MICROCOMPUTERS IN EDUCATION

compiled by
Duncan Sledge

CET
Council for Educational Technology

MICROCOMPUTERS IN EDUCATION

A SELECTION OF INTRODUCTORY ARTICLES

selected and introduced by
Duncan Sledge

CET Council for Educational Technology

Produced and distributed by the
Council for Educational Technology
3 Devonshire Street, London W1N 2BA

ISBN 0 86184-010-0

First printed 1979

Printed in Great Britain by Imediaprint Ltd

CONTENTS

FOREWORD

This pack aims to give a general introduction to the subject of microcomputers in education. It was commissioned by the Council for Educational Technology in order to provide background reading for anyone considering using or purchasing a microcomputer, and who has had little or no previous experience of such equipment.

The pack comprises a selection of articles with a linking introduction. Among the aspects covered are the nature of the microcomputer, its management and potential uses, the types and compatability of equipment, purchase and running costs, various programming languages, and the availability of software including user-written programs. The introduction does not make recommendations, but concerns itself with raising issues and offering a discussion in which some relevant factors and opinions are presented. For more detailed information and guidance regarding purchase, attention is drawn to User Specification 32: A Guide to the Selection of Microcomputers, published by the Council for Educational Technology.

The articles are a personal selection by Duncan Sledge, made after a comprehensive survey of publications during the past two years. His introduction reflects his experience, in particular his supervision of the Durham Microcomputers Project - a school-based investigation (established by Durham Educational Committee and now in its third year) into the use of the microcomputer in secondary education.

CET gratefully acknowledges Mr Sledge's assistance in producing this pack and the cooperation of Durham County Education Authority and the Faculty of Education, New College Durham, for agreeing to his involvement.

V W Thompson
CET Programme Manager
November 1979

EDITORIAL INTRODUCTION

'Non-expert' seems at first to be a most unflattering term, but on reflection embraces so many levels of readership that it has been used throughout this introduction. Whilst it must be clear that this pack will not convert the reader into an 'expert' (whatever that may be) it is hoped that it will provide the 'non-expert' with an introductory ticket to the arena of this rather complex circus - microcomputers in education.

A variety of questions could and should occur to the 'non-expert' who wishes to know more about microcomputers and their use in education.

To begin with, he/she may be concerned to know what a microcomputer consists of and what it can do, as well as the range of choice available and the likely initial and ongoing costs.

What is the significance of microcomputers for schools, how could they be used and managed and what comitment of time, effort and money would be required for various types of application?

Are there any possible problems with microcomputers that are not immediately apparent to the layman?

Although exhaustive answers to these questions would not be possible in the confines of this pack, its intention is to provide some thought-provoking information, opinions and ideas in these areas.

But before scrutinizing the microcomputer itself, a brief history of the relationship between schools and computers may halp to set the scene and put more recent developments into perspective.

Some years ago, the only computers available were large and expensive (often referred to as 'main-frames') and they were normally sited in town halls, local industry, large colleges and universities. Those that afforded any kind of contact with schools usually provided a visiting service for school parties and/or a 'batch-processing' service for pupils' computer programs. As the name implies, this latter service involved collecting pupil programs together in batches and sending them off to the main-frame installation for processing, after which the results would duly be sent back to the school .

Programs could be sent by post, by courier, or delivered personally by teachers or friends, but the essential feature of such an arrangement was usually that it was self-defeating because of the delays involved. After all, the speed of the computer can hardly be demonstrated by an arrangement that takes days, if not weeks, to produce the results!

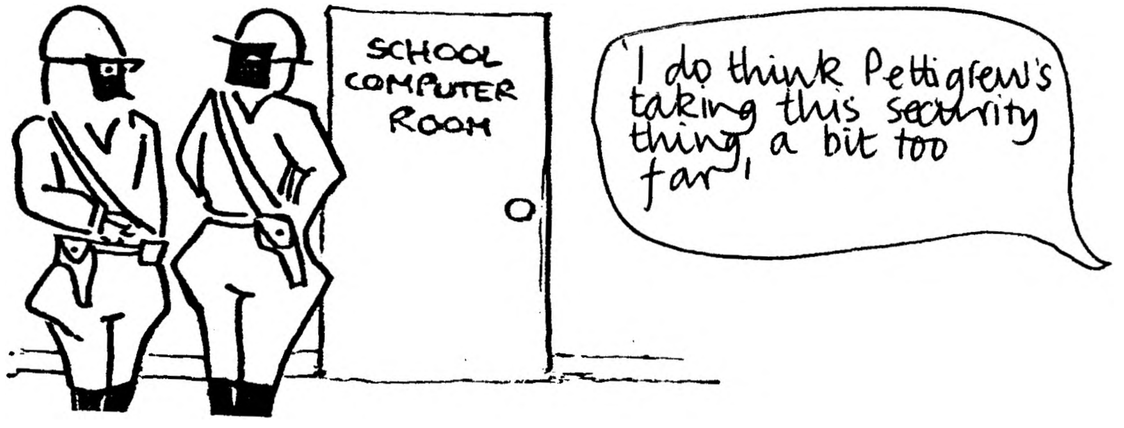
Although it was, and still is the most economical and practicable way to process large numbers of programs, a batch-processing service as the one and only computing facility available to a school was generally felt to be unsatisfactory and insufficient.

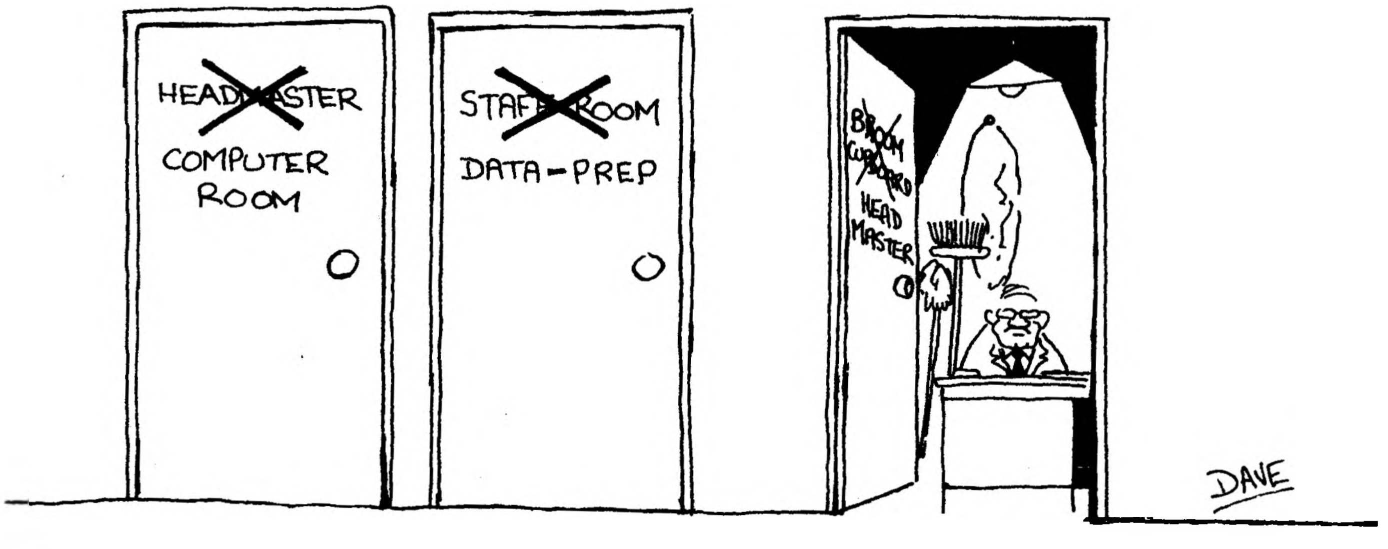
The development of 'multi-access' systems (sometimes called 'time-sharing' systems) was a significant departure from traditional batch-processing, since it allowed several users to have apparently simultaneous access to the computer via several terminals (eg, teletypewriters or visual display units), and the users could actually converse with the machine themselves (through the keyboard) in what became known as 'conversational' or 'interactive' mode .This meant that visiting school pupils no longer had to be content with viewing a main-frame computer from a distance, but could instead sit down at terminals and gain 'hands-on', personal experience of the computer's power, speed and accuracy.

More importantly, these terminals did not need to be in the immediate vicinity of the computer, but could, in fact, be sited in the schools themselves, provided that telephone links were available to connect them. This new possibility brought about the introduction of 'remote terminals' into a large number of schools throughout the country, especially in inner city areas.

The advent of the 'minicomputer' - smaller and much cheaper than traditional main-frames - drastically reduced the costs of multi-access computer systems that could support large numbers of terminals. This in turn meant that many smaller local colleges (but not really schools) could now afford their own computer, and, consequently, more schools were able to acquire remote terminal links and so have local computing power on tap.

There are, in fact, many advantages in being a remote computer user, not least of which is that the worry of running and maintaining the computer can be left to someone else. Virtually unlimited storage capacity, high speed, and a large library of available ready-written programs are further common features.


SCHOOL COMPUTER ROOM
I do think Pettigrew's taking this security thing a bit too far!


~~HEADMASTER~~
COMPUTER ROOM
~~STAFFROOM~~
DATA-PREP
~~BROOM CUPBOARD~~
HEAD MASTER
DAVE

But there are distinct disadvantages to remote terminals, and it is the weight of these that probably explains why most schools in this country still have no 'in-house'terminal provision.

For example, prohibitive ongoing Post Office costs (especially over long distances), the rationing of computing time (leading to bottlenecks in demand), the vagaries of PO telephone lines, reliance upon a central computer installation that is beyond the school's control or jurisdiction and considerable initial financial outlay are just some of the discouraging aspects of remote terminals.

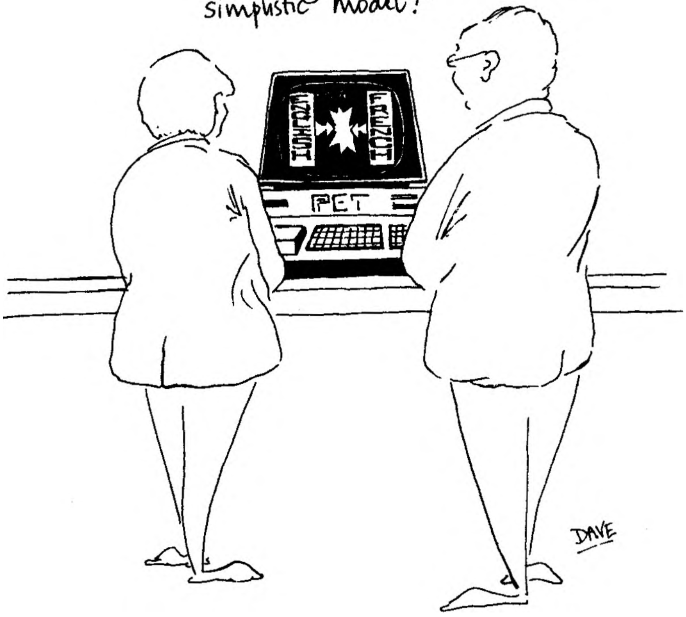
It is hardly surprising, then, that when microcomputers first appeared in the marketplace at a price that many schools could at least contemplate, the country's population of computer studies teachers and other interested parties were delighted at the prospect.

At last, here was a machine that was not just a peripheral device like the terminal, but an entire self-contained computer, all in the space of a desk-top. Besides not suffering from any of the disadvantages of batch-processing or remote terminals, it even had additional features not previously made available by the local main-frame or minicomputer. Graphics, light pens, fast printers and voice recognition are examples of the exciting extra facilities offered by various microcomputers. Moreover, their compactness and portability gave rise to hopes that they might also be used in a wide range of school subjects and activities outside and beyond computer studies classes.

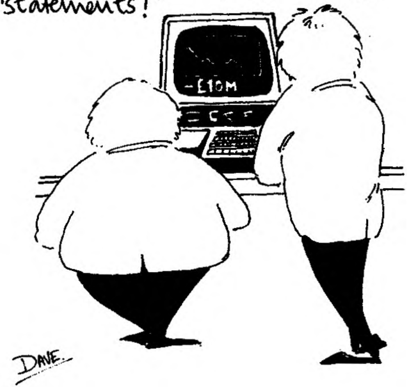
Anyone who has witnessed this sudden rise in hopes and expectations and who has noted the numerous predictions about the likely impact on future society is likely to wonder what manner of miracle is a microcomputer.

Perhaps as a first priority, our non-expert needs to become acquainted with the nature and composition of microcomputers in order to appreciate later discussions. In venturing into this area, we are forced to enter the realm of high electronic technology with its attendant mystifying jargon and, no doubt, some would question the wisdom of this excursion when for the purposes of most people it would probably suffice to describe the microcomputer as 'the black box'.

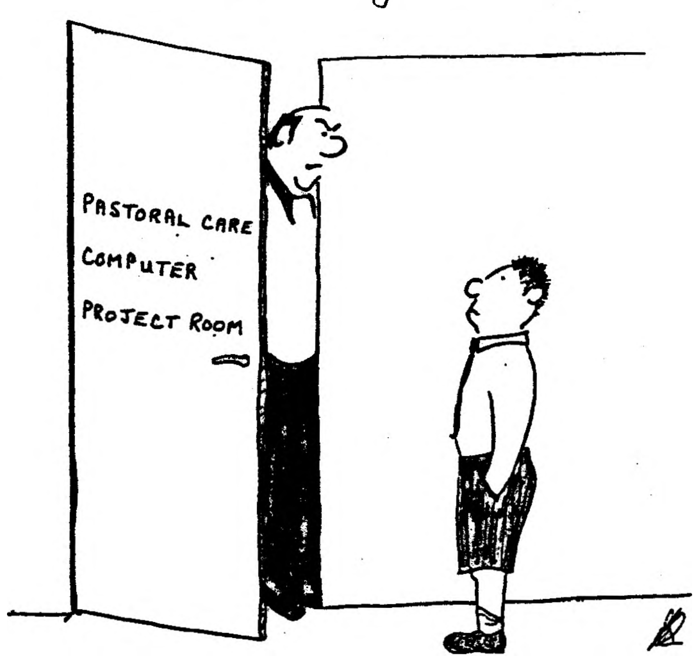
'Something tells me this is a rather simplistic model!'



'I never dreamed a simulation of the world's economy was possible in just 23 programming statements!'



'Look son how many times do I have to tell you? We are 'too busy' to talk to you right now - come back in a few weeks.'



However, many people seem to grow in confidence (and commitment) once they have gained some insight, however limited, into the nature of the phenomenon that they are dealing with. Moreover, there seems to be sufficient confusion in the non-expert camp over the terms 'microprocessor', 'microcomputer', and 'minicomputer' (not to mention a host of other terms) to warrant this excursion in its own right. The reader who seeks an effortless and painless conversion from near or complete ignorance of the make-up of the 'micro' to a state of reasonably informed grace may well find article A, 'A practical introduction', to be of value.

Indeed, this article goes much further than merely describing the microcomputer's innards, since it also examines the various items pf hardware eqipment that can make up a microcomputer 'system' or 'configuration' and discusses to some extent what these machines can do. It advises the reader on features to look for, and even on what to buy for particular purposes, although it must be noted that its intended audience was far wider than just the educational world.

The question of which microcomputer make, model and configuration would best suit the needs of schools is difficult to answer. Sadly, many of those who take part in this debate seem to forget that a chef usually decides exactly what he is going to cook before he gathers the ingredients and utensils, otherwise a culinary disaster could occur. In the same way, without knowing precisely what a school intends to do with a computer, it is not really possible to specify the right machine to buy.

In this respect, the principles of the Berkshire Working Party as laid out in article B, 'A computer for schools', while being an excellent effort to construct positive guidelines, cannot be regarded as general principles for all schools because they presuppose certain kinds of use for the computer. Some articles and arguments making out a case for one machine or another seem to overlook completely this fundamental question of intended use.

A further questionable supposition is that the idea of gradual expansion from a minimal equipment system to a more powerful system is either desirable or acceptable.

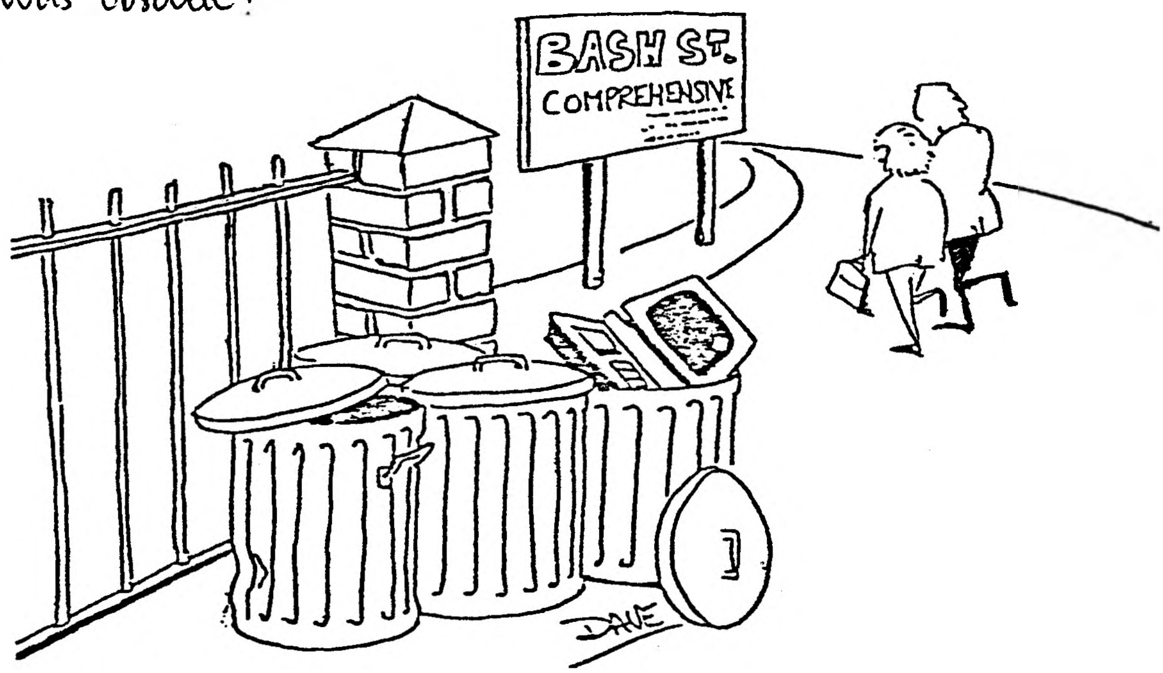
The idea of repeatedly interrupting development to go out and buy the next required ingredient would seem inefficient and unlikely to inspire good results. This kind of disruption could easily occur where a school might purchase a minimal system and subject it to regular expansion by adding on extra items as money becomes available, or as the need arises, especially if this would entail sending the machine away, possibly to the other end of the country, for this 'upgrading'. Disruption may be caused by the temporary loss of equipment (possibly at vital times), the administration effort involved in each enhancement, or by the necessity to spend a great deal of time modifying existing programs to take best advantage of the enhancement. Perhaps it would be better to ignore talk of minimal systems and instead concentrate on 'desired' or 'required' systems. This may mean, sadly, that many schools may define required systems that they can not afford, but then, if that is the case, such schools will surely be wasting their money if they buy anything less.

The dialogue between the Berkshire Working Party and a microcomputer manufacturer did, however, mark a unique and valuable collaboration between industry and education in this field, and the resulting microcomputer has already been adopted by some education authorities.

The adoption of one or more standard microcomputers within an area, a county, or the country (that is, standardization) is yet another interesting talking point. Unfortunately most microcomputer makes are incompatible with each other, in that programs produced one one are not readily transferable to another. Transfer of some programs can be effected, but only with significant effort. For the ease of inter-school cooperation, program exchange, coordination of development, in-service training, and maintenance, the attraction (indeed, necessity) of standardization within an area is obvious. A 'buy-what-you-like' policy amongst institutions in an education authority could presumably be catastrophic in the long term. On the other hand, the monopolistic dangers of an official or even unofficial national standard microcomputer for schools should be equally obvious. Currently, however, this is hardly a major problem, since the market is young, competitive, and full of vigour, and seems likely to present fresh options and new developments for some time to come.

All the discussions so far have assumed that the microcomputer is, in fact, viable as a piece of school equipment and that, once purchased, there is an end to the problems. But some would disagree, and article C, 'Microcomputer systems: revolution or revolt?', points out a few reasons for dissent. In it, Michael Coleman rightly identifies some of the problems that will have to be faced - the rising price of software, the disruption caused by system failure, disruption caused by program failure, the eventual need for a full-time school 'computer manager', and the ultimate inadequacy of a minimal system, if purchased.

'And by the time we'd got it to print out the average of ten numbers, the thing was obsolete!'



All these issues will need to be looked at very carefully, and are, for the most part, aspects of the management of school microcomputers - a much-neglected subject.

The proposed solution (mobile computer units) to these problems may suit the Polytechnic in question, but would seem unlikely to be popular with schools, since each school would only have irregular access to the computing facilities.

In any case, few people are likely to enthuse over the implications of such a solution. The need to spread demand evenly by synchronizing school timetables is an awesome prospect and would anyway cater only for computer studies classes. A whole fleet of vans could service only ten schools per week (assuming one visit per school per week of a half-day duration). The cost of these vans, garaging, van maintenance, drivers' wages, standby vans and standby drivers could be phenomenal. Add to this the fact that the constant movement would inevitably take its toll on the microcomputer equipment, and the original set of problems almost seem to become positively inviting by comparison.

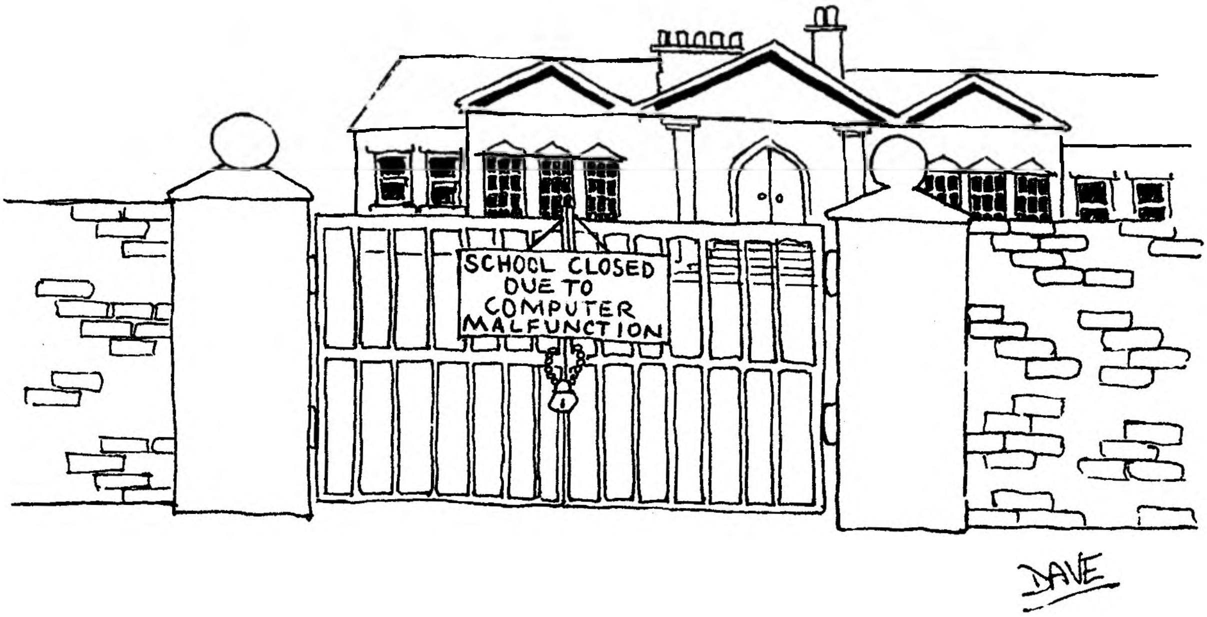
This is not to say that there could be no place for or value in mobile computer units. There may be scope for such units organized on a regional basis to tour schools giving prearranged displays and demonstrations to teachers of all disciplines of the type of educational software produced in their subjects. Such an approach stands a fair chance of getting through to the majority of the teaching force, as opposed to traditional in-service courses which tend only to reach a small minority, and more often than not constitute preaching to the converted. Mobile computer units could, perhaps, play a useful role in dissemination in this way.

But are school-based machines really likely to encounter significant difficulties? There is so much speculative writing on microcomputers, and so many writers are eager to tell of their successes while conveniently omitting to mention the headaches, that it is almost refreshing to read article D, 'The real world' - editorial, in which the experiences related are both lamentable and actual. This catalogue of failures bears out many of the points made by Michael Coleman in the previous article, and is a cautionary tale for any prospective buyer of microcomputers, educational or otherwise.

