

M I C R O -

S C O P E



MICRO-SCOPE - 2

(MICROcomputer Software Co-operation for Primary Education)

<u>CONTENTS</u>	<u>PAGE</u>
Editorial	2
Reader's Input - Letters	3
Yealand - a Computer Project	6
"Close Encounters" R Butcher	8
Act now R Keeling	11
Problem Specification	13
"PHRASES" - Program Listing and Documentation R Keeling	14
A Primary B.A.S.I.C. - Part 2.... J.Fair	18
The M.A.C.E. Project C Watkins	24
A Scrap-Box Computer A James	26
Software Review D Fitcher	34
Puzzle Page	37
More Hints on Hardware R Keeling	38
Courses at Newman College	40

EDITOR: John Lane

BOARD: John Fair, Alan James, Roger Keeling

DESIGN: David Barlow

TYPING: Chris Wilkinson

Published at NEWMAN COLLEGE, Bartley Green, Birmingham B32 3NT
Tel. 021-476 1181)

This second issue of MICRO-SCOPE is sponsored, like the first, by Newman College as a service to teachers. It is still circulated, free, largely by courtesy of L.E.A. mailing. We are seeking sponsorship for the future and hope to report progress soon. The June issue should reach you by the same means as this one. Meanwhile, there are no subscriptions. Requests for software should include 30p to cover postage etc.

Contributions and letters for publication should reach us by May 15th. Please let us know how MICRO-SCOPE can become more useful to you, and share your ideas and developments through us.

EDITORIAL

In this issue we are able to extend some themes from MICRO-SCOPE 1 and to initiate some new ones. We are delighted to have received so many articles and letters from schools, both from established users and from teachers who are just dabbling their toes. The water is warm! Providing a forum for an exchange of ideas and experiences will keep us busy!

Of course, many primary teachers still have negative reactions, ranging from caution to fear. One common misconception is that computers are for mathematicians and deal only with numbers. In this issue we have selected language uses for special emphasis. This may well prove the broadest and most valuable category for primary schools. We are investigating programs and specifications for early reading, phonetics, vocabulary, spelling and story-writing, and will report on progress soon. Please send in ideas.

Many schools about to take the plunge have sought our advice on hardware. Some take the view that they must choose now, so as to steer development. Others see advantages in waiting for at least a local consensus, backed by the L.E.A., on the "best buy". Latest news, that the BBC has ordered a new design from "ACORN", may be significant. Roger Keeling discusses this issue, which is highly-charged politically, on P. 38.

In our next issue we hope to begin a survey of the correct scene in a national context, following the Easter conference at Exeter. Our mailing list is growing rapidly, and will help us to form a User's Directory.. All we need to know initially is the address and telephone number of your institution, a list of your hardware, and whether you act as a local centre already. We shall also list some useful books and journals. There are many initiatives under way, on large and small scales - one enterprising head has just sent us his staff newsletter on micros!

Many teachers who have no hardware yet wish to get involved with their pupils in ideas associated with computers. Alan James' article on P26. initiates this theme, which we shall develop.

READERS' INPUT

Readers' letters will form an important part of each issue of MICRO-SCOPE. We had a large and encouraging response to the first issue, suggesting that our initiative is welcome and relevant. Many letters had constructive suggestions which we will take up as time and space permits. Some were short articles - some, longer! Again, we are grateful for them all and will try to find space for more in our next issue. Here is a small selection of extracts.

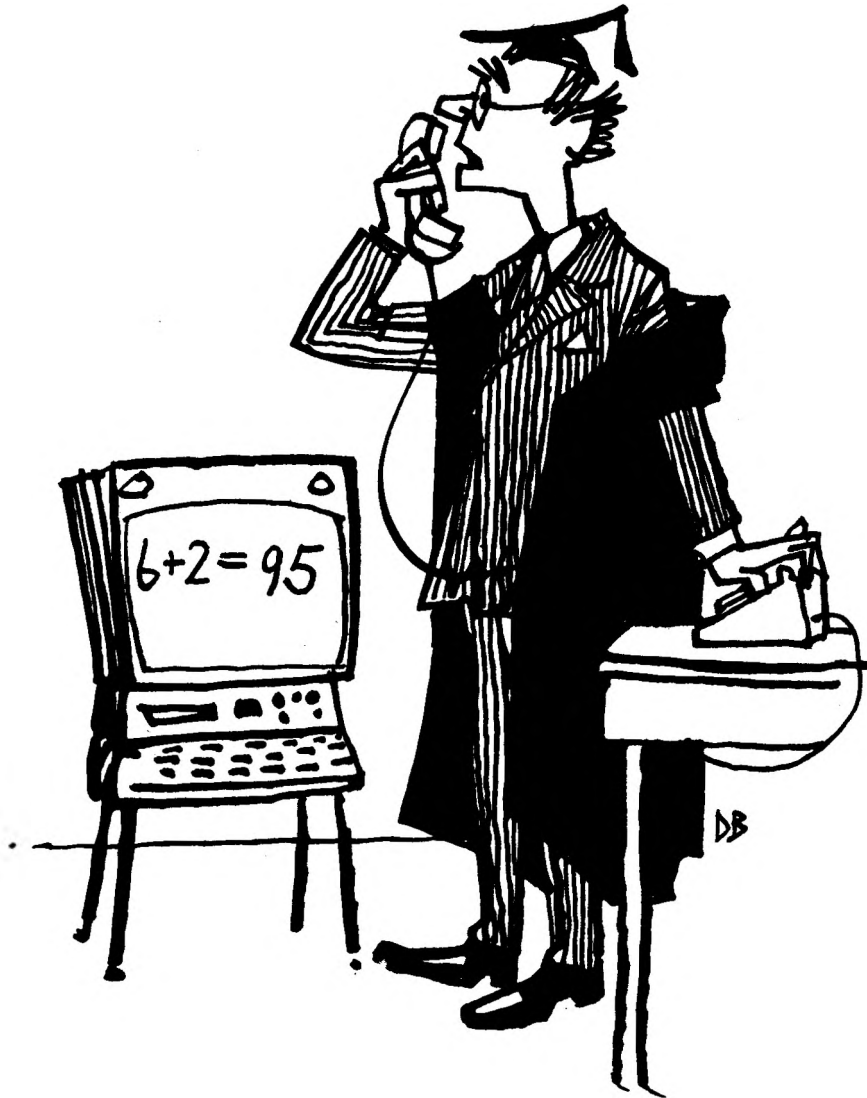
....You can imagine my surprise when at the beginning of the week I was asked by the headmaster to see him at breaktime; only to be asked how I saw a micro being used within the school! The adviser had come to school to give me the copy of MICRO-SCOPE that you had sent. The head and adviser had discussed my interest in micros and asked if it would be a feasible thing to introduce at Rush Common.

