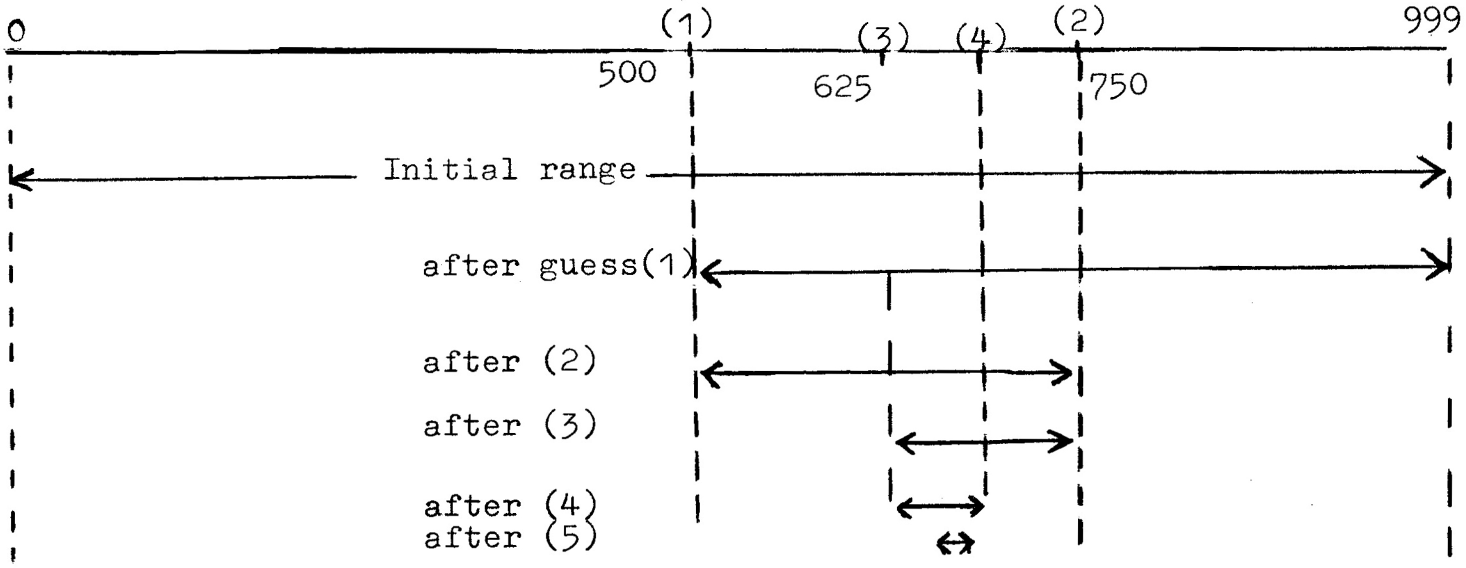
Two children play: one uses a calculator to help guess a number which has been chosen by the other child. Let us suppose that John can select any number from 0 to 999. When Mary makes a guess, John's reply states whether the guess is too high, too low or correct. Mary is allowed to use pencil and paper to keep a record. At first, Mary may guess wildly and the calculator is of little use at this stage. Later, as Mary begins to search for a good strategy, the calculator becomes more useful. Since each reply reduces the range of numbers in which the unknown number can still lie, Mary may eventually decide to divide the range by two at each successive guess. A careful record of the current upper and lower limits of the range can be kept with just paper and pencil, but a calculator with two memories will efficiently calculate and store the new limits after each guess. An example illustrates the best strategy:-

Play 1. John chooses 671.

Mary's guess	John's reply	Mary's record		Mary's calculation
		Higher limit	Lower limit	
(1) 500	Too low	1000	500	(1000+500)/2=750
(2) 750	Too high	750	500	(750+500)/2=625
(3) 625	Too low	750	625	(750+625)/2=687.5*
(4) 687	Too high	687	625	(687+625)/2=656
(5) 656	Too low	687	656	(687+656)/2=671.5*
(6) 671	correct			

* Rounded down to whole number below.

Mary's guesses and how she divides the range successively in two are illustrated below:



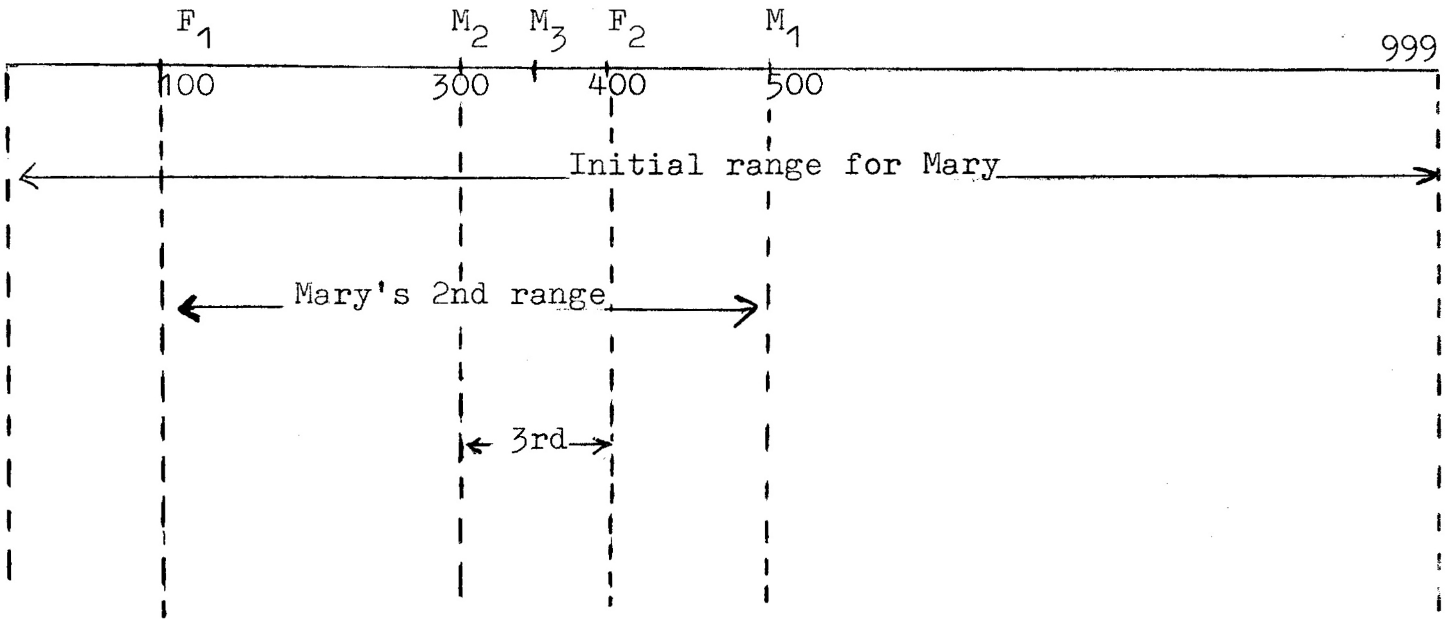
Mary has taken six guesses. Clearly the least number of guesses possible is one - but what is the least number of guesses Mary must make to guarantee success, whatever number is chosen within the range? The answer to this is left for the reader's pupils to provide.

When both players know the best strategy the chooser may experiment in selecting numbers to see if the task of the other child can be made more difficult.

A variation on this game involves a number of pupils guessing in turn. A child who knows the strategy can make use of information gained even from the wild guesses of those who don't. This is illustrated below, with Mary knowing the strategy and Fiona guessing wildly.

Play 2. Let John choose 326.

Mary's guess	Fiona's guess	John's reply	Mary's calculation
500		Too high	
	100	Too low	(500+100)/2=300
300		Too low	
	400	Too high	(300+400)/2=350
350		Too high	
	310	Too low	(350+310)/2=330
330		Too high	
	322	Too low	(330+322)/2=326
326		Correct	



A further development of this game requires the children to investigate how many extra guesses are required to guarantee success if the range for the choice of number is doubled, quadrupled etc. They will be surprised at the results. A calculator will be a great boon in averaging two four- or five-digit numbers.

* * * * * *

Game B - Guessing the operation

John and Mary have a calculator each and John chooses an operation like "add 5", "multiply by 3", or even "double it and subtract 1". (Division is not recommended for primary school pupils without additional constraints on the game.) Mary then names a number and John calculates the result of applying his operation to this number and gives Mary the result. Mary records this on paper and then names another number, noting John's reply. This cycle is repeated until Mary manages to guess (or deduce) John's chosen operation. John will be using the calculator continually whilst Mary will probably use it only occasionally:-

Play 1.

Mary names	John calculates		John replies
10	10+20	=	30
20	20+20	=	40
25	25+20	=	45
12	12+20	=	32

At this point Mary can compare her record of named numbers with the replies and will (hopefully) guess that the operation is 'add 20'. A rule involving subtraction alone will give rise to a game similar to the one above. No illustration is given here for operations which are mixtures of addition and subtraction - children readily deduce that every such case reduces to a single addition or subtraction e.g. "add 15 and then subtract 7" is equivalent to "add 8".

The operation of multiplication, taken on its own, does not present much more difficulty, especially for able children. A far greater challenge is the mixing of multiplication with either addition or subtraction and this is illustrated in the following example:

Play 2.

Mary names	John calculates	John replies
5	2x5+3 =	13
8	2x8+3 =	19
10	2x10+3 =	23

etc.

Mary may not make much progress until she realises that, for example, the increase from 19 to 23 is caused by the multiplying factor alone acting upon the increase from 8 to 10, i.e. 2⟶4. This helps Mary to deduce that the multiplying factor is 2. She can use this with her first chosen number (5x2=10) and note that this is 3 less than John's reply. Hopefully Mary will deduce that the addition operation is "add 3" and that the complete operation is "multiply by 2 and then add 3". (Of course, "add 3 then multiply by 2" is different.) Before informing John of her guess Mary can check the guessed rule with her own calculator. Mary may make many attempts involving sheer guess-work before she starts to apply a good strategy. It may be advisable initially to restrict the range of operations available from which John can choose.