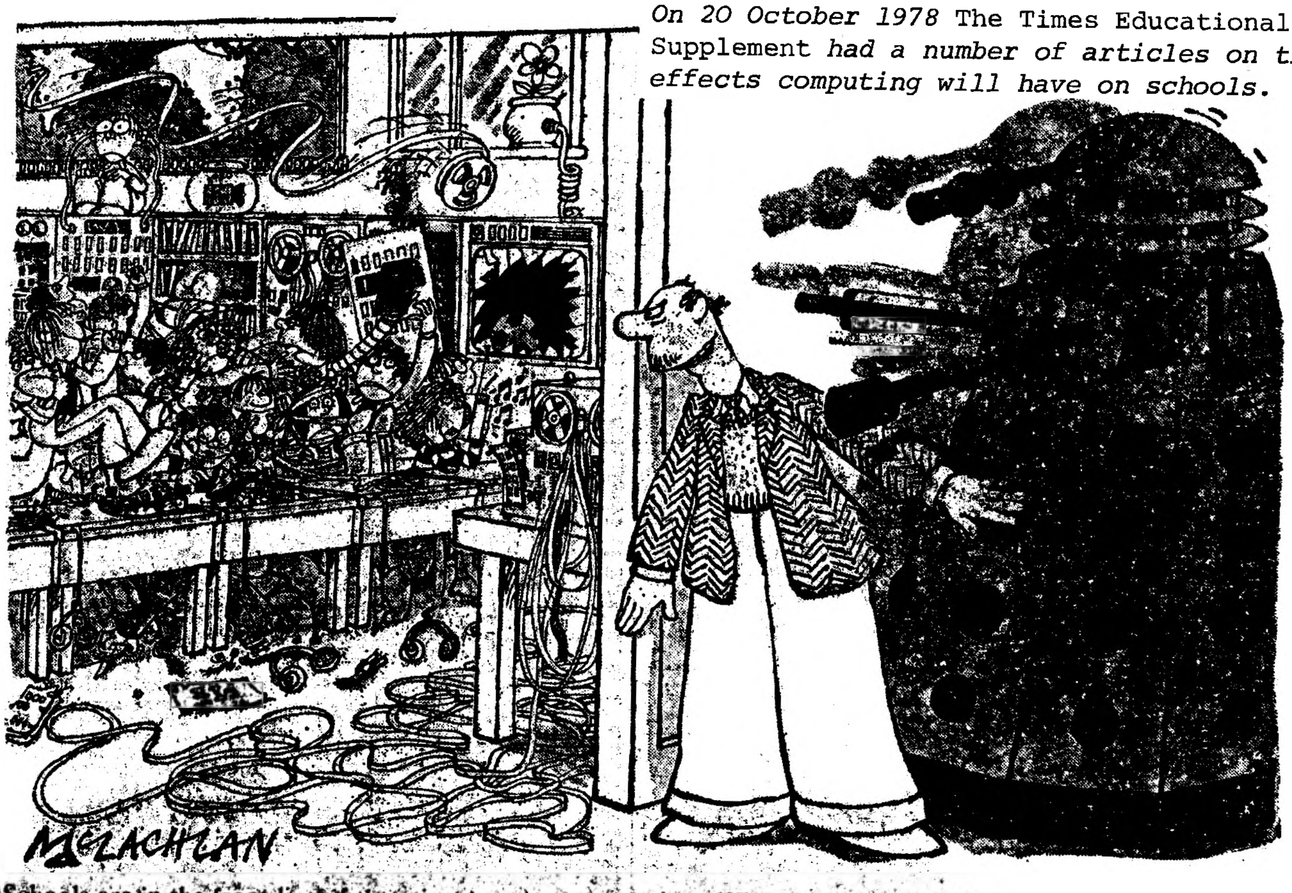
Two final points on the management of school microcomputers may deserve special emphasis. Firstly, good organization is important to the beneficial exploitation of these machines in schools or anywhere else. A computer of any kind is a very sophisticated resource, and a school is a very complex environment. Merely to throw the two together without making any further supportive or organizational provision could be a sad and, possibly, damaging mistake. This clearly does not happen in industry, and would be regrettable in education. (The editor would be interested to hear of any unmanned, unplanned computer installations in industry or commerce!)

The second point is that adequate and suitable accommodation would seem to be essential if the microcomputer is to be used satisfactorily as a broadly based educational tool, regarded by all staff as a general school resource. The fact that a microcomputer system is small enough to fit into a broom cupboard does not necessarily mean that that is where it belongs. The word 'microcomputer' need not imply 'microaccommodation' and clearly there are already enough barriers to break down in introducing the widespread use of computers in schools without the need to introduce any more - that is, by discouraging access through inadequate provision of space.

Having emerged from the issues, considerations and characteristics of the microcomputer hardware jungle, the non-expert's curiosity might next turn to the thicket of programming languages.



On 20 October 1978 The Times Educational Supplement *had a number of articles on the effects computing will have on schools.*



Schools are in the front line of the microprocessor revolution. What can they expect?

15

Just as the world has many natural languages and many dialects within each, so too, for better or worse, the world of computing has many computer langeages and again many dialects.

To begin with, any individual make and model of computer will have its own 'native tongue' called its 'machine code' and usually 'assembly language' (which will be very close to the machine code in nature, but slightly more intelligible to humans). Generally, the machine code and assembly language of one make or model of computer cannot function on other makes or models, because they are tied by design to the internal physical structure of one particular machine.

Such languages are termed 'low-level' and are complex and tedious to work with (novice programmers should studiously avoid them), although they do offer, to those who can use them, the advantages of higher speed, more economical use of memory and closer control over all the computer's activities. In a school, only experienced programming enthusiasts and GCE A-Level Computer Science students would need or be able to use low-level languages.

High-level languages (like COBOL, FORTRAN, ALGOL, BASIC) on the other hand, are much more friendly and are universal in that they are largely independent of particular makes and models of machine. In theory, this means that transfer of programs from one type of computer to another should be easy, obscured only by minor differences in dialect.

Unlike low-level languages which are 'machine-orientated',that is, designed for a particular machine, high-level languages tend to be 'problem-orientated', that is, designed with a specific purpose in mind. Thus FORTRAN was intended primarily for scientific calculations, while, in contrast, COBOL was designed to manipulate the large volumes and complex structures of business files and records. BASIC was designed as an introductory language for beginners - easy to learn and easy to use.

In fact, there are scores of high-level languages to serve scores of purposes, and this accounts for the tremendous diversity of computer applications in the world at large. Yet, at the same time, it also accounts for a rather confusing, almost chaotic state of affairs - something akin to a United Nations assembly during an interpreters' strike. It is ironic to think that a technology/ industry that is so dependent upon creating logical and orderly systems as its bread-and-butter should be so disorderly in its own development - rather like a carpenter whose own front door is falling apart.

Fortunately, schools and other levels of education in this country have so far escaped this fate of Babel that is suffered by industry because BASIC has over the years become the adopted standard language within education.

Although this pack would not be an appropriate vehicle for a full-scale programming course, it may help the non-expert's feel for the subject to at least sample the flavour of programming. Article E, 'BASIC - the first steps', by Philip Couzens, provides an excellent opportunity for the uninitiated to do just this without risking indigestion.

The importance of BASIC as the computing *lingua franca* of education is that it has enabled through the years the development of some uniformity in educational software production, in-service training, and communication of ideas, materials, and teaching methods, locally and nationally. These are some of the benefits of all parties speaking the same language.

By a happy coincidence, BASIC has also been adopted by microcomputer manufacturers as a standard feature, and it was this, coupled with low prices, that originally caused microcomputers to be of such immediate interest to schools in the UK. With some microcomputers, other high-level languages are available, but these are generally less important to schools than is BASIC.

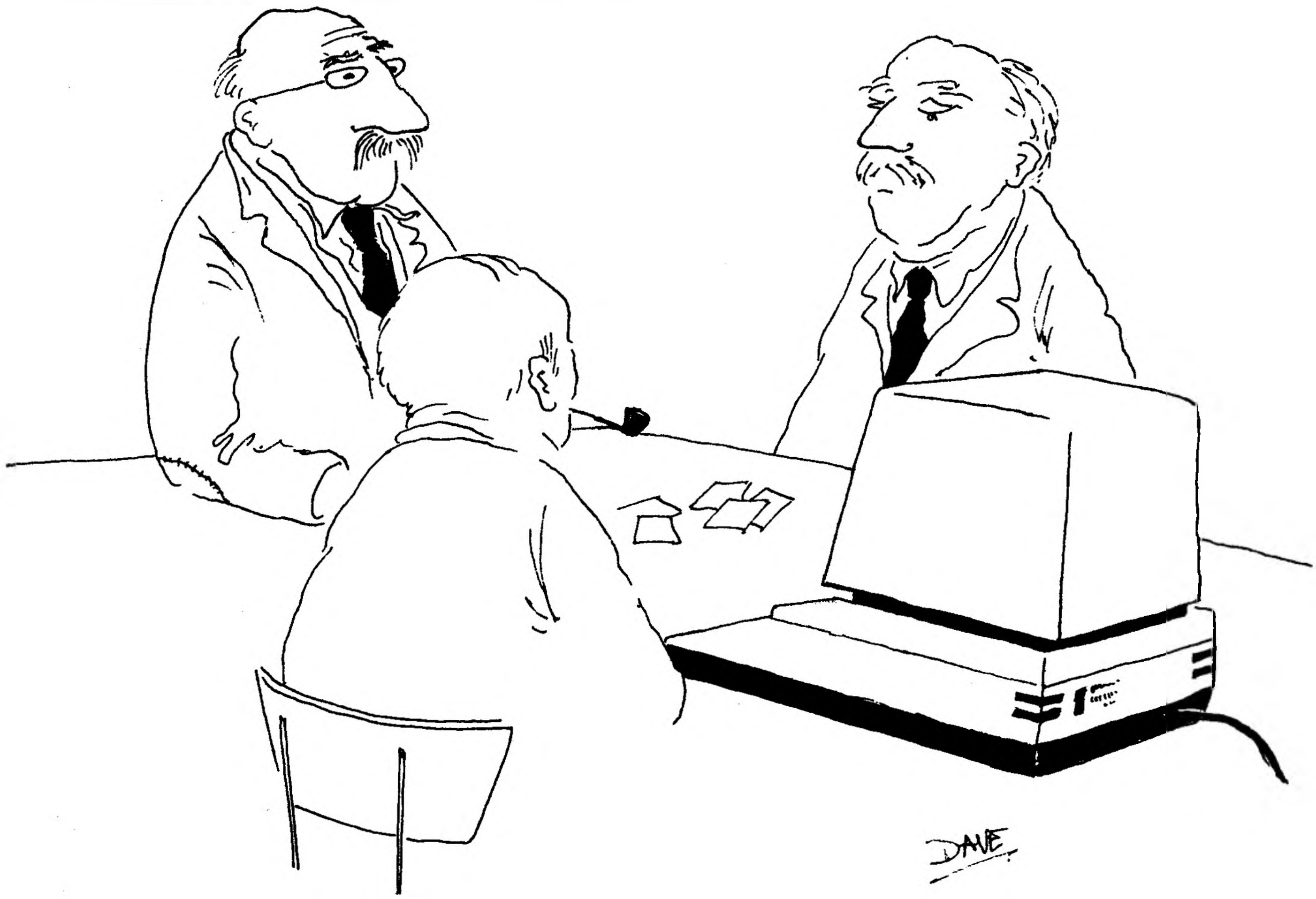
It is small wonder, then, that many shrink from recent suggestions that BASIC should be replaced in schools by some slightly superior language. Computing in UK schools could be far too young to survive the disarray that such major surgery would cause.

Certainly, interesting languages such as PASCAL, COMAL, PILOT and others should be carefully examined by those educators with the enthusiasm and the wherewithal, and, if they are as useful as their supporters claim, they will, no doubt, be gradually assimilated and eventually take their place in educational computing.

The range of languages offered by various microcomputers may, therefore, be of definite interest to some prospective buyers, especially if they are intent on experimentation, but, in the main, many schools' immediate concern is probably the provision of a good quality version of BASIC, and, as lesser considerations, the availability of a useful assembly language (for development or GCE A-Level work) and of CESIL (for CSE or GCE O-Level work).

The one fundamental question that must concern any interested party is, 'How is the microcomputer going to be used in school?'. The answer to this, if it is known, largely determines the size and type of equipment required and most other aspects, such as the necessary level of management, choice of software, and accommodation.

'Well yes, I suppose it does play a better hand of bridge than George, but I still miss his conversation!'



One object of this pack is to provide some guidance on the possibilities of computer use in schools.

Care should be taken, at this stage, not to confuse 'modes of operation' with 'types of application'. As for modes, the microcomputer is capable of being used for calculation, data processing, information retrieval, audio/visual display, conversation, data capture, process control or text manipulation and word processing, each of which represents a different type of operation.

No doubt, more modes could be listed, but these are all essentially facets of the microcomputer's potential. An 'application' of the microcomputer may employ one or more of these modes to achieve a particular end. Table 1 below outlines some of the possibilities.

Table 1. Microcomputers: potential applications in schools

Administrative context

Potential applications here could be many and varied, but might include book-keeping, pastoral care, record-keeping, examination result processing, fixture lists, mailing lists, timetable-related tasks, school register.

Academic context

Applications here could include the following.

1. *Simulations and models* could be in scientific, mathematical, historical, geographical or other contexts and could model systems, processes, experiments, apparatus or events.
2. *Reference source*. This involves the task of putting a large volume of data into the computer (a databank) followed by the simpler task of accessing it in a variety of interesting ways.
3. *Problem-solving*. The computer is employed to calculate the answers to programmed problems.
4. *Educational game-playing*. Many computer games can be used as enjoyable ways of learning, provided that the teacher is clear about the educational benefits.
5. *Demonstration*. Concepts, principles and techniques can be displayed or illustrated.
6. *Consolidation*. Drill and practice exercises, revision material, testing and marking and performance summaries are possible.
7. *Computer aided instruction*. Course material can be imparted through the medium of the computer, but development is involved and requires more effort than most schools can provide.
8. *Computer managed learning*. Pupils can be tested and directed to the appropriate next level of study in a course. Again, development is too involved for most schools.
9. *Computer assisted learning*. This term could really cover most of the previous terms.
10. *Control device*. The microcomputer could be used to take readings from, or control the operation of, say, equipment in a science experiment, if sufficient expertise were available.
11. *Programming tool*. The machine could be used as a practical aid to pupils and staff who are learning a programming language like BASIC. Computer studies is an obvious context.

Two articles (F and G) illustrate the kind of teaching material that could be developed and used on a microcomputer in school. The first, 'Historical simulation' by Richard Ennals, is an excellent example of how a computer can be used imaginatively in a non-numeric context. It may be worth noting, however, that an abundance of experience and enthusiasm seems to have been invested in this development, and no doubt accounts for its success.

'Escalator packages' by David Pegg describes a numerical type of application, reproducing coin-tossing experiments and providing statistical summaries. This application relies heavily on the computer's speed of calculation and its ability to generate random numbers to act as sample data. The idea of starting with a simple computer program and in stages slowly building it up with additional statistical calculations is again an imaginative idea and appears to be a successful teaching technique.

These are just two of many similar educational software developments in a whole range of subjects.

However, it would be misleading to suggest that the purchase of a microcomputer would give instant, fruitful results in all or even a few of the activities in Table 1. Far from it, most of these activities are still in their infancy and besides, there are many other factors involved, not least of which is the provision of the appropriate software.

Software can be imported (at a price, usually) from outside sources as 'packages' or can be written within the school as tailor-made programs. Opinions differ as to the best course of action here. Certainly, packages from outside sources (even where the source is another school) can be disappointing. The programs may not work, and if they do, they may be badly written, misguided in design, inadequately documented or lacking in defined educational objectives. Some of the so-called 'educational software' emanating from American microcomputer manufacturers follows these regrettable traits. The problem is that it is normally impossible to assess such packages in advance of payment and so, at present, purchasing educational software is rather risky.

Moreover, there could be something inherently undesirable about the thought of, for example, all geography teachers in the country using exactly the same program to teach 'location of industry' or any other topic.

Unquestionably, it would be preferable for a program to match an individual teacher's own needs, style and lesson content, and since he is the best judge of his own requirements, then ideally he should write his own programs. Few would disagree with this assertion in principle, but in practice it cannot work at present for all teachers in all schools, because the required time and expertise is just not available. In fact, it takes so much time and effort to produce just one worthwhile item of software that even with help from some professional programmers, a school would still be unlikely to attain self-sufficiency in its software development.

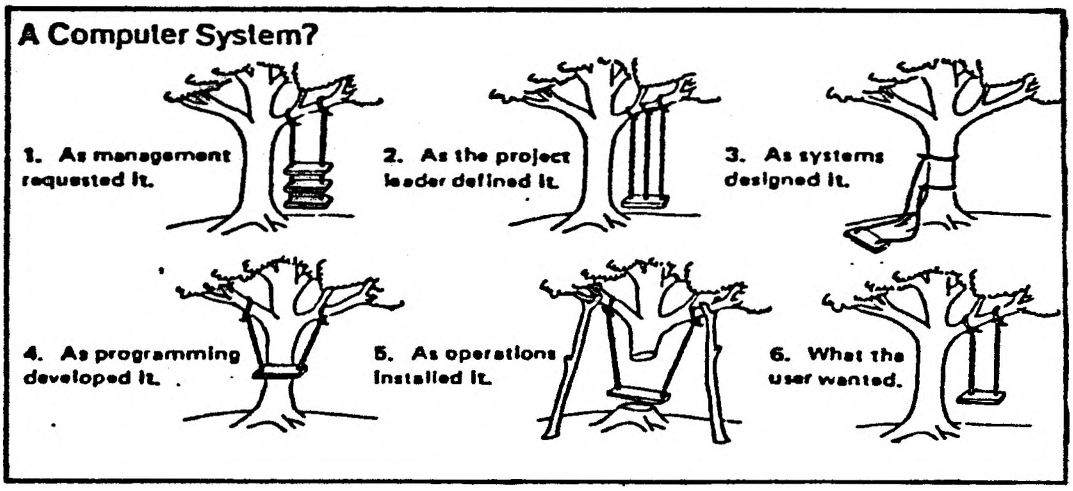
So the use of packages (that is, other people's programs) despite their shortcomings, will probably win the day by default, as the only practicable way of maximizing the benefit from this not inexpensive resource - the microcomputer. Individually, schools may not develop very much software, but collectively, what they do produce could form substantial program libraries. A danger here is that without recognized national standards for programs and documentation (that is, if they do not all follow the same familiar pattern) teachers, and especially non-experts, may well become confused and disillusioned. A further danger is that without national coordination of effort and dissemination of information, schools and colleges in every part of the country may be reinventing the wheel.

In article H, 'Computer programs for schools', John Turnbull elaborates upon these and other points concerning the far from trivial business of producing valid educational software.

It is comforting to observe that some agencies, notably the Schools Council Computers in the Curriculum Project, are attempting to produce or collect educational software that conforms to certain standards of programming and documentation. The work of the Computers in the Curriculum Project is summarized in article I ('Schools Council Project') and gives some idea of the range of school subjects being explored. It is perhaps worth noting that in this project, roughly 25 people took five years to complete 52 programs. Assuming that all concerned were enthusiastic about the task, the reader can draw his own conclusions about the time-scale of software development.

Currently, efforts are being directed to the transfer to some microcomputers (that is, rewriting) of scores of educational programs that were originally designed to be used with remote terminals. Prospective buyers in education would be well advised to identify which of the microcomputer makes are to be the lucky recipients of this transfer process before making a decision.

Meanwhile, one may well ask: 'What proof have we that these programs are educationally more effective and desirable than traditional teaching techniques, and what are the full practical implications of introducing microcomputers into schools?' Some investigative projects are trying to provide more information and guidance for schools and education authorities on the best courses of action to take, by undertaking in-depth studies of these areas. Ongoing research of this kind is clearly necessary to help provide some answers.



Of course, in the final analysis, it is people and not machines that make the educational world go round. It is fitting, then, that the next article in the pack should be concerned with people and their attitudes to microelectronics.

Article J, 'How will we cope with the micro chip' by Vince Houghton, classifies the attitudes of educators to the new technology under five headings - ostrich, Luddite, apocalyptic, cooperative, and imaginative.

Throughout the coming years, education in the UK desperately needs to undergo a phase of adjustment to the new high technology (of which microcomputers are a part) that is sweeping the world at large, and this implies a significant amount of change. But without the wholehearted support of all the people concerned - education officers, advisers, examiners, administrators, and especially the teaching profession - these changes will only be frustrated in one way or another.

Vince Houghton's comments on attitudes are highly pertinent, therefore, to the shole school microcomputer debate and are, at the same time, very disturbing. He highlights the fact that we are faced primarily with a human problem rather than a technological one.

The general implications of microelectronics and the urgent need for action, especially in education and training, are amply expounded in the final three articles, which, taken together, give a composite picture of what is in store and the necessary preparations to be made. They are representative of a multitude of articles from well informed sources, all of which stress the urgency of the situation in the UK.

Article K ('The race to train the brains that work the future' - a *Times Educational Supplement* special report by Philip Venning) is a perceptive examination of the general implications of microprocessors for employment, training and educational patterns in the UK. This exposition of the scale and scope of the across-the-board changes required in education and training indicates how serious the situation is.

Ironically, *The Times* and Mr Venning's colleagues have subsequently discovered, to their cost, one of the implications of the introduction of new electronic technology, namely industrial strife.

In article L - 'Microelectronics: their implications for education and training' - Geoffrey Hubbard analyses in more detail the problems facing education together with possible solutions and the new facilities and techniques at out disposal. His realistic view is that educational change, being a 'response' to the effects of microelectronics on society, will vary according to whether these effects are sudden and radical or a 'slow and wholly contained modification'. This latter is certainly worth mentioning as a distince possibility. Effects may well prove to be slow and marginal, but if that were to be the case, then it would mean that this country had failed to capitalize on the amazing opportunities presented. Where we might fail, other nations with progressive policies would almost certainly succeed. Radical change must, then, be the target, even though this may throw up mammoth problems of a social and industrial nature for politicians to solve.

In theory, education need not be a mere response but could instead be a catalyst. Massive and immediate investment and restructuring in appropriate areas of education could force the issue and help to ensure that radical change took place in industry. Education in this sense would be the determinant of destiny rather than its whipping-boy.

In the final article (M - 'A plan for feeding the hungry sheep ...') David Pegg states the case for some kind of national policy to be formulated on computer education in schools. There can be little doubt that the sentiments expressed here are largely shared by the majority of those involved with schools' computer education.

The implementation of measures in the spirit if not the letter of the proposals given would be a major step forward. Three further areas are not explicitly mentioned by David Pegg and ought, perhaps, to be included in the list. Firstly there is a need for a single national centre acting as a source for all valid schools programs, and secondly a corresponding need for a national body to define and maintain a set of software standards to ensure that these programs are worthy of use. Third, projects engaged in research and investigation into the use of microcomputers in schools need to be sustained and encouraged if future development is to be on an informed basis.

Plans of action, then, are needed, but not in such haste that they are ill-considered. Whether one is looking at microcomputer hardware selection, computer languages, software selection, software development, management of computer facilities or dissemination, none of these areas of concern seems to be straightforward.

In fact, it is conceivable that, having read through this pack, our non-expert is somewhat nonplussed. No slick answers have been given because there are none. Clearly, there is a need for more information on microcomputers, their usefulness in schools, and the practical and educational implications of their use.

It is probably safe to say that where a school intends to use a microcomputer purely as a programming tool within computer studies classes, then the problems will be relatively few and the situation will be greatly simplified.

But where a school hopes to exploit the full educational potential of the microcomputer throughout the curriculum and administration, then there are many more considerations involved and a successful outcome will depend on how well these considerations have been dealt with in advance.

It is hoped that this pack at least has helped to direct the non-expert's attention to some of these areas and provided sufficient information and opinion to stimulate further questions. The prospective buyer would always be wise, before making a commitment, to seek further information and advice from as many quarters as possible, especially advisers and those with practical experience in this field.

Although the issues may be many, the microcomputer seems to hold so much promise for education, in so many ways, that it just cannot be ignored.

LIST OF FURTHER REFERENCES

'To work on a chip - the mixed blessings of miniaturization', *The Listener*, 6 April 1978
(Development background information and industrial applications and societal implications of microprocessors.)

'Developments in computing at King Edward's Five Ways', A Clements, *Computer Education*, No 30, November 1978
(Microcomputers in the context of previous computer-based education in a Birmingham school.)

'The race to train the brains that work the future', P Venning, *The Times Educational Supplement*, 20 October 1978
(The problems facing society and our educational system in coping with changes brought about by microelectronics.)

'Microelectronics: their implications for education and training', Geoffrey Hubbard, *CET News*, January 1979
(A summary of the educational implications originally published in the Council for Educational Technology statement, *Microelectronics: their implications for education and training*, 1978.)

'A plan for feeding the hungry sheep', David Pegg, *MUSEletter*, No 3, April 1979
(Proposals on national policy and support for computers in education.)

'Computer in the classroom: teaching packages', C Sweeten, *Personal Computer World*, No 2, June 1978
(The potential of the teaching package - a documented teaching program - as an educational method as instanced by a particular example.)

'Educational computing in the local authority sector - the next ten years - a future study report', *NDPCAL Technical Report*, No 15, 1977
(Views on educational computing in general, and descriptions of a range of computer assisted learning packages.)

'Computer languages - the multilingual machine' , G J Marshall, *Personal COmputer World*, Vol 2, No 2, June 1979
(An examination and comparison of a variety of computer languages.)

'Is there a computer in the house?', Guy Kewney, *New Scientist*, 19 October 1978
(A general opinion of the microcomputer market, presented and projected, with some notes on costs.)

'MUSE software library - mini and microcomputer users in secondary education', *MUSEletter*, see most recent edition (Details of programs with special relevance to secondary education available through this recently formed library.)

'A glossary of computing terms - for introductory courses', British Computer Society, 1977
(A usefully consise summary of key terms.)

'Teaching with microcomputers: the ITMA project', *CAL News*, No 12, July 1979
(A report from an investigative project looking at uses of microcomputers in the classroom.)

