hard core

THE JOURNAL
OF THE
BRITISH APPLE
SYSTEMS
USER GROUP

December 1984 £1 VOLUME 4 No. 6

hardcore

THE JOURNAL OF

THE BRITISH APPLE SYSTEMS USER GROUP

P.O.BOX 174 WATFORD WD2 6NF

BASUG Ltd. is a non-profit-making company limited by guarantee.

Basug Committee

****

BASUG Telephone No.: [redacted]
Please note: This number is administration ONLY.

Bulletin Board: [redacted]

Founder members of the Association of Computer Clubs.

CONTENTS

INTERNATIONAL APPLE CORE TM

MEMBER OF THE INTERNATIONAL APPLE CORE

HARDCORE is produced using Applewriter II, and printed on an APTEC Flowriter with a Madeleine Daisy-wheel

Front Cover: A seasonal Macintosh greeting from Norah Arnold.

Editorial

As most of you will be aware, this is the last issue of which I will be the editor. Peter Baron will be taking over the reins and I hope you will all give him the help and support you have given me. Please continue to send in your tips, hints, problems, articles (learned or otherwise), reviews, news, small ads., etc. and don't forget to let us know what is happening so that we can keep the diary up to date and pass on information about local groups and special interest groups.

Also, please support our advertisers. Without them there would be no magazine and they are unlikely to keep advertising if they don't have any response. So if you deal with one of our advertisers, tell them you saw their ad. in Hardcore. It helps us and them.

Following on from that we would like to take this opportunity of thanking our advertisers for their support over the past year and look forward to working with you again next year. We wish you all a prosperous New Year.

Regarding sending things in for the next issue, do remember that the next copy date is January 4th. In view of the fact that there may still be delays in the post after Christmas as well as before, it would be a great help if you could send things to us early. Apart from the convenience to us, it may be inconvenient to you to miss the next issue. We cannot guarantee space to anyone but do try to get all entries in the first possible issue. However, there is little chance of late entries getting in. If you know you would like something in Hardcore but cannot produce your entry in time (for whatever reason), just let Peter know how much space your entry will take and when it will arrive. If he knows it is on the way, he will be able to consider whether to include it or not and can give a couple of days grace on receipt of your item.

I wish everyone a Merry Christmas and a Happy New Year.

Roger has an Apple /// for which he wants to make custom hardware interfaces. Apple will not give information about the signals available on the slots. At one time they advertised an OEM prototyping card which included full hardware details (Apple part no. A3B0001). Can anybody help with information? Please contact Roger Jenkins on [illegible]

Chairman's Corner

It has been very heartening that, following our appeal for a new editor, we had four candidates. I hope that this reflects an increased feeling that BASUG is a club, and that members always feel that if they have something to contribute, they can. I would like to wish Dr Peter Baron, a member for some time of BASUG, success in his role as editor. Without <u>your</u> help, he will not be able to do such a good job. If you can help in any way, please let us know.

I would also like to thank Yvette on behalf of all the members for the work that she has put in on the magazine. The standard of presentation is surely the highest of any of the user group magazines.

Thanks are also due to Jim Panks who has been unable to carry on as Software Librarian because of work commitments. Graham Attwood has taken on the full job. As you may know, the Software Library is currently being upgraded to DOS 3.3, and soon will consist of fewer, but better disks. Recent additions such as those from Washington Apple Pi are already to a very high standard.

We are currently looking for a volunteer to man the BASUG hotline, a job that involves pointing people to an expert within the group who may be able to help with a problem. If anybody is interested in helping with this valuable service, please contact Martin on the hot line to discuss what is involved or Fran to put your name forward. We are always looking for more 'experts'. It doesn't matter how specialised your knowledge, we can use it.

BASUG has a number of areas for 'first aid' with technical problems. We would like to establish a Technical group who could help solve more detailed, or written problems. If you are interested in acting as the coordin-ator, or as one of the group, let me know.

Please can I stress again that the BASUG committee and helpers are volunteers who have full-time jobs. Unless we print otherwise, please do not call outside weekday evenings.

Finally, some of will already know that this is my last Chairman's Corner. At the time of writing, my successor is not yet elected, but I would like to wish him (or her) the best of luck, and ask all of you to give the same support and encouragement that I have received in my time as chairman.

Expert Systems

EXPERT SYSTEMS : an overview

by Leo Crossfield

Computer Assisted Learning Group
Department of Educational Studies
University of Surrey
October 1984

INTRODUCTION.

Since the late 1960s a sub-discipline of main line artificial intelligence (AI) research has developed. This research has tended to concentrate upon the development of programs which capture a level of expert performance within a small field of speciality (domain). Expert system research is a sub-set of intelligent knowledge-based systems (IKBS) research; thus the main emphasis of this research effort lies upon the concept of 'knowledge' and its representation within the computer.

The programs that were the outcome of AI research during the late 1950s and early 1960s, were attempts to develop domain independent problem solving routines, that is, programs which are capable of solving problems of a general nature. These early programs were found to be limited and the general approach was abandoned, although it still stands as an ideal. The new emphasis has tended to focus upon programs which are domain specific problem solvers that have a knowledge base of domain dependent facts and procedures (relevant to the specific problem domain) available to them. This new approach can be seen as a fundamental change in the principles of the approach towards the development of research into intelligent programs.

WHAT DO WE MEAN BY THE TERM 'EXPERT'?

There are many problems that are associated with what we understand by the concept of 'knowledge'. Also there are problems as to what we mean when we say that a person is an 'expert' or that a person has 'expertise'. If a person is said to have expertise, what does that person actually possess? Firstly we can say that the person is knowledgeable about a certain field or a given topic, secondly we would usually feel quite justified to say that the person is capable of understanding and solving many of the problems which relate to the field or topic. Also, if such a person did not know the answer to a problem they would usually have the skills and knowledge to seek out appropriate journals, books, articles, or other experts in an attempt to find an answer, or they would at least have some idea as to where an answer might be found. So, our expert has two main areas of knowledge available to him - private and public knowledge and also two levels of knowledge, that which is specific to the problem and that which allows the expert to gain extra knowledge (usually when the private knowledge is either exhausted or insufficient).

THE EMPHASIS OF KNOWLEDGE IN EXPERT SYSTEM RESEARCH.

"... expert performance depends critically on expert knowledge." (Hayes-Roth, Waterman & Lenat 1983 p 5).

The move towards an emphasis on knowledge is of fundamental importance to AI research for a number of reasons; first, the most difficult problems do not have tractable algorithmic solutions for they are usually resistant to precise analysis and description, unlike the problem areas of traditional data processing (i.e., stock control, accounts, etc.). Secondly, such problems tend to have different levels of abstraction; this joined with an imprecise nature of the problem requires mechanisms which are capable of dealing with imprecise facts and thereby making inferences on the basis of the knowledge which is represented within the system. Furthermore, by emphasising the role of a knowledge base, within a system, we can see that there are many complex areas of human endeavour which might well be represented such as aspects of legal reasoning, medical diagnosis, criminal analysis, geological exploration as well as a number of military applications. The most immediate significance of this research effort has major relevance for industry, the military and the government in general; although there has been a large number of developments in medical consultation. To date expert systems have tended to focus on several application areas which fall into the main categories of diagnosis, monitoring, control, interpretation and prediction.

The main concern of expert system research is in the area of private knowledge with the main question being 'how can such knowledge be extracted from the expert?' and if it can be extracted 'how can this knowledge be represented within a computer?'. This

approach has tended to view knowledge as some form of inexhaustible entity which people possess. Private knowledge can be viewed as 'rules of thumb' (procedures) by which an expert arrives at a conclusion, with the expert sometimes not knowing what rules, if any, were actually applied to the problem. Such rules of thumb have been termed as heuristics which enable experts to deal with incomplete information yet still arrive at a meaningful conclusion. The use of heuristics play a major role in AI programs for they allow the program to limit the search space by differentiating between paths which could lead to a valid or reasonable conclusion from paths which are obviously irrelevant. The use of these strategies aim at reducing the possibility of an exhaustive search and by so doing reduce the amount of processing required.

WHAT CONSTITUTES AN EXPERT SYSTEM?

As we have briefly discussed above, an expert system is a computer program which embodies the expertise of one or more experts within a particular, usually quite small, domain. This expertise is seen as both factual knowledge and procedural knowledge. Therefore an expert system, as well as having facts about the domain, also encompasses stategies for applying the knowledge to specific problems, within that domain. Such strategies also allow the system to make useful inferences, but the development usually moves along a development path which is quite different to standard data processing (DP). Because of the conceptual complexity of AI systems their development can often go through a number of cyclic development phases. Such a procedure could involve the results of each phase being discarded and the experience gained constituting the basis for the next phase. This approach to software development perhaps highlights the fact that AI programs can be considered as working hypotheses; consequently such an approach to software development is time consuming.

The development of a useful expert system can represent a couple of years work for a small research team, but such time scales are difficult to assess because of the individual problems which are associated with each particular domain. These problems tend to vary considerably and depend very much upon the availability of experts in the field and/or expert information from other sources. The other major factor in development time comes down to the amount of effort which is required in extracting knowledge and representing it in a machine-like form. (The title of 'Knowledge Engineer' has been termed for the individual or individuals that are involved in this process (Feigenbaum 1977). The amount of effort depends very much on the nature and accessibility of relevant software tools (and methodologies) that the team has at their disposal.

SOFTWARE TOOLS.

An important focus of expert system research (like any good software development effort) is the development of software tools which aid the development of intelligent knowledge-based software. Artificial intelligence research has spent a number of years in developing appropriate tools for specific domains. Such tools are intended to ease the process of building an expert system with the focus upon knowledge elicitation, the knowledge representation process, and the actual development of the software. The main area that has been found to be problematic is that different areas of expertise, when being represented, often require different tools and methods.

Tools for expert system research range from languages that have been developed for symbolic manipulation, such as LISP, (which is an acronym for LISt Processing) to programs which aid the design, construction and testing of an expert system. This software sometimes taking the form of a powerful programming environment.

SHELLS.

Expert system shells (or frameworks) are also considered as tools. A shell is in effect an empty expert system, that is, it is empty of any knowledge and therefore domain independent. The user is expected to fill the shell with knowledge from the relevant area of interest. Such shells include set search strategies (which the user cannot change) which are often based on some specific existing expert system, such as MYCIN or PROSPECTOR. This, in effect, defines the nature of the system by restraining its overall function towards a specific activity such as diagnosis, monitoring, control, etc.

The main idea of a shell is that a potential expert system user can buy a shell 'off the shelf' (so to speak) and adapt it to their own specific needs. This approach has received some criticism in recent years for although it is possible to buy a shell for

as little as £400 to run on an Apple or approx. £4000 to run on a more substantial machine such as a VAX. It is generally agreed that there are a number of problems associated with expert system shells. First, there is a consensus of opinion that a microcomputer with less than 512Kb of RAM is impractical for any serious application of this technology. This fact stands whether the development involves using a shell or writing the entire system in LISP. Generally AI tools are quite large and consequently heavy on memory, also a reasonable expert system might entail anything from 150 to 2000 rules. Secondly there are problems in the fact that the knowledge engineer's role involves the difficult process of knowledge elicitation which is undoubtably a difficult process. This process involves a number of skills in interviewing techniques although the main difficulties arise in regard to the nature of expertise, for often even the expert is not exactly sure of what he or she knows, or what procedures are used to solve specific problems.

KNOWLEDGE ELICITATION.

Knowledge elicitation is therefore usually a long cyclic process of eliciting the knowledge in the form of key elements and rules, extracting certainty factors from the expert, representing the knowledge, discussing the representation with the expert and making the necessary modifications. The entire process involves a number of cyclic paths of refinement until an agreement is eventually arrived at and the knowledge base appears to be stable. But, this is rarely the end of the story, for when the knowledge base is represented within the computer the performance of the expert system may still show errors and inconsistencies. Consequently the knowledge base will have to be 'tuned' (small or major corrections made, perhaps only to some aspect such as the confidence factors) until a specific performance level is achieved. The tuning of the expert system is usually done in consultation with the expert until the expert feels that the system's performance is similar to how the expert would have performed on similar problems.

Overall it seems to be generally considered that the main use of an expert system shell is for the first step in developing an expert system, constituting a valid feasibility study before a company needs to fully commit themselves to the development of a fully fledged system. In effect, shells can be viewed as a useful and practical method of assessing the feasibility of what this technology can offer over traditional data processing techniques. (For a more in depth discussion of software tools see Hayes-Roth, Waterman and Lenat (1983) Ch: 6 and also O'Shea and Eisenstadt (1984).

SYMBOL MANIPULATION AND THE REPRESENTATION OF KNOWLEDGE.

Perhaps the most distinguishing feature of expert systems, that has been briefly discussed above, are the techniques for representing knowledge (discussed in detail below), symbolic inference and heuristic search. Each of these features depend heavily upon symbolic manipulation and consequently the language LISP (which was originally developed by John McCarthy) allows the implementation of symbol manipulation on a computer. It was mentioned in the previous paper 'ARTIFICIAL INTELLIGENCE' (in Hardcore June 1984) that AI attempted to move the use of the computer away from its traditional role of being a purely mathematical tool towards a tool for the manipulation of symbols which could be realised in the 'real world'. Consequently the role of symbol manipulation has come to play a fundamental role in all aspects of AI research from natural language understanding and visual perception to robotics and expert systems. Specifically, expert systems are seen as a method of implementing symbolic reasoning, and the recent move towards the representation of knowledge could not have been achieved without a symbol manipulation paradigm.

The area of knowledge representation is a complex field which holds many problems with some being seen as effectively in the domain of philosophy; problems such as 'what do we mean by knowing', although AI research seems to be founded on the premise that 'knowing' must begin with some form of symbolic representation of facts about the world. From this view point it can perhaps be argued that AI programs differ from conventional data processing programs because they utilise symbolic representational inferences and reasoning.

There are a number of ways in which knowledge can be represented within an AI program and basically these can be narrowed down to four main methods: formal logic, associative networks, frames and production systems.

SOME KNOWLEDGE REPRESENTATION TECHNIQUES.

We have discussed (above) that for a computer program to use knowledge we must first find a convenient way to represent that knowledge so that the program can make full use of it. Such a usage involves the manipulation of the specialised data structure (knowledge base) so that the program can make use of it when making intelligent inferences.

In general each of the knowledge representation schemes (which are outlined below) although seemingly un-exciting, have had influences on issues in cognitive psychology for they touch upon some major issues and concerns for the study of cognition and intelligence in general.

1. LOGIC - Declarative Representation.

Logical representation can perhaps be considered as the classical approach of AI to the problem of representing knowledge. This approach has been termed 'declarative' representation, which means that the knowledge is declared in a formal logic, such as predicate calculus, which allows inferences to be made from the declared facts. This approach guarantees that any deductions are true if they are based upon true premises. This is perhaps the strongest reason why formal logic has been found to be useful in AI research.

An example of this approach can be seen in the following example:

All swans are white

which can be translated into predicate calculus as:

∀ x. Swan (x) => Are white

which is read as:

(∀ x.)	for all x which exists in the world and,
(Swan (x))	if x is a Swan,
(=>)	then (implies),
	x is white.

Within the framework of formal logic there are specific rules called 'rules of inference', which state that if the facts (premises) of an argument are true, then they can be used to derive other facts which will, of necessity, also be true (deduction). A typical example is found in the form of syllogisms:

1. All men are mortal. (premise)
2. Socrates is a man. (premise)
therefore
3. Socrates is mortal. (conclusion)

where 1. and 2. are declared facts and 3. is the deduction which MUST be true, as long as the premises are true.

Predicate calculus has been used successfully in many AI programs, and has been popular because new facts can be derived from old existing facts (or existing knowledge base). But, these techniques were found to be limited because there was a need for the program to have access to procedural as well as declarative knowledge, therefore allowing the system to infer how relevant certain facts were to a given situation. (For a more detailed discussion of predicate logic see Quine (1980).

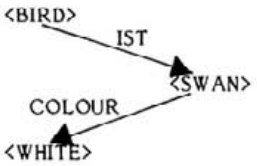
There is a controversy in AI between the declarative and procedural representations of knowledge. Winograd (1975) explains that the debate can be viewed as an incarnation of the old philosophical distinction between 'knowing that' and 'knowing how'. "The declarativists ... do not believe that knowledge of a subject is intimately bound with the procedures for its use." (Winograd 1975 p 186). On the other hand the proceduralists believe that "... many things we know are best seen as procedures, and it is difficult to describe them in a purely declarative way." (Winograd 1975 p 189). So how can knowledge be represented in a procedural form? There are three major forms of procedural representation: associative networks, frames, and production systems.

2. ASSOCIATIVE NETWORKS.

The associative network (or semantic network) was developed by Quillian (1968) and others as a psychological model of human semantic memory, and has been further developed in AI research in general and the field of natural language understanding in particular.

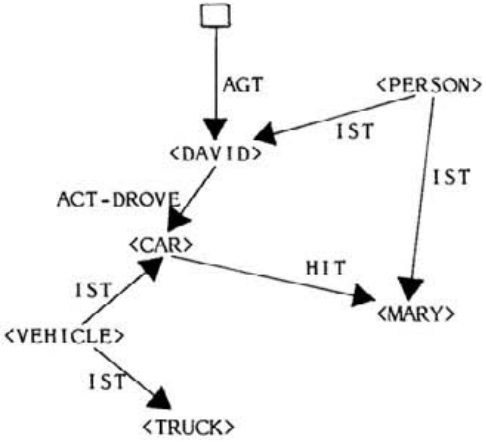
The overall structure of an associative network can be considered as a network, or graph, where each main item that is specified (node) is linked to other items by lines (arcs). The following diagram is an example of a simple associative network. Some points to note are that arcs are usually labelled, in our example we have a label IST which stands for 'instance', e.g., a swan is

an instance of the category bird.



We could make a more complex network by increasing the associations:

AGT = AGENT
IST = INSTANCE



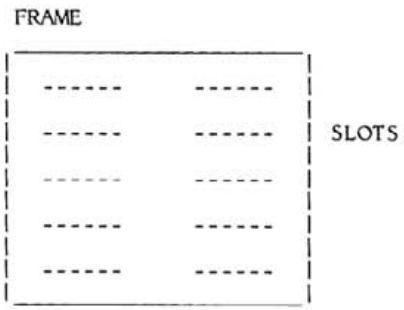
This network suggests that <DAVID> was the main agent who committed an act of driving a <CAR> (which is an instance of <VEHICLE>) that hit <MARY>. Where both <DAVID> and <MARY> are instances of <PERSON> and <CAR> and <TRUCK> are instances of <VEHICLE>. We can see from these examples that it is fairly easy to represent a quite complex set of associations within a given situation.