The adviser asked me, after a lengthy debate on the values and developments of micros, to find out more; with a view to introducing a micro in September 1981.

C.J. Ellis, Rush Common P.S., Abingdon, Oxon.

....I think MICRO-SCOPE will prove invaluable for two reasons. One is that one is very isolated in a primary school and the other is that primary schools are always short of money. It is one thing asking a P.T.A. to buy a computer - it is another to keep asking for more and more money for books, tapes, etc. and your magazine will help enormously as a source of primary-orientated programs (the only really appropriate source I have found yet).

Our school has 500 pupils and we have a PET 2001 8K. I am teaching our 57 10-11 year olds simple programming which they learn remarkably easily. I have done this in the previous two years and taken children to use the local Technical College computer, but now we have our own PET. I can see they are really going to achieve a level of understanding higher than I expected by the end of the school year. I am anticipating some hard work to come - for me!



' Doctor... He seems to be getting worse.'

The younger children (and the 10-11 year olds) also use it as a teaching-aid. So far, the only programs we have used are ones I have written myself. They are only simple as I wanted to build up a few quickly. We have programs to aid spelling, number bonds, multiplication tables, square numbers and square roots. We also use it to record the children's swimming ability so we can change the ability groups week by week.

I don't really anticipate writing any "flashy" programs; for one thing I love programming but find lengthy graphics and "tarting-up" as we call it, rather boring!!

At the moment, the PET is used every minute of every day by children so I haven't really time to teach teachers as well!!

Mrs D. Chaplin, Reffley Co. P.S., Kings Lynn, Norfolk.

.....I'm not too sure that I want lots of programs to "teach" (c.f. last sentence of report on Newman's November 5th course). I feel it is vital with young children for there to be interaction between teacher/child and too many good teaching programmes can destroy this. I prefer the use of our local teachers is making of it, as a "marking" machine which stores childrens results, so he can spend his time teaching and yet analyse each child's mistakes calmly at the end of day and plan for next day/week.

Margaret Adams, Rolle College, Exmouth.

..... At Primary level the Micro gives

- 1) Instant check, right or wrong - with no cheating
- 2) Speed of presentation
- 3) Variability of presentation
- 4) Incentive

It falls down in that it doesn't speak - yet! So use the Language Master - something already familiar to most children - and for a start concentrate on practice and testing programs.

I would suggest Spelling, Dictation, Word Recognition both look and say and Phonic and some Listening and Doing. Using a tape recorder a set of Listening and Understanding programs are possible.

Tom Barlow, St George's J.I.S., Newtown, Birmingham.

- ...(1) Before purchasing a microcomputer, it would be ideal if Primary schools in an area formed a group.

This group could then decide to buy identical microcomputers, and be able to exchange programs with one another and build up a library of programs. The group could also purchase in bulk and get a better price from a dealer, and be in a good position to haggle over service contract terms.

- 2) When teachers have used programs for a period of time, they will then become aware of the programs' shortcomings and be able to indicate to the programmers their exact requirements.
- 3) Microcomputers are tremendously adaptable, but it takes imagination to put them to good use. Imagination plus a thorough knowledge of the job they are required to do is the province of the serving teacher.

M.E.Fowler, Short Heath P.S., Erdington, Birmingham

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.)

CLOSE ENCOUNTERS

Seeing is believing - and doing is understanding! Teachers of young children are rightly sceptical of new approaches and materials for which extravagant claims are made with little basis in classroom practice. Take care, then, with the micro! - its appeal seems to be almost universal and, once used, irresistible. Here is another account of a staff's rapid progression from caution to commitment.

Beginnings

Early in the school year the staff were thinking in terms of introducing some teaching machines into the classroom. To this end we purchased a "Speak and Spell" and a couple of "Little Professors". Whilst we were assessing their possible uses and abuses, I attended a computer course at Newman College. Even at this early stage the vast potential of the micro was obvious. Having already observed how the smaller teaching machines motivated the children to do repetitive number practice and spelling practice, the use of the microcomputer in the classroom seemed to have endless possibilities.

I reported back to the staff about the course and also the offer Newman College had made, of the loan of a micro computer. The staff were obviously cautious in their response but it was agreed that we take up the offer and borrow a Research Machine 380Z for a week. The date was fixed for the third week in January.

Encounters

The Research Machine arrived in school as promised along with ten programs which included: Train, F.Train, Coin, Diagram and Grid. Mr R. Keeling of the College came into school one evening and gave a brief demonstration on the use of the micro for the staff.

The demand by the staff was such that the micro was timetabled into each class for half of a school day.

We had a brief "hands on" training session for staff (about 45 minutes) and then the micro was in use in the classroom. The hardware easily fitted on a trolley and could be wheeled from class to class as the need arose. When I first wheeled the micro into the classroom several children seemed rather disappointed. "Is that all it is?" was the comment. I think they were expecting a cross between Metal Mickey and K9! However their disappointment soon vanished and they very quickly became adept at handling the hardware. Many children gave up play times, dinner times and after-school time to use the micro. It was quite the norm to see children still in school at 5.p.m. or later using the micro (we finish at 3.25 p.m.)

Although we had ten programs only two were used to any great extent - "Trains" and "Diagram". Both of these programs were very popular with the children, and so took up the very limited amount of time each class was allocated on the micro. Some staff did run through other programs in their own time, whenever they could get hold of the micro. (which usually meant telling the children to go home!)

At the end of the week the microcomputer was reluctantly returned to Newman College and the staff then assessed the week.

To this end we had several informal discussions and I also issued the staff with a questionnaire to fill in. From these several points arose.