'Durham Microcomputer Project - an interim report', D Sledge, Durham County Education Authority, 1979
(A full report of two-and-a-half years' investigation of microcomputers in secondary schools.)

'USPEC 32: Guide to the selection of microcomputers', *CET User Specification*, forthcoming
(A practical guide for those intending to buy a microcomputer.)

A practical introduction

THERE ARE too many myths about computers. Too many people hold too many rigid and unsubstantiated beliefs about what computers are, what they can and cannot do, how and why they work. This introduction will change all that. For a start, let us summarise the myths:

- **computers can think.**
- **computers are large.**
- **computers are bureaucratic.**
- **computers are expensive.**
- **most computers are run by government organisations.**
- **most of the rest are run by large companies.**
- **computers are electronic brains.**
- **computers are incomprehensible unless you are some kind of genius.**
- **computers are best left to someone else.**
- **computers fail all the time.**

They are all fundamentally incorrect. Computers are not necessarily like that, though some people would like you to think so. It's worth looking briefly at how the myths evolved before deciding what is wrong with them.

The whole computer business, like most others, is driven by technological and economic forces. When the components were expensive, not many people could afford computers but for the people who could bear such an outlay, there were considerable benefits of scale, so computers tended to be complicated, expensive and big.

Two reasons

The other main prop of the myth is the need for dedicated acolytes. While computers were complicated, it's true that to understand, organise and manipulate their complexity a bunch of specialised jobs had to be created. The need for computer training and the arrival of computer jargon both served to separate computer people from the rest of us.

Some computers are still big and complicated; they still require specialised staff; they still cost a good deal of money to buy and to run; they still operate at unimaginably fast speeds.

That kind of computer is now in the minority. They are bought for only two reasons – to solve gigantically complex problems or to allow a large number of people to use a little bit of the computer, all at the same time.

Weather forecasting or space exploration are obvious candidates for the first category. The second is called time-sharing and it is an alternative to giving the same number of people a small computer of their own.

A small computer of your own is what *Practical Computing* is about. That mixture of technology and economics has worked to produce a breed of computer which does not differ in kind from the multi-million-dollar megaliths with their over-qualified minions, their impenetrable forests of new and mis-spelt words, their general inaccessibility.

What is a computer?

Technology has provided a cheaper and more compact type of computer. The economic factors dictate that more people can afford them, more individuals want them, more businesses need them; so they can be produced and sold fairly cheaply. Computers are becoming accessible.

A computer is a fast, rule-following idiot machine. It is fast because it is electronic and electrons are speedy. It follows rules because that is all a computer does; you can alter its set of rules more or less at will, you can add rules, you can complicate rules. It's an idiot because it simply can't think for itself – not in terms of original, creative thought, anyway. It follows those rules.

That applies to all computers irrespective of their size, shape, colour, capacity, nationality or the uses to which they might be put.

The essential element is that you provide rules for it to follow and that you can change them. All talk of 'rules' is a little abstract, so let's give the business a name – *programming*.

A program is a set of instructions for the computer to follow. It will try to follow them blindly, typing errors and all. If you change the program, or substitute another, the fast idiot will go through the new instructions as coldly and as logically as before.

Remember this; the freshly-arrived computer is blank, it has no intrinsic purpose until you give it one, and the

(continued on next page)

(continued from previous page)
word computer is incorrect, a historical accident; it doesn't necessarily do computations at all. It's just that the first computers spent all their time doing calculations – for shell trajectories and census returns, initially.

System mix

It's helpful to start calling the computer a 'system'. Most computer people do and for a change this is a meaningful use of a jargon word. A system is a set of components which can be combined to produce effects none of the individual pieces could manage alone.

The components of the computer system are a mix of *software* – which is a group name for all your programs – and *hardware* – all the pieces you could stub your toes on; literally, everything that is hard, tangible and visible.

You have to relate that to what a computer does when it is computing. Four things happen:

- information (or words, or data, or instructions, or whatever) goes in; and that is *input.*
- the input is decoded, acted upon, massaged, manipulated; that is *processing*
- it may be stored for future use; so may whole programs and any results of processing
- alternatively, or subsequently, the results of that processing may be displayed, printed, or in some other way proclaimed to the outside world, usually you; that is called *output.*

A conventional computer, then, has facilities for:

- input
- processing
- storage
- output.

What happens is that you, or someone else, *inputs* a program which is stored until it is needed. Subsequently some data is input, the program is activated and processes it, and the results are output.

The idiot in the middle

Obviously the piece in the middle of all this computing will be the processing component. It will be no shock to discover that this is called a processor – or central processing unit, or CPU.

Time for technology. What happens inside a processor is that electricity moves down one circuit or the other. The complexity of the alternatives – how fast the choice can be made, how quickly the electrons can follow the chosen route, how small the whole thing can become, is what distinguishes one processor from another.

There are dozens of processors, incidentally, and there are many more products for the end-user which have found ways of incorporating the same processor into identifiably different computer systems.

Computers are really a whole series of electronic switches. Like any switches, they can be ON or OFF. There isn't any other possibility. As it happens, there's a neat way of expressing this ON/OFF business – binary numbering.

Don't be worried by this but there are many numbering systems other than the one we use, which is called decimal, because it uses 10 digits. The binary system uses only two digits, which for the sake of argument are '0' and '1'. If '0' corresponds to 'off' and '1' to 'on', obviously you have a neat way of representing the internal operations of the computer.

The electronics can decode a string of 0s and 1s as a series of off/on combinations and you have a way of communicating with the electronics. You can tell it

that certain types of 0/1 patterns will be program instructions; other binary structures will be information to be processed by programs.

Suspicious characters

It is possible to write programs which give instructions for any computer encoded as 0s and 1s. Provided you and the computer both know what the binary sequence means, it is possible to hold any information in this form, even alphabetic characters.

It is extremely tedious to communicate with the computer in this way, though, not least because any normal person would have to keep checking on the binary codes for alphanumeric characters. You would go out of your mind.

So you give the computer a special manufacturer-supplied program which will convert a more intelligible way of expressing information into the binary digits a computer can use. That way you can give the computer a number or a letter and with a quick piece of internal transformation it can understand what you mean.

Most computers translate characters according to an eight-bit code, a 'bit' being a binary digit. An eight-bit code comprises a string of eight digits, each of which can be 0 or 1. That gives a total of 128 possible combinations, enough to give patterns for each letter of the alphabet, each number, and a few punctuation marks and arithmetic symbols, too.

So if you key-in a particular sequence of characters at your computer keyboard, it will decode them into a group of eight-bit codes, and they are the binary sequences it can understand.

It can hold them in its memory, too. The storage capacity of a computer, the amount of information it can keep in memory, is expressed usually in characters – or bytes. A 'byte' is eight bits, so generally one byte is the equivalent of one character.

You will also encounter the cryptic symbol 'K'. That is shorthand for 1,024 – don't worry about why K means 1,024, it's just one of those things. So '8K' means 8 x 1,024 = 8,192 characters.

Chips that pass in the night

Electronics these days is about switching streams of electrons (or electricity) and it is only 60 or 70 years old. In early days a kind of switch called a *relay* was used; they were comparatively slow to operate, though.

'Slowness' here means a few thousandths of a second, which sounds fast until you realise that even a simple internal operation looks complicated when you reduce it to a number of switches opening and closing – and for one operation that typically means several thousand, several million switchings. They all mount up.

So the advent of *vacuum tubes* in the 1950s pleased everyone. They operated rather faster. *Transistors* followed, a few years later, faster still and more reliable. The major breakthrough of the early 1960s was the *integrated circuit,* and that's where we are now.

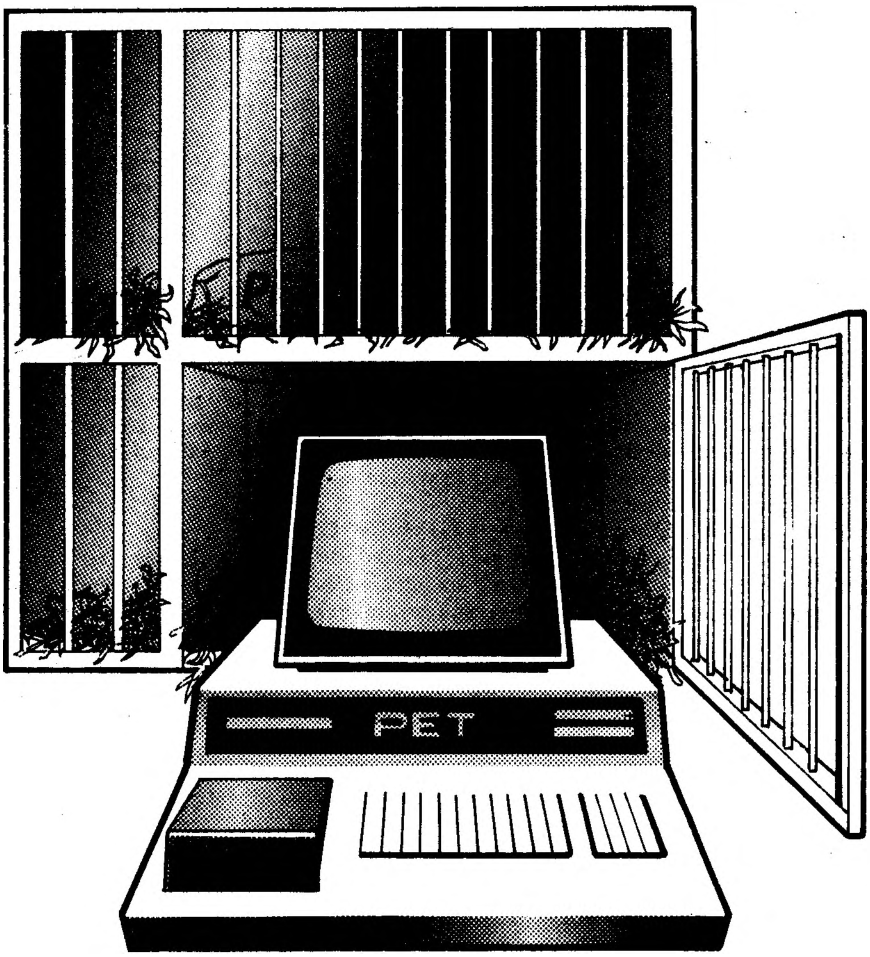
Even faster and even more reliable, integrated circuits were also considerably cheaper and much more compact. They used the relatively new technology of semiconductor materials to cram an increasingly large number of electronic switches on to a decreasingly small silicon chip.

So you've heard of silicon chips? Silicon materials happen to be the best way at present of putting the maximum number of circuits – at least 100, more usually several thousand – on to a really small area of crystalline material.

That, in turn, is encased in a block of plastic with legs; each leg corresponds to and is connected to one of the circuit ends on the chip. The little lot plugs into, or is soldered into, a socket on a printed circuit card, which is sometimes called a printed circuit board and abbreviated to the PCB.

Those boards are fibre-glass or plastic rectangles with circuit lines printed on to them. The lines are gold or silver or some

Your first computer



other electrically-conductive metal and they run between the socket holes; put the correct semiconductor packages into the correct holes and the chips can pass electronic signals which mean actual data to each other.

You can't put all the chips you need on to a single PCB – yet. So the chips on one board have to have some way of communicating with the chips on another. They also have to have some way of getting information to and from the rest of the system, and they need some electrical power to work in the first place. So a PCB has a line of circuit connectors along one or more edges; they are the other end of the circuit lines which connect the socket holes.

Processor power

You plug the circuit boards into slots in a kind of metal skeleton frame. This has built-in wires connecting one slot with another, and all slots with the electricity supply and the other parts of the system. The connectors along the edge of the PCB mate with connectors in the frame, so there's a way of passing signals to and from the PCB via the frame.

A processor isn't a computer, obviously; it is just one component but it is an important component because it decides exactly what the computer can and cannot do.

A *microprocessor* is a small processor. That doesn't do it justice – a microprocessor represents a major technological advance, because on one or two chips the designers have managed to cram all the circuits for which conventional computers have several chips.

There are three important implications of this. Microprocessors are cheap to mass-produce, they are fairly small and, for various technical reasons, they can't be as powerful or as fast as their upmarket brethren.

You might meet four names among microprocessors. None is a computer, but they are the most widely-used microprocessors which feature in small computers.

- **Intel 8080** – and some family relations like the 8048. Intel made the world's first microprocessor in 1971, and this is now the most popular micro. It is used, among others, by Processor Technology, the Compelec Altair system, the Heathkit H8 kit computer, the Imsai line, and the Compucolor II.
- **Motorola 6800**: Motorola is probably No 2 in the microprocessor business; like Intel and most of the micro manufacturers, it sells the bulk of its production to industry and other users outside the kind of computing of which we are talking. The best-known 6800-based system in our field is the SWTP computer sold by Computer Workshop.
- **MOS Technology 6502:** This micro bears a passing resemblance to the 6800 and it is used widely in so-called personal computers. Among those are the manufacturer's own Kim-1, the similar Superboard II from Ohio Scientific – other Ohio computers also incorporate it; and above all the Pet, whose maker, Commodore, owns MOS Technology; and the Apple II.
- **Zilog Z-80:** Zilog was set up by some people who left Intel with the aim of building a similar microprocessor – only better. To some extent they succeeded. The Z-80 is well-liked by those who know about such things and is used by the North Star Horizon, Tandy TRS-80, and the Nascom-1, among others.

Thanks for the memory

A particular arrangement of particular chips will provide the functions of a processor. The same kind of technology is applied differently to provide other parts of the computer system, including the memory.

There are various ways of storing information – especially programs – for future use. They differ primarily in speed of access. You read information fastest from read-only memory or ROM. It is so-called because you can't 'write' new information on to it.

ROM is physically one or more of those plug-in semiconductor packages. Its contents are usually fixed by the manufacturers and consist generally of frequently-used programs without which your system couldn't really operate.

Then there is read/write memory, whose contents you can alter. Sometimes it's called random-access memory, or RAM. This is the main 'user' memory of the system, sitting there waiting for you to fill with your own programs or the data on which your programs will operate.

Finally there are various external storage devices, the slowest for the computer to get at but also the cheapest. These are connected usually by cable to the box which houses PCBs of the processor.

The two external storage media you will encounter are tape and disc and the slowest and cheapest versions of each are cassette and floppy disc.

Cassettes as used with computers are much the same as ordinary audio cassettes; on the cheaper computers they are audio cassettes sometimes. Certainly the cassette units are sometimes off-the-shelf portable tape recorders.

Cassettes are obviously a cheap form of storage; you can buy one of those recorders for well below £30, after all, and the cassettes aren't expensive. They are, however, limited. You can't read from or write to cassette at the speeds

(continued on next page)

(continued from previous page)
possible with disc. More important, cassettes store data serially. That means one piece of data is stored after another and if you want to reach a particular item you have to pass over everything which precedes it on the tape.

You don't have that disadvantage with floppy discs. They are in two sizes, one about the diameter of a 45 rpm record and other around two-thirds of it. They really are discs, and they really are floppy, though to give them at least some rigidity and some protection they arrive in cardboard envelopes. They have slots cut in them to expose the disc surface so that the read/write head can make contact with it.

Faster

Discs are much faster at getting data to and from the processor. They also allow what is called 'random' access to data stored. It isn't really random – it just recognises the fact that you can tell the read/write head to move over the disc before it does any reading or writing.

With audio tape, there is no simple way of getting to the start of a particular taped song on cassette. The same applies to data on a cassette but just as you can move a record arm over an LP to the correct point, so the computer can move the read/write head to the proper point

on a floppy disc. That can be very important, as we shall see later.

There is one other form of storage which you might come across – paper tape. It stores data in a form you can see. A reel of paper tape contains holes punched across the tape, there can usually be up to eight of them corresponding to the eight-bit binary code and each hole denotes a '1' position in that code. No hole means a '0'.

Paper tape is very slow to read and slower still to punch and the special reader/punch unit which does that is expensive; it is also noisy. Still, it's a clear and simple method of storing data and if you already have the reader/punch mechanism, it might be worth considering.

I/O is input/output and the two are usually bracketed because the one device often provides both functions; it's easier to build it that way.

The visual display unit or VDU is the classic example. It comprises an input device (the typewriter-like keyboard) and an output device (the television-like screen), but a VDU manufacturer finds it convenient to provide one cable for connection to the processor.

In fact, that cable contains some wires specific to the imput function and some for output; since the computer knows which is which, the VDU isn't really a single device at all. In any case, many of today's smaller computers make a physical distinction between a display and the keyboard.

Alternative

Another I/O device encountered frequently is the keyboard/printer terminal, an alternative to the VDU but with a printer instead of a screen. You might also have a totally separate printer with no keyboard; you will certainly need one if you intend to use your computer system for your business.

The are other more esoteric forms of I/O. One which appeals particularly to technology buffs is speech; the computer recognises what you're saying, and it replies vocally, too. The voice output part is more or less possible now, though it's not exactly broadcasting quality; voice input is proving more of a problem.

What is a personal computer?

You can forget most of what has gone before because today you can buy off-the-shelf a fully-fledged plug-in-and-go computer system which requires you to understand as much about electronics as the buyer of a music centre knows about hi-fi. Frankly, though, in both cases you will have more pleasure from your purchase if you know what's going on inside.

That off-the-shelf buy is what is usually called a personal computer – the emphasis being on the individual user. Though you might buy one for your business, you will also be the principal user. By comparison, larger computers sometimes occupy full-time staff who do nothing but work with the computer but who didn't select it and who didn't sign the cheque.

Let's start at the bottom. The most basic personal computer looks like this:

- **Input:** typewriter-style keyboard.
- **Processor:** totally invisible, probably buried somewhere inside the keyboard on a couple of circuit boards.
- **Storage:** internal memory is probably there, too, on one or two more PCBs. External storage will be a cheap cable-connected keyboard.
- **Output:** a screen, possibly an ordinary portable TV set slightly amended.

The three best-sellers at the bottom end of the personal computer market exemplify different approaches.

- **Commodore Pet:** less than £500. One unit containing keyboard, built-in cassette unit, and screen.
- **Tandy TRS-80:** less than £500. Four separate units – screen, keyboard (incorporates the processor and memory), cassette player, power supply.All computers need a black box to convert mains voltage to the current they use; in most computers this transformer is invisibly inside another unit.
- **Apple II:** less than £1,000. Three separate units – colour TV, keyboard (incorporates processor, memory and power supply), and cassette unit.

As for software, all have an *operating system* of some kind – typically a ROM chip or two which contain all the low-level binary decoding functions which make things work; you won't need to know anything about it, though. You will also have a *programming language* called Basic.

Remember all that decoding the computer is doing to save you having to communicate with it in binary? Well, that conversion process can be extended and most of today's computers allow you to use a near-English 'language' called Basic.

It is fairly easy to learn and it's easy to understand. In general, the syntax and vocabulary of a programming language like Basic are simpler to grasp than the rules of a foreign human tongue.

Basic has become popular partly because it was devised from the start as a beginners' language. It had little competition; and the business of translating it into binary digits for the computer – the so-called 'machine code' – didn't require as much space or effort from the computer as other programming languages.

Today you might also hear about PASCAL, a language with similar aspirations but it is only just starting to appear on small computers. The only other serious contender is Cobol, a long-winded language for business use which really scores only when you are already familiar with it, perhaps by using it previously on a big computer.

What can it do?

Peter Ustinov's biography compares TV to telephones. If someone asked you whether you like telephones, you would have to say it depends who is on the line and what they are saying. The same applies to TV and it's also true of computers.

A small computer can do something for everyone. That is an over-simplification, of course, but in practice it can probably do something for you, provided you tell it what to do. If the job you have in mind can somehow be expressed as a series of rules or instructions in the form of a computer program, the computer can do it.

There are two ways of putting in a program. You can key it in yourself, or you can 'load' it from cassette, disc or paper tape, in which case someone else will probably have written the program and sold it to you ready for loading in that form.

So you can put in a program you've written yourself or you can load someone else's. What those programs do is limited only by your imagination – within reason.

Here are some examples we've heard of:

- **Games.** It's easy to regard computerised games as trivial and irrelevant. In fact, game-playing obviously can be intellectually and emotionally stimulating as well as merely diverting.
- **Simulations.** There's a cross-over point which illustrates the value of games. Simulating the economy of Sumaria or the starship *Enterprise* might be games but there's little which is different about planning the future of your company or looking at alternative ways of getting you and the family to Dubrovnik this summer.

A good example is in education, where a history teacher might use the classroom computer to decide 'what if' questions and thus bring historical situations to

(continued on next page)

(continued from previous page)
life. A recent prizewinner in a *Practical Computing* competition is doing that with a variety of situations, including the Norman Conquest – the pupils take parts, make decisions based on the historical situation, and watch the computer decide what the outcome would be.

● **Education.** This could be applied equally to geography and science subjects. The computer can also be used with obvious benefit in complicated calculations at school; but most teaching and much school administration could also gain from some automation of the more routine functions.

● **Business.** The same applies to the administration of business, though the returns are visible and financial. Stock control is an obvious example. A small computer could tell the shopkeeper or a retailer the current stock position at any time on all items, which were selling fast or slowly, which were approaching re-order levels, and how quickly the supplier could deliver. With that amount of information you ought to be able to cut back inventory levels and save money.

You could well do the same for debtors if you have a big sales ledger, and you should certainly look to save time by having the computer produce invoices with VAT analyses as an automatic by-product. It might also produce 'personalised' form letters or quickly-updated price

lists if your business involves those.

● **Home.** There is more rubbish talked about computers in the home than any other area of this subject. True, a computer can run your bath, feed the cat, switch on the TV and change the record but the extra items of hardware you would need could become complicated and expensive. In any case, who needs it?

There are, however, some sensible home applications. Playing games and doing household accounts are obvious candidates. If you're writing a novel, the computer might help, too. You'll need some skill to put a microprocessor into a vacuum cleaner but that's a chore worthy of automation. There is also some scope for having the computer control your home while you're away – feeding and watering your houseplants, perhaps, or turning a video recorder on and off at longer intervals than the VCR manufacturer allows.

● **Art.** Computer-generated art is not to everyone's taste, but at the very least you can have plenty of innocent fun persuading your computer to produce patterns, poems, animated cartoon-like sequences and even music, though you might need special hardware. Some highly serious work has been produced with the computer, so don't take it too lightly.

What to look for

As with motor cars or hi-fi systems or sorbets, there are no hard and fast rules which do not rely ultimately on personal preference, but here are some points to consider:

● **Processor:** It doesn't matter whose processor is inside your computer unless you want to get into it with your soldering iron, in which case you shouldn't be reading this.

● **Standardisation:** Much more important is to consider what you might want to add in the future. Some personal computers follow a standard arrangement of connectors for the slots in their metal frames; so into those slots you can put any PCB which obeys the same standard arrangement.

The best-known standard is called S-100. If you think you'll want to add goodies, you might well opt for a computer with the S-100 standard. Don't be too dismayed if your favourite computer doesn't have it; it reduces choice but it's not a disaster.

The other important standard to ask about is called RS-232, also known as CCITT V24. This refers to external connectors, the means whereby you can attach printers and other peripherals to your computer. Again, RS-232 increases the options and it's probably more important to have it than S-100.

● **Read-only memory:** ROM is generally a good thing. If you have to load the basic system software into the computer each time you want to use it, you don't have ROM in your computer. Having these functions pre-programmed and ready to go in ROM modules saves time and you won't have any load problems – they happen occasionally. Still, this isn't something for which there is much choice about; either you have it, or you don't have it.

● **RAM:** Read-write memory is much more important. The pressing question is how much do you need? And there's no easy answer.

Determine how much memory you can have for your programs and data, because most personal computers put some of their basic system software in RAM, whether you like it or not.

Then you might look at what your input takes up. For instance, if you wanted to put in a full A4 page of text you would need almost 4K of memory to store it. A relatively complicated game with many twists and turns, like most of the versions of *Star Trek,* will need 7K or 8K.

There's a variant on Parkinson's Law here – you almost always use all the memory you have, whatever size it is. Aim for at least 8K and try for 16K if you can afford it; you'll want to run fairly sophisticated programs sooner or later.

● **Programs:** Go for Basic. All personal computers have a Basic and it has become the *lingua franca* in which programs are exchanged. Beware, though, variants of the language are not interchangeable, and you can't load a Basic program written for a Pet into a TRS-80 and expect it to work.

If you buy one of the more popular personal computers – Pet, TRS-80, Apple II – you will find there is a good deal of ready-made software on sale in the form of cassettes (£3 to £25) and diskettes (usually £10 to £50).

Check whether your computer uses an operating system called CP/M or a floppy disc drive made by North Star; again, there is plenty of off-the-shelf software available (on floppy disc) for these.

● **Keyboards:** Unless you have a computer with a decent keyboard, you can have all kinds of problems trying to type-in a program. A good all-purpose keyboard follows the QWERTY typewriter layout. It has a solid, chunky feel when you depress keys and it has a big, unmissable RETURN key – this you use to tell the computer you have concluded one line of input and want to start the next, so it is used frequently.

Extras on it might include a separate numeric keypad on the right (speeds the entry of numbers) and a CAPS LOCK key in addition to SHIFT LOCK – locking into capitals means only that you can press all the non-alphabetic keys and still get whatever is in the lower-case position.

● **Display:** Go for a big display if you can. A good-sized display produces more information more quickly and in more alternative shapes and sizes than a printer or a small display. 'Big' in this context means it should be able to show at least 16 lines of 64 characters.

Displays are in three varieties. You ought to obtain the best possible results from a purpose-built visual display unit (VDU). They normally display 24 lines of 80 characters, but they will generally add at least £500 to the cost of the system.