One major problem of associative networks, unlike predicate calculus, is that there is no intrinsic guarantee that a given inference which is made from the representation will be valid. Therefore although we have a more flexible representation scheme, which can be easily manipulated, we have a scheme which depends solely upon the program which manipulates the data structure. (For a more in depth discussion see Barr and Feigenbaum 1981 pp 180-189).

3. FRAMES.

Frames are the most recently developed scheme (see Minsky 1975). Frames can include both declarative knowledge and procedural knowledge, but both of these forms of knowledge are contained within a data structure which gives the situational context. A frame can be viewed as a block of information about a specific area which contains information that can be either declarative facts or procedural information; these areas are called 'slots'. It is common to have general frames as well as specific instance frames, for example a general frame could contain general details of a person with a specific frame being the details of some specific person.

A frame can be conceptualised as:



For example a frame could be developed to contain information about a specific area and within the frame the slots (or knowledge hooks) will contain knowledge of the area that is represented by the frame. Kuipers (1975) details a number of properties which he sees as fundamental. He also gives a detailed scenario to explain the main idea behind a frame representation; this example scenario suggests that whenever we meet a new situation we take to that situation a number of pre-conceived ideas, notions and expectations. Any reasonable expectation that we hold is usually based upon both our previous general experience of the world as well as our experiences of similar situations. These expectations might be 100% correct, totally wrong or somewhere between these two extremes. When we are confronted with our actual observations of the new situation we might well change our expectations to encompass the newly gained knowledge. In effect we start with generalised frames and gradually either modify these existing frames or pull in more relevant frames - often discarding the general frame completely.

It could be argued that we code many stereo-

typical situations in this manner and modify them on the basis of our experience. Indeed it could also be argued that within these frames we might well store actual facts such as 'dogs have four legs' (in a general dog frame) as well as procedural knowledge such as 'if I want to go through a door my first action must be to open it'. One can see that this simple rule involves other knowledge such as knowledge that doors can be opened, and knowledge of facts such as doors can be opened if they are closed except when they are locked. If they are locked we will need information about locks, therefore the door frame could involve a pointer to a lock frame. Within the lock frame there would be information about keys which would allow us to gain the knowledge that 'one must find the appropriate key for the lock'.

One can therefore see the importance of knowledge representation schemes which can handle both declarative and procedural knowledge. Barr and Feigenbaum (1981) give a good example of a generic frame for a dog:

Generic DOG Frame
Self: an ANIMAL; a PET
Breed:
Owner: a PERSON
(If-Needed: find a PERSON with PET=MYSELF)
Name: a PROPER-NAME (DEFAULT=ROVER)

DOG-NEXT-DOOR Frame
Self: a DOG
Breed: MUTT
Owner: Jimmy
Name: Fido
(Barr & Feigenbaum 1981 p 159)

Barr and Feigenbaum (1981) suggest that frame based processing allows the system to determine whether a given frame is applicable within a given situation. Also that a likely frame can be selected to aid in the process of understanding the current situation. If this frame finds that it is not appropriate, it could transfer control to a more appropriate frame.

It is also possible, from a frame based system to implement other knowledge representation schemes within a given slot, such as associative networks or rules; thus creating a 'hybrid' representation scheme.

4. PRODUCTION SYSTEMS.

Production systems were developed by Newell and Simon in 1972 and represent the scheme which is most often used in current expert systems. In essence a production system is a scheme in which knowledge is represented by rules in the form of: IF <condition> THEN <action>

This structure is formally known as 'antecedant-consequent pairs' where the antecedant can be either a simple condition or a more complex condition.

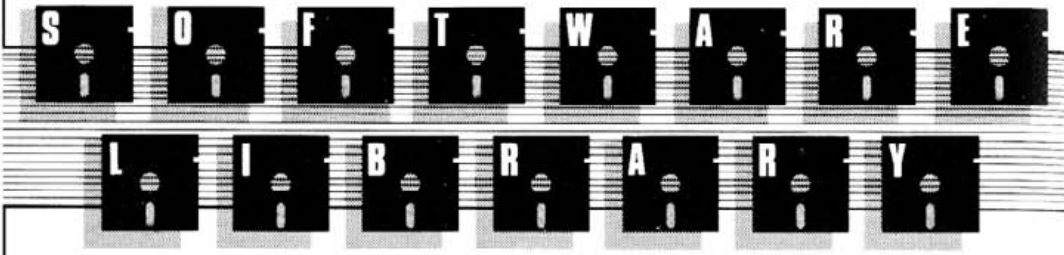
A knowledge base is made up of a number of rules (production rules) and are used in an attempt to mechanise complex decision making. Barr and Feigenbaum (1981) explain that a production system consists of three parts: a rule-base, a context list, and an interpreter. The production system's interpreter will use a specific control strategy, and (depending upon the control strategy which is implemented within the expert system, in the case of a forward chaining strategy) will first take the user's input and try to match the condition (left hand side of the rule) of the first rule in the relevant section of the knowledge-base. If the interpreter fails to find a match it will then try to match the condition of the next rule (and so on through the rule base). Alternatively, if the first rule was a successful match, then the rule will be stored as the first element in the context list (but the action, or right hand side of the rule will not yet be acted upon). The interpreter will then try to match the condition part of the second rule. If this second match is also successful, then this rule will also be added to the context list and so on until the interpreter fails to find any further candidates. When the rule base (or relevant section of the rule base) has been exhausted, there may be either a number of rules in the context list, or only one. If it is the latter then there is not a problem and that rule can be 'fired', i.e. the action (or right hand) side of the rule can be implemented. On the other hand, if the context list contains more than one rule then there is a problem. This problem is termed as a 'conflict' and it must be resolved before a rule can be selected to be fired. The criteria which the interpreter will use to resolve this conflict can vary tremendously depending on the actual application and requirements. For instance, the criteria for conflict resolution can be as simple as 'take the first rule that was found to be appropriate', or it could take a more complex form such as

'take note of all the certainty factors, which are associated with each rule, and fire the rule which has the highest

RECENT ADDITIONS

disk
88 Utilities Visicalc (Magicalc etc) and spreadsheet models
89 Personal/educational programs for domestic use
90 DOS 3.3 utilities and disk maintenance programs
91 Action and adventure game mixture
92 More personal/educational/domestic programs
93 Educational tests of maths
94 Educational tests of algebra, spelling, geography and maths
95 Textfile reader/mailing list/invoicer/grades database
96 Four different pinball games from 'Pinball Constructor Set'
97 More educational maths and arithmetic programs and tests

* all the above have full instructions and documentation on the disk *



Disk 99 Printer utilities
Apple DMP/Imagewriter **NEW**

Make proper use of this superb printer's facilities which are not well explained in the manual. (How unusual!)

Disk 100 DIVERSI-DOS **NEW**

Formats disks for DOS, ProDOS, Pascal and CP/M
Fast copies disk-to-disk with 1 or 2 drives
Checks drive speeds.
Includes free 'Dog-fight' game.

Disk 101 POM's 6 **Disk 102 POM's 7** **NEW**

Parlez vous francais?
Two disks from the French Apple Club.
Good graphics and various utilities.

■ All disks are £5.00 inclusive.
Send your order form and cheque (or Access No.) to P.O. Box 174, Watford, WD2 6NF or ring Fran on 0635 46385 for Access sales.

NEW . NEW . NEW . NEW
EAMON scenario disks

E13 Treasure Island
E14 Furiso
E15 Heroes Castle
E16 Mondamen Caves
E17 Merlin's Castle
E18 Hogarths Castle
E19 Death Trap
E20 Black Death

Special Offer:

EAMON Starter Pack
Disk E1 (Eamon master) + 4 scenario disks E2 to E5 for
■ £20 . . . save £5

B · A · S · U · G

certainty factor. If two rules have the same certaintly factor then fire the rule with the highest certainty factor which was first found on the context list'.

Once a rule is selected from the context list it is then fired.

The control strategy, which has been detailed above, is known as 'forward chaining', that is antecedant to consequent. A number of expert systems have been developed using this chain of reasoning, the most classical example being MYCIN (Shortliffe, 1976). Forward chaining usually involves the user to enter a specific enquiry including as many details as possible; such enquiries could obviously be highly detailed. Once the entry has been completed the expert system will attempt to solve the problems which are associated with the enquiry, eventually displaying an answer. The main problem which is associated with this strategy is that some of the enquiry details might be found, by the expert system, to be irrelevant. An example of this point being that to make a detailed enquiry about a person's child including age, name etc., is irrelevant if the system finds that the person does not have any children. An alternative control strategy is backward chaining, that is from consequent to antecedant, an example of this control strategy can be seen in PROSPECTOR (Duda et al., 1979). Backward chaining starts with a goal, if it cannot solve the goal it will break it down to smaller sub-goals. It will continue to further sub-divide the goal until it eventually reaches a level of detail where it succeeds. Once the system has been successful, at the lowest level, it will work up the list of sub-goals until it can solve the main (top level) goal. This strategy involves the user typing in the minimum of information, with the system asking the user for further, more detailed information, as, or if, it requires it.

An ideal control strategy needs to encompass a flexibility which will allow it to switch between these two classical strategies as the need arises. (For a more detailed discussion of production systems see Barr & Feigenbaum 1981 pp 190-199).

SELF KNOWLEDGE AND JUSTIFICATION.

We have seen how knowledge can be represented within a system but generally it would be agreed that people have the very special ability of being able to reason about their own thought processes. This is a specific problem area for AI in general, for if we could develop a system which could reason about itself we would be brought sharply into the realms of cognition and self-knowledge. An important point to make is that AI research is explicitly interested in this area, for if an IKBS could implement some form of self-knowledge then it would be possible to develop expert systems which are capable of rationalising their decisions as well as offering the user valuable explanations of how it had reached a specific conclusion.

The need for reliable explanation functions has been found to be fundamental to the development of expert systems. For example, when an expert system is being used as a professional consultancy system one can see that any diagnosis or recommendation that the system makes will need to be justified. Generally current expert systems do involve a fairly simple explanatory mechanism which, upon the user's request, can offer the user a backtrack over its decision tree. Such systems may present the user with the previous rule, or the chain of rules that it was attempting to satisfy. In a fairly crude sense this can be seen as a form of explanation, but it begs the question 'can a rule stand as its own justification'. The alternatives are to either have the system generate an explanation automatically (this would involve the system 'knowing' the basic principles on which the explanation is founded) or perhaps a less grand alternative would be to associate some text with the relevant high level (or meta) rule. This latter solution would, of course, only provide an approximation, but might serve as a better explanatory function than relying upon the generation of the system's historical path through the knowledge base.

ARE EXPERT SYSTEMS 'INTELLIGENT'?

Whether one could consider an expert system as a truly 'intelligent' system is an interesting question. One might perhaps expect an intelligent system to be capable of learning from its experience, and it would be reasonable to expect that such learning should enable the system to not only improve its overall performance (which could involve adding new facts to its knowledge base) but it should also be capable of improving its method of performance. Such a learning system would involve a great deal of high level knowledge, possibly involving another expert system to manage the re-organisation, connections and relations of the existing

and newly acquired knowledge.

"An 'expert' system can use a knowledge system which is hierarchically organised. An 'intelligent' system can 'detect' hierachical structures (and build them) from knowledge-items which are connected by relations in a tangled network." (Moore, 1984).

This suggests that the ability to reorganise a knowledge base is a creative ability, and also that machines will never be capable of creative action. (This issue is a complex area of debate in its own right).

CONCLUSIONS.

The area of expert systems is becoming a major focus of government funds even though this research is still in its most elementary experimental stage. Yet, even though this research is still in its infancy it has grabbed the imagination of many people in both industry and academia, perhaps due to the important role that it could play in assisting control as well as legislation based organisations. Expert systems have evolved to a stage where development is quite well understood and considered as feasible to many areas of commerce and industry. This research area is therefore seen as the first major practical contribution that AI research has offered to the world at large. Along this line of thought we can see that government bodies are actually recognising the potential of expert systems in many diverse fields and to that extent the British government have actually financed a large number of 'demonstrator' projects, via the Alvey programme, to demonstrate the use of this aspect of high technology in specific fields of endeavour.

There are a number of reasons why expert systems research is important, but primarily, expert systems encompass a number of important features (some of which have been mentioned in this paper). There is little doubt that the research into expert systems has somewhat enriched our understanding of the cognitive processes, as well as offering us a product which has practical implications for industry and commerce. We have also gained a great deal by having the tools and methodologies which have been developed by this area of research with such tools inevitably aiding us in the task of developing more powerful and 'intelligent' systems in the future.

REFERENCES AND RECOMMENDED READING.

** Barr, A. & Feigenbaum, E.A. (1981). The Handbook of Artificial Intelligence. Vol 1. Pub: William Kaufman Inc.

** Barr, A. & Feigenbaum, E.A. (1982). The Handbook of Artificial Intelligence. Vol 2. Pub: William Kaufman Inc.

* Bobrow, D.G. and Collins, A. (1975). (Eds.) Representation and Understanding. Pub: Academic Press.

** Cohen, P.R. & Feigenbaum, E.A. (1982). The Handbook of Artificial Intelligence. Vol 3. Pub: William Kaufman Inc.

** Dehn, N., and Schank, R. (1982). Artificial and Human Intelligence. In Steinberg (1982), pp 382-391.

Duda, R., Gaschnig, J., and Hart, P.E. (1979). Model Design in the PROSPECTOR Consultation System for Mineral Exploration. In Michie, D. (Ed.) Expert Systems in the micro-electronic age. Pub: Edinburgh University Press, pp 153-167.

Feigenbaum, E.A. (1977). The Art Of Artificial intelligence: Themes and Case Studies in Knowldege Engineering. In IJCAI, pp. 1014-1029.

* Hartley, R.T. (1980). The Competent Computer. Technical Report MCSG/6. Department of Computer Science, Brunel University, Uxbridge, Middx.

** Hayes-Roth, F., Waterman, D.A., Lenat, D.B. (1983). Building Expert Systems. Pub: Addison-Wesley 1983.

* Johnson, L., and Keravnou, E. T. (1984). Expert Systems Technology : a guide. Pub: Abacus Press.

** Johnson-Laird, P.N. (1983). Mental Models. Pub: Cambridge University Press.

* Kuipers, B.J. (1975). Representing Knowledge for Recognition. In Bobrow and Collins (1975).

Minsky, M. (1968). (Ed) Semantic Information Processing. Pub: Massachusetts Institute of Technology.

Minsky, M. (1975). A Framework for Representing Knowledge. In Winston, P. (Ed.) The Psychology of Computer Vision. Pub: McGraw Hill, New York.

Moore, L. (1984). The Unknown Generation. In Burns, A., (Ed.) New Information Technology. Pub: Ellis Horwood Ltd., pp. 159-169.

** O'Shea, T. & Eisenstadt, M. (1984): ARTIFICIAL INTELLIGENCE: Tools, Techniques and Applications. Pub: Harper Row.

Quillian, M.R. (1968). Semantic Memory. In Minsky (1968).

* Quine, W. V. O. (1980). Elementary Logic. Pub: Harvard University Press.

Shortliffe, E.H. (1976). Computer-based Medical Consultations: MYCIN. Pub: American Elsevier.

** Steinberg, R.J., (1982). Handbook of Human Intelligence. Pub: Cambridge University Press.

** Weis and Kulikowski (1984). A Practical Guide to Designing Expert Systems. Pub: Chapman and Hall.

* Winograd, T. (1975). Frame Representations and the Declarative/Procedural Controversy. In Bobrow and Collins (1975).

Winston, P.H. (1970). Learning Structural Descriptions from Examples. In Winston, P.H. (Ed). The Psychology of Computer Vision. Pub: McGraw Hill, 1975. pp 157-209.

* = Recommended reading
** = Essential reading

Special Offer

BASUG SPECIAL OFFER FROM ELITE

ITEM	R.R.P. (INC VAT)	DISCOUNTED PRICE
Ramview	£ 69.00	£ 55.00
Ramview 64k	£138.00	£110.00
Wildcard	£ 92.00	£ 70.00
Wildcard Plus	£136.85	£110.00
S-C Macro Assembler	£ 79.35	£ 62.00
Wild-Word	£ 40.25	£ 33.00
Appli-Kit	£ 28.75	£ 23.00
Edit-]I[	£ 28.75	£ 23.00
Symbol-77	£ 28.75	£ 23.00
Password-69	£ 28.75	£ 23.00

Edit-]I[is a line based editor which can be co-resident with Appli-Kit. Symbol-77 is a graphics utility which allows text and other symbols to be displayed on the hi-res screen. Password-69 allows a password to be placed on a disk - until the password is given to unlock it, the disk is almost unusable. The other product descriptions can be seen in Elite ads in this issue.

Any order for ten or more of any one item received from BASUG will be given the discount. Elite will be happy to mail direct to the people concerned and postage and packing are included in the price. However overseas members should add £2 for postage. To take advantage of these offers, send a cheque made payable to BASUG Ltd. for the appropriate amount to BASUG at the P.O. Box, stating clearly your name and address and what you require. As soon as Fran has the orders, she will send the list of names and addresses to Elite, assuming that there are enough to warrant the discount. If not, she will let you know. We must have your orders by 15th December.

Small Ads

WANTED

I would like to buy a copy of Stoneware's Professional Graphics Processing System or any other software that is compatible with a Symtec light pen. Details to David Steward, [illegible].

****

For Sale. Apple cooling fan made by MR Engineering, Chicago. Sits on top of power supply, 240V. Originally $69 - sale price £20. Mike Watson. [illegible] (day) [illegible] (evening).

****

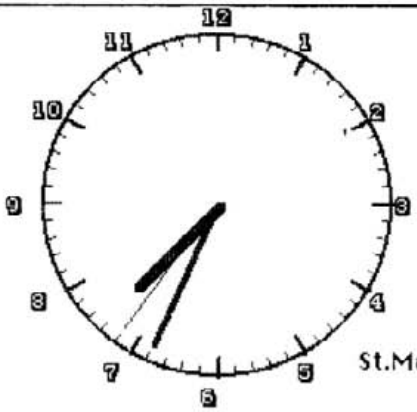
FOR SALE.
Parallel Printer Card, suitable for Apple DMP or C.Itoh. Built in text & hi-res dumps. Special character design & driver software. Complete with manual and demo disk. £60 o.n.o. Ring Clive Benjamin on [illegible].

****

ITT 2020 fitted with Apple motherboard, Digitek RAM Master, Kaga Monitor, 2 disk drives (1 brand new) plus manuals, disks, etc.