1. The initial response was favourable. However, it was felt that we should know more about the long term value of the micro, when the novelty wore off.
2. How available was good quality software?
3. Many of the programs were maths orientated - what about programs in other subjects?

4. If we bought a micro, how long before it became obsolete?
5. What facilities are there for in-service training?

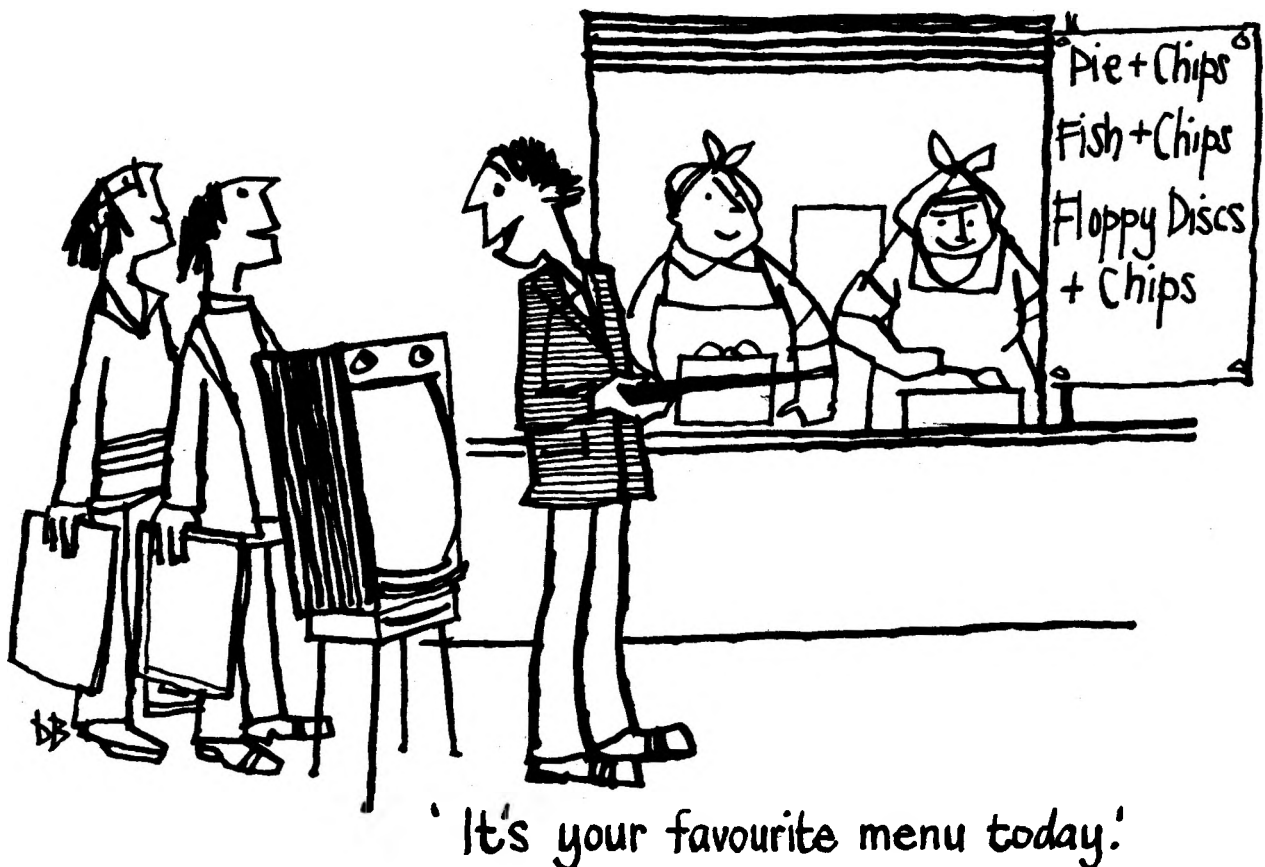
We discussed all these points. However, one factor seemed to point the direction in which we as a staff would progress - the comment from the staff and the children alike: "When can we borrow the micro again?"

Closings

As a staff we have decided to purchase our own microcomputer, using P.T.A. funds. The maths adviser informed us of other local primary schools which were acquiring machines through the L.E.A. If we selected compatible hardware, we could form a user's group and a profitable exchange of programs and experience could follow. Accordingly, we decided on the "Newbrain", and hope to have it in school by May.

However, May seems too long to wait - so we have arranged with Newman to borrow their micro again!

R. Butcher
(Deputy Head, Rounds Green Junior School, Oldbury)



ACT NOW!

I received a letter recently from an Inspector of Schools who wrote:

"....we are currently putting the primary computing work to a very low priority. The reasons for this are largely lack of finance.... linked with this problem of finances is the possible lack of suitable hardware and indeed software. In a few years one may see a good deal of involvement of computing equipment with very young children in their basic numeracy and literacy work."

In the current economic climate such a stance is perhaps predictable. It may be a few years yet before microcomputers become a regular feature in primary schools. However, the point I wish to make is that we must not waste the interim period, but take positive steps to prepare for the day when the hardware arrives. A planned programme of action could include some of the following steps:-

- 1) to learn about the broad principles of computing - and some of its jargon! (A floppy disc really isn't a spinal vertebra turned to putty);
- 2) to examine those areas of the curriculum which lend themselves to applications of the computer. This could lead to a draft programming specification (see example at end of article);
- 3) to discuss how to integrate the microcomputer into the school curriculum;
- 4) to consider different teaching strategies;
- 5) to ensure that the teaching environment steers the development of software and not vice-versa;
- 6) to arrange for demonstrations and/or loan of equipment (West Midland teachers can do so through Newman College);

- 7) to make contacts with other interested teachers;
- 8) OPTIONAL - to learn BASIC programming - this will never be a prerequisite of using a microcomputer, but for those interested teachers with access to hardware, now is the time to start. Practice makes perfect!

Every primary school could consider these aims without the necessity of spending money on hardware. You may need help and guidance and in this respect I suggest you approach:

local authority advisers to put on courses,
workshops and demonstrations (for staff);
local colleges to make their hardware facilities available;
a local comprehensive to give help and advice, and to stage demonstrations and workshops (but this time for the primary school pupils).