A built-in screen like that on the Pet or Compucolor II might not display so many characters but you will get special graphics symbols and no need for cabling.

The third type is simply a converted TV, the simplest and the cheapest kind of display, though it might not produce the sharpest image.

● **Power supply:** Translating the mains power into the electricity your computer needs is the job of a chunky component usually called the power supply. In fact, it's a voltage transformer like the one used by a model train or racing car layout.

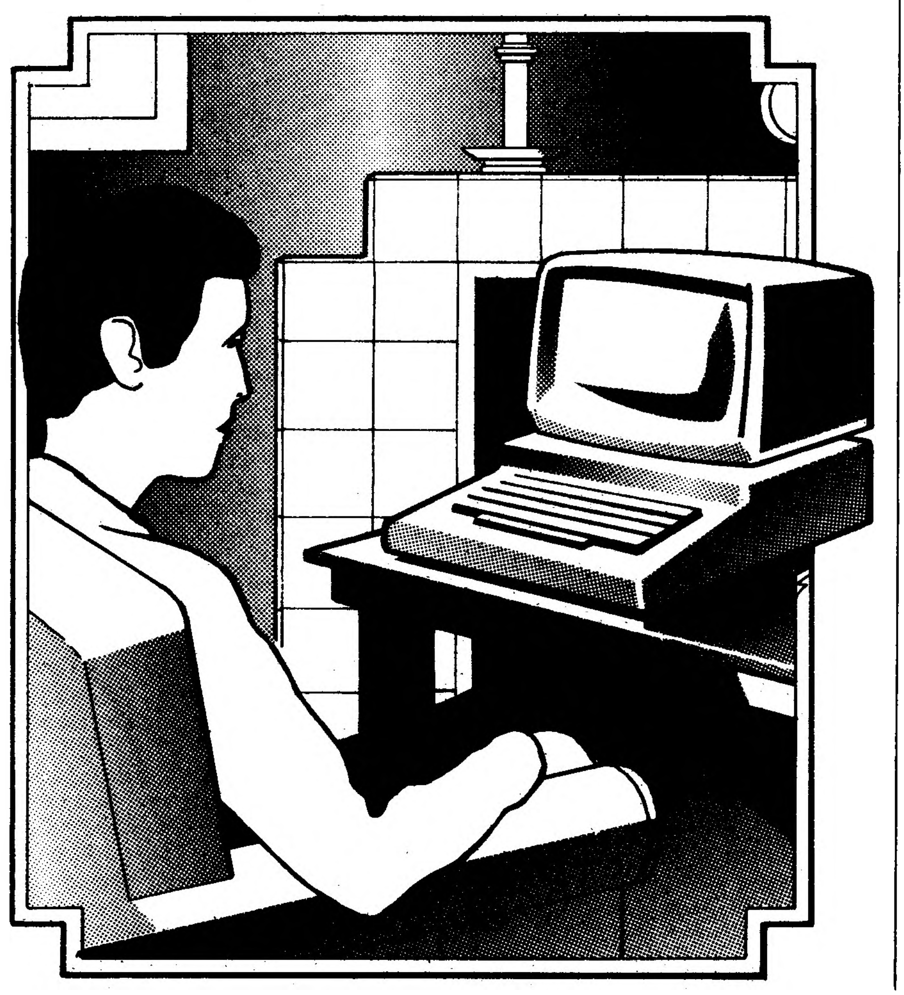
It is possible to overload the power supply, in which case things get hot and/or frail. So look for really beefy transformers and hefty wiring. Also get some advice about how much you can plug into the system's existing power supply before it needs a hand.

Computers dislike variations in the electric current. This may result in the appearance of wobbly characters on the screen, or data being lost between the processor and cassette or disc, or at worst, some component failing.

Voltage variations are inevitable in mains electricity and if your computer is plugged into the ring main you may compound the problem by having other electrical appliances switching on and off – refrigerators, stereo set-ups, irons, heaters.

Ask someone's advice about *voltage regulators.* For an average personal com-

(continued on next page)



(continued from previous page)
puter you could buy one for £25-£50; it plugs into your mains socket, and you plug the computer into it. The regulator evens out the dips and peaks in the power supply cleverly, delivering a smooth flow of electricity to your computer.

● **Printers:** Sooner or later you'll want a printer to keep a record of your programs – you might lose or damage a cassette, after all. For some applications, like business uses or word processing, a printer clearly is vital.

The immediate problem is connecting a printer to your computer. Having specified an RS-232 interface means you can attach almost any printer; otherwise your options are more limited.

There are two types of printer. There is high-quality, typewriter-like printing from daisywheel printers (£2,000 or more) or from converted IBM golf-ball typewriters (£1,000 or more but rather slower). Dot-matrix printing is a technique whereby characters are built up from dots, and because those printers are mechanically simpler, the prices are lower – down to around £300 for a very small and slow printer which can manage only 40 characters to a line. In the range £700 to £1,000 you will get a faster printer (60 to 120 characters per second) which is good enough for anything but top-quality printing.

● **Storage:** Using cassette tape or floppy disc for storage gives you a cheap and easily-expanded alternative to keeping data and programs in RAM.

The cheapest kind of cassette system loads at something around 50 bytes per second, and the fastest rarely exceed 300 bytes per second; it could take several minutes to transfer a complex program.

If you can afford it, choose floppy discs – their chief virtue is that they operate at much higher speeds, taking far less time to transfer information. This saves on boredom but it also allows you to make better and more imaginative use of your system.

In any case, your computer may well be doing jobs all the time which involve looking-up records; you need the speed of disc storage for this.

● **Documentation:** Personal computers generally have inadequate user instructions and reference manuals. These days the accompanying documentation tends to be better-produced and some of the learner-level starter manuals are really good. Even if you are sure that all the information is there somewhere, it can still be very difficult to read via an index or the contents page. Quantity is no substitute for quality.

● **Users:** You should also look for an active users' newsletter, perhaps even a specialised user group. Both are media for exchanging opinions, advice, notifications of errors, and potentially useful programs.

● **Supplier:** You'll come across three kinds of supplier – single-system specialists, single-system generalists, and computer stores.

The last category is probably the most important. The computer store sells several types of computer, several types of printers and disc drives, and almost everything you need for your computing – paper, discs, tapes, books, magazines, the lot. Those places should serve as local social centres for the personal computer community, repositories of knowledge and advice and notice-boards for exchanges, advice and information – or that is how it should be. If your local computer store doesn't seem like that, tread warily.

The second group comprises mainly Pet and Tandy dealers, retailers who handle other products – typically audio systems or hobbyist electronics, sometimes other consumer electrical goods –

but who sell only one brand of computer, normally at rock-bottom prices.

The single-system specialists typically make their money from knowing a great deal about the one brand they sell and by selling you many extras for it, including perhaps some programs written by them to your specifications. Some computers you cannot obtain anywhere else. If you need a relatively complicated system to do certain specific tasks, that kind of supplier is your best bet.

What to buy?

Ensure you know what computers can and cannot do; then decide more or less what you want from a computer. The first decision should be what type of computer to consider, and there are four categories to look through:

● **Hobby computers:** Typically costing less than £300, they require some technical knowledge. They are either build-it-yourself kits – not too difficult to assemble – or ready-built computers on a single printed circuit board – with some memory but not much, and with no protective casing around it.

They may have a built-in keyboard for input; they may have a small display for output. Some have neither, some have both, one has a keyboard and a tiny printer on the PCB.

All have connectors to attach cassettes, printers, or other external devices. Most have Basic; some have only their own cryptic programming languages. Examples of this breed are the Kim-1, Aim-65, Nascom-1, and Ohio Superboard II.

● **Games growing up:** Again less than £300, they are a spin-off from the TV games you see for a few pounds. The more complicated allow you to plug in new cassettes for more games. Since these so-called games centres are essentially microcomputers with the games cassettes being programs, you could add a keyboard and program them yourself. Some of the games manufacturers now allow you to do that. It's already happening in the States, and there will be similar products this year in Europe, from Philips and others.

● **Appliance computers:** Denotes computers designed to be sold, taken home, plugged in, and used just like any other domestic appliance, TV or washing machine. The Commodore Pet was the first and the best example. It's compact, simple to use, and is in one piece. Against that, it's not particularly expandable. The newer model is more expandable but you have to add extra external items, like a printer and floppy disc drives.

A more recent arrival in this genre is the Compucolor II – more expensive but it incorporates floppy disc rather than cassette and it gives you a full colour display rather than black and white only. Further up-market there are several plug-in-and-go computers, so the price spread is from £500 to more than £5,000. The more expensive ones are designed for business use, of course, and they assume generally that you will be attaching at least one external device – a printer.

● **Building-block computers:** They separate the I/O, external storage and processor functions into different boxes and connect them by cable. The processor box contains the memory and may contain floppy disc drives, too, as in North Star Horizon. The Apple II, Tandy TRS-80 and Processor Technology Sol exemplify an alternative design trend by putting the processor and memory into the keyboard.

The attraction is simple. To uprate one area of the system you can disconnect the existing unit and plug in a better one, and/or you can attach more of the same, and/or you can add extras.

So you could swop a slow but cheap cassette for a fast but expensive floppy disc when you can afford the difference. You can attach more memory or a second disc if you want more storage. You could plug in a special typewriter-quality printer or one of those voice output devices if you want some extra facilities.■

A Computer for Schools

by Roy Atherton, Bulmershe College, Reading

In the last edition of Computer Education much was said about microprocessor-based computers. But even before that edition was published a computer had come on to the market which matched closely the minimum system specification as agreed by MUSE and the Berkshire Working Party.

There is no need to call it a microcomputer because in all important respects it performs like a moderately powerful minicomputer. The title refers to the Research Machines 380Z microprocessor-based computer system but what matters most is not what is in the box but how it performs in a classroom, and whether or not it maintains its standards of reliability after it has been taken to a teacher's home a couple of dozen times in the boot of his car.

Other microcomputers are possible as school systems and some are further ahead than the 380Z in terms of floppy disc hardware and software but a careful look at all aspects including prices and quality led the Berkshire Working Party to recommend the 380Z. Although it did not come off the production line until October, 1977, my first feeling of confidence in its future occurred in June when I gazed in wonder at the eight small chips which comprised 16K bytes of memory. They cost a little over £300 (now appreciably less) and symbolised the 380Z secret weapon. The "Auto Refresh" feature of the Z80 microprocessor encourages the use of these dynamic RAM chips. They are simpler, cheaper, and have greater capacity than static RAM chips and this simplifies everything. Only two small boards are needed for the minimal system as defined by MUSE. The power supply copes easily and heat generated is negligible. Reliability should not be, and is not, a problem, and one can feel relaxed about maintenance costs. A safe design became an easy task. Further bonuses are an 80 x 72 graphics facility and expandability to 32K bytes of memory by just plugging in eight more chips.

It is a common myth that software for microcomputers is not advanced. But the "Software Front Panel" of the 380Z represents a leap forward in the teaching of machine code and machine organisation. The display of memory contents, registers, loading and running programs is a delight to anyone who has had to work with binary switches, paper tape or teletype output.

Other languages on Z80 machines include BASIC, ALGOL, FORTRAN, COBOL, though the last two are not promised for the 380Z. However, it would be strange if someone does not make them available in the not too distant future. A further point is that since the Z80 includes all the Intel 8080 instructions, software from that machine is transportable to any Z80 based system.

Perhaps the main disadvantage of Z80 based systems is that the machine code is made complex by the fact that it serves the twenty-two special registers of the central processor and also because it retains some of the idiosyncratic structure of the Intel 8080 from which it was developed. One also has reservations about the official Zilog Z80 assembler language. The solution to this problem is available in a carefully defined subset of the machine and the machine language together with a more conventional assembler language. These and other educational materials are available through Educational Micro Users (EMU). Chris Robinson plays a leading role in this work as he has in the whole microcomputer in schools development. There is always a gap between what manufacturers do in terms of software and support materials and what schools need. It is hoped that EMU will fill the gap.

The most obvious deficiency in the minimal system as defined by MUSE is the lack of a printer and a number of ways of getting hard copy have been suggested including: acquiring an old teletype; getting cassettes copied to a printer on a centrally placed system; transferring to mark sense cards in the final stages of development of a program if such a service is offered in the area. Obviously the most desirable method of getting hard copy is to acquire a good quality printer such as the 30 characters per second dot matrix printers provided by DEC, TREND, TRANSTEL or DATA DYNAMICS. Prices vary in the range £870 to £1,050.

On the question of printers, however, a note of caution should be sounded. It has been observed that when a computer system is enhanced by getting a faster printer, say twice as fast, the major measurable effect is that twice as much paper is used. In schools particularly, users will waste as much paper as they are allowed to by getting more frequent listings and runs of programs or dumps of files, core images, etc. This is not only a waste of machine time and paper; it develops bad habits which can be continued through higher education and are observable in industry. There are, therefore, educational reasons for restricting access to a printer as well as financial ones. A splendid solution would be for perhaps three or four schools to share a robust lightweight printer at a cost to each in the range of £220 to £350. This is an opposite arrangement to an old Berkshire idea of giving schools a teletype each and allowing a minicomputer to circulate.

A microcomputer at about £1,100 and a share of a printer at about £300 can give a school thirty or forty hours of computing each week, perhaps more. If a mark sense card reader is available somewhere in the area as well, batch processing can also be done at low cost. While there are still things that one cannot do, the technology is moving on and systems will get better. At this time, many schools will wish to own a 380Z or something similar and the detailed comments of the Berkshire Working Party are appended.

Whether or not they have opposed the idea of computers in schools in the past, many computer educationists would now accept that computers are coming into classrooms and that this is a good thing for individuals and for the nation. The task of the next few years is to make sure that the machines are used wisely.

REPORT FROM THE BERKSHIRE WORKING PARTY ON MICROCOMPUTER SYSTEMS FOR SCHOOLS AND COLLEGES, December 1977

The members of the working party were:

Roy Atherton	(Bulmershe College)
Chris Davies	(Wellington College)
Bernard Foot	(AWRE, Aldermaston)
Linda Foot	(Kennet School)
Pat Galindo	(Easthampstead Park School)
Marion Holden	(Bulmershe School)
Chris Robinson	(LEASCO Software Ltd.)
John Shearwood	(Waingels Copse School)
Clive Snashall	(Theale Green School)

1. The Berkshire Working Party agrees with the MUSE principles.

2. It is difficult to be sure that information is up to date and in these circumstances it seems inappropriate to make confident judgements about the "Best" system available. However, we have considered a number of systems and we believe that the 380Z 16K system represents good value and conforms reasonably closely to the MUSE principles. We recognise that other systems exist in schools and that these will be developed and others will be purchased. We see no harm in this and there are positive advantages, provided that satisfactory maintenance and other support can be arranged. The case for the 380Z is stated below in the light of the agreed principles.

3. 3.1 **The Closed Box Principle**

   The 380Z is an excellent example of a coherent and economic design. The two circuit boards contain 16K of memory, CPU and control circuitry and interfaces for keyboard, TV and cassette. External wiring is minimal.

   3.2 **The Minimal System**

   The 16K 380Z is a minimal system as specified, Its 8K BASIC is excellent as an interpretive system. Particularly good features are the string handling and machine code subroutines. CESIL and other languages are expected. The ZILOG assembler is regarded as unsuitable for schools. An educational subset (ED80) has been defined and will be available. We feel that this subset is suitable for A-level work.

3.3 **Expansion**

There are nine empty slots available for expansion. A serial interface and floppy discs are expected soon. Multi user BASIC is not yet available. There are no immediate plans for a mark sense card reader interface.

3.4 **Safety and Security**

The box seems satisfactory but the lid needs to be locked or screwed down and some form of fixing device is necessary to prevent unauthorised removal.

3.5 **Maintenance**

Experience to date is most encouraging and maintenance arrangements are satisfactory. Boards may be posted or the complete machine can be sent fairly easily to the makers.

3.6 **Standardisation**

There is little doubt that the majority of microcomputers purchased in Berkshire will be 380Z systems. Establishments in Lancashire, Hertfordshire, Devon, Somerset, Buckinghamshire, Oxfordshire and ILEA have shown positive interest. While the pressures against standardisation are strong it seems that there is enough interest in the 380Z to make it viable from this point of view.

4. **M6800 Systems**

   Schools which cannot find the money for a 380Z system, or for other reasons, may decide to purchase another system. The M6800 systems are widely used and offer a machine structure and assembler which are very suitable for educational purposes. In these circumstances we feel that particular attention should be paid to the coherence of the design and the quality of the support services available.

5. The state of the microprocessor market is developing rapidly and is likely to continue changing up to the nineteen eighties and beyond.

Notes

(1) The Berkshire Working Party is an agency of the Berkshire County Council Working Party on Computer Education and of the School of Inservice Education and Research of Bulmershe College.

(2) Full details of the 380Z are available from SINTEL, P.O. Box 75, OXFORD.

(3) Details about EMU are available from Mr. C. P. Robinson, 6 Stonehaven Drive, Woodley, Berkshire.

Micro-computer Systems: Revolution or Revolt?

by Michael Coleman, Portsmouth Polytechnic Computing Centre

'Revolution' is a strong word; a word to be used sparingly. There is, however, ample evidence to suggest that, in the field of Computing, we are indeed experiencing a revolution. A revolution founded upon the low-cost microprocessor.

The effects of the phenomenal development of these devices are likely to be felt as strongly in our schools and colleges as anywhere. Changes are inevitable, both in what is taught about computers, and how it is taught. Indeed, the microprocessor is being proclaimed in many quarters as the saviour of schools computing — the wherewithal for a computer in every school.

But revolutions have a nasty habit of producing as many problems as they solve. The French revolutionary Pierre Vergniaud wrote: "There was reason to fear that the Revolution, like Saturn, might devour in turn each one of her children". Perception such as this (justified in Vergniaud's own case — Robespierre had him executed!) seems to be in little evidence with regard to the microprocessor revolution.

Computer Education 28, for instance, contained a number of articles on micro-computer systems; each article applauded the advantages of such systems and failed to mention the disadvantages. But disadvantages there certainly are.

Consider firstly the question of finance. It is certainly true that the price of micro-computer systems now put them within the reach of a school's computing budget. But the prices of such systems are being deflated by two factors: the global decrease in the price of hardware, and the absence in the system of any significant amount of software. It is this latter point that is the problem.

If the computer is to be used to support a range of courses, from computer appreciation to 'A' level, say, then a further capital outlay is likely to be necessary to augment the software provided. Moreover, as demand increases, and manufacturers produce bigger and better applications packages, compilers and so on, the school will require funds to keep their system up to date. Now, the price of software is most certainly *not* decreasing. Its continued increase is, in fact, guaranteed simply because software is produced by people not machines.

As a simple example, consider the Commodore PET computer. The BASIC ROM supplied with the system is insufficient to support any GCE course. Step number one, therefore, must be to purchase an Assembly program — at about 10% of the cost of the whole system! Multiply this by the number of schools into which such a computer has been installed and the figures start to mount.

Finance also raises its head when considering a second problem area, the provision of a computing service. Once a school takes the major step of basing its computer science teaching on its own micro-computer system then, most importantly, that system must be reliable. Time, tide and GCE examinations wait for no man; a computer that is out of action for a length of time causes immense disruption in an educational environment. Thus some sort of maintenance contract will need to be paid for, simply to ensure that hardware failures can be overcome quickly.

And how about software? That, too, can go wrong, or, more accurately, not work according to its specification. The present situation is not too clear regarding many manufacturers' attitude to software maintenance. I cannot believe that software disclaimers such as *"The complex and extensive software of the PET computer has been thoroughly tested and is believed to be entirely reliable. However, no responsibility is assumed by Commodore or your sales agent for inaccuracies"* will be with us for long. Disclaimers are not good for business. No, the likely development is the emergence of companies (such as Petsoft) who will develop and maintain software for particular micro-computers — at a price.

Another requirement in running a computer service is, of course, manpower. The advocates of 'a computer in every school' talk, quite rightly, of future expansion (future money!) toward batch processing and multi-terminal capabilities. Equally correctly they look towards the system servicing the needs of hundreds, rather than tens, of students. All of which points, inevitably, towards the expense of appointing somebody whose *sole function* it is to run the computer. Even the most enthusiastic band of teachers will eventually tire of the slog of running a computing service.

Consider, finally, the educational problems. How far will a minimal micro-computer configuration of a keyboard, 16K bytes of memory and interfaces for a T.V. screen and a domestic tape recorder really serve the needs of a voracious computing population? Certainly its very existence gives the student an opportunity to gain hands-on experience. But a single computer can only have one, or at most two, pairs of hands on it at a time. This implies that each student operates the machine for a small period of time; in practice, since computing tends to be addictive in all its forms, the enthusiasts will use the computer for large periods of time and the not-so-enthusiasts will lose out.

Moreover, should the teacher be satisfied with such a facility? Surely the demand should be for a service that aids the teaching of his subject in the way that it is actually conducted, namely, to a class, not to an individual. Should the minimum not be a batch processing capability? Or a multi-terminal, or multi-computer configuration giving hands-on experience to every member of the class at the same time?

One clear indication of the micro-processor revolution is that it enables a relatively inexpensive Computing Centre to be created for schools use. This, for instance, certainly seems to be a way in which the costs of running a service could be spread over a number of sites instead of being incurred individually. Furthermore, it provides a realistic way in which a classroom's worth of terminals could be made available to every school. Computing Centres could be based at selected schools and colleges or, as the result of some collaborative arrangement, at local computer installations. Polytechnics and Universities, for instance, would be ideally suited to such an arrangement. Wherever it was located, the centre would be responsible for serving the requirements of a number of schools, both in terms of facilities and advice.

The use of the term 'centre' has also, in the past, implied that the user visits the facility, the latter being immovable. Again, the advent of micro-systems has been particularly notable for the fact that they can be transported from place to place without problem. There seems no reason to doubt the viability of Mobile Computer Units — vans equipped with anything from a simple system to multi-computer configurations which take the service to the user. Used in combination with a Computing Centre, this approach could be a highly cost-effective way of providing class teaching systems for schools use.

In this way, a school with the desire, capacity and expertise to run a computing service can do so — and receive the necessarily increased financial backing required. Those schools not wishing to cope with their own computer can still benefit from the right sort of educational facilities at a local centre.

The Microprocessor Revolution should be of immense benefit to computer education in our schools and colleges. However, recall if you will a conversation regarding another Revolution:

Louis XVI: "Is it a revolt!"
Duc de la Rochefoucald-Llancourt: "No, Sire, it is a revolution."

Did Louis get it right?

The Real World

WE HAVE five computers in and around the *Practical Computing* office. At the time of writing, only one is working. To give credit where credit is due, that one is the Pet. What of the others?

We have a £2,000 household name system which is supposed to be doing a useful job in the office. For many weeks, nothing has emerged from it but garbage and the sour smell of burnt ROM. It has been twice already for repair and will no doubt soon be making the journey a third time. We have a second household name with discs. Sometimes it loads, sometimes it doesn't. "Put your finger in the hole and stir", say its masters; we stir and it loads not.

For three days we have been trying to get another cheap but trendy computer to drive a popular matrix printer. Those days have been filled with endless three-way telephone conversations between the computer people, the printer people and ourselves.

One says one thing, the other something different, but it is of no importance — the machine gives not a click. Then a colleague pulled the computer power plug out of the wall accidentally and ended the debate, for now the machine will not work at all.

We have a mini which behaves reasonably well, as it should, considering how much it cost, but on some days it gets into a loop and has to run through its half-million irrelevant calculations before it will re-emerge several hours later into the real world.

Even if the machinery were working correctly, would we be happy? That is a big question and the answer for today must be in the negative. If the machinery is fragile, the software it runs is worse. The Pet works but the batch of sample tapes we are trying from a major software supplier often do not.

We have a good Life game but there are certain input patterns which lock-up the screen and the only remedy is to re-load. The business system alleged to be running on the second household name has no visible escape from many of its routines. It will sit, showing what you don't want to know, its discs buzzing and clicking like demented chickens, and no combination of keys will budge it.

In the wider world, some canny fellow has offered a prize of £100 for a British business system running on a microcomputer which will run for a month, or perhaps a week, without crashing. We think he may not have to pay in the foreseeable future.

Even programs which run are subject to problems. For instance, we published a Zombie game in our June issue. It was written in Basic on a PDP, and you would think that since Basic is a universal language, anyone who wanted to use it could do so.

Yet the office telephones rang for a month with the plaints of unhappy readers who could not get the wretched thing to run. The trouble, as it emerged after many aggravating hours, was in the random number generator. There are dialects of Basic, and one machine's RND is another machine's poison.

The whole thing puts one in mind of the early days of motor cars — well, what one has been told about them, since we're not that ancient. Cars then were very like microcomputers now. There were dozens of different kinds, allied only by their unreliability, incompatibility and the fanatical enthusiasm of their owners.

Those owners would speak disparagingly of "horse and buggy" ideas and would happily spend six hours travelling six miles because they had had the fun of fitting new big ends after the second mile, re-wiring the ignition after the fourth and then retrieving — as Kipling so movingly describes in one of his stories — the contents of a ball-race dropped along six furlongs of un-metalled road.

Well, so it is now. The difference is that the fans of microcomputing have a more gullible public. Many people are being persuaded to ride in our new bangers. They are holding their hats and waiting for the big thrill. It may come, or, what is more likely, the whole mess of machinery may blow up, cover all concerned with hot oil and sit lifeless by the roadside.

In plain terms, microcomputers and their software are not yet robust enough for the real world. At best, they are being built and sold by people who are fascinated by them, take great care how they use them, and expect others to do the same.