Offers to Mrs. J. Brooks on [illegible].

Mac Clock



St.Mac Clock

This clock is produced from the following listing in MS-Basic:-

```
10 CLS : PI=3.1415926#: HR = 12
20 OS =-1
30 DEF FNX (R) =110 +100*SIN(R)
40 DEF FNY(R)=110 - 100 *COS(R)
50 FOR R = 2 * PI TO 0 STEP -PI/6
60 CALL MOVETO(FNX(R),FNY(R))
65 CALL TEXTFONT(9):CALL TEXTMODE(2):CALL TEXTSIZE(9):CALL TEXTFACE(16) :
70 PRINT HR;: HR=HR-1
80 NEXT R
81 DEF FNA(R)=125+90*SIN(R)
82 DEF FNB(R)=105-90*COS(R)
83 DEF FNC(R)=125 +85*SIN(R)
84 DEF FND(R)=105 -85 * COS (R)
85 FOR R = 2 * PI TO 0 STEP -PI/30
86 LINE (FNA(R),FNB(R))-(FNC(R),FND(R))
87 NEXT R
88 DEF FNE(R)= 125 + 80 * SIN(R)
89 DEF FNF(R) = 105 -80 * COS(R)
90 FOR R = 2 * PI TO 0 STEP -PI/6
92 CALL PENSIZE(2,2)
94 LINE(FNA(R),FNB(R))-(FNE(R),FNF(R))
96 NEXT R
98 CIRCLE (125,105),90
100 T$=TIME$
110 M=VAL (MID$(T$,4,2))
120 H=VAL(MID$(T$,1,2))+INT(M/5)/12:IF  H>=13 THEN H=H-12
130 S=VAL (MID$(T$,7,2))
140 IF S=OS THEN 100
150 H1 = H*2*PI/12
160 M1= M*2*PI/60
170 S1=S*2*PI/60
180 CALL PENSIZE(3,3)
190 IF OM <>M THEN LINE(125,105)-(125+80*SIN(EM),105-80*COS(EM)),30
200 CALL PENSIZE(5,5)
210 IF OH<>H THEN LINE(125,105)-(125+60*SIN(EH),105-60*COS(EH)),30
220 CALL PENSIZE(1,1)
230 : IF OS<>S THEN LINE (125,105)-(125+80*SIN(ES),  105 - 80 * COS (ES)),30
240 LINE (125,105)-(125+80*SIN(S1),105-80  * COS (S1))
250 CALL PENSIZE(5,5)
260 LINE (125,105)-(125+60*SIN(H1),105-60*COS(H1))
270 CALL PENSIZE (3,3)
280 LINE (125,105)-(125+80*SIN (M1),105-80*COS(M1))
290 OS=S: OM=M: OH=H: EM=M1: EH=H1: ES=S1
300 GOTO 100
```

From a listing in ST Mac enhanced by Peter Trinder

Mac Notes

The following is a discussion posted by a Compuserve user pointing out some of the improvements to Microsoft Basic for the Macintosh:

I received my upgrade (Version 1.01) to MS Basic by calling Microsoft some weeks ago, getting a return authorisation number and mailing my old master back. The upgrade contained the following 'enhancements':

LPRINT no longer causes buffer overflow when Imagewriter DIP switch is changed from DTR handshake to XON/XOFF.

Files can now be edited with MacWrite and moved from MW to Basic if they have been saved with 'text only'.

'Command-' now stops program and is equivalent to 'Command-C'. Hopefully at some point they will pass Command C through; it still stops program in the update.

Running on Lisa under Macworks, CIRCLE statement is ok and full screen is now available.

A dumb terminal demo program is included and shows quickdraw calls.

SQR is nearly twice as fast.

The following corrections have been made:

A SAVE no longer saves only part of the file if you go to the FILE menu before the disk write is complete.

NEW now resets the output window title.

A bug in passing a numeric display to a ROM function or using it in a GET or PUT statement is cured; it produced a fatal system error.

A bug causing 10 GOTO 30; 20 GOTO 40; RENUM to cause undefined line number messages has been fixed.

Files now keep their folder identity when opened.

A bug causing two scroll thumbs with three list windows is fixed.

NOTE: The bug that caused some icons to become 'generic' reported in several messages here has apparently been fixed, although that's not reported on Microsoft's list. One way to update old MS Basic program disks so that icons are correct is: load the new basic; make a copy to a clean, newly formatted disk; delete any files you don't want to end up on all your upgraded disks from the copy; if you have Fontmover, remove any excess fonts from the system folder of the copy to save space; put everything left in the copy in one folder. Then use the copy disk as follows: make it the boot disk (restart); copy everything except system and empty folders and MS-Basic from each old disk in turn to the copy disk; use either the 4-pass (if you only have one drive) or the disk icon move to copy the entire copy disk back to the old disk; trash everything on the copy disk except the system and empty folders and their contents; empty trash; continue.

Another tip: If you use File, the sample text editor from the Mac Software Supplement on many of your disks, as I do, since it saves a lot of space compared with MacWrite if you only want to do program editing, then you will find that if you remove all the fonts except the system fonts from those disks to save space, File may produce funny-looking characters. If it does, a solution is to use Resource Mover to change the FREF entry numbers for File to match two system font FONT entries in System.

Wildcard Plus

by Mark Whelan

Judging from recent letters to Hardcore and other publications, there is a fair amount of interest in 'Copy-cards', or the hardware devices for copying your precious software. However, there is little in the way of practical help published. I hope the following may help those trying to decide if they need a back-up device, as well as those who already have one.

The obvious application is in backing-up protected software, so that you don't have to wait months for that replacement disk that is always an American product. The WC+ will back up 48K, 64K, or 128K programs, giving the added advantage of faster loading times too.

An example of the card's use in speeding things up is the ability to do away with pre-boot disks. Simply boot the pre-boot, then the main program. When everything is as you want it, press the WC+ button and dump it as one file that is unprotected and auto-booting. This is fine for memory-resident programs which load fully first time.

For ones which access the disk after booting, a combination of hardware and software is required. It is often the case that the most heavily protected part of a program is the boot routine. Copying this with the WC+ is accomplished readily. As before, simply get your program through to the required menu, and make a copy at this point. From now on this disk can be booted to get up to the point where data is accessed. The data which is referred to from time to time is very often unprotected, or can at the very least be handled by a bit copier. By copying the disk itself with some piece of software, the data is readily available and can be swapped for the boot disk at the appropriate point. This may be a little inconvenient, requiring two disks. How much more inconvenient though to lose the program entirely, through carelessness or simply overuse.

A less obvious use is in controlling your Apple. An example that will appeal to games fans is one I used on a Wizardry scenario disk. These had an annoying habit of going down just as I was getting a decent party of characters built up. And of course my back-up scenario was sufficiently out of date as to be useless. Whilst there is a 'Recover Out Characters' utility, this requires a readable disk. Often mine were so corrupt as to be missing the vital areas. Perhaps a good programmer could sort this out with enough utilities. If not, try this:

1) Run the Recover utility.
2) Put in a good scenario copy and press <RETURN>
3) When it gives the message "Recovering" press the WC+ button.

At this point the drive continues to turn, but nothing is happening. Now put in the corrupt disk. More often than not (mine only failed once) the characters are readable, only the disk recognition fails. So once the damaged disk is in place, press <R> (for RESUME). The WC+ now returns control to the Apple, the recovery utility loads up your characters, and they can be dumped to a good disk.

Another use for the card turned up when I was sent a copy of RAMDRIVE e/c by CCS & S, for evaluation. This is the latest release of the //e only disk emulation package (now //c compatible) as reviewed in August's Hardcore. This is a great package for anyone unable to warrant the extra cost of a second drive. It readily took all the little utilities I seem to spend hours looking for, because they are always on unlabelled disks. A 128K WC+ dump, and I was left with a quick loading package of my most often used software, without the bottom 64K being touched, as it all lies in the //e 80 column card's extra RAM.

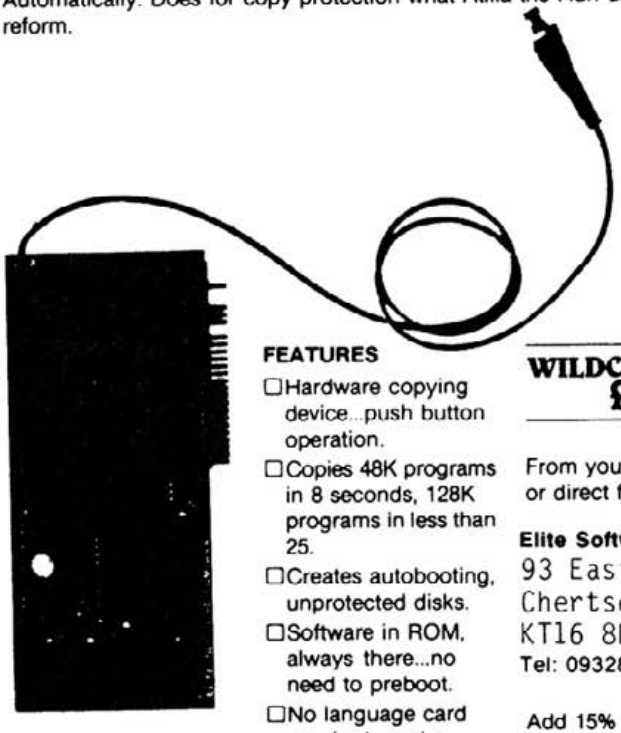
At the other end of the disk storage spectrum lies the hard disk. These show particular problems in getting protected software transferred onto them. Using a copy card such as WC+, the program can be copied, turned into a binary file if necessary (using the WC utilities), and put onto a free volume of the hard disk.

These are areas in which I have found the card useful. There are bound to be others not covered here, and any suggestions would be appreciated.

WILDCARD PLUS

COPIES 64K PROGRAMS IN 12 SECONDS

The new **WILDCARD PLUS** is the card that thinks for itself. One push of the button, one simple menu - with its own 6502 processor **WILDCARD PLUS** makes a perfect copy on any Apple, from a 48K][to an extended 128K // e. Automatically. Does for copy protection what Atilla the Hun did for social reform.



FEATURES

- ☐ Hardware copying device...push button operation.
- ☐ Copies 48K programs in 8 seconds, 128K programs in less than 25.
- ☐ Creates autobooting, unprotected disks.
- ☐ Software in ROM, always there...no need to preboot.
- ☐ No language card required, works equally well with or without.
- ☐ Utilities include: compression software - several backups may be stored on a single disk, recover BASIC programs, disassemble machine code software, screen dumps, etc.

WILDCARD PLUS
£119

From your local dealer, or direct from:

Elite Software Company
93 Eastworth Road
Chertsey, Surrey
KT16 8DX
Tel: 09328 67839

Add 15% VAT for delivery in UK P & P included.

Special Offer
see page 15

Dealer enquiries welcome.

Beginners' Page

by John Sharp

When I'm CALLing you.

When we dealt with PEEKs and POKEs, we also mentioned that sometimes you might POKE in a machine language program and then CALL it. The word CALL suggests that you can call on something that is already there. There are some routines in the Apple that are waiting to be used which are not commands in BASIC. In fact in Integer Basic there is a need to carry out certain operations for which there is no command, even though it might be there in Applesoft. One such case is the HOME command in Applesoft, which clears the screen and sets the cursor to the top left of the screen. If you are in Integer Basic you have no alternative but to use CALL -936. This will work in Applesoft as well and you sometimes still see it in Applesoft listings where an Integer programmer has not lost the habit when moving over to Applesoft. There are two other CALLs which are very useful for which there are no commands in either Integer or Applesoft. One clears to the end of a line (the equivalent of ESC E if you are in the editing mode), and the other clears to the end of the page (the equivalent of ESC F in editing mode). But first let's look at an example of why you may need to use them.

Suppose you have an INPUT statement which you then check to see if a number or string has been input correctly, and if not repeat the input until it is right. You might write a program thus:-

```
10 TEXT : HOME
20 PRINT "IF YOU HAVE TEN CATS AND
   THREE DOGS"
30 INPUT "HOW MANY ANIMALS DO YOU
   HAVE ALTOGETHER? ";A$
40 IF A$<>"13" THEN GOTO 30
```

If you run this program and the person answering types in the wrong answer, another input line will come up on the screen. Eventually the PRINTed statement in line 20 will scroll off the top of the screen and the condition will disappear. There will be a list of wrong answers with the question before them. We could rewrite the program, so that by using VTAB we write the question (i.e. line 30) on the same line of the screen each time, overwriting the last one. So insert a line

```
25 VTAB 10
```

and alter line 40 to:-

```
40 IF A$<>"13" THEN GOTO 25
```

Now if you run the program there will not be scrolling and lots of answers, if a wrong result is typed in. However, if the first answer was 1111, when you come to type in the second answer, the 1111 is still sitting on the screen, although the cursor is waiting on the first 1. In order to clear the line first you could use a PRINT statement, but this would produce complications in VTABbing. However, if we put a CALL -868 in line 30 before we carry on the problem is solved. Line 30 would now read:-

```
30 CALL -868: INPUT "HOW MANY ANIMALS
   DO YOU HAVE ALTOGETHER? ";A$
```

Try the program as outlined making the changes and see the result. I much prefer this sort of program instead of a screenful of wrong answers.

CALL -958 is useful in the same way. You might have a screen which is full of text but which has some instructions at the top, and you want to print only on the bottom half of the screen. There are two ways to do this. One is to alter the TEXT WINDOW. If you POKE 34,T where T is the number of lines down you do not want to print. You can only write below this line. So the following short program will print four lines then only print below those lines. If it is necessary to clear the screen below the line, the HOME command will do this.

```
10 TEXT : HOME
20 FOR N = 1 TO 4
30 PRINT "THIS IS LINE ";N
40 NEXT N
50 POKE 34,4
60 FOR M = 1 TO 50: PRINT "A LOAD OF
   RUBBISH AND YET MORE RUBBISH":NEXT
70 HOME
```

Note that line 10 has a TEXT : HOME to clear the screen, whereas line 70 only has HOME. If you put a TEXT command in a line it will automatically remove any TEXT WINDOW settings you may have. This is why it is often a good idea to use it as the first line of your program, so that anything left over from another program does not cause problems. To help others a TEXT:HOME in the last line when you exit your program will help when running another program where it is not the first line.

But suppose you wanted to clear the screen

starting at the middle of a line. You now need to use a CALL, CALL -958. This is a less likely problem, but could occur if you want to delete the text starting from the beginning of a particular sentence. It is so unusual that I cannot think of an example program. It will allow you to clear the screen and still keep the scrolling, whereas the text window setting will keep above the line you have designated until you alter it.

If you want to print on the left of the screen or just in one corner, you can set the sides and the bottom of the text window with other POKEs. Look in the Applesoft Reference Manual for more details.

DIFFERENT CATALOGS.

When I wrote last time about using the cursor to run over the catalog title to run it, I assumed many people were as lazy as me. This doesn't appear to be so. So let me extend it to let you put inverse or even flashing catalog titles.

To recap first. Let me go over the method of moving the cursor over a title to run a program. Type CATALOG and get the catalog up on the screen. If you have a very big catalog, you will not be able to use the method if the program has disappeared off the top of the screen. Press the ESC key and then the I key until you are on the line where the program name sits. Now type R (twice so that first you break the control over the cursor, and then print an R) then U and N. You will probably find that a number still exists from the file length, so use the space bar to type over with a blank. Now using the right arrow, run over the rest of the name of the file. Press return and you should be away.

When you carry out the following sequence of events, to give you flashing or inverse titles, this is the only way (or the modification I shall describe) of running the program. So if you are sitting comfortably in front of your machine, I'll begin.

Suppose we want to save a program under the name "TESTING INVERSE". Type the following in immediate mode:-

```
PRINT "SAVE ";: INVERSE : PRINT
  "TESTING INVERSE": NORMAL
```

The words

SAVE TESTING INVERSE

will come up, but with the last two in inverse of course. Now press the ESC key and as with the method outlined for running the CATALOG program, copy over this line with the right arrow once you have positioned the cursor on the S of SAVE. Now catalog the disk and there will be a program called TESTING INVERSE written up in inverse. To run the program, use either the method of writing a line (but with LOAD or RUN instead of SAVE) or copy over the CATALOG heading as described. If you just type RUN TESTING INVERSE then FILE NOT FOUND will come up because the comparison does not match. To FLASH the name just write FLASH instead of INVERSE when writing in immediate mode.

As a last note, you may like to try the following CALL to see what it does:-

CALL -1184.

Correspondence

Gerry Corti has sent us the following:

I think members of BASUG might like to see the following exchange of letters. Having spotted some material in our competitor 'Apple User', I thought it right to try and send a letter to Mr. Steve Jobs. This is quite normal practice in the oil industry where clients - even retail clients - do not hesitate to test out the attitude of a firm by writing to its Chairman or Chief Executive. I'm afraid the response conveys an extremely poor impression. Neither Jobs nor his office replied (nowadays this would be almost unthinkable in the biggest oil companies) and I finally got a telephone call from Mr. Kissach in the U.K. a month after I had written. He seemed surprised that I should press for a written reply but he eventually agreed to provide one. As you will see it hardly addresses itself to my letter, and I had to prompt him on the point covered in the last paragraph.

I think this is where our hope lies, i.e. with vigorous independent software companies who will spot the hole in Apple's development. For the present I do not think we can look for much from Apple on this front. Incidentally if any members know that the 80 column card //e and c is in any way adaptable to the][+ then I think we should let Apple Computer (UK) know.

****

22nd August 1984

Dear Mr. Jobs,

You gave an interview to the British magazine 'Apple User' in June 1984. In it you laid out how Apple Computers will fight and survive IBM. You went on to say that it is innovative software which will keep Apple there. "I would never start a hardware company now - I would start a software company", and then later "we think that the Apple 2 operating system is standard ..... we have shipped over 2 million Apple 2s in several years".

All this is fine and after a deal of market research I invested in an Apple 2 a couple of years ago, to be specific an Apple][Europlus. Now the very next month Apple User had an article on AppleWorks, the software program. It said "now the bad news. It will only work on //e, //c and /// - not on the Apple][or][+." I know you are not responsible for Apple User or their lighthearted dismissal of nearly 2 million Apple users, but I got my dealer to approach Apple Computer UK Limited and they tell me they got a reply that there was no intent to make AppleWorks available for the Apple][and they frankly did not seem very interested.

I know that your company has developed some excellent new hardware, having seen Mac and the adapted Apple 2, named //c. Fine, but on your own say-so it is the software that counts and I cannot see IBM making a marketing error of this sort. Quite a few Apple][owners in Britain are in a dilemma at the moment and they will not thank Apple for being forced into a re-investment in their primary hardware as opposed to an adaptation or upgrading of it.