An important point here is that most primary schools don't necessarily need programming courses yet, but do need "awareness" courses. This will lead to an informed choice of suitable hardware later, and prepare the ground for using it sensibly. Find out the implications of the new technology now, the potential, and how to exploit the machine to the full. Above all we need to make sure we are guiding development as we wish to see it, and not being dictated to - but this presumes a certain fundamental understanding. Hence the need for action now.


Roger Keeling.

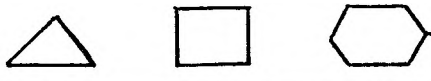
PROBLEM SPECIFICATION On the next page is an example to show some possibilities. Many variations are possible, and special features can be built in to suit the teacher's needs and intended uses. Please send in ideas for programs.

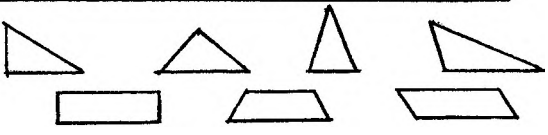
Aim: To provide practice in matching for 4 or 5 year olds at graded levels of difficulty.

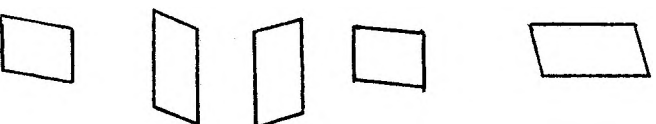
Use of the keyboard to be kept to a minimum; only necessary to use 'return' and 'space bar'.

Levels of Difficulty:

- 1) topologically different- 

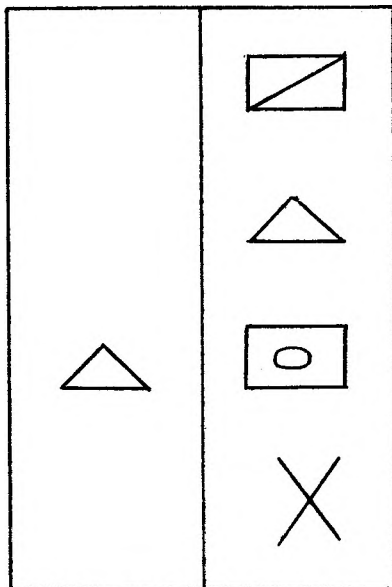
- 2) geometrically different- 

- 3) different triangles or quadrilaterals- 

- 4) mirror reflections of a shape- 

The program automatically moves up one level of difficulty in response to 5 consecutive correct answers.

Display:



These 4 figures are drawn automatically. One is then copied at random onto the left hand side of the screen. The arrow starts at the top and moves down in response to "RETURN" (screen wraparound). When he thinks there is a match, the child presses the space bar and the selected figure moves to overlay the original and indicates whether or not they match. If it is correct a happy face is displayed followed by a new problem. If it is wrong a sad face appears and the problem is repeated.

"ESCAPE" will terminate the program at any stage. At each new level of difficulty an example is displayed.

RK

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 all 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
```

Commas separate the phrases, 0 terminates the sentence, and 1 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 1230
130 DIM A$(20),B$(10),S$(20),R(20),S(20),B(10):RANDOMIZE:?"CHR$(12):RT=0
140 GO$="123456789":SP$=""
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)="Z" THEN RESTORE:GOTO 210
250 GOTO 220
260 NS=1:B$(1)=LEFT$(GO$,NN)
270 NS=NS+1:READ B$(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 20,24,"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$(0)
390 IF ASC(I$)=13 THEN 600
400 IF ASC(I$)=27 AND G$<>"" THEN ?CHR$(12):GOSUB 1230:GOTO 360
410 IF LEN(G$)=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$(R2):A$(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$(ZZ$,1,PZ-1)+CHR$(48+NO)+MID$(ZZ$,1,ZL-PZ)
660 PLOT 2*TL,PP,ZZ$:PLOT 2*TL,PP-3,U$:TL=TL+ZL+2
670 RETURN
680 REM ***** GUESSER *****
690 ?CHR$(12)
700 IF VAL(G$)=0 THEN ?"A NUMBER BE.G. ";LEFT$(GO$,NN):" ";:GOTO 1110
710 IF LEN(G$)<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=L
750 NEXT U
760 NEXT L
770 IF R=1 THEN ?"DON'T REPEAT ANY PHRASES ! ";:GOTO 1110
780 CT=0:NV=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$(12):WT=WT+1
870 PLOT 0,30,63:LINE 78,30:LINE 0,58:LINE 0,0:LINE 78,0:LINE 78,30
880 IF WT<3 THEN 1020
890 UU=GET(100):GRAPH 1:FF=0:?"CHR$(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(0):?"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$(12)
1140 FOR U=1 TO 11
1150 L=(U/2-INT(U/2))*12+192
1160 PLOT 0,30,L:LINE 78,30:LINE 78,58:LINE 0,58:LINE 0,1:LINE 78,1:LINE 78,30
1170 NEXT U
1180 FOR U=1 TO 5:UU=GET(20):GRAPH 2:UU=GET(20):GRAPH 3:NEXT U:UU=GET(300)
1190 GRAPH 1:PLOT 15,24,"YES, THAT IS ONE ANSWER !":UU=GET(300)
1200 GRAPH 1:PLOT 0,24,"CAN YOU SEE ANY OTHER ANSWERS ?"
1210 RT=RT+1:UU=GET(400):GOTO 340
1220 GOTO 340
1230 NZ=NZ
1240 NEXT BL
1250 REM ***** DATA *****
1260 ?CHR$(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

```

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 trying 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 W\$, A\$.

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.

Microelectronics and Computers
in Education

You may have heard of the £9,000,000 that the Department of Education and Science has put aside in order to encourage the study of Microelectronics and Computers in schools and colleges. One Project which has received a grant from the D.E.S. is the Microelectronics and Computers in Education (MACE) Project which is being run in the West Midlands.

The Project is supported by the eleven Local Authorities in the West Midlands region. They are: Birmingham, Coventry, Dudley, Hereford and Worcester, Sandwell, Shropshire, Staffordshire, Solihull, Walsall, Warwickshire and Wolverhampton. Each of these Authorities has seconded one or two teachers to the Project for the academic year 1980/81, making fifteen teachers in all. The intention of the Project is to train these fifteen teachers so that they may be able to help in the development of Computer Education in the years to come. The Project is based in Birmingham so that it can make use of the expertise available at both Birmingham University and the Birmingham Educational Computing Centre (BECC).