There is, of course, also a fair share of cowboys who do not care about these things, but we shall not mention them. The trouble is that the population of micro-freaks, who have the brain power to wrestle with raw micros, is almost used up. If our industry is to develop, the machines must be sold to people who want answers to their problems, not more problems still.

So far we have managed to make micros do more or less what we want. What we have not yet done is prevent them doing what we do not want, and there is a huge difference between these two states of affairs. For, if there is one way of doing something correctly, there are a hundred ways of doing it wrong.

If it costs £1 to make the thing which works correctly sometimes, it costs £10 to make it work most days and £100 to make it work properly always.

That is particularly true of software. Any intelligent fool can write a program to add a firm's invoices and deduct 15 percent for VAT. It takes a clever chap to write a program which will load and run, time after time; which will not crash or lock up no matter what silly key Minnie the office mouse hits. But it must come.

Next month — Teletext

In a roundabout way, that leads us to the subject of next month's delights. In the October *Practical Computing* we shall be devoting a good deal of space to the important subject of Teletext, or data-at-a-distance.

We shall look at the dissemination of software by television and wire, at *Ceefax*, *Oracle* and *Prestel*; at the coming generation of home data stations with micros built into television sets, which can take-in data and programs from central banks, or transmit them back along the line to friends, colleagues or for general use.

We shall be looking, in short, at the techniques and hardware which will link the many individual micro-revolutions into one — which may well, in time, disperse cities and jobs as we know them today.

It seems likely that the most important effect of this development for *Practical Computing* readers will be to create a vast new market for software. Reliable, never-crash software. It may be that many people who are now hobbyists will work full-time at home, writing programs to satisfy this market and having a better life than they do now. We hope so, anyway.

PROGRAM OF THE YEAR COMPETITION

This competition has now closed, slightly late because of postal problems, and we are happy to report a heavy crop of entries — at first glance of a high standard. It was not possible to do them justice in time to include the results in this issue. All entrants will be informed of the results by post and we hope to announce them in the October issue.

E

BASIC THE FIRST STEPS

Philip Couzens

Step after step the ladder is ascended
George Herbert, *Outlandish Proverbs* (1640)

BASIC, Beginners' All-purpose Symbolic Instruction Code, was originally designed at Dartmouth College, U.S.A. to be a simple introductory language from which students would progress to the more powerful high level languages such as FORTRAN, COBOL, ALGOL and PL/I. The grammatical rules of BASIC are so few and simple that the user can quickly program the computer to solve his problems.

Although BASIC is so simple, it is still a very useful and powerful high level language suited to both mathematical and business situations alike. Due to its simplicity, interpreters and compilers for the language are small, making BASIC particularly suitable for microprocessor based systems in which core storage may be limited.

If one has experience of one of the popular high level languages, particularly FORTRAN, then converting to BASIC should take only a few hours. The remainder of this article is intended for those with no previous programming experience, but might also be useful for those wishing to change from another language.

BEFORE PROGRAMMING

Before programming can begin, the system has to be switched on and the BASIC interpreter has to be loaded into memory. The method of doing this varies considerably from system to system, obviously instructions for doing this cannot be given here. It will be assumed that the system is ready and working in BASIC.

At least two of the control keys should then be identified. These are the 'carriage return' key which is pressed every time something is entered at the keyboard, and the 'escape' key which is used to stop the computer doing a program. If no particular key is marked 'escape', then the instruction is probably given by holding down the 'control' key while pressing another key — check in your manual for this.

One further key that is useful is the 'rubout' or 'delete' key (sometimes back-arrow or underline). This has the effect of erasing the last character which was typed. Pressing it twice will erase the last two characters, and so on.

WRITING A PROGRAM

Any line which is now typed in at the keyboard will be treated as a command, and this will be executed when the 'carriage return' key is pressed. If a line begins with a number (any integer between 1 and 10000), then that line will be stored as part of a program.

So, for example, the first line of a program is typed in:

```
10 PRINT "HELLO"
```

This, of course, is followed by a carriage return. *Every program in BASIC should be terminated by the instruction END, so the last line is typed in:

```
20 END
```

There now exists a complete program which will be typed out at the terminal if the command LIST is issued at the keyboard; the lines of the program are typed out in order of their line numbers, not in the order in which they were typed originally. To execute the program the command RUN is given. This program simply prints HELLO and then stops.

If it becomes necessary to insert further lines into the program, then this is done by choosing a suitable line number which has not already been used. For example, a line is inserted between the two existing lines:

```
15 PRINT "HELLO AGAIN"
```

If the command LIST is now given, the whole program is displayed at the terminal:

```
10 PRINT "HELLO"
15 PRINT "HELLO AGAIN"
20 END
```

*This is not necessary on all systems, but it is useful to conform to standards.

If it becomes necessary to change a line, then this is done by creating a new line with the same number as the one to be replaced and retyping the whole line. A line is deleted by typing in the line number followed by 'carriage return'.

So it is seen how a program is built up. It is usual to choose 10 for the first line number and to go up in tens, that is 10, 20, 30 . . . This leaves plenty of unused numbers so that lines can be inserted if required. The command RENUMBER, which is supported by most systems, changes the line numbers to fit this pattern.

When it is wished to lose the existing program and start a new one, the instruction *NEW is given. The old program is lost completely and a new one may be commenced.

The important BASIC commands which we have met so far are:

(i)	RUN	— the current BASIC program is executed
(ii)	LIST	— the current BASIC program is displayed at the terminal
(iii)	NEW	— the current BASIC program is deleted so that a new one may be written
(iv)	RENUMBER	— the lines are renumbered to fit the pattern 10, 20, 30 . . .

ARITHMETIC IN BASIC

Numbers are expressed in BASIC in three ways:

(a) Integers — written without a decimal point, e.g. 4395

(b) Decimals — written with a decimal point, e.g. 31.416

(c) Numbers written in standard form. $1\cdot08 \times 10^{-4} = 0\cdot000108$ is written as 1.Ø8E-4

A negative number is preceded by a minus sign in the usual way, for example −4395, −31.416, −1.Ø8E−4.

These notations can be used as and when convenient.

*On some systems the command is SCRATCH

Numbers will be typed out by BASIC in the most suitable form, the user is normally not able to choose. Unless a number is particularly large or small the integer or decimal form will be chosen.

All arithmetic expressions have to be typed in at the keyboard, and therefore the notation is limited by the facilities of the keyboard. Addition and subtraction are denoted by the usual signs + and −. Multiplication is denoted by the sign * and division by the symbol /. The up-arrow ↑ is used for exponentiation (raising a number to a power), and brackets are used as in ordinary arithmetic.

As an example, the expression

$$\frac{(3\cdot14 + 2 \times \frac{1}{2})^4}{9^2 - 4\cdot6 \times 10^6}$$ could be evaluated

in BASIC by running the program:

```
10 PRINT (3.14+ 2*1/2) ↑4/ (9↑2 − 4.6E6)
20 END
```

BASIC uses standard algebraic hierarchy when performing arithmetic, the order of operations being given by BEDMAS (brackets, exponentiation, division = multiplication, addition = subtraction). Operations of equal priority are performed from left to right. So the answer to 1 + 2 * 3 is given as 7 since the multiplication is done before the addition. Notice that 12/3*4 is given as 16. The order of performing operations is in fact exactly the same as one would use in ordinary arithmetic.

LIBRARY FUNCTIONS

All BASIC systems (except those called Tiny Basic etc) should support at least the following set of library functions:

Function	*Meaning*
SIN (x)	the sine of x, where x is an angle in radians
COS (x)	the cosine of x, where x is an angle in radians
TAN (x)	the tangent of x, where x is an angle in radians
ATN (x)	the principal angle (measured in radians) whose tangent is x
SQR (x)	the square root of x
LOG (x)	the natural logarithm of x
EXP (x)	the value of e^x
ABS (x)	the absolute value of x, ABS (−1.9) = 1.9, ABS (1.9) = 1.9
INT (x)	the largest integer not greater than x, INT (6.3) = 6 INT (−6.3) = −7
SGN (x)	the sign of x, has value 1 if x is positive, 0 if x is 0 and −1 if x is negative x is positive, 0 if x is 0 and −1 if x is negative
RND (x)	a random number between 0 and x

In the above, x, called the argument of the function, represents an expression of any complexity, and may include other functions. Note that all angles are measured in radians so that degrees have to be converted (multiply by $\pi/180 = 0\cdot0174533$).

It should now be possible to use BASIC as a scientific calculator.

EXAMPLES.

(1) Evaluate $\sin 30° + \sqrt{2}$

```
10 PRINT SIN (3Ø * Ø.Ø1745) + SQR (2)
20 END
```

(2) Evaluate $\frac{2}{3}(\ln 3 + e^{\sqrt{2}})^3$ ↑

```
10 PRINT 2/3 * (LOG (3) + EXP (SQR(2)))   3
20 END
```

Note that several levels of brackets are allowed, but one should be careful that they are closed correctly.

VARIABLES IN BASIC

Variables are used in BASIC in a way not dissimilar to that of ordinary algebra. The equals sign, however, is used in a very different way and this is stressed by the use of the word *LET. For example, consider the BASIC statement

```
10 LET A = 4.2
```

This assigns a value of 4·2 to the variable A. The meaning is made clear by reading the instruction as 'let A become 4·2'. The expression on the right hand side of the 'becomes' sign can be of any complexity, and can contain any number of variables (including the variable on the left hand side). The left hand side must contain only one variable.

Consider the statement 3Ø LET A = A + 1. Although this would be nonsense in mathematics, it is allowed in BASIC, meaning 'let A become what it was plus one' and has the effect of increasing A by one. It is, indeed, a very useful and much used facility.

*In many versions of BASIC the word LET may be left out.

Ordinary variables which represent numbers in BASIC can be given any of the 286 names which consist of a single letter: A, B, C, . . . Z, or any letter together with a single digit:

```
AØ, A1,. . . A9
BØ, B1,. . . B9
  :        :
ZØ, Z1, . . . Z9
```

So to evaluate the expression given earlier,

$$\frac{(3{\cdot}14+2\times\frac{1}{2})^4}{9^2-4{\cdot}6\times10^6},$$

the program given here would also be suitable:

```
10 LET A = 3·14 + 2 * 1/2
20 LET B = 9*9 − 4.6E6
30 LET C = A ↑ 4/B
40 PRINT C
50 END
```

THE PRINT AND INPUT STATEMENTS

The PRINT statement has already been met, but is now considered in greater detail. One or more items can be printed by a single PRINT statement. The line 1ØØ PRINT 2/3, X, 2.3 will type out the value of $\frac{2}{3}$, that is 0.666667, followed by the numerical value of X followed by 2·3. Since the items in the PRINT statement are separated by commas, they will be typed out in columns across the page. Alternatively, if they are separated by semi-colons then they will be squeezed together. Anything which appears between quotation marks in the PRINT statement will be printed out exactly as it appears in the program. Plain numbers and arithmetic expressions are allowed in the PRINT statement. A typical line in a program would be:

```
5Ø PRINT "THE ANSWERS ARE: ", X; Y, 39/3; Z
```

This will cause the following to be typed when X = 0·333333, Y = 100, Z = 6

```
THE ANSWERS ARE: 0.333333 100  13 6
```

The effect of 6Ø PRINT is to advance one line of paper. If a PRINT statement ends with a semi-colon then the printing head will remain in its last position until another PRINT statement is encountered.

The INPUT statement is a method of assigning values to variables from the keyboard *as the program is running.* When an INPUT statement is reached during the execution of a program, the program will stop and continue only *after* a number has been typed in at the keyboard. The variable mentioned in the INPUT statement will then have the value which was typed in. The input of a number is prompted by a question mark being typed.

So then, if it is wished to read in a value for X during the execution of a program, this is done by the statement:

```
1Ø INPUT X
```

More than one variable may be assigned a value by a single INPUT statement:

```
2Ø INPUT U, V
```

When this line is reached in a program, then it will stop and continue only after two numbers have been entered. U will have the value of the first one and V the value of the second one.

On many systems it is possible to change the prompt offered by the INPUT command by placing the required replacement in quotation marks in the statement thus:

```
1Ø INPUT "TYPE IN THE VALUES OF X AND Y: ", X, Y
```

Note that the comma is required after the last quotation mark.

Example: This program will read in two numbers and type out their average.

```
1Ø INPUT A, B
2Ø PRINT (A+B)/2
3Ø END
```

To execute the program, the command RUN is given. At line 10 values for A and B are required. The request for these values is indicated by a prompting '?'. The values must now be entered with a comma between them, followed by 'carriage return'.

```
RUN

?4,6
5

END AT ØØ3Ø
```

Now to change the program some of the lines are retyped:

```
1Ø INPUT "TYPE IN THE TWO NUMBERS: ", A, B
2Ø PRINT "AVERAGE IS: "; (A+B)/2
```

If the command LIST is given again, a list of the amended program is obtained. A run of the new program will produce this:

```
RUN

TYPE IN THE TWO NUMBERS: −4.2,3.9
AVERAGE IS: −Ø.15

END AT ØØ3Ø
```

THE GOTO STATEMENT

A program is normally executed in order of the line numbers. The GOTO instruction, however, causes this order to be changed. For example

```
5Ø GOTO 1Ø
```

This statement will cause the program to jump to line 10 rather than carry on to the next line after 50. Any existing line number can be used in the GOTO statement.

Example: This program will type out the natural numbers, together with their squares and square roots.

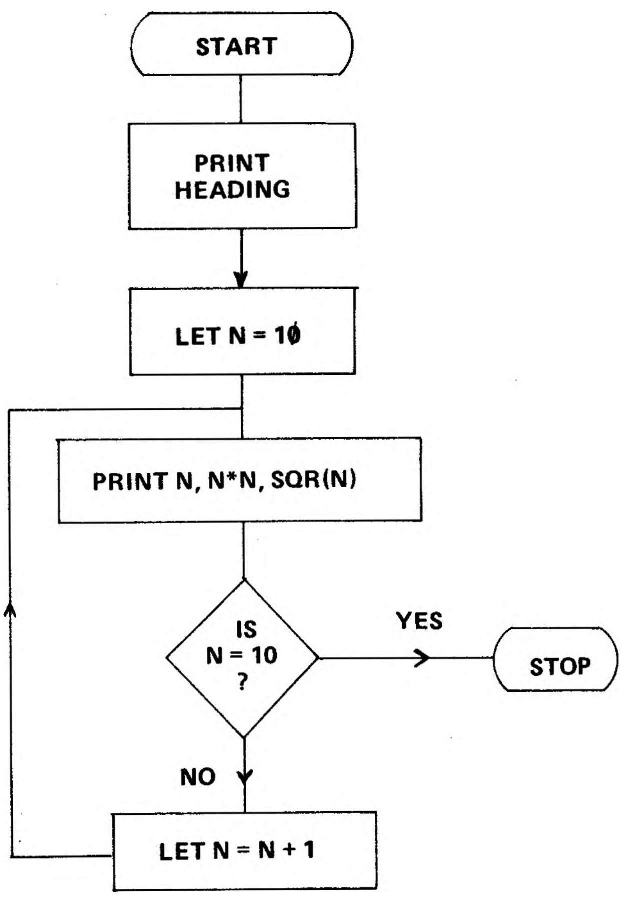
```
1Ø PRINT "NUMBER", "SQUARE", "SQUARE ROOT"
2Ø PRINT " ____________________ "
3Ø LET N = 1
4Ø PRINT N, N*N, SQR (N)
5Ø LET N = N + 1
6Ø GOTO 4Ø
7Ø END
```

Work through this program and you will see that it contains an infinite loop (it will keep running for ever). The only way of stopping this program is to press the 'escape' key during execution. Naturally, this is considered bad programming technique, but it has served a useful demonstration purpose here.

CONDITIONAL BRANCHING
To GOTO statement provided us with unconditional branching, but it is possible to ask a question before branching, and branch only if the answer to that question is yes. It would be useful in the example above to ask at line 45 the question 'Is N equal to 10?' — this would avoid the infinite loop. If the answer to this question is yes, then the program will stop. Otherwise the program continues to run as though the question had not been asked. The question is asked in BASIC by the statement.

```
45 IF N = 1Ø THEN STOP
```

The flow-chart for the program now looks like this:



The program is:

```
1Ø PRINT "NUMBER", "SQUARE", "SQUARE ROOT"
2Ø PRINT " ______________________ "
3Ø LET N = 1
4Ø PRINT N, N*N, SQR(N)
45 IF N = 1Ø THEN STOP
5Ø LET N = N + 1
6Ø GOTO 4Ø
7Ø END
```

In general, a question may be asked containing any of the relational operators:

=	equal to
>	greater than
<	less than
>= or =>	greater than or equal to
<= or =<	less than or equal to
>< or <>	not equal to

A test of this sort can result in any of the BASIC statements being executed,

e.g. `2Ø IF A> = B/C THEN PRINT A`

This will result in A being printed if and only if A is greater than or equal to (B divided by C).

```
11Ø IF A*SIN(T)<=1Ø THEN GOTO 21Ø
```

If the result of this test is true, then the program will jump to line 210, otherwise execution will continue as normal.

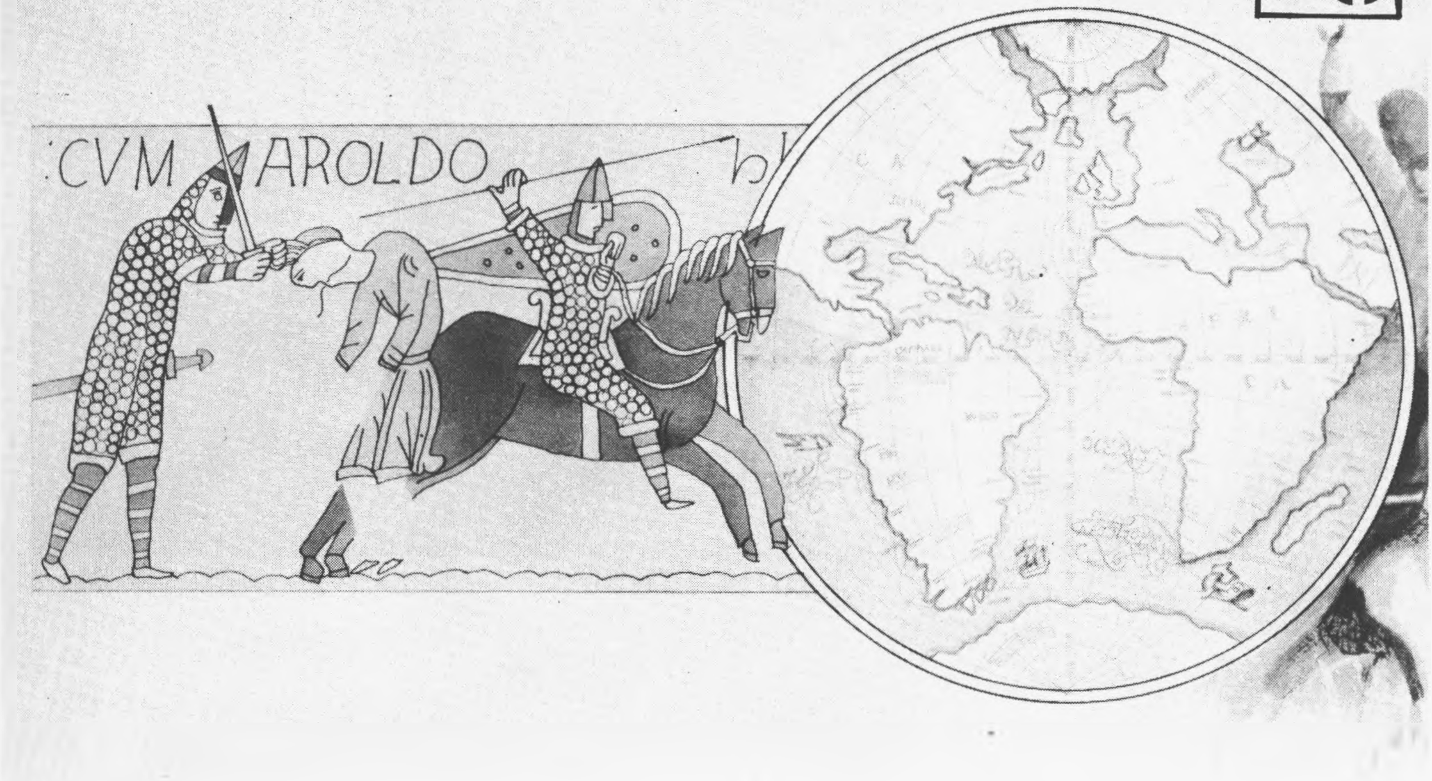
THE REM STATEMENT
When writing a long and complicated program, it is often useful to have explanations along the way actually written as part of the program. This is done in BASIC by the REM statement. If a line begins with the letter REM (after the line number), then that line will be ignored when the program is executed.

e.g.

```
1Ø REM THIS IS A REMARK WHICH WILL NOT INTERFERE WITH THE PROGRAM
2Ø PRINT "THE SAME AS USUAL"
3Ø END
```

FROM HERE?
Although only a few of the facilities available in BASIC have been described here, the reader should now be able to write programs to solve even quite complicated problems.

BASIC was designed for the user in the hope that the computer should be used as a tool to find solutions to what might otherwise be insoluble problems or at least problems which would involve large amounts of time when tackled by hand. But the user often becomes interested in computer programming for its own sake, finding it intellectually satisfying to have written a concise and effective program. By writing simple programs in BASIC one can begin to appreciate the pleasure which can be obtained from computer programming.



Historical simulation

Richard Ennals writes about the ideas which won top prize in the schools section of our Christmas competition. The entry was from the history and computer studies departments of Sweyne School, Rayleigh, Essex. Ennals is developing some of his history simulations on the Research Machines 380-Z he won.

AT FIRST SIGHT it might seem far-fetched to use the computer to assist in historical simulations in the classroom of a comprehensive school but I have been brooding on the variety of ideas involved over a period of at least 10 years.

During my own sojourn in a VIth form I became interested in the links between the disciplines. A scholarship to Massachusetts gave me the chance to learn Basic and an introduction to Probability and Decision Theory.

Later, at Cambridge, I spent a year immersed in the linguistic philosophy of Wittgenstein, especially his theories of games and languages. My studies included mathematical logic before I switched to a history degree.

While a student, I wrote and produced a number of plays on historical situations, all starting from the basis of simulation and role-play. Teacher training followed and on teaching practice in Southfields, I developed my first simulation 'kit', based on the United Nations Organisation and how its members handle a series of possible crises.

Based on briefing

My first teaching post, in Mitcham, was in a department whose head, John Waddleton, had considerable skill and experience in simulations; this produced further kits on the House of Commons and the workings of a factory.

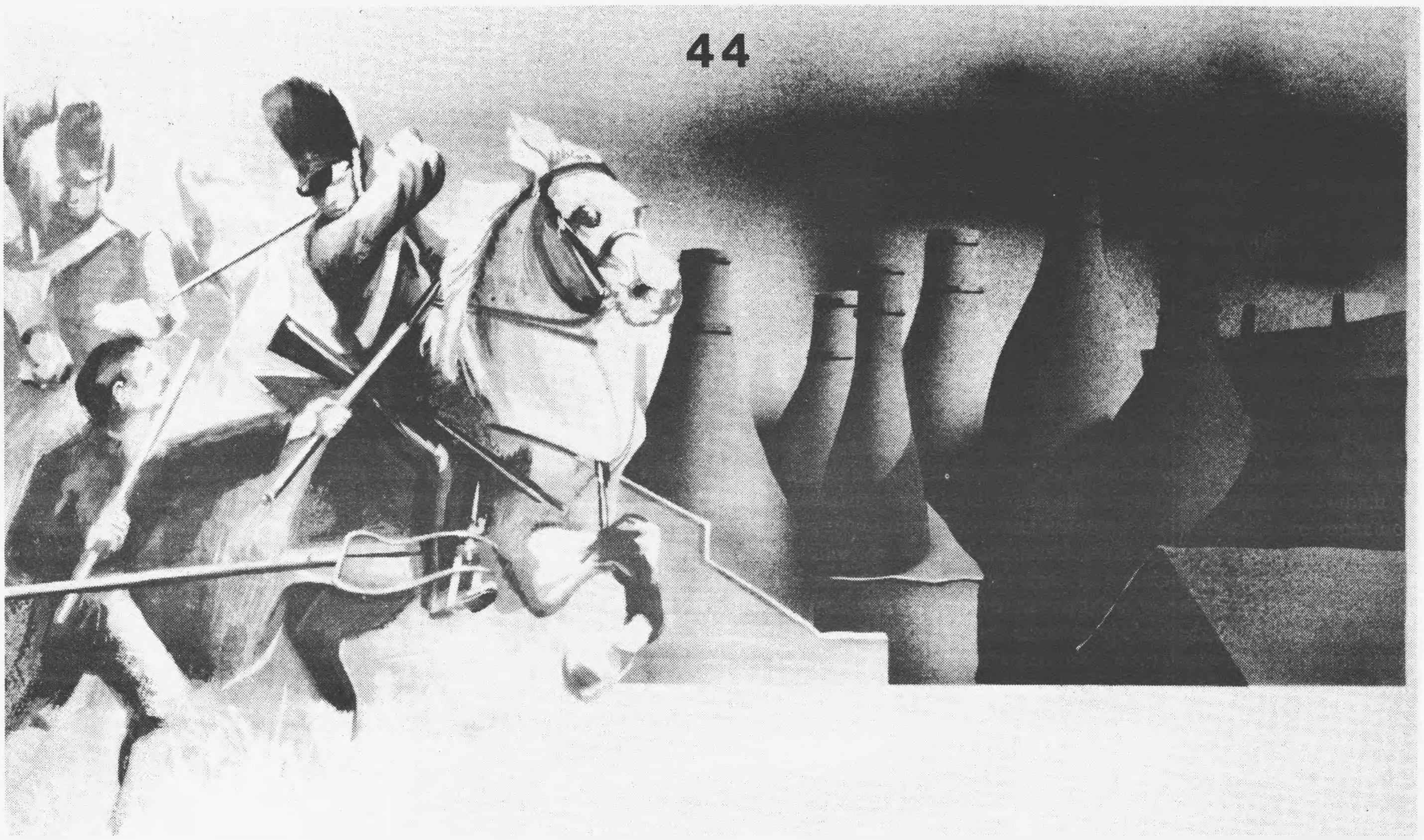
No computer facilities were available for these; the simulations were based on the briefing of the individuals, and the free development of the situation by the group in their roles.