Are we being correctly informed about Apple's intention on AppleWorks and its future availability in Britain? Are you going to introduce a four way package including graphics, which will be "multi task retrievable"? If you are not going to do either of these things in Apple, can you recommend any software house which is likely to do so or has done so?

I would prefer to make the contents of your reply known to fellow members of the British Apple Systems User Group, but as this is a personal letter should you prefer not I would respect your request.

Yours sincerely,

G. Corti

****

25th September 1984

Dear Mr. Conti (sic)

Thanks for your letter of 22nd August addressed to Mr. Jobs.

I understand your concern with the availability of AppleWorks on the Apple][Europlus. AppleWorks makes use of the firmware on the 80 column card which is available as an option on the Apple //e and is built into the Apple //c.

There is vigorous software development activity ongoing in the Apple // family and should I hear of an integrated package which functions on the Apple][Europlus, I'll drop you a line.

Yours sincerely,

W R Kissach
UK Marketing Manager.

****

Ed. - We know there are a number of allegedly integrated packages on the market. Has anybody used one?

Rosco Ltd

FREE Free 80 col. card when purchasing software over £300. Free 80 col. + 64k when purchasing software over £500. Valid until 31 Dec.

PRINTERS

	OUR PRICE
BROTHER	
EP 22	£129
EP 44	£219
HR 5	£139
HR 15	£399
Keyboard for HR 15	£149
HR 25	£695
HR 35	£895
EM 101	£849
CANON	
PW 1080A NLQ 80 Col	£299
PW 1156A NLQ 156 Col	£399
EPSON	
RX 80T	£189
RX 80FT	£239
RX 100FT	£399
MX 100 III	£379
FX 80FT	£359
FX 100FT	£479
DX 100	£419
LQ 1500	£999
JUKI	
6100	£359

MONITORS

KAGA	
12" hi-res (green)	£109
12" hi-res (amber)	£109
12" norm-res (PAL/RGB)	£199
12" med-res (RGB)	£259
12" hi-res (RGB)	£359
SANYO	
12" norm-res (green)	£69
12" hi-res (green)	109
14" norm-res (RGB)	£239
14" med-res (RGB)	£349
14" hi-res (RGB)	£499
NORMENDE	
14" TV/RGB monitor	£219

DISC DRIVES

Apple Compatible (full height) ... £149
Apple Compatible (half height, direct drive) £159

PERIPHERAL CARDS

NB: Rosco printer cards are CP/M, Pascal & Applesoft compatible + full graphic dump.

Printer Centronic	£38
Printer + 16k buffer	£89
Printer + 64k buffer	£129
Printer Cable	£12
80 column (Not IIe)	£48
80 column Inc inverse video chip. (New version. Not IIe)	£54
Expandable 80 column (IIe)	£49
Extended 80 column (+64k)	£89
64k upgrade kit	£49
Disc controller	£38
PAL (+ modulator)	£49
16k RAM	£44
Eprom writer (up to 2764)	£54
Eprom writer (up to 27256)	£99
RS232 (serial card)	£42
Parallel/serial interface	£59
Z80 CP/M	£44
A to D (16 channels)	£69
Real time clock (battery backup)	£59
I/O interface (4 ports)	£49

ACCESSORIES

52 key Ascii encoded keyboard (upper and lower case)	£59
Power supply	£59
AC cooling fan	£29
40/80 column switch	£9
Inverse video chip (not req. for new 80 col or IIe 80 col)	£6

SOFTWARE

Format 80	£109
MagiCalc	£59
Multiplan	£149
Sage Accounts + Payroll	£375
Merlin	£49
Routine Machine	£39
Microsoft Compilers	£119
Copy II Plus	£39

P & P: Computer, printer, monitor, sheet feeder £8 + VAT. Other under £100 add £3 + VAT, over £100 add £5 + VAT.
All prices subject to 15% VAT.

Rosco Ltd

289 Birchfield Road, Birmingham B20 3DD. Tel: (021) 356 7402.

Telex: 334303 TXAGWMG

Access VISA

AW II & Graphics

APPLEWRITER II WITH GRAPHICS

by Dr. M. G. Johnston

On countless occasions in the past (so many that it would be pointless to cite any particular references) the problems associated with producing graphics together with text from within a word-processor has been highlighted and, as yet, no overall solutions to these problems has been put forward. Sure, we could all go out and buy a Lisa tomorrow, probably much to the bank manager's annoyance, but I think that constant updating of one's system is expensive and, in general, unnecessary - let's make the most of what we've got. There's no great mystery as to why these problems occur - quite simply, word-processors deal with text files and not binary graphics files - but they are not insurmountable problems given the graphics facilities offered by certain printers which have the ability to convert text characters (or more accurately ASCII codes) into graphics patterns to produce pictures, logos, graphs etc. whilst printing text files. So what's the problem?..... Well, there are several which I shall outline with particular reference to the EPSON FX-80 dot-matrix printer and Applewriter II, although much of the article is of a very general nature applicable to all other EPSON dot-matrix printers and in all probability to all other printers with graphics capabilities (although I am unsure of the exact details involved with these).

The most simple way to produce graphics within text, no doubt used by many, is to print a text file, leaving gaps for the graphics, and then to insert these later using HIRES screen dumping methods, or, indeed, to insert them by hand. This is a tedious business if more than a few diagrams are involved.

A neater solution is to create a text file, from one or both of the HIRES screens, which is compatible with the Epson bit-image print mode, and to save this to disk for subsequent use with Applewriter II. I shall not enter into a description of bit-image printing here as it has already been adequately covered in previous articles by Mike Glover in Windfall and by Norah Arnold in Hardcore. See refs. 1) 2) 3). These articles outlined several problems encountered with 8-pin bit-image printing with the Epson printers, not least being the inability of the interface cards to pass all of the codes they receive onto the printer. It is not possible to use a by-pass routine from within Applewriter so these limitations must be accepted. Including the codes which Applewriter cannot generate itself there are 20 codes which fall into this category and any bit image column requiring any of these cannot be printed in 8-pin bit image mode. This makes for very scrappy pictures.

However, there is one very important point about the Epson 8132 interface which is not mentioned in the interface manual even though it is VITAL to the operation. This is the 'Transparency' character (CTRL-T). I call it this because the normal mode of operation of the interface is to mask off the high bit of all characters it receives before sending them onto the printer but issuing a CTRL-T unsets this condition and makes the interface transparent thus giving it the ability to send characters with their high bit set. A subsequent CTRL-T resets this condition. So now, at least we can get 8-pin bit image printing from within Applewriter (which sends all codes with their high bit set) simply by enclosing some characters between CTRL-Ts. e.g.

CTRL-@ sends 0 to the printer
CTRL-TCTRL-@CTRL-T sends 128 to the printer

But 8-pin bit-image printing is still severely limited because of the characters which cannot be sent from Applewriter II. 4-pin bit image printing, however, is not! (Don't go rushing to your manuals for notes on this because there aren't any!). Pseudo 4-pin bit-image printing is actually 8-pin in which four of the bits are not used, specifically the low order bits. Line spacing is normally set to 1/9-inch for 8-pin bit-image printing in order to connect dots in the vertical direction. For 4-pin bit-image printing the line spacing is set to 1/18-inch using the command from Applewriter II:

[ESC] 'A' [CTRL-D]

and subsequent bit-image codes are sent in which all of the four low order bits are set to zero (i.e. $00, $10, $20..... $F0) and all of these codes can be sent from within Applewriter although, as outlined above, the ones with their high bit set must be sent after a CTRL-T has been issued to the interface. So there we have it, an unbroken series of sixteen vertical dot patterns (including the null pattern) which can be built into an image quite easily using standard Epson bit-image printing

techniques. This can be quite easily implemented directly from Applewriter II even though it is rather a large task to type in a complete picture in this way. Remember, of course, to reset page length and interval to take account of the lines being printed 'on top of each other'.

Converting information from the HIRES screens to this 4-pin bit-image from is not an easy matter. However, much of the work involved has already been done by Epson (and others before them) since the Epson 8132 interface ROM contains a routine to convert the HIRES screens to a series of 8-pin bit-image columns ($CA00 to $CCFF) - How else could it dump the HIRES screens? So, the procedure is fairly straightforward - simply copy this routine, together with the character output routine ($C96F to $C9A5) to a convenient area of memory ($6000 is best) and then modify it to route its output to another convenient area instead of to the printer (this is rather more involved than it sounds). After running the program the 8-bit codes necessary to generate the graphics screen on the printer are stored in memory and it is a simple matter to convert these to two sets of pseudo 4-bit codes ready to be sent to the text file together with any necessary CTRL-T characters and Escape code sequences necessary to set the printer into bit image mode. The whole HIRES screen cannot be sent in this way because Applewriter has a maximum line length of 254 characters (compared to 284 needed for one complete graphics line) but a substantial part of it can. I find it easiest to keep all 'graphics files' separately from the text and to execute a WPL file to insert them where necessary.

I shall make the conversion programs available to the BASUG library as soon as I have made them user friendly. For further information on the Epson 8132 interface and Applewriter II I refer readers to an article by myself in a forthcoming issue of 'Laboratory Microcomputer' (vol.3 No.4) due out in November or December of this year.

References:

1) N. Arnold. Epson Pages.
Hardcore (vol.3. No.4.) August 1983.

2) M. Glover, C. Roper. Epson pages.
Windfall May 1983.

3) M. Glover, C. Roper. Epson pages.
Windfall June 1983.

Courses

By Patrick Bermingham.

Thanks are due to Richard Beck, BASUG's organiser of courses, for the excellent one-day course on the Apple's Assembly Language, held at the County Hall, London, on Saturday October 20.

Lecturer Ken Kelso began by outlining the main advantages of Machine Code over Basic: its economic use of Ram space and very much greater speed of program execution. After explaining the fundamental architecture of the Apple's 6502 microprocessor, its registers and supporting elements, Ken guided the group through Binary and Hexadecimal arithmetic onto the use of Assembly Language Instruction Codes. By the time the morning session was over the group had reached the stage where they could understand a simple bubble sort program written in Assembler Code.

The afternoon finished with each course member keying-in on the available Apple micros the sort program; and using a Prodos Editor/Assembler disk, loaned by Ken to everyone for the day, to create the Object Code. It was a revelation of the power of machine code programming to compare the speed of execution of the bubble sort program written in Applesoft basic with the same sort program written in machine code. The Basic program took 16 minutes to sort 200 random alphabet letters. The machine code program took less than 1 second!

As with all BASUG meetings that I have attended, new friendships were struck up, and arrangements made to help each other in various ways. There is nearly always someone who has the solution to the problem that you are currently despairing of ever solving. The room at County Hall was an ideal place. It was crammed with all kinds of microcomputing equipment and had many visual aids to assist a lecturer. I think all of us attending the course agreed that it would be a fine thing to have a follow-up course in the future. OK Richard? Incidently, Ken Kelso will be taking up a lecturing post at Watford College after Xmas, specialising in aspects of Computing.

****

From Phil King of March, Cambridgeshire.

I would like to thank everyone concerned with the Assembly Language Course which took

place at London County Hall on the 20th October.

Everything went extremely well. The only problems were minor problems due to the electricity supply and unfamiliarity with the equipment which I suggest should be rectified by having more courses.

As with all things it is easier to see things after the event, and looking back on the day I find that Ken Kelso (our tutor, who coped admirably and in a highly professional manner) may have been helped a little if each "student" had previous to the course answered a questionnaire on what they did or did not know.

Questions such as;
1. Are you familiar with hexadecimal notation?
2. Can you do basic hexadecimal arithmetic?
3. Do you know how to enter the Monitor?
4. Are you familiar with the Monitor commands?

This I would imagine would apply to most of the one day courses held by BASUG. That having been said I am looking forward to the next instalment.

Computer Music

by Roger Harris

Months ago - it was in June - I attended a performance given by the Electro-Acoustic Music Association (EMAS) in the Almeida Theatre in Islington, North London.

EMAS "encourages membership from anyone with an interest in electro-acoustic music: this is electronic and tape music, live electronics, computer music, indeed, any interface of science, technology and music" and its membership includes those "who have a stake in the creation and performance of music using electronics or computers".

They presented several works: all involved some form of electronic equipment. Some even had human accompaniment. Whatever the merits of the music, the musical scores can often be very beautiful for those of calligraphic bent.

Membership costs £6.00, (students, £3.00).

Details from the Hon. Secretary,
EMAS,
72 Hillside Road,
London
N15 6NB.

Book Reviews

Title: Mind and Media - The Effects of Television, Computers and Video Games
Author: Patricia Marks Greenfield
Publisher: Fontana
Price: £2.50
Paperback : 193 pages

reviewed by Danielle R. Bernstein

This very readable book is part of a series on the developing child. It compares the "new" media of TV and computers with print and radio. Why the feeling that you can only learn by reading? Reading is a serial media (one thing at a time) good for memorising facts. With TV and computers, many things happen at the same time (much like life).

But it is when Dr. Greenfield (Professor of Psychology at the University of California) writes about the merits of video games that she really started me thinking. This is the first serious discussion of the benefits of Pac-man and other such games that I had ever seen. She observed her son playing Pac-man and assumed that she would be able to play it but she was in for a surprise. She explains Pac-man and her attempts at it. For one thing, no one tells you the detailed rules of behaviour for each monster.

In Pac-man, you must deduce the rules from observation (again like the real world). Is that why there is no documentation? Video games teach parallel processing and interaction between things. In contrast, a game like chess is one-dimensional with no speed factor. Players cannot change the characteristics of the pieces in chess. In the fantasy (adventure) games, you add new dimensions to the characters by using parameters and levels of difficulty.

She thinks that the most harmful aspect of violent games are that they are solitary in nature and that two-person games are best. Children teach other children when computers are used. Computing is really a social activity, especially when you need to share computers in school.

How are girls to get interested in action games? For it is action, not violence, that both boys and girls of the TV generation enjoy. Unfortunately she does not spend much time on any practical answers. Instead, she goes on to discuss the educational values of programming and Word Processing. She finishes up by recommending a multi-media approach to education. A recommended book.

Title: Kids to Kids on the Apple Computer
Authors: Billy Saunders and Sam Edge
Publisher: Prentice/Hall (Reston)
Price: £8.95
Paperback : 8.25" x 11" : 168 pp.

Reviewed by Patrick Bermingham, and son Sean.

If you have an Apple][,][+ or //e and if you are one of those kind (but foolhardy?) parents who let your children use it occasionally for games, then why not give them the chance to broaden their interests and skills by putting a copy of this excellent "Kids to Kids ...." in their Christmas pillowcase. Well written and cheerfully illustrated, its chapters progress from using Print and Maths statements to Loops, Low-Res Graphics, High-Res Graphics and Sound, How to Make a Game to How to do your Homework on an Apple. My 13-year-old son, Sean, rated the book very highly.

Title: Astounding Games for your Apple Computer
Authors: Hal Renko and Sam Edwards
Publisher: Addison-Wesley
Price: £3.95
Paperback : 6" x 6" : 147 pp.

Reviewed by Patrick Bermingham

If you have decided to buy "Kids to Kids..." reviewed above, then craftily slip into one of the family's Christmas stocking a copy of this feast of Games Programs, for yourself! The 31 programs are well chosen and cover a wide spectrum of challenge and excitement. As the blurb on the back cover truthfully says "Battle with GALACTIC MONSTERS and ZOMBIE IN THE SWAMP, compete in a thrilling ROAD RACE, defend yourself against a SHIP'S ATTACK and set out on a nerve jangling SHARK HUNT. Gamble at LAS VEGAS and KENTUCKY DERBY, stretch your mind with board games like ONE TO FIVE and SHAKESPEARIAN SHUFFLE; and test your intellectual capabilities with VOWELS AND CONSONANTS, KEYBOARD-MEMORY and AT THE MARKET".

Apart from the fun of playing, a lot of programming tricks can be learnt through keying in the programs. Good value for money.

Title: Applesoft BASIC Toolbox
Author: Larry G. Wintermeyer
Publisher: Addison-Wesley
Price: £10.95
Paperback : 23.5cm x 18.5cm (9.25" x 7.25") : 514 pp.
ISBN: 0-201-14775-0

Reviewed by Dave Miller

"Stymied by the complexity of more advanced programming? Unsure what you can do with a beginner's knowledge of BASIC? Looking for a thorough and practical guide to using Applesoft BASIC? This book is just what you need." So "Applesoft BASIC Toolbox" proclaims on its back cover. It goes on to say that this book is ideal for the advanced programmer or the rank beginner and that this book will teach the following:

to design more efficient BASIC programs
to organise, store and retrieve data
to use serial, sequential and random access disk files
to create and use a complete data-file system.

The book is split into two main parts: the first section deals with Applesoft while the second deals with disk files. The part on Applesoft is split into 33 sections which cover either a single command or a group of similar commands. Commands are explained by using the following format:

instruction name & format
examples of instruction usage
purpose of instruction
rules for use
illustrations of these rules.

Section one titled "Getting started" is an introduction to the Applesoft part of the book. The author advises the beginner to study not only the text but the program code as well while the advanced programmer, who is not familiar with Applesoft, should look at the coding rules. The following sections introduce each BASIC (and DOS) instruction. There seems to be no specific order in introducing the instructions except that the easier instructions tend to be at the beginning. There are, though, one or two anomalies such as dealing with DOS commands before even starting with BASIC and introducing ON...GOTO and ON...GOSUB before IF...THEN. It would serve no useful purpose to list the contents of each section, suffice it to say that most of the Applesoft instructions are dealt with in some detail.

The second part of the book covers using disk files. It is split into six sections. Section one, called "Information Storage on Disks", introduces the basic structure of the Apple disk and such concepts as a 'track' and 'sector'. A diagram of a disk neatly shows the main parts of the disk including the position of the catalog and VTOC track. Although there is no mention in the text,

DOS 3.3 is assumed and those readers with DOS 3.2 will find that some of the information given will be incorrect although it will not affect the example programs given. A small program which creates a text file is fully explained.

Section two is called "Introduction to DOS disk instructions" and, not surprisingly, covers the commands available from DOS 3.3. All the commands except INIT are covered in quite some detail, including the use of ONERR GOTO to trap DOS errors. This is unfortunate since on the first page the author says that a disk is available with all the example programs used in the book (good idea, especially as many of the later programs are quite long) but, since the disk is full, it has to be copied to another disk which should be initialised before running FID. INIT is covered in the first part of the book but it would have been nice if the author at least referenced the user to this section without just saying "make sure to initialise your disk before running FID".