Most of the Authorities involved have decided that the secondary sector takes priority, and so they have seconded secondary teachers. However, Birmingham has a well organised support service for Secondary Computing at BECC, and consequently has been able to look at the Primary sector. They have seconded a Primary teacher to the Project. I am extremely pleased to say that I am the Primary teacher they seconded!

In order to have "hands on" computing experience, we have been equipped with an RML 380Z with 56K RAM, High Resolution Graphics, twin double sided disc drives, and a generous supply of discs! It is hoped that these machines become the basis of a network system for downloading software from a mainframe computer to a local centre. In short, it could make it easy for anyone to get programs!

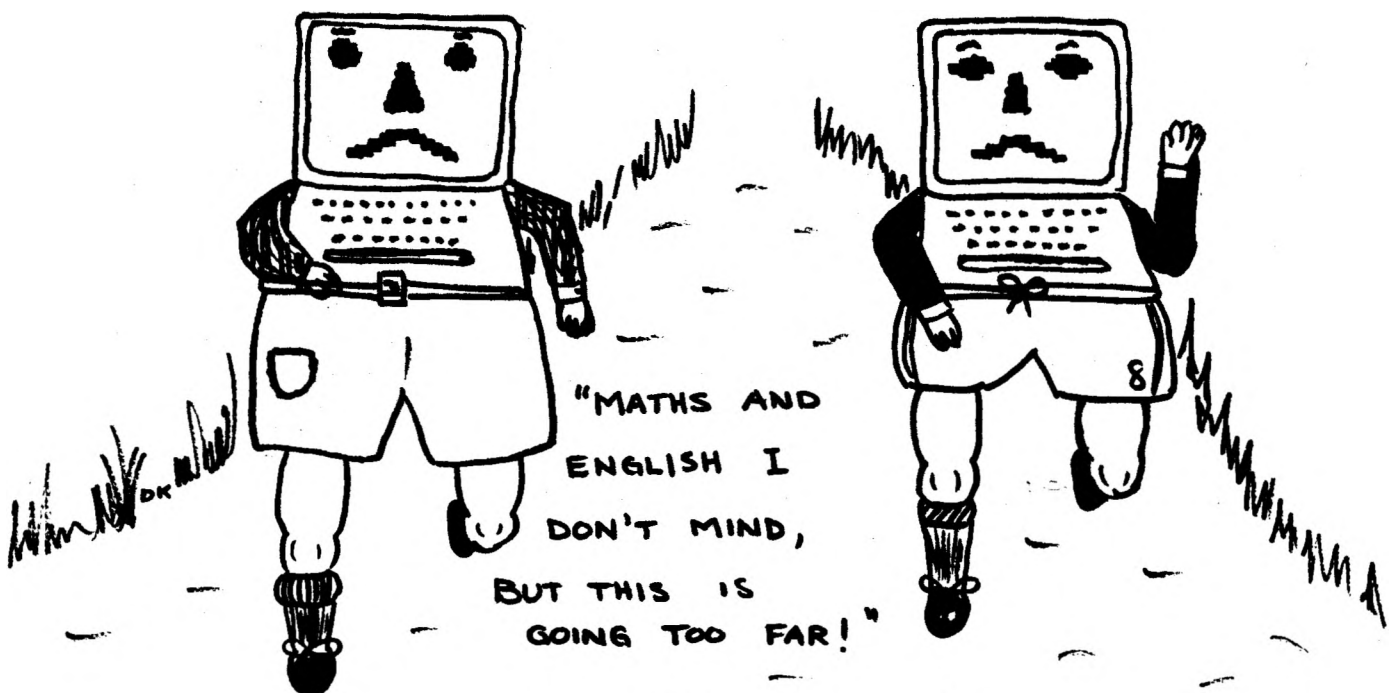
My speciality on the Project is, of course, Primary Computing. I have been trying to find other Projects around the country which are also involved in this field. This work is going to last a long time! There seems to be a great deal of interest at present, although I feel that someone, somewhere, must coordinate our efforts so that we can learn from each other rather than re-inventing the wheel. (How many tables-testing programs are there around the country?)

The MACE project still has six months to run, and I am sure that during that time we will have learnt a great deal, and maybe we will be able to see where our efforts ought to go in order to make the best use of the amazing machine which the electronics industry has made available to us!

Our thanks ought to go to Newman College for opening up some lines of communication for us!

Colin Watkins
MACE Project
Birmingham Educational Computing Centre.

COMPUTER TRAINING!



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.

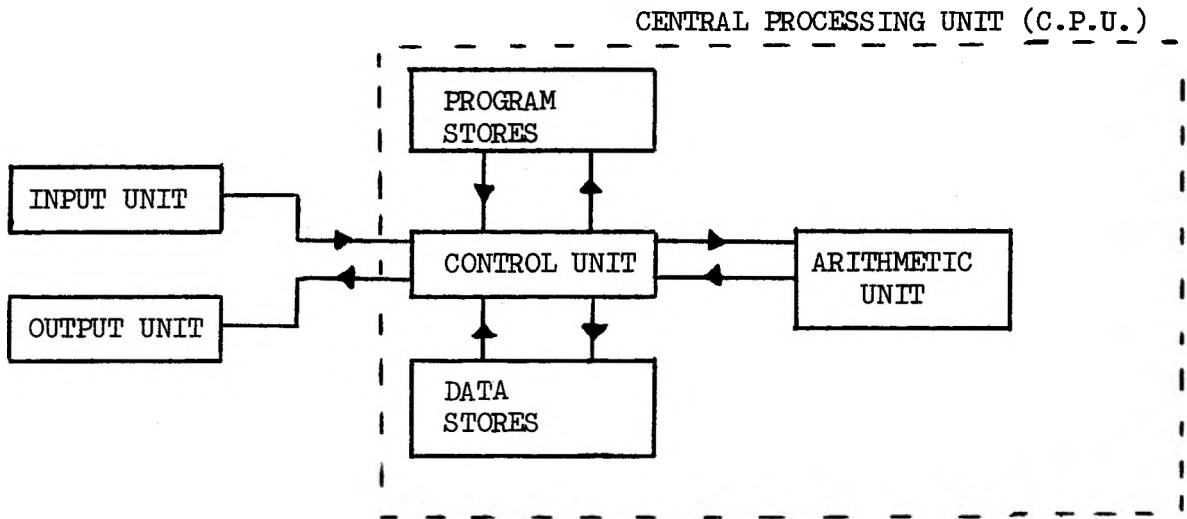
Dawn: (her eyes alight with excitement): Please tell us how a computer works, Miss.