Last September, in my present post as head of the history department at Sweyne School, I found an ideal opportunity to develop my ideas further. My colleague, Martin Frampton, shared my interest in games, and I developed a range of new

HOW THE IDEAS WORK

Our use of the computer for history simulations on a set of principles from the classroom teaching situation:

- The logical structure of a computer analogue of a historical situation is similar to the equivalent classroom simulation, with the possible difference that the simulation game normally involves decisions by a number of participants rather than by a single player. So it should be feasible to make use of the computer analogue to enhance the classroom simulation, and vice versa.
- If the appropriate programming and role preparation are provided in advance, the input of data and classroom role-play can be carried-out without over-direction from the teacher.
- The computer can aid the simulation participants in carrying-out historically credible decisions, given the accuracy of information already stored.
- There is philosophically a clear link between mathematical and computer logic, language structure and the rules of games — as outlined, for instance in Wittgenstein's *Philosophical Investigations* and Braithwaite's *Theory of Games as a tool for the Moral Philospher).* This has clear, if undeveloped, practical implications, especially for people like teachers who have the task of explaining complex processes.
- The computer should not be used simply for its own sake of so doing, and it should not perform functions already carried-out adequately by other means. It should help clarify the decision processes involved and enhance the learning experience of the class. It should enable the participants to be aware of irrelation of two or more variables, and the consequences of their combination—hard to convey by conventional means.
- The computer work involved in class should not be too complex, with easily-understood input data and comprehensible conclusions printed-out or displayed on a screen.



kits based on the Russian Revolution, the Age of Discovery, the Wedgwood firm in the Industrial Revolution, the Norman Conquest, and the League of Nations—all for mixed-ability classes.

Next to my classroom at Sweyne is the computer studies room, run by John Ward. I showed him a range of programs I had devised for use in the games; he detailed three able 'O' level students—Andrew Wood, Martin Attwood and Keith Stewart—to help me as part of their computer studies course-work. They came to history lessons where the games were being used, and devised programs to facilitate the simulation process and enhance the students' understanding.

In an evening

Then Ward spotted the competition run by *Practical Computing*. The initial entry was written in an evening but a good deal of hard work was put in by all five of the team in developing the ideas, flowcharts and programs for the final submission, which ran to about 40 pages of typescript.

We have had to work so far with strictly limited access to a time-sharing terminal on-line to the county computer centre but our students have enthusiasm to start work at 7.30 each morning.

Helpful to all

The prize of a Research Machines 380-Z should expand our ability enormously to produce innovative materials for use with our classes, and in other schools. Essex County Council has been encouraging computer education for some years and recommends the 380-Z. It is helping us to modify our rooms to gain the most advantage from this new acquisition for our two departments.

Use of a television screen should enable a whole class, rather than just one individual, to benefit from the output of the computer; and the aim in our historical work is for the computer to become a major classroom teaching aid. *Practical Computing* and Research Machines have given us a chance to show how this can be done.

PUTTING THE PRINCIPLES INTO PRACTICE

Some examples of how the computer can be of use, based on classroom experience with simulation materials devised at Sweyne School.

- **Wedgwood Potteries**
  We use a simulation kit for third-year, mixed-ability secondary students based on Josiah Wedgwood and his pottery firm in the 18th century. Class members are allotted a separate role among the people connected with the Wedgwood Pottery. Each receives a historical briefing on an individual part so that decisions will be historically authentic. A range of Basic programs has been developed to aid the management in planning of wages, prices and research policy. Data input is straightforward, and the printed conclusions go beyond the complexity of reasoning and the calculation of which the students are capable. Participants react to the computer findings, and the game continues. This relies, in part, on simple economic theory and, in part, on the provision of historically-accurate background material.

- **Explorers**
  Second-year, mixed-ability history students have been using a simulation kit based on the Voyages of Discovery in the 15th and 16th centuries; the results have been very encouraging in terms of historical authenticity, the quality of written work and the level of enthusiastic participation. A fifth-year student, Andrew Wood, has written a program which clarifies the process of trading and makes possible many of the authentic variations in commodities and prices. Further programs will clarify the choices between routes, and their consequences, and enable the 'explorers' better to plan their strategies in the light of detailed historical knowledge.

- **Russian Revolution**
  Two fifth-year students, Keith Stewart and Martin Attwood, have devised a program based on decision theory to analyse the choices facing a Russian revolutionary in 1917. This will enhance the effectiveness of our existing fourth-year simulation kit for the Revolution. As with the other kits, each member of the class is allotted a different role through whose eyes they see the events of 1917 as they unfold. Properly used, the computer can enhance the authenticity of the simulation and throw the decision involved into sharper focus, so that a complex historical situation can be brought to life in the classroom.

Involving the student

EXISTING history games for the computer rely on the interaction between the individual and the computer but the games at Sweyne School are designed to involve a whole class of mixed-ability students, and so have to use somewhat different techniques. They are not designed solely for older and/or abler students, so they cannot depend on an advanced knowledge of computers.

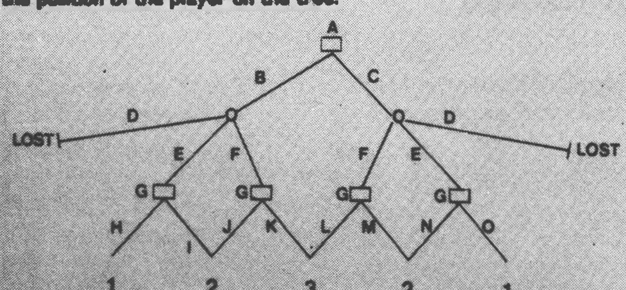
In a previous article, I described the ideas behind the use of the computer in historical simulation games at Sweyne School, Rayleigh. I described the range of games and outlined the ways in which the computer could help. This article gives more details of the way the computer works.

Applications of the computer for secondary school history teaching have been described in a recent Historical Association pamphlet edited by Joseph Hunt and commissioned by the National Development Programme in Computer Assisted Learning — *Computers in Secondary School History Teaching*. In the suggestions for simulations they have still been restricted in their range. The authors say they cannot involve the student emotionally through the use of the computer but adhere to intellectual involvement in a game played with a computer keyboard.

Our aim is to involve the student both emotionally and intellectually by using the computer to provide a simulation based on members of the class playing different historical roles.

FLOWCHART FOR RUSSIAN REVOLUTION DECISION THEORY PROGRAM

The program is designed for interactive use and works as a tree diagram. The tree diagram is terminated by the awarding of points depending on the position of the player on the tree.



□ Decision
o Random choice
| Terminator
A How do you intend to gain power?
B By legal means.
C By illegal means.
D Complete failure to gain support.
E Moderate success in early campaign
F Great popular success in early campaign.
G How do you intend to continue your campaign for power?
H Rely on the will of the people in elections.
I Improved use of legal means.
J Continued use of legal means.
K Seek support from other groups.
L Direct appeal for public support.
M Seek support from other groups.
N Continue organising popular campaign.
O Rely on spontaneous popular support.

by Richard Ennals

Our approach relies partly on theoretical and partly on technical innovation. The theoretical innovation is the application of basic decision theory to historical situations, such as the Russian Revolution, enacted in the classroom. The decisions facing key individuals are analysed, separated chronologically, and simplified into an overall structure — a tree diagram or critical path analysis corresponding to the choices made and outcomes of actions taken.

A parallel critical path analysis follows the progress of the overall simulation by stages. If this is done appropriately, the same critical path model may serve as a guide. Or, translations may be added, perhaps as PRINT statements, to a common core of logical symbols.

The computer is not being used to make decisions in the simulation. It is there to increase the ability of the players to decide, make decisions, or perhaps to comprehend or explain decisions of historical characters. A problem in historical teaching is to give the student enough information to enable him or her to make decisions. Normally, this is difficult but the computer can be used to call-up further information when required by the participants which is available only at certain stages of the simulation and once certain alternative courses of action have been followed.

Incidentally, the determining of what alternatives are followed can be done retrospectively by an observer or as laid down from a choice of alternatives provided by the computer. This can be printed; or, more suitably, it can be displayed for class use on a large-view screen.

So far, this is all theory on paper but the provision of a Research Machines 380-Z through the competition in *Practical Computing* gives us the chance to put theory into practice far more than was possible when we relied solely on a time-sharing terminal. Our new microcomputer is connected to a large black and white television screen and a small monitor, and it is about to be interfaced to the tape and printout facilities of our terminal.

We are working on the use of the television screen for the class. In our General Election simulation on May 3, once the computer had processed the results of the school it displayed the national consequences which would follow if the voting tendencies at Sweyne were repeated throughout the country.

In our simulation on June 7 of the election for the European Parliament, the results of the school-wide vote were processed and the screen provided briefing information to the (School) Members of the European Parliament when called to speak in their debate, information which enabled the participants to reach decisions on the admission of Spain and Portugal to the EEC.

One of the first games to be given computer assistance will be the Norman Conquest Game used by first-year classes. Using the principles described above, the computer will display further information as and when required by the players, to enable them to make up their minds on the basis of sufficient information.

The computer is also being used for more straightforward assistance to young historians. In the Explorers Game, for second-year students, a program simplifies the complex process of trading in different commodities. Each participant inputs what is being bought or sold and what price is being offered; they can also read a display of prices offered elsewhere.

In the Wedgwood Game for third-year students, straightforward mathematical routines help the young managers to calculate wages and prices and assist in their determination of policy.

Participants in historical simulations can carry-out decisions of a greater complexity and subtlety than normally would be possible. An individual version of the Russian Revolution game has been devised following the decision theory model outlined. It points quickly to the historical consequences of different courses of action.

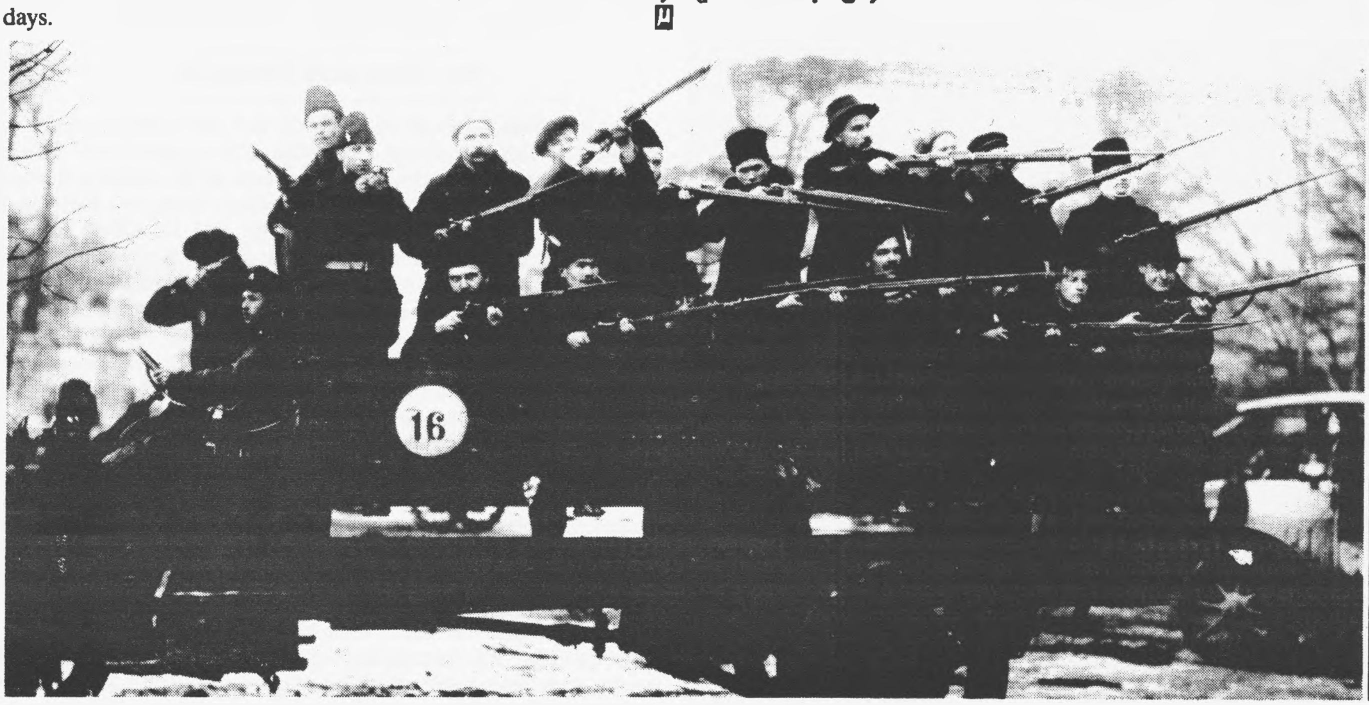
The same model can be adapted using historical translations for the rise of Hitler, and it could be developed into a game of several rounds for the individual power-seeker.

I am more interested in developing the approach for use by a group. We are trying to develop the consequences of the interaction of two or more participants using the decision models outlined, with different outcomes for the matrix of their decisions. These are early days.

TCHEIDZE, Social Democrat, right-wing president of the Workers and Social Council until suspended by Trotsky, May 12, 1917 (above).

RED Guards shooting from an armoured car in Petrograd (below).

GENERAL disorder when Leninists besieged Duma in July, 1917 and a group of Georgian officers intervened (previous page).



G

ESCALATOR PACKAGES

D PEGG
Bedford School

1 INTRODUCTION

The type of package with which we have become familiar in Computer Education is a rather bulky program or suite of programs written to teach a particular topic. There is no need for the user to have any idea of the logic behind the program or indeed to be capable of understanding the computer language. This is acceptable in most subjects, but mathematical packages may lose some of their value if they are incomprehensible to their users.

At the other extreme the pupils may be guided so that they can write their own programs in order to illustrate and reinforce a mathematical concept. This is ideal, provided that all of them have a good grasp of programming, and there is sufficient time for program-writing.

There is a third method which is a compromise between the first two. It may be called the 'escalator' method, because the user can both walk up the steps himself and get a certain amount of external help. Briefly, the teacher provides a series of elementary programs on which the pupils build. Any new concept is best illustrated at first by a simple program. Once this has been grasped, the teacher (or preferably the pupil) can suggest elaboration and generalisation, and these can quite easily be made by modifications to the program. This process reinforces what has been learnt; and makes the student feel that he or she is contributing personally.

2 PAST EXPERIENCE

I have used this principle for three years to develop a number of topics on our PDP8E three-terminal system:-

(a) Solutions to equations using Newton's method.

(b) Introduction to integration.

(c) Areas under curves, by the trapeziodal and Simpson's rule.

(d) Introduction to Differential Equations (based on David Edsall's ideas)

But I have used it to a greater extent as the basis of a course in Statistics for 1st year Sixth Formers who are not taking a Mathematics 'A' level. Broadly, the course takes two periods a week for one year. The first term is concerned with theory, the second with experiments, and the third with group projects. We use three terminals and an off-line punch, and students work in pairs. This year there are eight in the class, which is an excellent number. A maximum number would be twelve, as part of the time is spent in program-writing, so that full-time working on a terminal is not essential for each pair. Most of the pupils have no previous knowledge of BASIC, and this is acquired as the course progresses. A number of copies of Blakely and Lewis' book 'INTRODUCTION TO BASIC' are available to be read by those not using the machines.

I introduce each topic with a demonstration of the V.D.U. to the whole class using a simple taped program which they then take away and elaborate. I have copies of the elaborated programs for those who are having difficulties. Some pairs may progress faster than others and they can be guided into a deeper study of the topic, but the overall pace is kept the same for all.

3 RECENT CHANGES

Last year a four-user micro-computer system was used, with cassette backing-store by pupils. Demonstrations were given on a single-user disk-based micro-computer. The new system has been an improvement on the mini-computer, and shortly a disk backing-store will replace cassette for the terminal users.

Program examples of the method are given at the end of the article. These relate to Statistics and can be used either in the initial teaching of particular topics or as a reinforcement of earlier instruction. The programming style is at times clumsy, but ease of understanding by pupils is the first requirement. The progressive modification can be done by most of the pupils, though some weaker ones may have to be issued with the program example. It is important that pupils have a good understanding of the program-structure, as this gives confidence in the results which emerge from the running of the programs.

4 DISTRIBUTION OF COIN-THROWS

The action of throwing a number of coins is like taking a sample from a very large population of 0's and 1's. The larger the number the closer will the mean approach the value of $\frac{1}{2}$. The deviation of each value from this will also be $\frac{1}{2}$, and so the sum of the squares of the deviation will tend to n/4 where n is the number of coins. The variance will therefore always be $\frac{1}{4}$. These facts are illustrated by COINDIS1, which models the throwing of n coins, simulating the number of tails thrown, the mean number of tails, and the standard deviation.

The question then arises - is there a mathematical way of making a good estimate of the theoretical mean? COINDIS2 illustrates the action of throwing no coins m times. The number of coins per throw is printed and finally the mean and standard deviation are given. You should carry out the following exercises:

(a) Set m = 10, n = 20. Study the variation in the number of coins per throw, and the final mean and SD.

(b) Increase m to 20, keeping n at 20. What do you notice about the number of coins per throw, mean and SD?

(c) Increase m to 40. Before running the program decide what changes you will see. Then run it and see whether you were right.

(d) Repeat (a), (b) and (c) but with n = 40. What further changes do you observe?

(e) Write briefly on the expected consequences of increasing m and increasing n as separate operations.

It will be seen that variable results will occur, but the variations will become less with increasing m and n. In order to study this a further extension is necessary (given by COINDIS3) showing the results of l experiments of m throws of n coins. The mean, SD and proportion of tails in each experiment are given, and finally the mean and SD of individual means, and the mean and SD of the whole distribution of all experiments lumped together. You should carry out the following exercises:

(a) Set e = 5, m = 20, n = 20. Note the results.

(b) Change n to 40. Compare the results with (a). What has been the effect of increasing n?

(c) Change m to 40, and re-run with n = 20 and n = 40. While the program is running predict the results.

(d) What have you noticed about the SD of means?

(e) What have you noticed about the mean proportion of tails as m and n increase?

(f) Why are the mean of means and the overall mean always the same?

(g) Go over all your results, making theoretical calculations of expected mean, SDs and SDs of means. Make comments on the differences between observed and theoretical results.

5 FREQUENCY DISTRIBUTION OF DICE-THROWS

If one dice is thrown a very large number of times, we would expect a distribution which is very close to uniform. Suppose now that we throw n dice at a time. This is equivalent to taking samples from the uniform distribution. We can then calculate the mean of the sample and repeat the operation m times. We will then have distribution of means which common-sense tells us will not be uniform. The means will be clustered more closely around the theoretical mean. The SD of the distribution of sample-means will therefore be less than that of the parent population (σ). It can be shown that for large values of m the distribution of means approximates closely to a Normal distribution.

In DICFREQ1 this experiment is simulated. The mean and SD of each throw are printed, and at the end the overall mean and SD of means is given. Afterwards a frequency distribution of means is tabulated. You should try the following:

(a) Set m = 10, n = 20. Note the wide spread of means. How do the SDs compare with the predicted value?

(b) Change n to 40. What changes have occurred?

(c) Set m = 20, n = 20. How do the results compare with (a)?

(d) Change n to 40. Study the results as they are printed, and forecast the final distribution.

DICFREQ2 is the same program as DICFREQ1 except that the intermediate results of throws are not printed. This enables larger values of m and n to be tried. You should do the following:

(a) Set m = 20, n = 40. As the program runs, calculate the theoretical value of the SD of means, and form a picture of the likely distribution.

(b) Increase m to 40. Work as in (a). After the results have been printed compare them with (a).

(c) How closely do the distributions approximate to a Normal distribution? What do you think would be the results if m were increased with n held at 40?

COINDIS1

```
0005 PRINT
0006 PRINT "DISTRIBUTION OF COINS IN A SINGLE THROW"
0007 PRINT
0015 INPUT "NUMBER OF COINS THROWN",N
0020 FOR I=1 TO N
0030  X=RND(0)
0035 IF X>=.5 THEN 45
0037 PRINT "T";
0040  T=T+1
0042 GOTO 50
0045  H=H+1
0047 PRINT "H";
0050.NEXT I
0052 PRINT
0060 PRINT "TOTAL OF TAILS=";T,"TOTAL OF HEADS=";H
0070 END
```

```
DISTRIBUTION OF COINS IN A SINGLE THROW

NUMBER OF COINS THROWN ? 50
THTTHHTHTTHHTHTHHHTTHTTHHTHHHTHHTHHHHHTTTTHHTTHTTT
TOTAL OF TAILS=24                TOTAL OF HEADS=26
```

COINDIS2

```
0005 PRINT
0006 PRINT "DISTRIBUTION OF COINS IN A NUMBER OF THROWS"
0007 PRINT
0020 INPUT "NUMBER OF COINS THROWN",N
0025 INPUT "NUMBER OF THROWS",M
0027 PRINT
0028 PRINT "NUMBER TAILS PER THROW"
0030 FOR I=1 TO M
0035  T=0
0040 FOR J= 1TO N
0050  K=RND(0)
0060 IF K>.5 THEN 80
0070 GOTO 90
0080  T=T+1
0090 NEXT J
0095 PRINT T;
0100  Q=Q+T
0105  R=R+T^2
0110 NEXT I
0115 PRINT
0116 PRINT
0119  Y=Q/M
0120 PRINT "MEAN=";Y,"SD=";SQR(R/M-Y*Y)
0130 END
```

```
DISTRIBUTION OF COINS IN A NUMBER OF THROWS

NUMBER OF COINS THROWN ? 20
NUMBER OF THROWS ?      20

NUMBER TAILS PER THROW
 8 11 13 9 6 11 9 14 8 9 10 8 6 7 13 7 8 12 13 13


MEAN=9.75        SD=2.507489
```

COINDIS3

```
0001 LINE= 80
0003 PRINT
0004 PRINT "DISTRIBUTION OF MEANS FROM A NUMBER OF EXPERIMENTS"
0005 PRINT
0015 INPUT "NUMBER OF COINS PER THROW", N
0016 PRINT
0018 INPUT "NUMBER OF THROWS",M
0020 PRINT
0022 INPUT "NUMBER OF EXPERIMENTS",L
0023 PRINT
0024 PRINT "EXPT. NO.","MEAN","S.D.","PROP. OF TAILS."
0025 FOR H=1 TO L
0027  Q=0
0028  R=0
0030 FOR I=1 TO M
0035  T=0
0040 FOR J= 1 TO N
0050  K=RND(0)
0060 IF K>.5 THEN 80
0070 GOTO 90
0080  T=T+1
0090 NEXT J
0100  Q=Q+T
0101  R=R+T*T
0102  S=S+T
0103  Z=Z+T*T
0110 NEXT I
0115  U=Q/M
0117  W=W+U
0118  X=X+U*U
0120 PRINT H,U,SQR(R/M-U*U),U/N
0130 NEXT H
0145 PRINT
0150 PRINT "MEAN OF MEANS=";W/L,"SD OF MEANS=";SQR(X/L-(W/L)*(W/L))
0155 PRINT
0159  Y=S/(L*M)
0160 PRINT "SD OF TOTAL DISTRIBUTION=";SQR(Z/(L*M)-Y*Y)
0170 PRINT "MEAN OF TOTAL DISTRIBUTION)=";Y
0180 END
```

```
DISTRIBUTION OF MEANS FROM A NUMBER OF EXPERIMENTS

NUMBER OF COINS PER THROW ?,20

NUMBER OF THROWS ?        20

NUMBER OF EXPERIMENTS ?   5
```

EXPT. NO.	MEAN	S.D.	PROP. OF TAILS.
1	10.2	2.088061	0.51
2	10	2.236068	0.5
3	9.55	2.224298	0.4775
4	10.45	2.376447	0.5225
5	10.6	2.497999	0.53

```
MEAN OF MEANS=10.16           SD OF MEANS=0.3679674

SD OF TOTAL DISTRIBUTION=2.318275
MEAN OF TOTAL DISTRIBUTION)=10.16
```

DICFREQ2

```
0015  W=0:X=0
0020 INPUT "HOW MANY DICE ",N
0025 INPUT "HOW MANY TIMES ",M
0030 FOR I=1 TO M
0035  S=0:R=0
0040 FOR J =1 TON
0050  K=INT(RND(0)*6)+1
0055  S=S+K
0056  R=R+(K*K)
0090 NEXT J
0095  M1=S/N
0100 GOSUB 200
0103  W=W+M1:X=X+M1*M1
0110 NEXT I
0115 PRINT
0120 PRINT "GRAND MEAN = ";W/M;"      ";
0125 PRINT "S.D. OF MEANS = ";SQR(X/M-((W/M)*(W/M)))
0130 FOR Y = 1 TO 9
0135 PRINT (30+Y)/10;
0137 IF B(Y)=0 THEN 155
0140 FOR Z= 1 TO B(Y)
0145 PRINT "*";
0150 NEXT Z
0155 PRINT
0160 NEXT Y
0170 GOTO 240
0200  B=INT(M1*10-30)
0205 IF B<1 THEN 220
0206 IF B>9 THEN 220
0210  B(B)=B(B)+1
0220 REM
0230 RETURN
0240 INPUT "ANOTHER RUN (YES OR NO) ",A$
0250 IF A$="YES" THEN 10
0260 IF A$<>"NO" THEN 240
0270 CHAIN INDEX
```

```
0026 FOR B = 1 TO 9
0027 B (B) = 0
0028 NEXT B
```

```
HOW MANY DICE?     40
 HOW MANY TIMES?    40

 GRAND MEAN = 3.514375       S.D. OF MEANS = 0.2153476

 3.1 **
 3.2 ****
 3.3 **
 3.4 **********
 3.5 ******
 3.6 *****
 3.7 *****
 3.8 ****
 3.9 *
 ANOTHER RUN (YES OR NO) ? ???????????????????????????????----
```

54
Computer Programs for Schools

H

by John Turnbull, National Computing Centre Limited

John Turnbull is currently Chairman of the BCS Schools Committee Working Party on 'Packages', referred to in this article. The article was written in an attempt to raise a few ideas – and hackles – within the Working Party.