Section three is called "The GET Subroutine"; most of this section deals in great detail with a useful BASIC and machine code subroutine which allows the programmer to circumvent some of the disadvantages and problems of using the INPUT statement. This is not only a useful utility but also very educational. The author advises the user to use this very comprehensive routine although it is compact and not very readable. The rest of section three goes into some length addressing topics such as string storage and others related to the function of the GET routine.

Section four is called "Serial and Sequential Disk Files". The author lists the programs from the optional program disk accompanying this book which will be used in this section. He also gives details on how to make the disk operate as a turnkey system although there is no explanation of what a turnkey system is and it seems out of place here. He makes a good job of distinguishing between serial and sequential files: most people use sequential when they mean serial. The section continues with a detailed look at the use of serial/sequential files by producing, stage by stage, a directory program storing names, addresses and telephone numbers The program allows data to be inserted, modified and listed and is quite sophisticated. Provided that the reader understands the text, he/she will be brought up to a high level of competence. Section five is called "Random Disk Files" and is of a very similar format to section four. Section six, called "Index Disk Files", uses the same methods but this topic is quite tricky and I feel that the author has handled it well.

Appendix A follows section six. It consists of an ASCII list giving the binary and decimal codes for the restricted ASCII set used on Apple Is. I would have liked to have seen a full set because many old Apples can use it all. I have doubts on the usefulness of the binary codes given.

The book is well presented but slightly hard to read. Program listings appear to have been produced on a daisy wheel printer and I could find no mistakes. The only fault was that GOTOs referenced REMs. The book is very large which enables the author to indulge in detailed descriptions of many facets of computing which smaller books could not hope to achieve. I found that this produced a feeling of being swamped with information. I think the book succeeds in three out of four of its main objectives but that not enough is said about producing well structured and efficient programs. I would have liked to have seen a Glossary even though most of the terms introduced are well explained in the text.

This book is a serious educational text book and I think that it is ideal for schools and colleges where the students are led through the book by their teacher. Outside guidance is necessary since there is little logical order in the first part of the book. I feel that the rank beginner would prefer a more easy going book but the experienced computer programmer not used to Applesoft BASIC will find this book very useful.

Title: Art and Graphics on the Apple II/IIe
Author: William H. Dewitt
Publisher: Wiley Press
Price: £12.25
Paperback : 10" x 6.8" : 128pp.

Reviewed by Ian Sidwell

This book is intended for the non-programmer who is interested in the graphics capabilities of the Apple. It starts from simple BASIC commands such as FOR-NEXT loops and thence onto LORES commands and use and then HIRES commands, including the use of shape-tables. It also has chapters on how to photograph your monitor/TV as well as the use of video-recorders.

The book deals solely with creating graphics

STOP PRESS
NEW MODEM
£115

Peanut

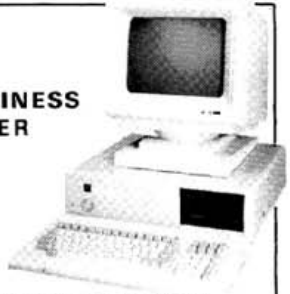
Apple Compatible Peripherals

£50 OFF
'MICROLEDGER' BUSINESS SOFTWARE!

Return a copy of this advertisement with any order over £50 and we will arrange to offer you this new business software system for £145 – a discount of £50 off list price. Spend £50 – get back £50!

Peanut

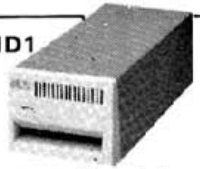
PC2000 BUSINESS COMPUTER



PEANUT are proud to announce their new full specification business computer with these features: twin built-in disk drives: Two microprocessors; 64K ram expandable. Multi-interface output drivers: RGB, PAL, monochrome monitors; parallel printer; RS232: all built-in Languages: Basic, Pascal, CP/M, Forth, Logo etc., available. Upgradeable with all standard Apple peripherals. Full and complete ready-to-go systems with various options of bundled business software. PC2000 . . . from **£795.00.** ASK FOR DETAILS.
STOP PRESS – See PC2000 in London at Textstore Ltd., Twickenham, phone 01-898 9911.
'Anyone familiar with the Apple computer will find learning to use this machine particularly easy'.

MONITORS

HARD DRIVE HD1



10Mbyte capacity unit running all DOS 3.3
CP/M, Pascal programs. Supplied complete with Keele Codes' CLIP software for CP/M file back-up. Full service contract available £1,095.00

DISK DRIVES

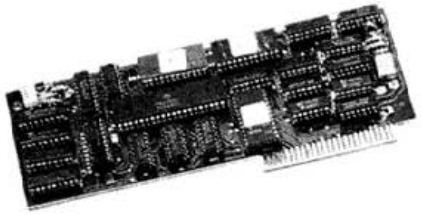
NEW LOW PRICES

AD1 Peanut full height drive, Shugart mechanism, 140K capacity £145.00
PDD2 Peanut half height, direct drive double sided, 280K capacity £149.00
PDD3 NEW Peanut half height, direct drive, double density, single side 320K capacity £175.00
PDD5 NEW half height, direct drive 140K capacity – lowest price anywhere £99.00

PEANUT GOES SOFT!

Top name software at PEANUT prices

PLUG-IN CARDS



80 COLUMN High resolution, 9 x 7 dot matrix fully CP/M DOS, Pascal compatible, II+ version £52.50

80 COLUMN IIe version £49.50
(IIe version with 64K ..... £89.50)

INVERSE CHIP for 80 column card, II+ version, suitable for use with, for example, Magicalc, Format-80, Multiplan, etc £9.50

40/80 column switch ... £12.50

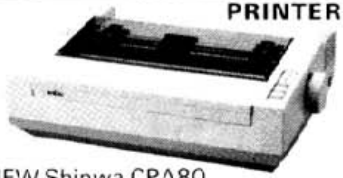
Z80 runs all CP/M £44.50
(ITT version, Z80 ..... £49.50

16K RAM adds 16K to memory so, for example, a 64K machine has 34K for Visicalc £44.50

PARALLEL PRINTER standard

Peanut high resolution monitors 20 MHz bandwidth, on tilt/swivel mount – green or amber screen £89.00

PRINTERS



NEW Shinwa CPA80, 100 cps £199.00
Daisystep 2000, daisywheel, 20 characters/second £239.00
Peanut KGP40, 40 column, two colour, complete with interface card £49.95
NEW 2 speed, near letter quality 65cps fast 165 cps. PEANUT NLQ £295.00

MICROLEDGER the perfect integrated accounts package for the small business, special offer with voucher £145.00
FORMAT-80 ENHANCED the ultimate word processor – prints what you see! £99.00
HILDERBAY Modules: Invoicer, Payroll, Book-keeper, SSP all at £69.00
HILDERBAY ACCESS database, power, speed, versatility, excellent value £69.00
MAGICALC the faster-than-Visicalc spreadsheet with many pluses – no pre-boot required for use with 80 columns and up to 512K RAM. Will read Visicalc files, uses Visicalc compatible commands £59.00

HARDWARE

KEYBOARDS: two types available: V3 Apple style unit giving lower case as an extra with 80 column card or special character ROM in 40 columns £59.50
V5 NEW detached unit. 24 function keys, separate numeric and operator section, independent four way cursor control – 96 keys in all, IBM style, fully Apple II+ compatible £99.00
JOYSTICK £11.95
FAN, UK manufacture £27.00
POWER SUPPLIES: two types: 5A heavy duty £56.50
7A extra HD, with fan £66.50
NEW PEANUT DISKETTES, own-brand. SSDD top quality 5 year guarantee. Packs of 10 (Add £1.00 carriage) £14.95

Centronics interface, recognises all control codes, including cable £38.50

PRINTER/BUFFER CARDS free computer of time-consuming delays – dump text straight into buffer printer takes it out at its own rate. Three capacities available:
16K £79.50
32K £89.50
64K £119.50

DISK DRIVE CONTROLLER 13 or 16 sector disks auto-run £38.50

SERIAL RS232 adjustable baud rate, half duplex, includes 25 way standard connector £42.50

I/O four port, 6522 chips, versatile card £39.50

EPROM WRITER programs popular eprom chips with compare, verify, copy, read etc. functions £54.50

128K RAM, blockbuster extra RAM, boosts Visicalc memory, for example, to 146K £150.00

NEW MODEM Prism 2000 complete with BT approved modem, interface card, cable and disk software. Prestel look out! £115

PEANUT COMPUTER

Unit S, FREEPOST, Dewsbury, WF13 1BR.

Phone: 0924 499366, 24 hrs. answering service. Ext. S.

Money back guarantee. Add 15% VAT to all prices

Carriage – orders under £100 – £3.00 orders over £100 – £5.00. Monitors £7. Printers £7. KGP40 £5. PC2000 £15.

from Applesoft (not machine code). It is full of sample programs and illustrations - a disk may be bought to save the labour of typing all the programs in. It is not an Applesoft tutorial, only referring to the commands needed, and in some cases (e.g. how to construct shape tables) it refers you to the Applesoft Manual. Indeed, a lot of the programs are written in inefficient BASIC, the author being an artist (albeit the President of Computer Art, having taught computer graphics at University of Rochester) rather than a programmer. Also, there are a lot of typographical errors in the programs and programs that do not match their illustrations (there are many illustrations!) - it seems as if there was a rush to get the book out and things were not checked - perhaps it is just a ploy to get the reader involved by altering the programs to do what they are supposed to do.

The various tutorials in fact are less fact-revealing than the Applesoft manual. For instance, it does not really teach you how to design shape tables and it does not tell you about the anomalies of Hires colour (e.g. when plotting white dots they do not appear white - this is mentioned, but the reason why it happens, or how to avoid it is not). It is more a book of applications using the simple tools of Applesoft.

Having noted the bad points of the book, there are quite a few good points. For example, it shows how to access the whole screen in HGR and Lores modes, it shows various tricks that can be integrated into programming (especially for non-mathematicians) e.g. drawing a circle, some of the example programs are really amazing to watch (the author suggests you can put some of the more animated ones to music). The language is definitely easy to understand for non-computerese literates, to the extent of being annoying to computerese literates. It goes through things in a systematic way, whereas the Applesoft manual is rather fragmented. However, by far the most useful part of it is how to photograph the screen, including suggested exposures for various 35mm film. It also has a video section.

In summary, this book is aimed at artists who do not necessarily want to learn about computers. For anyone else I would think twice, as it is not intended to teach BASIC - it gives ideas of what to do with the least number of commands (but not necessarily program steps) - it is a book of ideas. It is riddled with errors unfortunately (including a few factual ones). It is a very simple book. It requires colour to impress.

Title: Computer Playground
Author: M. J. Winter
Publisher: Prentice/Hall (Reston)
Price: £9.65
Paperback : 11" x 8" : 128 pp.

Reviewed by Elizabeth Raikes

It is a book that teaches you programming by giving you programming problems to solve. It has good pictures to colour. The problems are easy to do and interesting. It is easy to read and fun to look at. It is meant for primary children but my four-year-old sister likes doing it too. We like making "wallpaper" at the moment. I think other children would like this too.

Title: The Compleat Apple CP/M
Author: Steven Frankel
Price: £12.55
Publisher: Prentice Hall (Reston)
Paperback : 9" x 7" : 233 pages

Reviewed by Patrick Bermingham

Most people who decide to buy a Z-80 card for their Apple II do so in order to be able to use the more sophisticated software that opperates under CP/M: wordprocessors such as Wordstar, Perfect Writer and Spellbound; database programs such as dBase II and Condor; and spreadsheets such as Perfect Calc and Microplan.

The first problem they will have to overcome is which Z-80 card to buy. Then follows the task of "installing" CP/M so that it will work with the user's system; Apple computer, monitor, printer and the inevitable cards. Finally, each major piece of software will also have to be "installed", to make it compatible with the computer system.

Clearly, when entering the world of CP/M on the Apple, the most pressing need is going to be Good Advice and Help. Buying from a dealer who has the necessary knowledge (and not all dealers have) and who gives a good back-up service is always a wise decision, even though the dealer may make one pay the full retail price! Help and Advice from a dealer tends to decline in proportion to the amount the dealer "knocks off" the normal price of an item.

The next best thing to a good dealer, or a knowledgable friend, is a good book. Micro-computing is notorious for the terrible literature that has been turned out in the name of Information. I still recall with a form of horror the Epson Manual I searched through for information a couple of years

ago. Well, that preamble leads me to the excellent book by Steven Frankel.

He has divided the book into two parts. The first part deals with the CP/M operating system as such. He explains why CP/M is so popular and then gives a guide to the main CP/M cards available for the Apple II. The difficulties involved in assembling a computer system are outlined. He then gives a very clear and readable explanation of the major CP/M commands. The treatment of CP/M is limited to the most important aspects of the opperating system that the first-time user is likely to need. For more advanced features of CP/M the reader is advised to consult one or more books from a recommended list.

The second part of the book is a very useful review of over 40 programs, classified as Word Processors, Spelling and Word Use Checkers, Communication Programs, Spreadsheet and Statistical Analysis Programs, Accounting-Business Analysis and Tax Preparation Programs and File and Data Base Management Programs.

The Compleat Apple CP/M is well written in a friendly style and I can strongly recommend it to anyone who is considering using CP/M, or who has been using CP/M but who has never been able to make much sense out of manuals.

Title: Intermediate Apple
Author: Bill Parker
Price: £14.50
Publisher: Reston Publishing Co. Ltd.
Paperback : 8.25 x 5.25 : 221 pp.

Reviewed by G. H. Ashdowne

This book leaves you with a lot of decisions to make. And so it should. It is designed for the programmer who has reached an intermediate standard of success, and who needs that little push into the realm of machine language programming, with the resulting enhancement of his computing ability.

But first things first. Page flipping and index poking does not give a favourable impression. The Applesoft programs using cryptograms to define strings may be easy reading for someone who has adopted that style of descriptive programming, but for someone who already follows the advice given in the book of using the two operative letters to define the string, it gives rise to a suspicion that the device is used for page spreading. At the end of each chapter, a section for 'further reading' gives a list of books which are repeated chapter after chapter. Reference to different sections in those books would help dispel that suspicion of padding. Likewise the comic pictures squeezed into the text look as clumsy as the cover design and do not add to the brilliance of the text or the incentive to buy. Under these conditions it is easy to realise that it is a difficult book to get into.

So let us start, and discover within the first few pages that we are spaghetti programmers, working by the seat of our pants. Blunt and maybe truly factual, but who wants to be brought down to that size so quickly. Maybe this is the correct technique to get me motivated because I did not return this book to Yvette with a polite suggestion that somebody else have a go. No! Structured programming got at me and Flow Diagrams, as distinct from flowcharts, caused me to swallow the bait to further reading. This ***** book became my travelling companion, my bedside mate, my everything. Wifely comment was not a friendly 'What are you reading, dear?' but a resigned 'Still at it, then?'. After the last issue of Hardcore, I realised that Intermediate Programmers do it book in hand.

Many of my suspicions about programming were confirmed. Getting things into the computer and out again is not that difficult as long as you know the tricks. It is screen presentation which takes the time, and even this becomes a lot easier if you know how to control the cursor and modify the text window.

Speed has not been my concern because running a business on computer does require a lot of slow keying in and interpolation of data. However, a little bit of structuring and a little bit of machine language, which almost demands the inclusion of 'wait loops' in order to see what is happening, suddenly converts the program from an extremely effective calculator into a work of art. Oh joy, what can I do next? Yes! This is the book for the intermediate programmer provided he does not intend to remain at the intermediate level. It gets you nowhere unless the intention is to go on a lot further.

So what do you get for your money, apart from the desire to spend even more - first, five very good chapters on Applesoft improvement. The 'think first, design next and compose last' concept is clearly illustrated, covering all aspects of efficient computer use. RWTS in Applesoft,

many algorithm examples and ten file handling techniques, even if not new, look much easier when structured. Then there are three chapters on graphics, printers and tricks with peeks and pokes which are not overoriginal, a chapter on assembler magic which is limited to Merlin and two more chapters which will whet your appetite to further reading. A slight question will remain with you, that this padded out book, with a lot which could have been said more easily, gives you a feeling that you have been 'sales washed' into the author's, or his employer's, products.

When all is said and done, you will own a very good Ampersand Applesoft enhancing program which does help in structuring your own creations, as well as Bill Parker's Read Line technique for avoiding 'Extra Ignored' inputs. I hope you receive this book as a gift and it will GOTO making your Christmas a very happy one. There I have done it again; a GOTO is the hallmark of a spaghetti programmer.

Title:- The Epson Connection: Apple.
Authors:- W. H. Darnall and D. B. Corner.
Publisher:- Reston Publishing Company.

Reviewed by Norah Arnold

This book is the first in a series of printer and application books to be released over the forthcoming year by Epson in association with Reston Publishing Company. It is aimed at Apple owners who use an Epson MX, FX, RX or Comrex Comriter printer.

The book is divided into three major sections: tutorial, reference and applications, followed by several appendices. The tutorial section covers Epson and Comrex dot matrix and daisywheel printers, and their uses in business and at home. Particular features available on the printers are described in some detail. The need for a printer controller card is explained and the most popular cards are reviewed. Help is also given on the type of cable to get, and how to have one made, if the occasion arises.

The reference section begins with an overview of operating systems and languages. This covers Apple DOS, CP/M and Apple Pascal p-system operating systems and the languages normally associated with them. What I consider to be a vital point is brought out strongly, i.e. that it is essential to understand your operating system if you wish to make the hardware and software work together to perform tasks efficiently.

The text printing features of the printers are dealt with in detail, with many short Applesoft programs given to demonstrate specific points. Some of these examples are very similar to those given in the relevant manuals, but whereas the manuals only deal with the positive features, this book also speaks of 'double-strike creep', 'bidirectional skewing' and how to counteract these problems. The graphics features of the printers are also dealt with pretty thoroughly, starting with simple things such as making designs with asterisks and continuing with overstrikes, bit image graphics and user defined character sets. This section would possibly deter anyone from doing bit image graphics on the Comriter II or III. Apparently, on the daisywheel the period key is used to make the dots, and when you realise that an Apple screen dump could have 20,000 dots then a few screen dumps could cause excessive wear of that key. That is, of course, assuming that you were prepared to wait the three hours or so which the authors estimate each dump would take at up to 23 periods per second.

The application section begins with the most popular word processors, WordStar and Applewriter II. The WordStar section covers universal printer control codes, installing WordStar for your printer, safe and forbidden WordStar control codes and how to get your text file printed even if it contains forbidden codes. The Applewriter II section is fairly thorough and it would have saved me some time if I could have had it to hand when I first started using my Epson FX-80. It deals with built in print controls, the Control V (Insert) Mode, glossary files etc.