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)

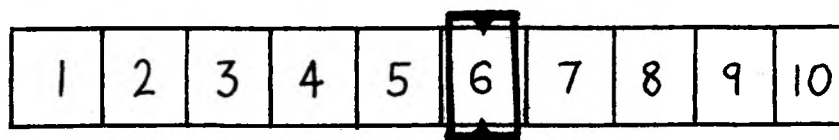
Miss J: A computer is made up of several parts and I have drawn this diagram showing how they are connected. The arrows indicate in which direction the information can pass. (She shows the children a large sheet containing the diagram below).



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.

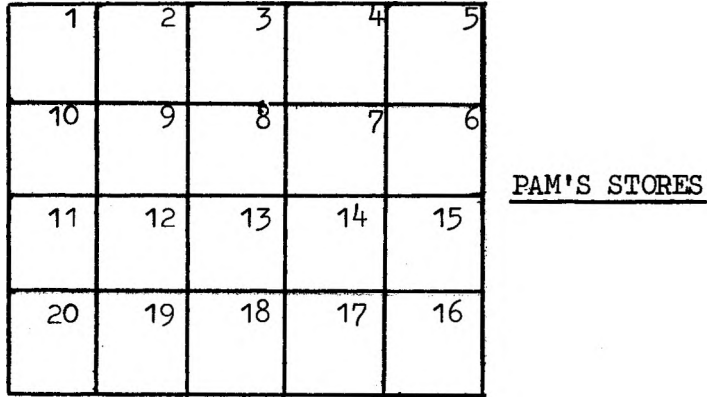
She allocates the items as follows:

- (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

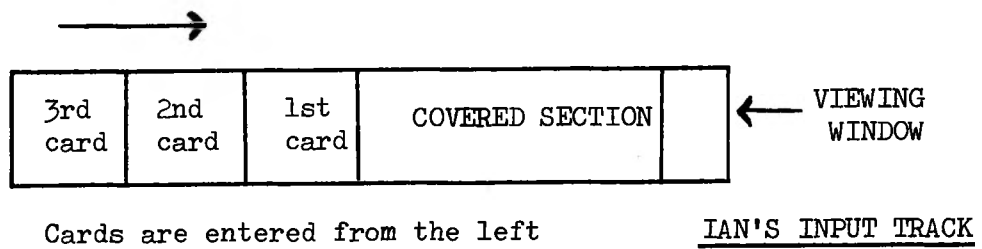
- (b) Pam, who is in charge of the Program Stores, is given a set of storage compartments which can be made from eggboxes, matchboxes or other suitable material. They are coloured blue to match the Program Instruction Cards, which will be stored in them. The individual stores are given numbers to identify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



- (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.



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.

```

```

graph TD
    L6[6 IF C = 3 THEN GO TO 8] --> L8[8 LET E = B/3]
    L7[7 GO TO 3] --> L3[3 INPUT A]
    L8 --> L3

```

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

```

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.