It is very easy to write a program and, for the teacher of computing, the exercise may have some of the fascination of the 'Times Crossword', a game of chess or a model railway operation. Most intellectual pastimes are relatively innocuous in their effect on others – an element of consent is required from the person who has to admire the completed crossword, participate in the game or marshal a train.

On the other hand, the 'programophile' teacher avidly seeks an audience for his work and SAVE, GET and RUN (but rarely KILL) become behavioural commands. The obvious audience is his own pupils but observing the interest of the enthusiastic few and, hearing the 'oohs' and 'ahs' through 'rose tinted ear muffs', he extrapolates and makes his own program 'available to others'.

So let us pause and reflect on the implications of such a process. After all we are using (or misusing) three types of powerful resources:–

– the intellect of teachers
– the capability of computers
– the learning experience of pupils.

We may trace through a development, implementation and transfer, questioning the actions, observing the omissions and examining the consequences.

The starting point comes before the development is even a twinkle in the author's eye – for it is the question of his competence. Rarely will the teacher be aware of dialects of the language he is using and he will probably be ignorant of procedures for documenting and structuring programs to facilitate maintenance or transfer. Maintenance, in fact, is rarely considered and yet, if a program is useful, it should be stored for subsequent use. What happens if the author leaves and a 'bug' is found or a modification is required? Indeed, with the passage of time, the author may have difficulty in following his own program – 'Why was that flag necessary' – 'surely it's I + 2 and not I + 3' – 'where was that variable initialised' . . . ? Unfortunately, even for a relatively small program written by a person who, by experience, instinctively programs in a well structured style, documentation is not a trivial task. In commercial computing it is recognised that the preparation of documentation is likely to take a similar effort to the preparation of the working program. Some idea of the care which must be taken can be seen in the developments of the Schools Council 'Computers in the Curriculum' project.

The first component in a development is the identification of the problem area – molecular movement, business games, numerical analysis, language appreciation, ... At this stage the actions to be questioned relate to the author's competence in the subject, the omissions are a lack of discussion with subject specialists or a lack of study of alternative resources and the consequences may be educational aids which are inappropriate and unnecessary. Should a teacher of computing – probably a mathematician by subject – be preparing a HAIKU program or a farm game?

In industry the problem area is identified by the user who will establish a dialogue with the data processing manager or computer user liaison officer. Specific problems will then be investigated by a systems analyst who will:–

– examine current procedures for tackling the problem
– consider alternative strategies and resources (not necessarily computing) for solving the problem
– design and formulate a specific system
– transfer responsibility to programmers.

All this will involve continued liaison with the user.

By analogy the educational computer program should stem from the need of the biology teacher, the physics teacher, the English teacher, ... This need should be discussed with the provider of the computing resource, who, in the case of the school, may be effectively the teacher of computing. This of course puts an onus on those involved in educational computing to make their subject specialist colleagues aware of the availability and capability of the local computing resource. This does not mean writing programs but it does require access to information on current developments. At local authority level it may also mean in-service activities for specific subject areas – such as Workshops for geography teachers, etc.

In the school there is no systems analyst but the capability to devise an effective resource should be an attribute of the subject teacher who should carry this responsibility. The computing teacher, acting as analyst, is likely to design a system which misses key educational concepts, fails to take proper account of the students' knowledge and ability, and ignores the non-computing resources available to complement the system.

But there will be some problems which relate to the subject and experience of the computer teacher. He or she may be a mathematician, qualified to identify a problem area in mathematics or statistics, to specify that problem and to perform the necessary systems analysis. In addition, experience of general educational practice and administration may qualify the teacher to determine needs relating to option analysis, pupil records, mark analysis, etc. However, even in these areas, it is easy to identify a problem by virtue of the fascination, rather than utility, of its solution. Is there really any point in marking objective tests of 30 pupils by computer when manual marking or the use of a response form and template is as quick as submitting data to the computer? Is there any point in generating Fibonacci sequences when any pocket calculator with x ⟷ y can generate successive terms by two key presses?

Despite these criticisms there will be some problems which are amenable to computer solution, for which the computing teacher is qualified to undertake analysis. There will be some teachers who undertake a thorough and valid analysis before embarking on coding. So, what about this stage in the process? If an on-line terminal is available, the teacher will probably jot down a rough scheme of procedures and start to key-in a program.

```
5 REM MARK STNDA ←←←ANDR← ARDISATION
10 PRINT "HOW MANY PUPILS",
11 INPUT N
15 FOR I=1 TO N
15 FOR I=2 TO N
14 LET A(1)=
14 READ ←←←←← INPUT A(1)
9 DIM A(40)
12 IF N=40 THEN 15
13 PRINT "TOO MANY PUI←PILS"
14 GOTO
```

. . . . . . .OOPS!

Interference with the thought process, tying up an expensive computing resource, performing a routine (keying) task without skill training – all exhibit marked inefficiencies. The only benefit is immediate syntax checking. Off-line preparation of paper tape or submission for batch loading will force the programmer to develop his program away from the machine. Possibly though, only batch is available or on-line is thought to be unsuitable. In fact most computing jobs require a fairly quick and reliable batch service, on-line being only necessary when there is a degree of immediacy or interaction. Thus the proper use of a terminal is its use as a calculating aid (most tasks at school level can be performed on a calculator), when output results dictate input parameters (as in some simulation games) or where there is a need for rapid interrogation and/or updating of files (rarely sufficiently urgent or complex at school level).

So the program is in and working (and 'bug' free?) – now for the kids!

"Sir! I have typed 'TWO NUMBERS' and it says 'ERROR 23 AT LINE 128', what shall I do?"

"Please Miss, where's the E?"

"Mr. Jones! It's not done anything for five minutes!"

So back to the drawing board (terminal) to change a line here, add a word there, set a trap, pin a flag, RESTORE a

DATA list, ... And even then, is it of educational value? The program is good, or seems so, – by chance or by planning. The author now has a dilemma. If it is good it is worthy of keeping and, possibly, making available to others. If it is kept or transferred, there is the need for documentation and some restructuring, discussed earlier. Even though sufficiently skilled the teacher will be reluctant to expend apparently non-productive effort. There is thus filed away an incomplete package and a listing (or tape) is made available to program jackdaws – for more effort to be spent and for more disc space to be taken.
Is this an exaggeration! There is one way to find out. Ask any computer installation providing schools computing for figures on usage of application programs – from library and users' files. If you get an aswer – and many installations are notoriously slack in monitoring usage of software – you may be surprised!

What is the answer! There are several, at various levels. The supplier of computing should be concerned that his system is used to effect. In most Authorities there is a Centre or person responsible for coordinating schools' computing activity. Perhaps there should be a regular report from the installation to the coordinator with items such as: "LUNLD has not been accessed for 2 years"
The coordinator of computing should be concerned with the need for computer awareness in subject specialists, the need to facilitate exchange of ideas and the need for a professional approach to program development. In-service courses and workshops, local software committees and reliable resources will help to provide effective classroom material.

The developer of programs should be prepared to reject a program at any stage, admitting that 'it was just a mental exercise' or 'it helped me to be more conversant with Algol'. Otherwise he must be prepared to devote time to re-writing and to make a more stringent assessment of the value of the resulting package.

At National level there is a need to promote awareness of the problems, to promote standards, and to support the distribution of software. Of particular note already, is the work in standards by NDPCAL, by NCC and for the Schools Council 'Computers in the Curriculum'. In Scotland, Scottish Computer Education Group SCEG has established subject and software committees to undertake some of the key tasks associated with problem identification.
Most of the activity at National level is known, but what of local activities? Do teachers in North Yorkshire adhere to any standards when they submit programs to County Hall – and how do these standards differ from those in Coventry? Are there program catalogues available for users of the Cambridge College of Arts and Technology system and how are the programs classified, what information is given, ...? Is there any attempt by Leicestershire teachers to examine the value of programs available on the County machine? One recent attempt to answer some of these questions has been the setting up of a Working Party of the BCS Schools Committee, with a remit to look at computer packages for schools. The aim of the Working Party is not to define standards (again) or compile catalogues or review software, but rather to gather information on various attempts to do this. Through members of the Schools Committee and through publicising the activity, it is hoped to collect catalogues and lists of programs and documents relating to documentation, standards and evaluation. Clearly the more information that can be obtained the better.
Documents or comments should be sent to Arch Coulson, Secretary to BCS Schools Committee, C/O BCS, 29 Portland Place, London W1N 4HU.

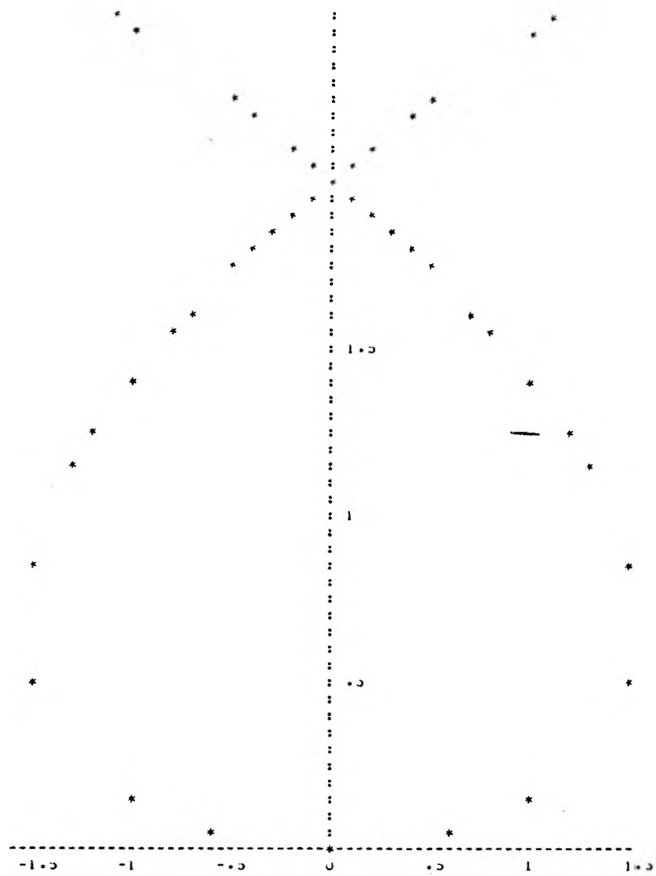
At a time of constraint and cut back, it is important to utilise resources effectively. At the same time there is a 'selling' job to be done to show that the computer can be a valid aid to education. The power and flexibility of the machine and the relative ease of transfer (compared with production costs and development time of audio visual materials and the added bulk of published materials) can be exploited if duplication of effort is avoided, effort is expended at the most appropriate point by the most appropriate person and review and assessment conclusions can be made readily accessible. Hopefully, the BCS Schools Committee initiative will be a step towards this.

Further Notes on 'A New Type of Graph-Plotter Program' by R. Ferguson, Cults Academy, Aberdeen.

Further to recent articles (C.E 19, C.E 21, C.E 24) on graph plotting, I enclose a graph of the function $2x^2=y(2-y)^2$, produced in Basic. The problems recently mentioned were avoided by treating that part of the graph where a multiplicity of points was likely, to a higher degree of accuracy.

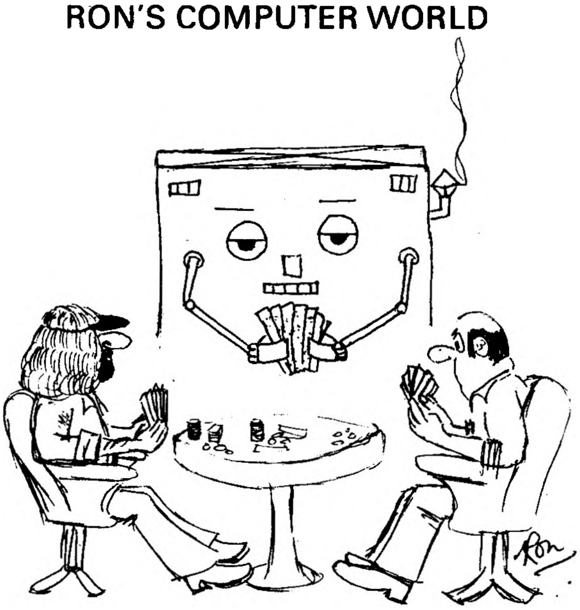
For example, for values of y, $1\cdot8 < y < 2\cdot15$, points were plotted from,
If ABS(2*X*X–Y*(2–y)*(2–y) <=0·002 THEN ...
For all other values of y (except 0), the difference was, <=0·02.

This has produced a satisfactory curve for a particularly awkward function.



GRAPH OF 2X↑2=Y(2-Y)↑2

RON'S COMPUTER WORLD



COMPUTER PUNCHED CARDS ARE AN IMPORTANT INPUT MEDIUM AND SHOULD NOT BE MISUSED.

Schools Council Project

COMPUTERS IN THE CURRICULUM

The first phase of the Project was supported by the Schools Council from 1973-77, during which time a number of units of computer assisted learning material were prepared by teachers from a variety of disciplines. These materials are currently being published by the Schools Council and details are given below.

The materials have been designed to integrate into existing curricular at sixth form level. A number in the former category should also prove useful in Colleges whilst a number of others have tried to tackle certain numerical difficulties faced by lower ability pupils. At all levels the aim has been to provide pupils with new opportunities for learning through investigations guided by their teachers. There is no need for either pupil or teacher to be familiar with computing and the units are not designed to teach this.

The publications are in the form of a loose-leaf pack in each subject area. Each contains (in 150-190 pages) notes on the topics covered for the teacher's own use and also student material which is included in one or other of two forms. The short form, appropriate for some of the topics, comprises a series of leaflets for students to be used in conjunction with lessons or laboratory work of the usual kind. The long form has been used for other topics where the content is best covered by a small booklet. A single copy of all this material is included in the packs and it has been agreed that teachers may copy the student pages for their own use using the reprographic facilities which are available to them. It will also be possible for teachers to purchase additional copies of the long form Students' Notes and this will be more convenient and cheaper if a number of copies are required for class use. The associated computer programs, in the language BASIC, are available free of charge to purchasers of the packs.

Phase One of the Project relied heavily upon collaboration with education authorities and this is to be continued in Phase Two which will run for three years from September 1978. The response from local authorities and from schools and colleges to the proposed further work has been most encouraging. As the Project gets under way it is hoped to be able to take up many of the offers of assistance which have been made.

Whilst some of the work in the second phase will be in those areas where the methodology for development has been established, namely in the sciences, economics and geography, the Project will also be investigating the use of computer assisted learning in new areas. The first phase included some preliminary work in history, part of this was concerned with decision games and part with analysis of census data, and it is hoped that this will become an established area of work and that materials in this field will be published. Work in mathematics has been excluded from the Project so far but it is hoped that an early investigation of the use of CAL methods in mathematics education will form a basis from which development can proceed.

Another important feature of the forthcoming phase of the work is to promote the use of the materials already developed through courses and workshops for teachers. It seems that there is a great deal of interest amongst local authorites and teachers generally about pursuing this line of action and members of the Project will be doing their best to meet all requests for help in this area.

FIRST-PHASE PUBLICATIONS

Inspection copies available from

Edward Arnold (Publishers) Ltd
Woodlands Park Avenue
Woodlands Park
Maidenhead
Berks

BIOLOGY

Human energy requirements; Pond ecology; Inheritance; Statistics; Transpiration; Predator-prey relationships; Countercurrent systems.

ISBN 0 7131 0227 6 - £13

PHYSICS

Capacitor discharge; Photoelectric effect; Gaseous diffusion; Mass spectrometer; Radioactive decay; Planetary motion; Gravitational fields.

ISBN 0 7131 0224 1 - £11.50

ECONOMICS

International trade; Multiplier; Fiscal policy; Elasticity of demand; Price fluctuations; Price stabilisation; Theory of the firm; Banking; Monetary policy.

ISBN 0 7131 0229 2 - £14

CHEMISTRY

Chemical equilibruim; Rates of reaction; Gas chromatography; Electrochemical cells; Lattice energy; Element game; Contact process.

ISBN 0 7131 0226 8

GEOGRAPHY

Human population growth; Joint-stock trading game; Drainage basin morphometry; Windmill game; Farm game; South Eastern railway game; Statistics.

ISBN 0 7131 0228 4

HOME HEATING

Unit on house insulation and costs (a smaller pack)

ISBN 0 7131 0225 X

Further information obtainable from:

Educational Computing Section
Chelsea College
Pulton Place
London SW6 5PR■

How we will cope with the micro chip

Vince Houghton plots the possible responses to new technology

When Coombs published his book *The World Educational Crisis* in 1968, he criticised educational systems throughout the world for their inflexibility, their unwillingness to change or even experiment seriously and most of all for their refusal to take note of changes which had already taken place in the various societies in which they were operating. A couple of years ago he reviewed his earlier findings and found that the situation had not changed in any major way. Since that time we have seen the advent of the microprocessor and a number of claims have been made for the probable impact of these devices which at least require a serious investigation from professional educators.

The claims for the social and economic impact of microprocessors and the associated technology have an extremely wide range. One author has stated that this technology is the most important to emerge since man has discovered fire. In a recent BBC Nation to Nation programme the general view was much less frenetic.

The speakers from the United States and West Germany seemed to believe that any unemployment caused by the introduction of silicon chip technology would be fairly easily absorbed by new kinds of jobs mainly in the service section of the economy. The speaker from Japan agreed with them as far as advanced countries were concerned but added that those Third World economies which were currently competing with the advanced industrial countries would quite likely see all their efforts come to nothing because the one commodity they had in abundance, cheap labour, would no longer be an asset and could even be a liability.

At a recent conference entitled 'Microprocessors and jobs' at the London Press Centre, the sentiments from the majority of speakers were similar. In other words most were wildly optimistic about the change which would be caused in employment patterns by the utilisation of chip technology. There were two important exceptions to this viewpoint.

These were Dr. Ray Curnow of Sussex University and Mr. Clive Jenkins of ASTMS. Both of these speakers predicted a massive increase in unemployment. In the United Kingdom figures of five million to eight million were mentioned. This is why, as educators, it is important that we should take notice of what may be happening.

It is possible that the mismatch between various educational systems described by Coombs would reach crisis proportions if the more pessimistic forecasters are even nearly right. Mr. Jenkins has a book being published this month in which he foresees the total decline of the work ethic. He also believes that trade unions will have to protect their members during the rough transitional phase.

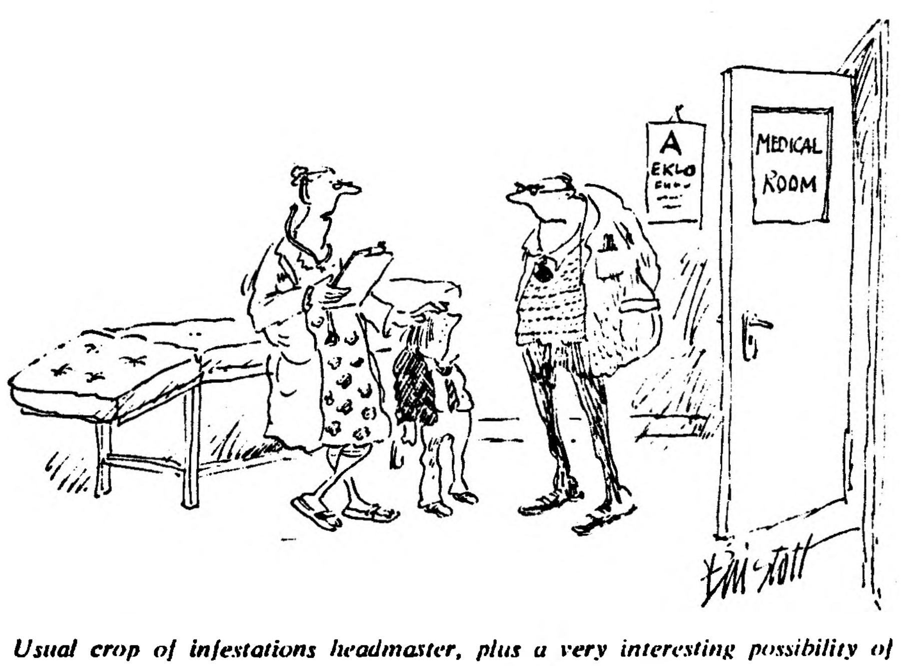
What will be the reaction of educators to the introduction and development of this technology? I would suggest that they will fall under five headings and I would like to present them in the most likely order.

1 The ostrich attitude Far from being a recipe for failure, this attitude, which consists of pretending that nothing important is happening, has a history of great success. Radio, television and programmed learning via main frame computers were all claimed as being about to revolutionise education but none have made more than a very marginal impact. Those who ignore the introduction of the silicon chip have history on their side.

2 The Luddite reaction This reaction would realise the potential of microprocessors as an educational tool but would also see that the change in information transmission would mean a decline in the numbers of teachers needed, and possibly because of increased access a lower status. The attitude would be hostile to the introduction of microprocessors into schools except for demonstration purposes. Like the ostrich attitude it is not silly and a serious case can be made for it.

3 The apocalyptic attitude This attitude may well be taken by those extremely in favour of the introduction of silicon technology as well as those people of an inherently nervous disposition. There is a body of opinion which sees the present changes as a watershed which is the beginning of a new society. With this vision your view of a new society may be optimistic or pessimistic, all one has to accept is that it will be sensationally different. Again one must remind oneself that the apocalyptic view is not always wrong. A number of societies not excluding Northern Ireland and British Leyland are sensationally different from what they were a dozen years ago. The sad thing about those who may see really fundamental shifts is that their insistence tends to reinforce the two previously described views.

4 The co-operative view This point of view accepts an historical inevitability about the advent of new technology. It suggests that while accepting and trying to understand the implications of the news that one should try to preserve all that is best in the present practice. This is the attitude most of us would claim to have but it is extremely dubious if more than an exceptional few really hold it. The reason for this is that there is no consensus about what should be preserved and anyway none of us relish the thought of carefully tended skills being



Usual crop of infestations headmaster, plus a very interesting possibility of dhobie itch. . . .

rendered obsolete either slowly or quickly.

5 The imaginative attitude This attitude is unfortunately even more unlikely. It consists of allowing those who are enthusiastic to experiment with microprocessors in a variety of educational situations to do so. Then if these were carefully and quickly evaluated we would have some real data to go on. It would, however, be necessary for the grant-awarding agencies to step outside the ranks of the magic circle of educationists who award and receive grants and to consider the claims of younger untried people and perhaps even a few elderly mavericks. It does not seem likely.

Politicians of all political parties would like to see both the co-operative and the imaginative attitudes prevail. The last government gave large sums to advance the knowledge of silicon technology, the Conservative Party is similarly committed.

Dr. Keith Hampson speaking at Exeter on 24 February referred to the DES sending out a circular calling for more attention to be paid to the educational implications of the micro-electronic revolution by remarking: 'What a feeble response – nothing more than a belated gesture to the task of facing up to a fast-changing industrial society.' He went on to say: 'If we are to have a future in electronics what is wanted is nothing less than a shake-up in the educational system.' He concluded by stating: 'The education system will have to become more computer orientated – not in the sense of teaching how microprocessors work (which, sadly, is what is currently the emphasis) but the application and means of using them. For teachers to gain such an appreciation, appropriate in-service training is a matter of urgency.' At the same time the French Government was arranging for 2,000 teachers and 10,000 executives to receive microprocessor awareness experiences during the year commencing in September 1979 and also announced the aim of introducing keyboarding as a skill in all French schools.

The argument in this short article is that the impact of microprocessors is likely to be felt in industry in the immediate future. The hard evidence for this is not merely the ex-mechanical watch industry, cab meters, scales and cash registers but also the changes taking place in any process concerned with hydraulics, office procedure (the word processor), as well as governors of washing machines, mixers, cookers and other household items. It is possible that when these jobs vanish others will take their place but there is the difference that whereas the automation revolution associated with Henry Ford took nearly 50 years to achieve completion, this one will take a decade.