Of course, examples involving small whole numbers can usually be worked mentally or using pencil and paper. The calculator, however, has some advantages even at an elementary level: (a) it speeds up the calculations and makes the game more interesting; (b) it helps to reduce errors in the calculations; (c) it encourages exploration of a wide range of "input" numbers, from which a good strategy can be developed. For children who wish to choose operations such as "multiply by 3.7 and then add 6.4" or inputs such as 2.5, the use of a calculator is very desirable. Teachers who structure the game more rigidly may help their children to discover many properties of decimals. Many children will eventually arrive at the very best strategy which guarantees the correct answer to any "multiply then add" operation in just two guesses. Readers are invited to see how long it takes their own pupils and the author would be pleased to learn of teachers' own experiences with pupils.

In an extension to this game children would deduce, for example, that "add 3 and then multiply by 2" is the same as "multiply by 2 and then add 6". This could lead to more complex equivalent operations such as "multiply by 4 then add 12 then divide by 2". The calculator is used at each stage to check the children's guesses.

Finally, a further investigation is to see for which numbers an operation such as "multiply by 3, then add 2" gives the same results as "multiply by 2,then add 3". The result will probably surprise most children, there being a unique result for "multiply and then add" and a different but also unique result for "multiply and then subtract".

Game C - Guessing a repeated operation

This game is related to work in the last year of primary school. The children are asked to find a single operation which is repeated a given number of times. The input number and the final result are also given. Examples are:"Use the same addition operation three times to get from 5 to 20", and "Use the same multiplication operation twice to get from 1 to 36". The calculator is invaluable in this work because many guesses may be needed before a good strategy is formed. The approach outlined below relies on the use of a calculator to check each guess rapidly - the results are used to indicate the next guess. "Use the same multiplication operation twice to get from 2 to 338".

Step	Guess	Calculation	Decision
1	6	2x6=12,12x6=72	72 < 338, 6 too low
2	20	2x20=40,40x20=800	800 > 338, 20 too high
3	12	2x12=24,24x12=288	288 < 338, 12 too low
4	14	2x14=28,28x14=392	392 > 338, 14 too high
5	13	2x13=26,26x13=338	Multiplier is 13.

This game has many possibilities which are left for the reader to explore.

Alan James.

'If he takes her out again I'll kick his fuse in !'

Useful Publications

1. FREE from Council for Educational Technology,3 Devonshire Street, London W1N 2BA:
   CET Information Sheets - No.1, "Thinking about microcomputers: first steps", provides a list of references and contacts; No.2,"Educational aspects of new technologies", summarises CET involvement in current developments; No.3,"Telesoftware", explains innovations of great significance (see P.9)

   "Protection of Computer Programs" attempts to clarify some problems of copyright.
   "USPEC 32" is a guide to the selection of microcomputers.
2. From Project Secretary ,I.T.M.A., College of St Mark and ST John, Derriford Road, Plymouth, Devon PL6 8BH:
   "Development and Evaluation of Materials and Methods of Use" - a discussion document (10p).
   "Teaching Style and Program Design" - analysis of design possibilities for class teaching material (£1.50)
   "Classroom Development of Teaching Material" (£2.00)
   "Practical Considerations in the Creation of C.A.L. Teaching Units (50p.)
3. From R.J. Margetts, Bristol Polytechnic, Coldharbour Lane, Bristol BS16 1QY
   A newsletter packed with up-to-date references (Sept.1981)
4. From Research Machines Mill Street, Oxford OX2 OBW, a newsletter (Sept.1981).

380Z Primary Software

In 'MICRO-SCOPE 1' we described a number of programs suitable for use in the primary school. They will only run on a 380Z or 480Z. We can now offer additional programs, which are also freely available to schools. However, as we believe in the free exchange of software, we would appreciate receiving at least one primary program in exchange for the ones we are sending out.

1) CLOZE - as described in more detail in this issue.
2) MATCH* - shape recognition for early infants.
3) KP - a contribution from Dave Fisher, Coventry, to provide keyboard practice for early infants.
4) ENGLAND*-suitable for top juniors, to develop atlas skills.
5) PUNCT - punctuation practice for juniors.
6) DIAGRAM*-similar to the previous version of DIAGRAM, but now uses high resolution graphics and any bearings.
7) HANDWRT*-for 4 to 6 year olds, showing dynamically the formation of letters of the alphabet.
8) ECHO -primary science, associated with illustrating the speed of sound.

These programs will be available as from December 1st. They require disc or cassette BASIC, version 5. Those marked with an asterisk need a high resolution graphics board. Send a 40-track disc or C60 cassette with 25p for postage (or 75p to include the documentation booklet) to Roger Keeling, Newman College.

Newman College with MAPE