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

```

A
SOFTWARE
REVIEW

ADDICTIVE	
SADISTIC	MATHS
MURDEROUS	GAMES
and DIABOLICAL	

It is often said that children and computers are difficult to part. Is it true, whatever the quality of the software, that children will push buttons in a computer assisted learning situation or an education game ?? Yes, maybe - but children are soon able to discriminate. Some programs offer that bit more, command more attention, and the children come back again and again.

We recently came across a suite of 4 programs from Premier Publications, (12 Kingscote Road, Addiscombe, Croydon) which did have that special something. They were entitled:-

ADDICTIVE ADDITION
SADISTIC SUBTRACTION
MURDEROUS MULTIPLICATION
DIABOLICAL DIVISION

Unlike most educational software these programs do not suffer from being designed for one machine. This suite of programs is available for 4 different machines - Tandy TRS-80 (level 2); Compukit UK101; Ohio Superboard; and Video Genie.

The programs (written by John Hooker, a teacher) have very good clear screen layout with an uncluttered display. Instructions are included within the program and keys for controlling the computer are well documented. I particularly liked the excellent scoreboard facilities. The children did too - they liked to see an updated picture of their results rather than wait till the end of their turn. Another nice touch is the offer of different levels of difficulty and the option of being able to move up and down between these within the program by keying in symbols. I did sometimes find that children were automatically moved up and down levels when they should have stayed for more experience in that level.

Let's have a look at what each game offers.

SADISTIC SUBTRACTION

This game stands out from the rest of the pack. It adds a number of twists to an important basic skill and keeps you thinking. You are shown a number from which you have to subtract a string of numbers. These appear in turn in the MINUS BOX.

The COUNT BOX tells you how many to expect. But the twist comes with the NO NO BOX ! For if the number appearing here is the same as in the MINUS BOX you leave it alone. The children really have to concentrate and watch the screen display carefully to ensure the computer doesn't catch them. It kept a group's attention for a considerable period and I felt the drill that the program provided within the game to be well reinforced.

ADDICTIVE ADDITION

This program has two modes of operation. In Game A the numbers you have to add appear on the screen together. In Game B the numbers are flashed on one at a time. Your ability to add numbers together quickly is certainly tested! ! Particularly as, if you get them right, the number series grows. Again it was a winner with the children both 8 - 10 year olds and the 10 - 12's. Great competition started to see who could build up to the longest sequence. It could become addictive, and could improve number bonds.

MURDEROUS MULTIPLICATION

There are many tables testers available, this one has a bit more. First you have to count the squares in each of two boxes on the screen, before they disappear. When the timer reaches zero, you then multiply the numbers in the two boxes together. A good variation on this basic game: two children work together - each counts one of the two boxes and they jointly work out the multiplication.

DIABOLICAL DIVISION

In this game you are given a division sum and a series of answers and have to choose the nearest right number. You are once again working against the count. A pleasing program but not as popular as the other two.

To sum up, these 4 maths games are very different from many I have seen. Staff and children gained from using them. They were fun, and made children of all abilities think in a new way about traditional four-rules maths. After a week with all sorts of right and wrong answers tried, they seemed bug-free, crash-proof and kid-proof. The pack, with clear teaching notes, is good value at £9.95 (including V.A.T.).

Dave Fatcher
(Head of Beaconsfield First and Middle School, Southall, Middlesex)

BOOK REVIEW

R. Jones, "Microcomputers: Their Uses in Primary Schools"
Council for Education Technology, £6 (111 pages)

Richard Jones has produced a very clear guide for those schools wishing to find out more about microcomputers. The first chapter describes the basic components of a computer system in everyday terms and this readable style is maintained throughout. Chapter 2 makes some very practical suggestions with respect to in-service training. Some chapters are not quite so useful. A schools survey is already out of date, such is the current rate of progress, and the software/applications side could benefit from one or two detailed case studies rather than summary titles which say very little. However, overall the report is well balanced, asks the right questions, and provides a useful source of information and addresses. To achieve the circulation it deserves, it could do with being half the price!

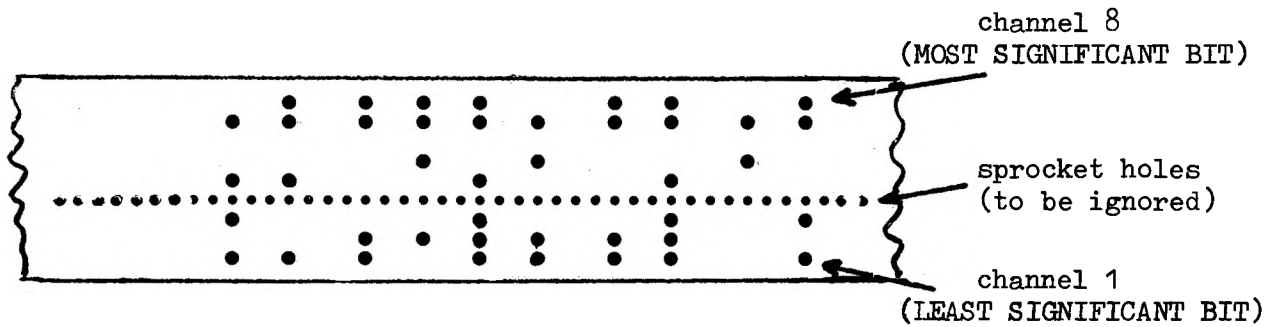
R.K.

A second useful publication, this time free, for anyone considering purchasing hardware is: "USPEC 32: A guide to the selection of microcomputers" from Council for Educational Technology, 3 Devonshire Street, London W1N 2BA.

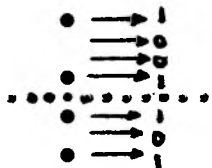
PUZZLE PAGE - a coding diversion

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.

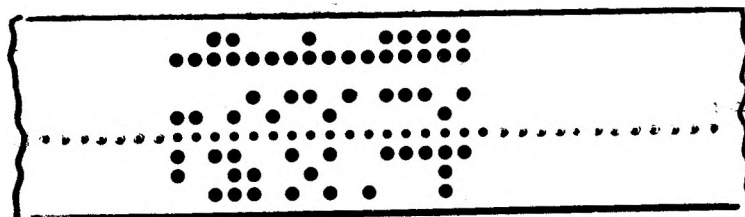