It would be fair to ask which of the viewpoints is adopted by the writer and the answer can only be pronounced with difficulty 'reluctantly apocalyptic', it seems that the evidence is strong enough for a major change to be imminent in industrial society. It would appear unlikely that any of our social institutions are sufficiently flexible to adapt quickly enough. Still, in a real crisis human ingenuity may frequently astonish and one should always remember ours is not the only corner of society which will be affected.

Perhaps all one is asking for is a sensible insurance policy approach in that a few experiments could be allowed to take place. There is little doubt that many of these would fail but experimental success can only be possible if there are many failures. The politicians have declared themselves, no doubt the institutions will try to be accommodating, the Press, both general and educational, has done a first class educational job. But is the will to be mildly co-operative present? Is the tolerance present to allow experimentation, particularly of the unorthodox kind? One wonders and remembers what they used to say in Western Canada: 'You can always tell the pioneers, they're the ones with arrows in their backs.'

(Dr. Houghton is senior lecturer in educational administration at the Open University.)

TES SPECIAL REPORT BY PHILIP VENNING

The race to train the brains that work the future

Electronics are about to transform our whole way of life but Britain could miss out on many of the benefits if our education and training system fails to produce enough skilled people.

In the next 25 years virtually everything we do - from the way we work to the goods we buy - will be blessed or blighted (depending on your viewpoint) by the microprocessor, a component so small and so cheap as to be almost insignificant, but with the power and versatility of an old-fashioned computer. The potential of the microprocessor, a tiny silicon chip on which thousands of circuits are etched, lies in the fact that it can control almost anything.

The technological ostriches admit its wizardry but doubt its effects. They expect its progress to be slow, its influence to be no wider than any other specialised technological advance. So far, its impact has been in trivial fields. Calculators, digital watches and television games may be little more than frippery, but they are the heralds of a trend - already planned in factories throughout the developed world - to include it in a vast range of normal consumer products.

But its real importance is in its power to make automation - that mirage of the fifties and sixties - an almost immediate reality. The Cabinet is now seriously worried about the danger of mass unemployment resulting from the rapid introduction of "chips" to manufacturing processes. The alternative is even less attractive. Japan, the United States and leading European countries are already so far committed to microelectronics that British industry must either follow or perish. We can go forward or back, but we cannot stay where we are. If we do decide to join the race, and the Government has accepted that we have no option, we have urgent ground to make up.

Education has been singled out as one of the crucial areas for attention, and the Department of Education and Science has already set up three working groups to look into the short, medium and long-term effects on education of the micro revolution.

There is, of course, a danger that once again too much may be expected of education. But there is no doubt that among the main things that will hold us back are a shortage of skilled designers, technicians, technologists and engineers to install the new automated machinery, and antiquated attitudes to the new hardware among management.

But, just because the most obvious areas for action are in the further, higher and adult education system, the schools should not be sitting back. Their pupils will increasingly be entering a world where traditional skills and attitudes are inappropriate. The answer is not simply a matter of shoving one new lesson on microelectronics into the general studies slot. Some fairly substantial changes in school, university and college teaching methods and syllabuses will be needed, and planning for these must begin without delay. The education system works too slowly to allow it time to wait and see.

Indeed the new technology may have an effect on schooling itself. Programmed learning may suddenly find a new acceptability in the home as the consumer market is increasingly flooded with cheap, do-it-yourself learning toys, games, and gadgets; while in the long term, as industry dispenses with labour, public services such as education may be expanded substantially and class sizes of half a dozen might become quite normal.

This report looks at how chips may affect educational technology; what measures the education and training services should take to ensure an adequate supply of people with the appropriate vocational qualifications and technical information; and how in the long term the education system may have to prepare young people for a drastically different society.

The Government has recognized that without immediate and drastic action much of British industry will be left stranded with outdated products and uncompetitive production processes. It has already started putting money into microelectronics (on a small scale compared with other industrial countries) - most of it on setting up a company to make the chips. But the real importance of chips will be in the way they are used.

Last month a cabinet advisory committee - the Advisory Council for Applied Research and Development - produced a paper recommending a strategy: *The Applications of Semiconductor Technology* (HMSO, 85p). It singled out education as one of the crucial areas for action, and the Department of Education as one of the key Government departments (the others were the Treasury, Industry, and Employment).

The education service has, on paper, a simple enough task. In the short term, in conjunction with the Manpower Services Commission, it should re-educate and re-train those already in industry. In the medium term it should improve technical and management training in further and higher education; and in the long term make sure schools produce enough people with new attitudes and new skills.

Other Government departments may be expecting too much. The education service is diffuse, slow moving and idiosyncratic. The DES cannot wave a magic wand and create a sudden increase in technologists and engineers (other than by poaching from abroad). The most able pupils have been shunning these subjects for too long, for familiar enough reasons. The best the DES can do is advise and encourage.

Its initial concern is comparatively straightforward, however. The Government must find ways of persuading thousands of small and medium sized manufacturers to include microcircuitry both in their products and in their manufacturing processes. This kind of consciousness-raising exercise can be slow and frustrating. It is essentially a matter of re-education and re-training.

Last month Mr Gordon Oakes, the Minister responsible for further and higher education, said that perhaps as many as 50,000 managers would need to hear the message over the next three years. University and college courses from one to three days should give them a preliminary introduction.

The Department of Industry has set aside part of its £15m Microprocessor Application Project to put on such short courses. They will pay for demonstration equipment and subsidize industrialists attending the courses.

Once the message does start getting through, are there enough qualified engineers and designers to alter or replace the existing machinery, or redesign the finished product?

The answer is undoubtedly no. One solution, proposed in the ACARD report, is an ambitious programme of retraining existing specialists, such as production and design engineers, and draughtsmen, in micro applications.

The report pointed out that this was principally a matter of organisation rather than money. Estimating that half of Britain's 260,000 engineers would need some retraining, the report calculates that it would be possible to create 50 centres, employing a staff of six or seven, running three-month courses, for a cost of only £5m over five years.

To mount these courses, as well as the short ones for managers, universities and colleges would be hard put to it to find enough suitably qualified staff of their own. This raises a tricky question of manpower priorities - should the small number of relevantly qualified people be out in industry getting on with it or should they be in classrooms training more? Obviously there is room for compromise.

The DES has already carried out an urgent review of existing relevant courses in further and higher education, and was due to report to the Government at the end of last month.

To be certain that enough new courses get into operation in time, the DES will have to enlist the help of the local authorities and universities, if necessary dispensing with some traditional slow-moving bureaucratic etiquette. In this race for economic survival, months count. The signs so far are encouraging. The Government appears to be in no doubt that more direct intervention is unavoidable, though at local level there is already a race among some colleges to become centres for microelectronics. American experience shows that firms tend to congregate where there is a pool of skilled manpower. Any college that can build itself a reputation will suck industry and jobs into the locality. One of the first off in the bidding is Devonshire, where the education committee is considering setting up a "national centre for microelectronic education" based on Plymouth Polytechnic and Plymouth College of Further Education. The estimated cost - £250,000 over three to five years.

In the medium term, however, Britain's ability to remain competitive in international markets will depend on rather more far-reaching changes than creating a few new courses. We may have to rely on a substantial increase in the technical expertise of large sections of the working population. Far more people will have to leave the education system familiar with micro-circuits and ways of using them to advantage.

The accepted wisdom is that it is only by investing in our brainpower, our education, that we will be able to compete with the American and Japanese. They may be good at the hardware, but they are not so good at applying it. Which is where we excel, or so the argument goes. Admittedly there is something gentlemanly sounding - a continuation of our old distaste for the practical - about the notion of surviving on our superior brains. But even if, faute de mieux, this is the way we eventually go it is dangerously complacent to assume that it will happen miraculously if we just carry on in the old way.

Already there is a shortage of engineers and technologists, and an acute shortage of really outstanding ones. Throughout the early 1970s the number of electrical engineering graduates from universities drifted down from nearly 2,000 a year to under 1,700 in 1975. The output of physics graduates was stagnant at about 2,000, and maths graduates at under 3,000. Even when CNAA degrees, and HNC/HNDs in similar subjects are taken into account, the grand total was virtually unchanged at about 14,000 a year. And of course many of these do not end up in jobs in relevant fields. University applications for science and technology have been improving slightly recently, but there is no evidence of the large expansion that will undoubtedly be needed. Once the delayed effect of this fall in the birth rate hits the universities in the 1980s, it may be hard to maintain present levels.

Nor is there any guarantee that those who leave with qualifications in, say, electrical engineering, will know much about microelectronic applications. The DES is planning to look at existing courses in further and higher education to see whether they need to be modified, not only for the scientists, but just as important, the future managers. Although computing as such is increasingly a normal part of most courses, many business studies students leave with little understanding of the potential of the new hardware.

The recent reorganization of business and technical education may have come just in time. The new structure of courses under the umbrella of the Technician Education and Business Education Councils may have just the flexibility needed to incorporate microelectronics modules.

Even the electronics specialists themselves may need totally different courses. No microprocessor can operate on its own - it needs all sorts of electrical and mechanical devices to feed in information or carry out its instructions. And one of the main advantages of the new microcircuits is that they enable the construction of ever more complicated systems.

The one person the country now needs is the "systems" man or woman, a person capable of thinking creatively about systems as a whole. But it may be a long time before the education system can find ways of producing more of this rare breed.

The Engineering Industry Training Board has started working with several firms on ways of identifying those with this ability to see what qualities should be encouraged. Chris Carrol, until recently head of the electronics sector of the EITB, said that out of 200 people they looked at in one firm only three could be described as systems men. "We need to pick those who perform at very high levels in all subjects." For this reason he wonders whether schools should try to delay specialization, though he admits that a really thorough grounding in physics and maths is fundamental.

The consequences for the school system may be unpopular. However much it runs counter to prevailing educational principles, schools may well have to do far more to single out their most gifted children for special treatment. HM Inspectorate has already recognised that not enough is being done for more able children even as low down the system as the primary school. They may well have to use all the direct and indirect power at their disposal to press schools to do more. And schemes such as the £500 scholarships for engineering students should, without any blushing or feeling of discomfort, be increased overnight to a point where they have a clear backwash on the school system.

In spite of a modest swing back to science in schools, the principal bottleneck - the lack of competent maths and science teachers - could well get worse as industry's demand for science graduates intensifies. The teaching profession and the education service must start talking very seriously about whether it is now time to consider unconventional, dramatic, and no doubt politically tricky measures to try to increase the supply of suitably qualified teachers in the long run.

One way out, though certainly not the only one, might be found in the A level in electronic systems pioneered by Essex University. Started by Professor Barrie Chaplin, it is designed to teach pupils some of the basic ideas of how any systems work, including biological and mechanical systems. It need not necessarily be taught by science staff and is not designed for science pupils alone.

The new A level was tried out in 10 schools in 1974 and was given full A-level status under the NEB in 1976. It is steadily growing in popularity. Mr Graham Bevis, of Richard Taunton College, a sixth form college in Southampton, who is the subject's chief examiner, said that 1,500 candidates were registered for the summer examination. More than 50 schools or colleges are likely to be offering it next year.

It is an attractive supplement to normal A-level maths or physics, but not a substitute, at least for potential university entrants. However, several firms have shown interest in it as a qualification for future technicians. Mr Bevis has found that its accent on problem solving may be more suited to pupils who are less obviously academic material. And Professor Chaplin is convinced that a steady flow of school leavers going straight into employment having studied the course may be just what the country needs. They will be able to spot what parts of a firm's production processes can be automated. The actual installation can then be farmed out to technical experts.

The main constraint on the more rapid expansion of the course is money. So far the curriculum development work has had no financial help from the education service. He is convinced that somewhere individual schools will have the right answer, and only when the micro revolution has gone a bit further will it emerge what that is. It does however seem prudent for the Schools Council and the inspectorate to start looking fairly closely at where curriculum development may be needed, as the ACARD report recommends.

This should extend from the primary school upwards. It should also look long and hard at ways of encouraging more women to do technical studies of some sort. Microelectronics is ideally suited to both boys and girls, and schools are full of girls with technical talent going to waste. Instead of making a spell in the metal workshop compulsory for all pupils, schools might usefully put practical electronics or technology on to the timetable.

The educational demands of the micro revolution will not be restricted to the sixth form. As another section of this article suggests, the future for traditional craft apprentices in manufacturing industry is uncertain in the long run. But there is no doubt that what craft jobs do remain will require a much higher level of skill. It is clear, for example, that as factories become automated it will be less and less in a firm's interest to tolerate "down" time - when expensive machinery is broken down. A new breed of super repair men, capable of solving (and permitted to solve) a wide range of sophisticated technical problems, will command high salaries and need a new kind of technical training and educational qualifications.

Microelectronics

their implications for education and training

This is an abridged version of the recently published Council statement Microelectronics; their implications for education and training. It was introduced at a press briefing on 30 November and reviewed in the Guardian and the (last) THES the following day. We have had an enormous number of requests for the statement, copies of which are available from Jill Coates, Information Officer.

Microelectronics - very cheap, very small, highly complex electronic control circuits on a wafer of silicon - are now accepted as being a potential agent of social and economic change comparable with the development of new sources of power which brought about the industrial revolution. The time scale of change, however, shortens rapidly with the advance of technology; if indeed we are on the threshold of a new industrial revolution, it may come upon us in decades rather than centuries.

The Council is no better placed than anyone else to foresee the effects of advances in technology on the social and economic future, but it does have a particular responsibility to consider the implications of technological developments for education and training. However, these implications can only be discussed against certain social and economic assumptions.

Assumptions

Economic and social

Just as mechanisation replaced human muscle as a productive force, so microcircuits are likely to take over functions in which the human brain has operated essentially as a programmable control device. Such a development will, of course, have large scale applications but it is important to realize that, because powerful electronic control devices can now be both small and cheap, many everyday products will shortly incorporate microelectronic control.

The immediate consequence is likely to be a sharp decline in the demand for labour, particularly in manufacturing and service industries and in any case, the rate of contraction of demand for labour, as a result of new technology is likely to be substantially greater than any rate of economic growth that can be achieved. In the transitional period there will be a sharp reduction in the demand for labour, whether or not this is a long-term feature.

Educational

Assuming this, the education and training system will have to face the following cluster of questions.

What can we do to help people to prepare themselves for a rapidly changing society?
What can we do to help people fit themselves for employment in new and technologically advanced occupations?
What can we do to help people to fill their leisure hours - whether the result of a reduced working week or the enforced leisure of unemployment?
What can we do to help people to maintain their self-esteem when there are no jobs for them?

In the schools there will be a need for more emphasis on attitudes relating to work and society. Young people will need to be helped towards views that will sustain their lives, whose opportunities will not be those that we know today. There will be a need to give them a balanced understanding of

- microelectronics and computer-related activities
- what such devices can and cannot do
- the difference between what is theoretically conceivable and what is currently attainable.

There will be a need to lay a foundation for later training in product design skills: development of an ability to conceive products and systems so that they can be made effectively and economically.

However, the most important task of the schools will be to instil in young people a general curiosity and a pleasure in learning. In their adult lives it may be very desirable for learning to play a continuing part, as vocational retraining, as a worthwhile activity replacing work, and to maintain self-esteem. There is a need for a deliberate emphasis on developing study skills, particularly those related to independent learning.

Post-compulsory education will need to develop as a complete system of continuing education, in order to cope with the demand for an increasing range of education and training courses, from an increasing range of students of widely differing ages and backgrounds.

For instance, if socially useful activity is to be a valued element in our solution to our problems, then there will be a need for educational efforts to develop both the skills and the attitudes required. There will also be wide scope for education for leisure and for pleasure.

To improve access to relevant courses for the wide range of people requiring them, there must be further development of open learning systems which make studies of their choice available at times, in places and in forms to suit their conditions.

Finally, we need to investigate possible schemes of study leave, or of a continuing education entitlement similar to the current mandatory grant system but increasing in value with time and encashable for any type of continuing education.

If microelectronics will have such a radical effect on society, we should look for their direct effect on education and training, particularly on teaching and learning methods. Let us first recognize that, as with previous technological developments, education and training will need to make use of systems and devices which have been developed for other markets. Teaching and learning methods will be expected by the public to use systems which are easier, cheaper, more reliable and more versatile, to keep pace with developments in commercial and domestic life.

Possible Developments

Application to existing equipment and systems

The development of digital recording techniques should eliminate compatibility problems in video-recording systems. The automatic recording of broadcasts during the night will become completely practicable. Remote control of a wide variety of electro-mechanical devices will be easier to achieve. Television receivers could also serve as receivers for a variety of data transmission systems via telephone lines.

The continuing trend to smaller, cheaper, more powerful computers means that every teaching and training institution could realistically contemplate having its own microcomputer. This will permit the use of electronic management systems for such tasks as library and resource management, and timetabling and scheduling, as well as the greater development of computer-assisted and -managed learning.

8

Information provision

The effect of computer-related systems of information provision on the education system could be as dramatic as was the development of the printed book. Already we can experiment with the two broadcast teletext systems (CEEFAX and ORACLE) and the telephone transmission viewdata system Prestel. Other computer-based information systems, such as the various specialist data bases and the British Library BLAISE bibliographic service, can be searched in a much more comprehensive way enabling the user to extract any available information in a form organized to suit his or her needs.

Computer-assisted and computer-managed learning

Computer-assisted learning and computer-managed learning could be much more widely applied in education and training.

Education and training need

(a) authoritative guidance on the available equipment and its suitability for particular tasks
(b) development of the current system of program directories and exchanges
(c) action to rationalize the programming languages of microcomputers
(d) arrangements for in-service training of teachers and local support services
(e) arrangements for the release of teachers and trainers for the writing of learning programs.

Opportunities for learning outside educational institutions

The modified television receiver is becoming a ubiquitous instrument, and so it is, therefore, important to explore the potential of TV 'games' packages in order beneficially to influence their future development for mass education.

Added to this development, the penetration of teletext and viewdata systems into homes, offices, public libraries and other public places will lead to an unprecedented increase in 'learning' situations completely outside traditional learning institutions.

Conclusions

There will, clearly, be many possible applications of microelectronics in systems of use to education and training beyond those which can currently be foreseen. CET has a particular responsibility to observe the whole field of developments in microelectronics and related communication sciences in order to identify their potential application to teaching and learning. In applying any new technology, the guiding principle must be to use it when, and whenever, it can serve the student's learning need. But the new technology itself also encourages us to look again at some of the ways in which we have conceptualized the education and training process, and particularly the ways in which information consultancy and advice, and teaching and learning might be provided as a single service.

Some might argue that the suggested major social change may not happen. Perhaps so; but all the indications are otherwise. In fact, it hardly matters. If microelectronics do have the effects we have assumed, we shall need, with desperate urgency, the changes in education and training here suggested. If there is no radical change, only a slow and wholly-contained modification, then we shall be able to afford the continuing education we have long agreed was desirable, on the scale for which there will be a demand.

There is not unlimited time. If change is coming it will come with unprecedented speed. We need to think and to act with courage and resolution.

Geoffrey Hubbard

A PLAN FOR FEEDING THE HUNGRY SHEEP ...

by David Pegg

A central policy is needed to cover computer education, says David Pegg, a master at Bedford School. Here he sets out several principles which should be applied in developing such a policy, and by way of provoking argument, offers ten proposals. These proposals, he says, would cost a lot of money; but would make the best use of the wide fund of expertise that already exists in schools in this country. The author is a member of the Schools Committee of the BCS.

"The hungry sheep look up and are not fed....
But that two-handed engine at the door,
Stands ready to smite once, and smite once more."
Milton, *Lycidas*.

It is fanciful to suppose that Milton had a vision of the power of the binary computer. But the fact remains that we now have a once-for-all chance to teach people of all ages about that power and how to use it. At last the government is ready to feed the hungry sheep, with the aid of £60 million.

What proportion of this vast sum will be spent on computer education in *schools* is not yet clear. But how will it be spent? Will it be largely wasted, in common with many other disbursements?

Firstly, let me make clear what I understand by "computer education". It would be more exact to say "education about, and with the aid of, information technology", if it were not such a mouthful. There are three main sub-divisions: Learning how to live with the new technology; learning skills in the development and applications of devices and systems; and harnessing information technology to aid education generally.

The first applies to everyone and extends to such subjects as the use of increased leisure and the need for continued education throughout life. The second relates to the need to produce a skilled force of designers and technicians. The third is concerned with a radical reshaping of education methods.

A central policy is needed to cover the three groups. So far this has not, for a variety of reasons, emerged. Warnings were given at the 1970 IFIP Conference in Amsterdam, but they were largely unheeded. In 1971 (Computer Weekly, June 17, 1971) it was prophesied that the failure of politicians to give adequate support to computer education might well prove to be a prescription for Luddism in the 1980s.

Since then much pressure has been applied from below with very little positive action except in computer aids to learning. But the failure of politicians is only a reflection of the ignorance and dislike of computers in society.

The outlook would certainly be black if there were not many teachers with plenty of experience in running courses on computers and how to use them. The growth of microelectronics has sparked off tremendous interest in schools. Many would like to have their own computer and quite a few have already bought them.

A branch of the Computer Education Group called MUSE (Minicomputer Users in Secondary Education) has expanded almost ten-fold in the last two years. Its purpose is to help schools and colleges to use small computers (minis and micros) effectively. It holds meetings throughout the country to give advice and to exchange experiences on hardware, software, teaching methods and uses of computers.

This is merely the tip of a large iceberg. In the major cities large networks have been developed with terminals (and in some cases micro-computers) linked to mainframes. In towns and rural areas there are more modest systems.

But there are many areas where little or nothing is being done because the LEAs do not have go-head advisers, and there are no experienced teachers in the schools.

Within schools themselves the great majority of teachers regard the computer as an unwelcome interloper. Higher authorities have little knowledge of where, or how much, teaching is being carried out. One urgent requirement is a country-wide survey on this matter.

What then can we learn from the above observations? First, that there is more knowledge of what is needed at the grassroots than among higher authorities. Secondly, that there is little communication between these two. Thirdly, that the resources available and the educational needs differ widely both geographically and in types of schools (of different sizes and age-groups).

Above all, there is need for a great increase in initial and in-service teacher-training for computer education. This should apply to all teachers. If almost every child should be given some contact with micro-technology, so should almost every teacher.

When it comes to taking decisions on how to spend the promised millions, the government should take into account the knowledge acquired by teachers over the last 10 years. There must also be opportunities for continued experiments by teachers.

With developments in cheap mass storage, many schools in 10 years' time will have their own distributed systems. Only by trying different methods now will we achieve the best results.

But there are many possible pitfalls. One mistake would be to equip very quickly a large number of schools with their own microcomputers, regardless of the lack of experience of the teachers in some of these schools. This may soon be happening anyway in default of a policy, because parent-teacher associations are increasingly buying micros for their schools.

Another pitfall would be to build very large educational networks using highly centralised software packages designed for computer assisted learning. The costs would be unacceptably high and the educational value limited.

The main need is to balance advantages of decentralised organic growth against those of centralised control and coordination. The latter gives economy of effort, pooling of information, avoidance of repeated mistakes, economies of size and provision of the best. Decentralisation, however, encourages individualism, variety, versatility and initiative and avoids rigidity, bureaucracy and over-complication. A judicious blending of the two is therefore required.

The Open University has done remarkable work in using computers in education. Its experience could be harnessed to the development of software packages, although it might not be feasible to form the equivalent of the Open University for schools use. However, the infrastructure of large-scale software and some limited use of computer networks would provide valuable economies. The decision on what to use and how to organise the work would be left to schools and local authorities.

It is not my intention to suggest how the money should be spent, but to propose principles on which allocation should be based. However, to provoke argument (and it is only from closely reasoned argument that worthwhile decisions will emerge) I will give a few concrete proposals on how the policy might be developed.

1 There should be a central council for schools education in information technology, consisting of government officials, businessmen, educationists, trades union officials, politicians and others. This body should advise the government on the general policy to be adopted, on how the money should be allocated and on how the work in paragraphs 2, 3, 4 and 5 should be split up.

2 The Schools Council should be asked to advise on ways to encourage curriculum development in computer education in schools.

3 The Council for Educational Technology should be given the tasks of encouraging the use of computers to assist learning and of proposing technological aids for the purpose.

4 The National Computing Centre should be given the task of encouraging computer education in those aspects outside the scope of the Schools Council and the Council for Educational Technology, with particular reference to hardware and software development.

5 The British Computer Society should monitor all this work, and advise the bodies concerned when it considers that urgent action is needed.

6 An advisory centre for computer education in schools should be set up, to act as a centre for information, coordination, and initiation and evaluation of projects.

7 A special priority is needed in the development of computer courses for teachers, both in colleges of further education and for in-service training. The Department of Education and Science should take an active role in this.

8 The Department should take positive steps to encourage LEAs to do more on computer education, to advise them what can be done, and to assist them with funds.

9 Local educational computer centres should be formed throughout the country using, as far as possible, facilities already available in colleges of further education. Their role should be to run in-service courses for teachers, to advise and help schools on educational and technical problems, and to provide a central computer system to which local schools can be linked, either by terminals or in-house microcomputers.

10 The generally agreed views of teachers should influence policy decisions, so some method of communication should be devised between them and the policy-makers. Further, opportunities for research and experiment should be available to teachers by financing their replacements while they study in universities, industry or in local computer education centres.

The above proposals would cost a lot of money, but they would have the great advantage of building on existing foundations, and of making the best use of the wide fund of expertise that already exists within schools in this country.

The initiative on carrying out these or other proposals must lie with the politicians. Let us hope that they will not make a party issue out of it, but will come to a general agreement.

(MUSELetter No.3 April 1979)

ISBN O 86184-010-0 £2.00 2000IM180