Printing from VisiCalc comes next, giving guidance on entering VisiCalc printer setup strings and printing spreadsheets to disk. Business applications of many kinds are dealt with, for example, printing on mailing labels and envelopes, custom forms and printing graphs and charts. The last few chapters of the book cover the use of printers in scientific and artistic applications, their use for hobbyist and professional writing and how to modify screen dump features to suit your requirements.

This is essentially a practical book, packed full of information. If you are just starting to use an Epson printer or are having problems getting the results you require then it would certainly be of use to you.

Epson Pages

USER DOWNLOADED AND SCALED CHARACTERS FOR EPSONS

by Ray Harris

Like most of its users I have found that the Epson MX-80 is an excellent printer, but I have no doubt that it is enhanced usefully in the FX-80 by the ability to use alternative character sets downloaded into it by the user. It was with this facility in mind that I set about developing the program that follows. While I was about it, it seemed worthwhile to add the ability to print new character sets at a chosen scale; why not make the horizontal and vertical scales independent? This would provide a new facility for all printers using Epson protocols.

I developed the program without being particularly careful how much space I used, and although there is room for reduction, especially where the length of the character set is required three times, it is convenient and runs fast enough for the printer not to have to wait. The source code is provided and includes instructions for use in its introduction. I have added an unsophisticated BASIC program (USER OUTPUT.A) for those readers who prefer to use it in that way; this could be improved by altering the message input routine to accept any input, using one of the published routines (for example, 'Windfall' vol. 2, no. 12, p. 30). Alternatively one of the following approaches may be useful:

(i) the buffer is at the usual address for an 'Applewriter' file. If the format of an A-W file can be changed to standard ASCII, and the closing byte changed to $00, it may be printed directly.

(ii) the instruction LTXT (Call-A.P.P.L.E. Sept. 1982 and in 'All About DOS') can be used to read a text file into memory starting at the appropriate point.

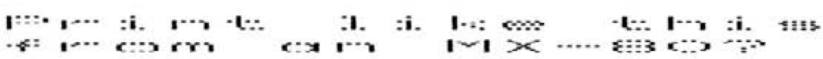
(iii) I have adapted the program 'Go-between' by Ian Trackman, so that 'Scale Output' can be used in conjunction with 'Applewriter' to change typeface in the middle of a line. I imagine the same could be done with the printer section of the original 'Applewriter' and probably with other word processing programs as well.

The program has the minimal text-writing facility of not stopping a line in the middle of a word (a facility similar to that suggested by Tony Game in the August 'Hardcore'). For pages narrower than 80 columns a byte ($9305) can be altered where the source notes "max. width". For wider pages than this the same byte should be altered, but for 128 columns or more the 'BPL PRNTBUF' in line 89 should be altered to 'JMP PRNTBUF'. The section labelled PRINT (lines 231-234) is the standard way to pass all eight bits through the Epson card to the printer. This may have to be adapted, although it works with some other interface cards.

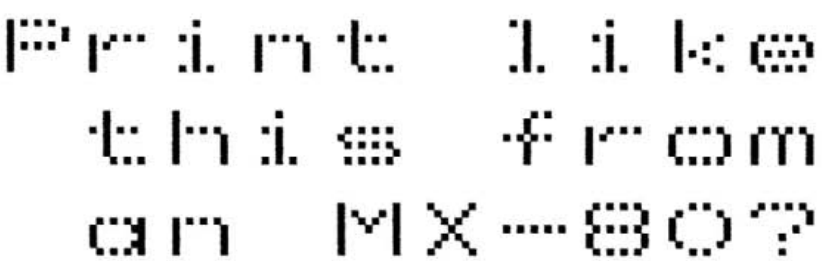
In essence the program first reads the next character in the buffer and works out how far through the character set it has to search (by subtracting $A0. If you wish to ignore numeric and punctuation characters and write a shorter character set the number of the new starting character must replace $A0 at addresses $9337, $9381 and $9389). By multiplying by 12 it finds the start of the required character and outputs the next 12 bytes. After a 'return' character is reached the pointer to the start of the buffer is reset to the following character.

Examples of the output are shown; all of these are scaled versions of an italic character set, for which the monitor listing is given. Note that descenders of lower case letters only extend one dot below the printing line as seven dots are used above. To produce the twelve bytes of code for one character it should be drawn on a grid, which can also be produced on the printer. The BASIC program 'GRID' produces a layout close in shape to the grid of the final character. The bytes to be sent to the printer in its 'bit-image graphics' mode are each one vertical column of the character, corresponding alternately to a vertical line of spaces and a vertical line of dots in the grid. The bytes are calculated in standard fashion, the top four dot positions for the high nibble and the bottom four for the low nibble of the required byte. The lower case 'a' is shown with its corresponding bytes. Twelve bytes are allowed for each character, using the double density graphics, although to provide reasonable spacing the last three of each character are 00, in the set listed. A set for a joined script would probably use all twelve bytes. The program could obviously be adapted to recognize characters with varying numbers of bytes to produce proportional spacing, but more calculation would be required within the program, both to work out the starting position of each

Print like this from an MX-80?



Print like this from an MX-80?



Print like this from an MX-80?

Print like this from an MX-80?

Print like this from an MX-80?

Print like this from an MX-80?

Print like this from an MX-80?

character and the total number of bytes needed for a line. No doubt more than one character set could be made available in memory at the same time: the samples illustrated also show text in a type similar to the Epson's ROM set.

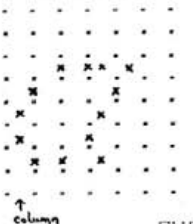
For those who wish to enter the program using the hex data rather than the source code and an assembler, the two bytes at $9469-946A can be entered as 00 00. The bytes after $94E8 need not be entered as the program puts its own information there. Finally BSAVE SCALE OUTPUT,A$9300,L$23F. The character set can be typed in starting at any convenient address. Using $2000 as suggested, complete the process by BSAVE CHARACTER SET,A$2000,L$474 (a character for $FF is not included).

With small scale factors (1 or 2 in each direction, say) the effect of leaving some empty columns (00) within character definitions is acceptable, and makes the appearance similar in tone to the printer's standard character set. With larger sizes the characters may appear less attractive at close range, but they seem very clear at a distance. Why not try important notices at an enormous scale factor of 80 in each direction?

```
1  REM  *** USER OUTPUT.A ***
2  REM  ***   RAY HARRIS  ***
3 :
10  PRINT  CHR$ (4)"BLOAD SCALE OUTPUT"
20  PRINT  CHR$ (4)"BLOAD CHARACTER SET,A$9E00"
30  HIMEM: 33791
40  TEXT : HOME
50  INPUT "HORIZONTAL SCALE FACTOR? ";HS
60  PRINT : INPUT "VERTICAL SCALE FACTOR? ";VS
70  POKE 252,VS: POKE 238,HS
100  HOME :BUF = 4096 + 9 * 256
110  PRINT "WHAT IS YOUR MESSAGE? (NO WORDS LONGER
     THAN " INT (80 / HS)" LETTERS) ";
111  INPUT I$
120  IF I$ = "END" THEN  END
130  FOR N = 1 TO  LEN (I$)
140 I =  ASC ( MID$ (I$,N,1)):I = I + 128
150  IF I = 151 THEN LC = 1 - LC: GOTO 190
151  REM   CTRL-W DURING INPUT TOGGLES LOWER CASE FLAG
155  IF I = 146 THEN I = 141
156  REM CTRL-R DURING INPUT IS CONVERTED TO 'RETURN'
160  IF LC AND I > 191 THEN I = I + 32
170  POKE BUF + X,I
180 X = X + 1
190  NEXT N
210  POKE BUF + X,0
220  CALL 37632
230 X = 0: GOTO 110
240  REM  RUN 30 AVOIDS RELOADS
```

hex bytes for lower case 'a'
(found at $230C in the dump)
0C 12 00 22 00 24 2A 10 20
00 00 00

lower case 'a'




```
1 REM ***    GRID    ***
2 REM *** RAY HARRIS ***
3 :
10 D$ =  CHR$ (4):L$ = ". . . . . ."
20  PRINT D$"PREI"
25  PRINT  CHR$ (0)
40  FOR I = 1 TO 9: PRINT L$: NEXT
50  PRINT D$"PREO"
```

CHARACTER SET

```
2000- 00 00 00 00 00 00 00 00
2008- 00 00 00 00 00 02 00 10
2010- 20 40 80 00 00 00 00 00
2018- 00 20 40 80 00 20 40 80
2020- 00 00 00 00 2A 04 38 40
2028- AA 04 38 40 A8 00 00 00
2030- 00 24 12 4C 30 44 90 48
2038- 00 00 00 00 00 42 84 48
2040- 92 24 42 84 00 00 00 00
2048- 0C 00 52 80 32 88 44 0A
2050- 10 00 00 00 00 00 20 80
2058- 40 80 00 00 00 00 00 00
2060- 00 00 18 24 42 00 80 00
2068- 00 00 00 00 00 00 02 00
2070- 84 48 30 00 00 00 00 00
2078- 10 04 18 40 38 04 30 40
2080- 10 00 00 00 10 00 14 08
2088- 10 20 50 00 10 00 00 00
2090- 00 00 01 04 02 04 00 00
2098- 00 00 00 00 10 00 10 00
20A0- 10 00 10 00 00 00 00 00
20A8- 00 00 02 04 02 04 00 00
20B0- 00 00 00 00 00 02 04 08
20B8- 10 20 40 80 00 00 00 00
20C0- 18 24 42 00 82 00 84 48
20C8- 30 00 00 00 00 02 00 06
20D0- 48 12 60 80 00 00 00 00
20D8- 02 20 46 00 8A 00 92 20
20E0- 40 00 00 00 04 02 80 12
20E8- 80 32 8C 40 80 00 00 00
20F0- 00 08 10 08 22 0C 50 28
20F8- C0 00 00 00 00 04 22 40
2100- A2 00 A2 04 98 00 00 00
2108- 0C 10 22 00 52 00 92 04
2110- 08 00 00 00 82 04 88 10
2118- 80 20 80 40 80 00 00 00
2120- 04 28 42 10 82 10 84 28
2128- 40 00 00 00 00 20 52 00
2130- 94 00 88 10 60 00 00 00
2138- 00 00 24 48 24 48 00 00
2140- 00 00 00 00 00 01 14 22
2148- 14 20 00 00 00 00 00 00
2150- 10 08 00 24 00 42 00 80
2158- 00 00 00 00 08 00 28 00
```

Concise Computer Consultants Ltd.

PC - 2000 BUSINESS SYSTEM

64K RAM / DET. KEYBOARD
TWO INTERNAL DRIVES
PARALLEL PRINTER/RS232 SERIAL INTERFACE
DISK DRIVE CONTROLLER BUILT IN
COOLING FAN/LOUDSPEAKER BUILT IN
12" GREEN SCREEN MONITOR
PLUS FREE DOT MATRIX PRINTER £895.00

APPLE

APPLE IIE	£495.00
APPLE IIC	£925.00
MACINTOSH	£1750.00

DISK DRIVES

H/H SINGLE DRIVE	£ 86.00
H/H DIRECT DRIVE APPLE II COM.	£135.00
H/H CUMANA DIRECT DRIVE	£150.00

FLOPPY DISKS

10 DISKS S/S S/D	£ 9.99
10 DISKS S/S D/D	£ 12.50
10 DISKS D/S D/S	£ 14.00
10 DISKS D/S D/D 80 TRK	£ 17.00
10 DISKS S/S D/D DYSAN	£ 17.00

MODEM

O.E.L. TELEMOD 3 (B.T. APP.)	£129.00

MONITORS

12" GREEN SCREEN ZENITH	£ 69.00
12" GREEN SCREEN DYNEER	£119.00
14" MICROVITEC RGB NOR. RES	£173.00
14" MICROVITEC RGB MED. RES	£288.00
14" MICROVITEC RGB HIGH RES	£416.00

PRINTERS

EPSON RX80F/T (DOT MATRIX)	£225.00
EPSON FX80F/T (DOT MATRIX)	£324.00
EPSON FX100F/T (DOT MATRIX)	£499.00
QUEN-DATA DMP-81(DOT MATRIX)	£135.00
QUEN-DATA DP-100 (DOT MATRIX)	£175.00
QUEN-DATA DAISY WHEEL (20CPS)	£225.00
QUEN-DATA TYPEWRITER/PRINTER	£350.00
DAISY WHEEL	

APPLE CARDS

16K RAM CARD	£ 42.50
128K RAM CARD	£150.00
Z80 CARD	£ 42.50
80 COLUMN CARD IIE	£ 48.00
80 COLUMN CARD II+	£ 49.50
80 COLUMN CARD EXT. 64K	£ 89.50
PARRALLEL PRINTER CARD	£ 38.50
RS 232 CARD	£ 40.00
COMMUNICATION CARD	£ 35.00
R.G.B. CARD	£ 55.00
DISK CONTROLLER CARD	£ 38.00