We can ignore the eighth channel which acts as a check against faulty data transmission. The standard paper tape code represents an A by the number 65, B = 66, .. Z = 90. A 'space' is represented by the number 32. Therefore to translate paper tape, write down each character as a binary number. A hole 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.



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

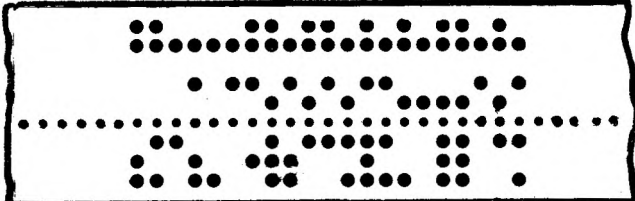
More Hints on Hardware

Through our in-service courses and correspondence arising from MICROSCOPE 1, we are hearing of primary schools which have the money available to buy a micro now, and the interest and enthusiasm of staff to support its acquisition. Schools are seeking guidance on which machine to buy. There is no clear cut answer, but the following points may be helpful.

I have already emphasised in the previous edition the importance of consultations with fellow primary schools, local secondary schools and LEA advisers. In many areas the authority will already have decided to standardise on a particular micro in the secondary sector. There are many advantages to be gained from choosing a machine from the same stable for the primary sector, not least of which is compatibility of programs and support from the more experienced secondary users. There is after all common ground between programs for upper juniors and slower secondary pupils.

For those authorities that have standardised on Research Machines 380Z, it may be wise to consider MINIM for the primary schools within the region. This is Research Machines new 'mini station'. It is a 'stand alone' computer with a cassette recorder for backing store. For the technically-minded, it is based on the Z80A processor and includes 32K RAM, 8K ROM (which does not yet include BASIC, but hopefully will in the future - i.e. - BASIC has to be loaded from cassette), keyboard, low resolution graphics, 40 character mode video and UHF outputs (you need to provide the TV set as an extra or find a parent throwing one out), cassette recorder, parallel and serial I/O parts. Future options include extra memory and high resolution colour graphics. At present the selling price is £550 with first deliveries due in September. As with the 380Z the price seems a little high, but Research Machines' track record may justify this extra expense. It is a British company, with a large following for the 380Z from secondary users, and a positive

* * * * *



Anagram of a television programme???

interest in the education market. I hope the MINIM will serve the primary area as well as the 380Z does the secondary area. Further details can be obtained from RML, P.O. Box 75, Oxford.

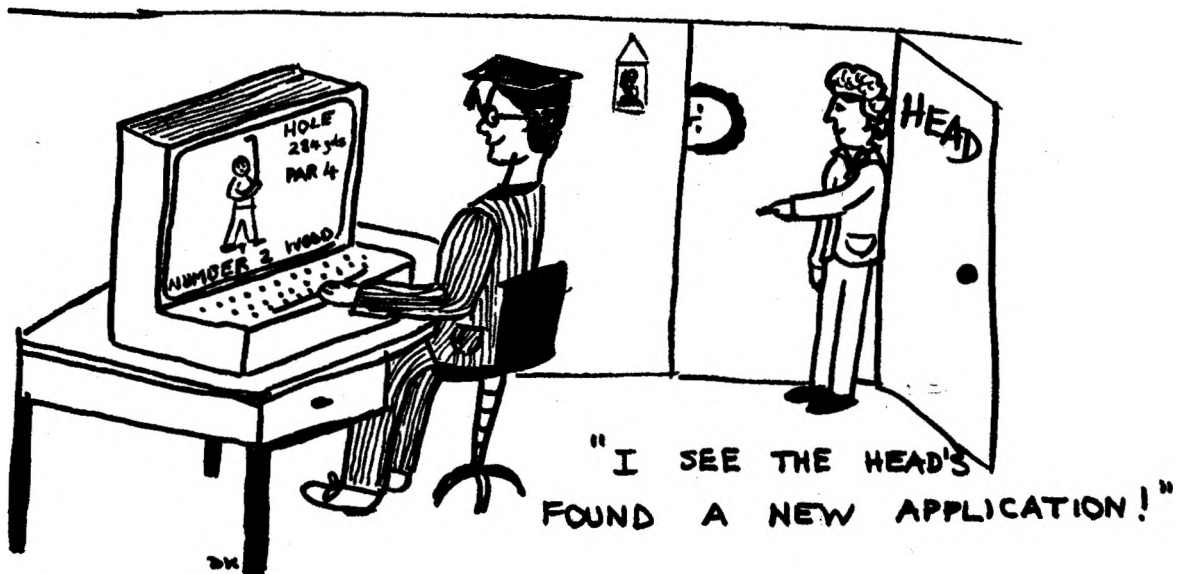
What else may be suitable for primary work? Those areas that have standardised on the Tandy TRS-80 could well be wise to use the same machine in the primary sector (or look at the Video Genie - a slightly cheaper but very similar micro). Walsall L.E.A. have already been brave enough to recommend the Tandy TRS-80 to its primary schools. This type of positive guidance, lacking in many other authorities, is precisely what the schools are wanting.

In a similar manner, secondary sectors using the PET could well be wise to recommend the same machine to its primary schools, although one must consider the VIC-20. This is PET's new 'baby' micro and will be out in about 4 months.

Some cheaper micros are now appearing, obviously with no proven record. The much awaited Newbrain still hasn't appeared, and we have heard a suggestion that unexpected problems have arisen - try to get information out of them if you can. Another two worthy of note are the Acorn Atom, with its non-standard BASIC, and the Transam Tuscan. It is early days to judge these machines, particularly as they still have to make inroads into the education field, but nevertheless they may have a valuable role to play.

In future issues we will continue to comment on new developments.

R.K



NEWMAN COLLEGE - COURSES FOR TEACHERS

A. MICROCOMPUTERS IN PRIMARY EDUCATION - Details from R. Keeling.

1. Microcomputers as a School Resource (Regional/D.E.S. Course)

Dates The course is of nine days duration -
5th, 6th, 7th, 14th, 20th May,
2nd, 9th, 10th, 11th June.

Target The course is designed for those teachers who wish to find out about the principles of microcomputers and how this new technology can be applied to their own teaching situation. Previous knowledge of computing is not a prerequisite. The course will cater for a group of primary teachers together with secondary teachers of biology, chemistry, geography and mathematics.

2. Microcomputer Workshop (Assessment and Development of Software)

April 30th, 4 p.m. - and monthly.

3. The next programming courses will be in the Autumn Term. There will probably be a beginners course and a continuation course to follow on from the present programming course.

B. DIPLOMA COURSES

1. Diploma in Science Education for Teachers in Primary Schools

Commencing September 1981, a course of two years (part-time), leading to a Diploma of the University of Birmingham. The course will enable teachers to develop an appreciation of the position of science in the Primary School. It emphasises the way in which science can integrate with and develop other areas of the curriculum.

Other Diploma courses are:

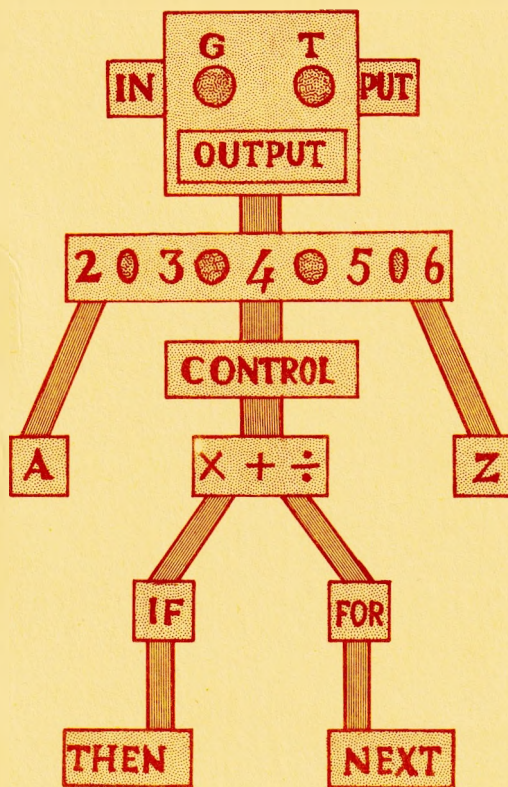
2. Diploma in Curriculum Development in Geography
3. Diploma in the Teaching of English
4. Diploma in Physical Education (5-13 years)

Application forms, obtainable from the Registrar, Newman College, should be returned before 30 May 1981. Tel. 021-476 1181.

C. IN-SERVICE B.ED

A handbook is available from the Registrar, or Miss J. Sugg will give details.

MICRO-



SCOPE