VISA

1 CARLTON ROAD, SOUTH CROYDON, SY. 01-681-6842 EXC. OF V.A.T.

```
2160- 28 00 28 00 20 00 00 00
2168- 00 02 00 84 00 48 00 20
2170- 10 00 00 00 00 40 02 80
2178- 08 80 10 80 60 00 00 00
2180- 0C 12 20 42 18 82 28 90
2188- 60 00 00 00 06 18 20 48
2190- 80 08 82 4C 30 00 00 00
2198- 02 8C 30 C2 10 82 10 8C
21A0- 60 00 00 00 0C 32 40 82
21A8- 00 82 04 C0 00 00 00 00
21B0- 02 8C 30 C2 00 82 04 98
21B8- 60 00 00 00 06 18 62 90
21C0- 02 90 02 80 00 00 00 00
21C8- 06 18 60 90 00 90 00 80
21D0- 00 00 00 00 0C 32 40 82
21D8- 10 82 14 C8 10 00 00 00
21E0- 06 18 60 90 00 12 0C 30
21E8- C0 00 00 00 00 02 00 86
21F0- 18 E2 00 80 00 00 00 00
21F8- 0C 02 00 02 80 0C B0 40
2200- 80 00 00 00 06 18 60 90
2208- 00 28 06 40 80 00 00 00
2210- 06 18 62 80 02 00 02 00
2218- 00 00 00 00 06 38 C0 20
2220- 18 20 46 38 C0 00 00 00
2228- 06 38 C0 20 10 08 06 38
2230- C0 00 00 00 0C 30 42 00
2238- 82 00 84 18 60 00 00 00
2240- 06 18 60 90 00 90 00 90
2248- 60 00 00 00 00 0C 30 42 00
2250- 82 08 84 1A 60 00 00 00
2258- 06 18 60 90 08 90 04 92
2260- 60 00 00 00 04 62 00 92
2268- 00 92 00 8C 40 00 00 00
2270- 00 00 80 06 98 60 80 00
2278- 80 00 00 00 0E 30 C2 00
2280- 02 00 0C 30 C0 00 00 00
2288- 10 68 84 02 00 04 08 30
2290- C0 00 00 00 06 38 C4 08
2298- 30 08 06 38 C0 00 00 00
22A0- 02 04 88 50 20 10 4E 80
22A8- 00 00 00 00 00 00 C2 24
22B0- 08 10 20 40 80 00 00 00
22B8- 02 04 82 08 92 20 82 40
22C0- 80 00 00 00 06 18 62 80
22C8- 02 80 00 80 00 00 00 00
22D0- 00 00 80 40 30 0C 02 00
22D8- 00 00 00 00 00 02 00 02
22E0- 80 02 8C 30 C0 00 00 00
22E8- 20 00 40 00 80 40 20 00
22F0- 00 00 00 00 01 00 01 00
22F8- 01 00 01 00 01 00 00 00
2300- 00 00 00 E0 00 80 00 00
2308- 00 00 00 00 0C 12 00 22
2310- 00 24 2A 10 20 00 00 00
2318- 02 0C 30 C2 20 22 00 24
2320- 18 00 00 00 0C 10 02 20
2328- 02 20 02 20 00 00 00 00
2330- 0C 12 00 22 00 24 0A 30
2338- C0 00 00 00 0C 12 08 22
2340- 08 22 08 20 10 00 00 00
2348- 00 00 22 0C 30 40 A0 00
2350- 80 00 00 00 01 0C 11 23
2358- 00 26 28 10 20 00 00 00
2360- 06 18 60 90 00 20 02 24
2368- 18 00 00 00 00 00 00 26
2370- 08 32 80 00 00 00 00 00
2378- 00 02 01 00 01 02 24 18
2380- A0 00 00 00 00 06 18 60
2388- 88 04 12 00 20 00 00 00
2390- 00 00 06 98 62 80 00 00
2398- 00 00 00 00 02 24 18 22
23A0- 04 38 02 24 18 00 00 00
23A8- 06 18 20 10 00 20 02 24
23B0- 18 00 00 00 0C 12 20 02
23B8- 20 02 24 18 00 00 00 00
23C0- 01 06 1C 22 00 22 00 24
23C8- 18 00 00 00 0C 12 00 22
23D0- 01 26 29 10 20 00 00 00
23D8- 02 04 38 00 10 20 00 20
23E0- 10 00 00 00 02 00 12 08
23E8- 22 08 24 00 20 00 00 00
23F0- 00 20 0C 32 40 A2 00 24
23F8- 00 00 00 00 00 0C 12 20
2400- 02 00 06 18 22 00 00 00
2408- 00 18 26 00 04 00 08 10
2410- 20 00 00 00 06 38 04 08
2418- 10 0C 02 04 38 00 00 00
2420- 02 00 24 10 08 04 12 00
2428- 20 00 00 00 01 18 25 02
2430- 00 04 08 10 20 00 00 00
2438- 02 00 26 00 2A 00 32 00
2440- 20 00 00 00 00 10 0C 62
2448- 80 02 80 00 00 00 00 00
2450- 02 04 08 00 20 40 80 00
2458- 00 00 00 00 00 00 02 80
2460- 02 8C 60 10 00 00 00 00
2468- 40 80 00 80 40 20 00 20
2470- 40 00 00 00
```

```
1  ********************************
2  *        SCALE OUTPUT          *
3  * TO USE VARIOUS TYPE-FACES    *
4  *         RAY HARRIS           *
5  *          20/09/84            *
6  ********************************
7
8  *Requirements are:
9
10 *a character set (of 12 bytes
    per character)
11 *loaded at $8E00(START-$500),
    starting with character $A0
12
13 *horizontal scale entered at $EE
14 *vertical scale entered at $FC
15
16 *text stored in the buffer
    (starting at $1900)
17 *ending with a byte $00.
18 *
19 *Any non-zero byte less than $A0
    will cause
20 *carriage return and line feed
    unless otherwise trapped.
21
22 *When ready reset HIMEM to
    START-$F01
23 *and CALL START from BASIC or
    'START'G FROM monitor
24
25
26 PRESHFT  =  $E0
   ;number of shifts done before
    this line
27 POSTSHFT =  $E1
   ;number of shifts done after
    this line
28 PRBYTE  =  $E2
   ;the byte to be printed
29 COUNTER =  $E3
   ;counts bits
30 CHAR1   =  $E4
   ;for counting character bytes
    and shifting them
31 CHAR2   =  $E6
   ;for storing 0 or 1 depending on
    carry from last shift
32 CHARLIN =  $E8
   ;max. value of Y for SCALE
33 BUFPNT  =  $E9
   ;points to start of present
    buffer
34 ENDFLAG =  $EB
   ;non-zero on EOF
```

```
35  REST    =   $EC
    ;used for each byte to show
     number of repeats for scale
36  RESTN   =   $ED
    ;stored from 1 block to next
37  *A block is a complete printed
     line of characters
38  HSCALE  =   $EE
    ;horizontal scale
39  LINES   =   $EF
    ;lines left per character
     (from SCALE down to 0)
40  BASE    =   $F9
    ;start of character set
41  YMAX    =   $FB
    ;one less than number of
     characters this line
42  VSCALE  =   $FC
    ;vertical scale
43  CHRSTT  =   $FD
    ;start of present character
     required
44  YSAVE   =   $FF
45  BUFFER  =   $1900
46  START   =   $9300
47  CHRS    =   START-$500
48          DO  0
49  GOSPC   MAC
50          LDA  ]1
51          STA  LINESPC+2
52          JSR  SPCPRT
53          EOM
54          FIN
55          ORG  START
9300: A9 00     56          LDA  #0
9302: 85 E8     57          STA  CHARLIN
9304: A9 50     58          LDA  #$50
                    ;max. width
9306: 38        59          SEC
9307: E5 EE     60  SC      SBC  HSCALE
9309: 30 04     61          BMI  SCEND
930B: E6 E8     62          INC  CHARLIN
930D: 10 F8     63          BPL  SC
930F: C6 E8     64  SCEND   DEC  CHARLIN
9311: A9 00     65          LDA  #<BUFFER
9313: 85 F9     66          STA  BUFPNT
9315: A9 19     67          LDA  #>BUFFER
9317: 85 EA     68          STA  BUFPNT+1
9319: A9 00     69          LDA  #<CHRS
931B: 85 F9     70          STA  BASE
931D: A9 8E     71          LDA  #>CHRS
931F: 85 FA     72          STA  BASE+1
9321: A0 00     73          LDY  #0
9323: 84 E8     74          STY  ENDFLAG
9325: 84 ED     75          STY  RESTN
9327: 84 E0     76          STY  PRESHFT
                77          >>>  GOSPC     E8
                    ;8/72 line spacing for graphics
9329: A9 08     77          LDA  #8
932B: 8D E9 94  77          STA  LINESPC+2
932E: 20 D9 94  77          JSR  SPCPRT
                77          EOM
9331: B1 E9     78  CHRIN   LDA  (BUFPNT),Y
9333: 99 EF 94  79          STA  BUFSTT,Y
9336: C9 A0     80          CMP  #$A0
9338: B0 0C     81          BCS  CHECKY
933A: B1 E9     82          LDA  (BUFPNT),Y
                83  *all characters <A0 are
                     terminators
                84  *except at start of buffer
933C: D0 02     85          BNE  TERMR
                    ;zero to end file
933E: E6 E8     86          INC  ENDFLAG
9340: 20 74 94  87  TERMR   JSR  YMULTTVE
9343: C8        88          INY
9344: 10 25     89          BPL  PRNTBUF
9346: C4 E8     90  CHECKY  CPY  CHARLIN
9348: F0 03     91          BEQ  CHECKY1
934A: C8        92          INY
934B: 10 E4     93          BPL  CHRIN
934D: B1 E9     94  CHECKY1 LDA  (BUFPNT),Y
                    ;OK if this character is a space
934F: C9 A0     95          CMP  #$A0
9351: F0 14     96          BEQ  CHECKY2
9353: C8        97          INY
                    ;OK if next character is space
                     or less
9354: B1 E9     98          LDA  (BUFPNT),Y
9356: 88        99          DEY
9357: C9 A1     100         CMP  #$A1
9359: 90 0C     101         BCC  CHECKY2
935B: 88        102 GOBACK  DEY
                    ;or go back as far as a space
935C: D0 03     103         BNE  GOBACK1
935E: 4C 5E 94  104         JMP  EXIT
9361: B1 E9     105 GOBACK1 LDA  (BUFPNT),Y
9363: C9 A0     106         CMP  #$A0
9365: D0 F4     107         BNE  GOBACK
9367: C8        108 CHECKY2 INY
9368: 20 74 94  109         JSR  YMULTTVE
936B: 98        110 PRNTBUF TYA
                    ;sets start of next buffer
936C: 18        111         CLC
936D: 65 E9     112         ADC  BUFPNT
936F: 85 E9     113         STA  BUFPNT
9371: A9 00     114         LDA  #0
9373: 65 EA     115         ADC  BUFPNT+1
9375: 85 EA     116         STA  BUFPNT+1
9377: A5 FC     117         LDA  VSCALE
9379: 85 EF     118         STA  LINES
937B: A0 00     119 NEXTLIN LDY  #0
937D: B9 EF 94  120 NEXTCHR LDA  BUFSTT,Y
9380: C9 A0     121         CMP  #$A0
                    ;ctrl-char gets through as
                     first of buffer
9382: B0 03     122         BCS  REDUCE
9384: 4C 18 94  123         JMP  ON
9387: 38        124 REDUCE  SEC
9388: E9 A0     125         SBC  #$A0
                    ;find character position in set
938A: 8D E8 94  126         STA  TEMP1
938D: A9 00     127         LDA  #0
938F: 8D EC 94  128         STA  TEMP1+1
9392: 20 AB 94  129         JSR  MULTTVE
9395: 18        130         CLC
9396: A5 F9     131         LDA  BASE
9398: 6D ED 94  132         ADC  TEMP2
939B: 85 FD     133         STA  CHRSTT
939D: 85 E4     134         STA  CHAR1
939F: 85 E6     135         STA  CHAR2
93A1: A5 FA     136         LDA  BASE+1
93A3: 6D EE 94  137         ADC  TEMP2+1
93A6: 85 FE     138         STA  CHRSTT+1
93A8: 38        139         SEC
93A9: E9 05     140         SBC  #5
                    ;approximately $500 bytes used
                     for character set
93AB: 85 E5     141         STA  CHAR1+1
93AD: E9 05     142         SBC  #5
93AF: 85 E7     143         STA  CHAR2+1
93B1: 84 FF     144         STY  YSAVE
93B3: A0 00     145         LDY  #0
                    ;print out bit-image
```

```
93B5: B1 FD     146 NEXT    LDA  (CHRSTT),Y
93B7: A6 E0     147         LDX  PRESHFT
93B9: 86 E1     148         STX  POSTSHFT
93BB: F0 04     149         BEQ  STORE
93BD: 0A        150 RESHIFT ASL
93BE: CA        151         DEX
93BF: D0 FC     152         BNE  RESHIFT
93C1: 91 E4     153 STORE   STA  (CHAR1),Y
93C3: 08        154         PHP
93C4: 68        155         PLA
93C5: 29 01     156         AND  #$01
93C7: 91 E6     157         STA  (CHAR2),Y
93C9: A9 08     158         LDA  #8
                    ;if using 8 pins per character
93CB: 85 E3     159         STA  COUNTER
93CD: A5 ED     160         LDA  RESTN
93CF: 85 EC     161         STA  REST
93D1: D0 0D     162         BNE  NEXTN
93D3: E6 E1     163 NEWSHIFT INC  POSTSHFT
93D5: B1 E4     164         LDA  (CHAR1),Y
93D7: 0A        165         ASL
93D8: 91 E4     166         STA  (CHAR1),Y
93DA: 08        167         PHP
                    ;check for carry
93DB: 68        168         PLA
93DC: 29 01     169         AND  #$01
                    ;mask other flags
93DE: 91 E6     170         STA  (CHAR2),Y
                    ;keep result of carry
93E0: A5 EC     171 NEXTN   LDA  REST
93E2: D0 04     172         BNE  GETBIT
93E4: A5 FC     173         LDA  VSCALE
93E6: 85 EC     174         STA  REST
93E8: B1 E6     175 GETBIT  LDA  (CHAR2),Y
93EA: 4A        176         LSR
93EB: 26 E2     177         ROL  PRBYTE
                    ;PRBYTE must be cleared
                178 ;<8 bits are being printed
                     each time
93ED: C6 EC     179         DEC  REST
93EF: C6 E3     180         DEC  COUNTER
93F1: A5 E3     181         LDA  COUNTER
93F3: F0 06     182         BEQ  ONPRINT
93F5: A5 EC     183         LDA  REST
93F7: D0 EF     184         BNE  GETBIT
93F9: F0 D8     185         BEQ  NEWSHIFT
93FB: A5 E2     186 ONPRINT LDA  PRBYTE
93FD: A6 EE     187         LDX  HSCALE
93FF: 20 68 94  188 OUTBYTE JSR  PRINT
9402: CA        189         DEX
9403: D0 FA     190         BNE  OUTBYTE
9405: C8        191         INY
9406: C0 0C     192         CPY  #$0C
                    ;number of bytes per character
9408: D0 AB     193         BNE  NEXT
940A: A4 FF     194         LDY  YSAVE
940C: C8        195         INY
940D: C4 FB     196         CPY  YMAX
940F: F0 03     197         BEQ  ENDLIN
9411: 4C 7D 93  198         JMP  NEXTCHR
9414: A5 EC     199 ENDLIN  LDA  REST
9416: 85 ED     200         STA  RESTN
9418: C0 00     201 ON      CPY  #0
941A: D0 05     202         BNE  ON1
941C: B9 EF 94  203         LDA  BUFSIT,Y
941F: F0 3D     204         BEQ  EXIT
                    ;saves a line if 00 at start
                     of line
9421: A9 8A     205 ON1     LDA  #$8A
                    ;end of line
9423: 20 6B 94  206         JSR  PRINT
9426: A9 8D     207         LDA  #$8D
9428: 20 6B 94  208         JSR  PRINT
942B: C6 EF     209         DEC  LINES
942D: F0 0A     210         BEQ  ENDBLOC
942F: 20 9B 94  211         JSR  BITSOUT
9432: A5 E1     212         LDA  POSTSHFT
9434: 85 E0     213         STA  PRESHFT
9436: 4C 7B 93  214         JMP  NEXTLIN
                215 ENDBLOC >>> GOSPC    #4
                    ;extra spacing between blocks
9439: A9 04     215         LDA  #4
943B: 8D E9 94  215         STA  LINESPC+2
943E: 20 D9 94  215         JSR  SPCPRT
                215         EOM
9441: A4 FC     216         LDY  VSCALE
9443: A9 8A     217         LDA  #$8A
9445: 20 6B 94  218 CRPRINT JSR  PRINT
9448: 88        219         DEY
9449: D0 FA     220         BNE  CRPRINT
                221         >>> GOSPC    #8
                    ;normal spacing again
944B: A9 08     221         LDA  #8
944D: 8D E9 94  221         STA  LINESPC+2
9450: 20 D9 94  221         JSR  SPCPRT
                221         EOM
9453: A5 EB     222         LDA  ENDFLAG
9455: D0 07     223         BNE  EXIT
9457: A0 00     224         LDY  #0
9459: 84 E0     225         STY  PRESHFT
945B: 4C 31 93  226         JMP  CHRIN
                227 EXIT    >>> GOSPC    #$0C
                    ;reset spacing
945E: A9 0C     227         LDA  #$0C
9460: 8D E9 94  227         STA  LINESPC+2
9463: 20 D9 94  227         JSR  SPCPRT
                227         EOM
9466: 60        228         RTS
9467: 1B 4C     229 ST      HEX  1B4C
                    ;ESC-L
                230         DS   2
946B: 2C C1 C1  231 PRINT   BIT  $C1C1
946E: 30 FB     232         BMI  PRINT
9470: 8D 90 C0  233         STA  $C090
9473: 60        234         RTS
9474: 84 FB     235 YMULTTVE STY YMAX
9476: F0 32     236         BEQ  RETURN
9478: 18        237         CLC
9479: A6 EE     238         LDX  HSCALE
947B: A5 FB     239         LDA  YMAX
947D: CA        240 DEC     DEX
947E: F0 04     241         BEQ  MULT
9480: 65 FB     242         ADC  YMAX
9482: 10 F9     243         BPL  DEC
9484: 8D EB 94  244 MULT    STA  TEMP1
9487: A9 00     245         LDA  #0
```

Elite Software Company

93 Eastworth Road
Chertsey, Surrey
KT16 8DX

Telephone: 09328 67839

RAMVIEW

This is Elite's own 80-column card for the Apple //e. It is completely compatible with Apple's own 80-column card. The only differences are the price and the fact that our board can be upgraded to be a 64K/80-column card by simply plugging in 8 chips.

RAMVIEW may also be purchased from Elite Software as an upgraded 64K board.

RAMVIEW is supplied with an instruction manual, is fully tested before despatch and is backed by Elite Software's **full** one year's warranty.

The price of a **RAMVIEW** is **£60** + VAT (£69).

The price of a **64K RAMVIEW** is **£120** + VAT (£138).

Postage and packing is **free!**

Special Offer
see page 15

Dealer enquiries welcomed.

```
9489: BD EC 94  246            STA  TEMP1+1
948C: 20 AB 94  247            JSR  MULTTVE
948F: AD ED 94  248            LDA  TEMP2
9492: 8D 69 94  249            STA  ST+2
9495: AD EE 94  250            LDA  TEMP2+1
9498: 8D 6A 94  251            STA  ST+3
949B: A0 00     252 BITSOUT    LDY  £0
949D: B9 67 94  253 GET        LDA  ST,Y
94A0: 20 6B 94  254            JSR  PRINT
94A3: C8        255            INY
94A4: C0 04     256            CPY  £4
94A6: D0 F5     257            BNE  GET
94A8: A4 FB     258            LDY  YMAX
94AA: 60        259 RETURN     RTS
94AB: 0E EB 94  260 MULTTVE    ASL  TEMP1
                    ;multiplies by twelve
94AE: 2E EC 94  261            ROL  TEMP1+1
94B1: AD EB 94  262            LDA  TEMP1
94B4: 0A        263            ASL
94B5: 8D ED 94  264            STA  TEMP2
94B8: AD EC 94  265            LDA  TEMP1+1
94BB: 2A        266            ROL
94BC: 8D EE 94  267            STA  TEMP2+1
94BF: 18        268            CLC
94C0: AD EB 94  269            LDA  TEMP1
94C3: 6D ED 94  270            ADC  TEMP2
94C6: 8D ED 94  271            STA  TEMP2
94C9: AD EC 94  272            LDA  TEMP1+1
94CC: 6D EE 94  273            ADC  TEMP2+1
94CF: 8D EE 94  274            STA  TEMP2+1
94D2: 0E ED 94  275            ASL  TEMP2
94D5: 2E EE 94  276            ROL  TEMP2+1
94D8: 60        277            RTS
94D9: A2 00     278 SPCPRT     LDX  £0
94DB: BD E7 94  279 SPCPRT1    LDA  LINESPC,X
94DE: F0 06     280            BEQ  OUT
94E0: 20 6B 94  281            JSR  PRINT
94E3: E8        282            INX
94E4: 10 F5     283            BPL  SPCPRT1
94E6: 60        284 OUT        RTS
94E7: 1B 41     285 LINESPC    HEX  1B41
                286            DS   1
94EA: 00        287            HEX  00
                288 TEMP1      DS   2
                289 TEMP2      DS   2
                290 BUFSTT     DS   80
```

New Opcodes for the 65C02

by Dave Miller.

Many readers will have noticed quite a bit of hype in the press about a new version of the trusty 6502 found in Apples and many other computers such as Ataris, BBCs, Orics and Commodore PETs (remember them?).

It is called the 65C02 and manufactured by Rockwell International. Unlike the NMOS-based 6502, the 65C02 is based upon CMOS technology. This has two benefits: reduced power consumption: from 450mW to 20mW, and reduced size. This reduced size allows for additional circuitry to give additional commands.

Fourteen new basic opcodes have been added along with two new addressing modes. Some existing opcodes have been extended to use the new addressing modes.

The fourteen new opcodes are :

MNEMONIC	OPCODE	ADDRESSING MODE	MEANING
BRS	0F-7F	zero page/ relative/ implied	Branch on Bit Reset - branch if bit is reset
BBS	8F-FF	zero page/ relative/ implied	Branch on Bit Set - see above
RMB	07-77	zero page/ implied	Reset Memory Bit - resets a particular bit
RMB	87-F7	zero page/ implied	Set Memory Bit - see above
TRB	14 1C	zero page absolute	Test and Reset Bits - allows bits to be tested & set with the accumulator
TSB	04 0C	zero page absolute	Test and Set Bits - see above
INA (INC A)	1A	implied	INCrement Accumulator
DEA (DEC A)	3A	implied	DECrement Accumulator
STZ	12 1C 64 74 9C 9E	(zero page) absolute zero page zero page,X absolute absolute,X	STore Zero - stores 0 at location
BRA	80	relative	BRAnch - unconditional relative jump
PHX	DA	implied	PusH X index register onto stack

```
PLX  FA     implied     PulL X index
                        register from
                        stack

PHY  5A     implied     PusH Y index
                        register onto
                        stack

PLY  7A     implied     PulL Y index
                        register from
                        stack
```

SMB and RMB are used in the following way:

```
opcodes   mnemonic

87 33     SMB 0,$33   ;set bit 0 of
                        location 33

37 56     RMB 3,$56   ;reset bit 3 of
                        location 56
```

I am not quite sure how BBS, BBR, TRB and TSB are used but I think BBS and BBR have three operands: BBS bitno, location-to-be-tested, branch-offset, e.g.

```
opcodes    mnemonic

2F 02 06   BBS 2,$2,BIT2SET
           ;bit 2 of location 2 set?
```

The two new addressing modes are:

1. zero page indirect or '(zero page)', e.g. LDA (0) : load accumulator with the contents of the address pointed to by location 0 and location 1.

This is the same as:

```
LDX #0
LDA (0,X)
```

or :

```
LDY #0
LDA (0),Y
```

2. absolute indirect indexed by X (only applicable to JMP) or '(absolute),X', e.g. JMP ($1234),X : jump to the value of the two locations starting at the address pointed to by adding the value of X to the value of the operand, so if X contains 2 and locations $1236 and $1237 contain $39 and $75 respectively then

```
JMP ($1234),X
```

will jump to location $7539.

The following opcodes have been extended by the addition of new addressing modes:

MNEMONIC	OPCODE	NEW ADDRESSING MODE
ADC	72	(zero page)
AND	32	(zero page)
BIT	89	immediate
BIT	3C	absolute,X
BIT	34	zero page,X
CMP	D2	(zero page)
EOR	52	(zero page)
JMP	7C	(absolute),X
LDA	B2	(zero page)
ORA	12	(zero page)
SBC	F2	(zero page)
STA	12	(zero page)

These new opcodes breathe new life into what is now an old processor although I would have liked to have seen an ADD instruction (i.e. add without carry).

Certain versions of the 6502 supported many opcodes undocumented by their manufacturers. The 65C02 does not support these undocumented opcodes. All existing assemblers for the Apple do not support these new opcodes but it is thought that the new assembler on the ProDOS Utilities disk will support them.

Four versions of the 65C02 are available: the difference between them is their clock speeds (from one MHz to four MHz). Only the one and the three MHz versions are of interest to Apple users. A one MHz 65C02 (coded R65C02P1) can be bought for about £18 and can replace the existing 6502 in an Apple II and //e. While the three MHz version is used in the Accelerator card.

Local Groups

Central London Group

The inaugural meeting has now been held. Meetings will be on the 1st Thursday of each month in Room 97, County Hall from 6pm to 9pm. Do come along and join us.

****

Other addresses:

Herts Group meets at:
The Old School
1 Branch Road
Parkstreet Village
St. Albans

Hants & Berks Group meets at:
Furs Building
Reading University

Essex Group
1st Wednesday
"Tree Top Hotel"
Epping

Essex Group
Alternating 3rd Wednesdays
Havering College of Technology
Hornchurch
and
ITEC
Harrow

Contact: Abe Savant [illegible]

MID-APPLE, for Birmingham and West Midlands.
Contacts:
Bill Watson on [illegible]
or Mel Golder on [illegible]

Kent Group is peripatetic.
Check with Jim Panks on [illegible]
or with Dougal Hendry on [illegible]

Croydon Group's meeting place:
The Shirley Poppy
Wickham Road
Shirley
Check with Paul.Vernon on [illegible]

Software Review

Super Editor.
by Ian Trackman.
Basug Special Release.
Price £11.50.

Reviewed by Patrick Bermingham.

If anyone is writing programs in Applesoft (or ROM Palsoft on an ITT) without the aid of an Editor then this one by Ian Trackman can be recommended. Three facilities are available: Find, Find String, and Find and Replace.

Find will very quickly list out in full every program line which contains a Basic command, function, variable or string, designated by the user. For example, to find all the lines of a loaded program in which (I) appears, the user simply types &"(I) and presses Return. Almost immediately the lines with (I) in them will be displayed, with the (I)'s highlighted in inverse. To find a string one uses &""string".

Find and Replace is invoked with the command &" followed by the item to be replaced separated by a slash from the item replacing it, e.g. &" COLOR=4/COLOR=6 would cause all lines containing COLOR=4 to be displayed, with the corresponding replacement line with COLOR=6, displayed underneath each original line. The user is given the choice of changing all of the lines or selected lines.

I tested out the program on a rather long Applesoft program that one of my students has been developing for his Masters degree and which I have to assess and mark. The Editor came in useful.

As with most things these days, you get what you pay for. At £11.50 Super Editor and its eight-page manual is reasonably good value for money; but its range of facilities is limited compared with such relatively exotic programs as GPLE. However, GPLE cost four times as much. Unexpectedly, I found Super Editor much faster than GPLE.

Utility

by Graham Ashdowne

Here is my most useful program. It enables me to read all of any seq. file and print out any portion for detailed study.

```
100 PRINT "SECFILEREADER"
110 PRINT "WHICH FILE ? ";: INPUT K$
120 PRINT "WHICH DRIVE  ";: INPUT D
130 PRINT
140 PRINT "HOW MANY RECORDS REQUIRED ?";:
    INPUT T
150 D$ = CHR$ (4)
160 PRINT "HOW MANY ENTRIES/RECORD   ?";:
    INPUT L
170 DIM X$(T,L)
180 PRINT : PRINT "HARD COPY Y/N (PRINTER)
    ?";: INPUT Q$
190 ONERR GOTO 270
200 PRINT D$"OPEN"K$",D"D
210 FOR I = 1 TO T
220 PRINT
230 PRINT "I=";I
240 PRINT D$;"READ";K$
250 FOR J = 1 TO L: INPUT X$(I,J): PRINT ,
    X$(I,J);: NEXT J
260 NEXT I
270 PRINT D$: POKE 216,0
280 PRINT D$;"CLOSE";K$
290 IF Q$ = "Y" THEN 340
300 PRINT : PRINT "COMPLETED - ANY MORE
    Y/N ";: INPUT Q$
310 IF Q$ = "Y" THEN 330
320 END
330 CLEAR : GOTO 110
340 PRINT K$: PRINT
350 PRINT "FULL RANGE Y/N ";: INPUT Q$:
    IF Q$ < > "N" THEN 380
360 PRINT "START .. ? ";: INPUT N1
370 PRINT "END .... ? ";: INPUT T: GOTO 390
380 N1 = 1
390 PR# 1: FOR I = N1 TO T: PRINT "I= ";I
400 FOR J = 1 TO (L - 1): PRINT J"- "
    X$(I,J)"    ";: NEXT J: PRINT J + 1
    "- "X$(I,L): NEXT I
410 PR# 0: GOTO 300
```

Epson HI80 Listings

Listings for the new Epson HI-80

by John Sharp

```
5 REM CIRCLE WITH SINE WAVE
6 REM WITH ROTATION AND REDUCTION
7 REM 42 MAR 84
10 PI = 4 * ATN (1)
15 PR#1
20 M = 100
25 PRINT "MA1255,960": PRINT "OR"
30 N = 3:A = 850:B = 100
40 W = 2 * PI / M
60 FOR PH = 0 TO PI STEP PI / 34
70 A = A - 25
80 FOR J = 0 TO M
90 R = A + B * SIN (N * W * J + PH)
100 X = INT (R * COS (W * J)):Y =
   INT (R * SIN (W * J))
110 IF J = 0 THEN PRINT "MA";X;",";
   Y: GOTO 130
120 PRINT "DA";X;",";Y
130 NEXT
140 NEXT
200 PR#0
```

*****

```
10 TEXT : HOME
20 PI = 4 * ATN (1)
25 INPUT "VALUE OF N ";N:ST = 144
   / INT (N * 4)
26 PR#1
30 PRINT "MA1255,960": PRINT "OR"
40 R1 = 900
45 T = 144 * 12: REM DAMPING
100 W = 2 * PI / 144
105 FOR J = ST TO 6000
110 R = R1 * (EXP ( - J / T))
115 ANG = J * W:ROSE = COS (N * ANG)
120 PH = PH + W * SIN (ANG) / 18
140 X = INT (R * ROSE * COS
   (ANG + PH))
150 Y = INT (R * SIN (ANG + PH)
   * ROSE)
160 PRINT "DA";X;",";Y
170 NEXT J
1000 PR#0
```

Try N = 3,4,5.

*****

```
10 TEXT : HOME
20 PI = 4 * ATN (1):XC = 140:YC = 92
25 INPUT "RATIO AS A,B ";A,B:N =
   A / B
27 INPUT "PHASE ";PH:PH = PI * PH
30 PR#1
35 PRINT "MA1255,960": PRINT "OR"
40 R1 = 900
45 T = 144 * B * 20:QZ = 3 * T
100 W = 2 * PI / 144
105 FOR J = 0 TO T * 2
135 ANG = J * W
137 R = R1 - (R1 * J / QZ)
140 X = INT (R * SIN (ANG * N + PH))
150 Y = INT (R * ( SIN (ANG) + SIN
   (A * ANG)) / 2)
155 IF J = 0 THEN PRINT "MA";X;",";
   Y: GOTO 170
160 PRINT "DA ";X;",";Y
170 NEXT J
180 GET A$
190 PRINT : TEXT : GOTO 20
200 END
```

Try A:B 2:3, 1:3, 2:5.
Try PH 0,0.1 -> 0.8.

*****

```
5 REM TRACTRIX GEOMETRICALLY
   DECREASING
10 M = 100:R = 800
20 F = 0.05
30 PI = 4 * ATN (1)
40 W = 2 * PI / M
50 PR#1 : PRINT "MA 900,960":
   PRINT "OR"
60 XN = 2 * R:YN = 0
100 FOR J = 0 TO M * 3
110 ANG = W * J
120 XP = INT (R * COS (ANG)):
   YP = INT (R * SIN (ANG))
125 IF J = 0 THEN PRINT "MA";XP;
   ",";YP: PRINT "DA";XN;",";YN:
   GOTO 160
130 XN = INT (XO - (XO - XP) * F)
140 YN = INT (YO - (YO - YP) * F)
145 IF Z = 0 GOTO 150
146 PRINT "MA";XP;",";YP
148 PRINT "DA";XN;",";YN: GOTO 160
150 PRINT "MA";XN;",";YN
155 PRINT "DA";XP;",";YP
160 XO = XN:YO = YN
165 Z = (Z = 0)
170 NEXT
180 PRINT "DF"
190 PR#0
```

Exec Tip

by Phil King.

Here is a small EXEC program for modifying the catalog display which I hope will be of use to new-comers like myself. The "old hands" will probably utter those dreaded words "Oh no, not that again", but please don't forget that we "youngsters" require these little snippets of programs to make life a little easier. Just because everyone seems to know, does not mean that everyone does know, so how about everyone dusting off those old diskettes and having a look for some old snippets.

CATALOG MODIFICATION

This routine will stop the CATALOG listing at its pause point if RETURN is pressed and continue on as normal if any other key is pressed.

Enter and RUN the following program in APPLESOFT to create the EXEC file CATALOG MOD. Then just EXEC CATALOG MOD to install the routine. (48K System. DOS located at $9600-$C000. If in doubt boot on an Apple Master Disk).

If disks are INITIALISED with the modification installed then they will also contain the modification.

```
10  D$ = CHR$ (4): REM CTRL-D
20  PRINT D$"OPEN CATALOG MOD"
30  PRINT D$"WRITE CATALOG MOD"
40  PRINT "CALL-151"     :REM ENTER MONITOR
50  PRINT "AE39:4C DF BC"    :REM JMP $BCDF
60  PRINT "BCDF:20 0C FD"    :REM JSR $FD0C
70  PRINT "BCE2:C9 8D"       :REM CMP #$8D
80  PRINT "BCE4:D0 03"       :REM BNE $BCE9
90  PRINT "BCE6:4C 2C AE"    :REM JMP $AE2C
100 PRINT "BCE9:4C 3C AE"    :REM JMP AE3C
110 PRINT "3D0G"      :REM EXIT TO LANGUAGE
120 PRINT D$"CLOSE CATALOG MOD"
```

Ribbons

Re-Inking An Exhausted Printer Ribbon.

By Roger Harris.

Why do fabric printer ribbons run out of ink so quickly? The black of a new ribbon soon turns to grey. The cost of a new ribbon cartridge is anything between £4 and £6. That seems a lot of money for a fast deteriorating product which is satisfactory only at the beginning of its working life.

I've been experimenting with ways of applying a liquid ink to an exhausted ribbon and it's really quite simple, clean and cheap too! First of all, the ink. Self-inking rubber stamps use a special NON-water based ink. I don't know the formula, but it does not evaporate from the exposed part of the ribbon. It is called "Faymus Fipi Ink" and is available from business stationers.

To apply the ink, mount a hand drill in a vice so that the axis of the chuck is horizontal. Insert the ribbon winding knob into the chuck and tighten lightly. The opposite end of the cartridge should rest on the work table with the ribbon uppermost. Wind the ribbon by turning the crank until you reach the join. Then, trail a thin line of ink onto the ribbon whilst turning the crank until you get back to the join. This takes a couple of minutes. Allow the ink to soak in. Then, print black, blacker, blackest!

Only use water-based endorsing ink as a last resort. It evaporates from the exposed ribbon and it is more trouble than it's worth to frequently re-ink the exposed part.

I have read elsewhere that the use of "non-standard" inks will result in rust, excessive wear and clogging of the print needles of dot-matrix heads. Perhaps it is true. I am still using the original ribbon and print head which were supplied with my Epson MX-70 in mid 1981.

Readers' Letters

Nairobi, Kenya.

Appleworks Tip.

There is a problem with Appleworks Spreadsheet program using a Microline 83A printer. If the Platen Width in the Printer Specifications is not correctly set to a maximum of 13.2" (as per the manual) and if the Platen Width is not set to accommodate the correct number of characters for the character-size chosen for the document (again, recommended practice according to the manual), the entire system may crash, with only one line of monitor code being printed, requiring a re-boot, perhaps losing the document if it wasn't saved.

For the Microline 83A, Orange interface, 17cpi, it has been found that the Platen Width should be set to 7.8 inches,

accommodating 132 characters.

Yours faithfully,

David F Wilson.

St. Albans, Herts.

Data-Pointer Restorer.

Dear Editor,

The routine below, employing the & command can be used to reset the Data-Pointer to any line in an Applesoft program. The routine is quite short and uses several existing routines in ROM.

To use it you set the & jump vector to the starting address of the routine, e.g. POKE 1013, LO-BYTE : POKE 1014, HI-BYTE.

POKE 1013,0 : POKE 1014,3 sets the jump vector to 768 (dec), $0300 (Hex). The routine is fully relocatable.

SYNTAX.

```
&              Gives an error; no place to go.
& 1000         Restores to line 1000.
& N            Restores to line number in N.
& I% + 1       Restores to value of function.
& "HELLO" Gives an error; no linenumber.
```

LISTING.

```
0300 : 20 67 DD              JSR $DD67
0303 : 20 52 E7              JSR $E752
0306 : 20 95 D9              JSR $D995
0309 : 20 1A D6              JSR $D61A
030C : A5 9B                 LDA $9B
030E : 85 7D                 STA $7D
0310 : A5 9C                 LDA $9C
0312 : 85 7E                 STA $7E
0314 : C6 7D                 DEC $7D
0316 : C9 FF                 CMP#$FF
0318 : D0 02                 BNE EXIT
031A : C6 7E                 DEC $7E
031C : 60           EXIT     RTS
```

Each time the function is used, a 'RESTORE' command must be issued beforehand.

Yours sincerely,

Jason W. Smith.

Croydon, Surrey.

Dear Sir,

People may be interested to know that there is an undocumented feature (i.e. a bug) in the Apple //c. If a Basic program is written as, for example:-

```
10 PR#1
20 FOR I = 1 TO 10
30 PRINT  "ABC"
40 NEXT
50 PR#0
60 END
```

This will produce an output on the printer of not just ABC 10 times but

```
#10 #20 ABC
#20 #30 ABC
#20 #30 ABC  etc. (i.e. the TRACE as well).
```

This is apparently due to a conflict between Applesoft and the Disk Operating System. It is necessary to output to the printer with a command of the form:-

```
10 PRINT CHR$(4) ; "PR#1"
```

for the program to behave as one might reasonably expect.

Incidentally, this feature might apply to an Apple II+ or //e running proDOS.

Yours sincerely,

Quintin Gardner.

Queensland, Australia.

Dear Sir/Madam,

I am a qualified Occupational Therapist currently undertaking tertiary studies in Engineering. My employer, The Queensland Spastic Welfare League, has steadily increased its use of computing and other technologies over a period of eight to ten years.

Our particular interest in the therapy related areas are:

1) as a biofeedback mechanism to enhance muscular control, especially for communication, posture, mobility training and analysis, upper limb and hand, oromusculature co-ordination and monitoring physiological changes, e.g. EMG, EKG.

2) as a medium for cognitive rehabiliation, e.g. visual perceptual skills.

3) in recreation, education, literacy,

numeracy, social and community skills and vocational areas appropriate for cerebral palisied people. In this area we are particularly interested in input on interactive interfacing methods, e.g. disabled persons accessing computers and the hard- and software modifications which are useful and viable.

4) in administrative duties within therapy departments such as writing reports, reecording information on test and treatment results and equipment and stock registers.

5) as an assessment/treatment tool for speech and language dysfunction. In particular, software available for assessment and treatment of communication disorders.

We have several years experience in the use of the following equipment, some of it with special software modifications and the usual adaptive keyboards:

PRINTERS

NEC Spinwriter.
GE 3404 Programmable Printer.
Star DP 515 (Dot Matrix).
MX/ and RX/80 Epson.
Qume Printer.

ELECTRIC WHEELCHAIRS

Sibbings.
Carter-tronic.
Gyrochair.
Vessa.
Avion.
Powerglide.
Lewis.
Orthopaedia.
Everest & Jennings.

COMPUTERS

Apple II Plus.
Sorcerer.
Vydec.
Intercolour.
NCR Decision Mate.

In the same period we have used dedicated lines to external mainframes at the University and other bureaux for both development work and running specific applications. These have grown to a level where we are now calling tenders to transfer these separated applications on to our own in-house small mainframe. Concurrently, we are calling tenders to upgrade the word processing equipment which is being operated by disabled people as an employment activity to a series of sixteen bit work stations linked by a local area network to a hard disk and a number of different types of printer.

Apart from your own interests I am sure you will be aware of others who are working in related areas. I would be very grateful if you would print this letter. We are happy to share our experiences with others working in related areas and are also concerned that we do not re-invent any wheels that have been better developed elsewhere!

Thank you for your assistance.

Yours sincerely,

Ross Black.
Therapy Technology Advisor.
The Queensland Spastic Welfare League
P.O. Box 386
Fortitude Valley Q4006
Australia.

Ulverston, Cumbria.

In reply to the letter in the October '84 issue, from Dr. A. Peter Smith, I would like to make the following points:

Adding double sided disk drives to an Apple is very simple if (like the one supplied by Peanut Computer) it is designed to imitate two separate drives. The one I bought from Peanut Computer plugs into both drive connectors on a standard controller and uses one side of the disk as drive one and the other side as drive two - this method requires no modifications to the operating system.

Some drives, including ITT 2020 drives and the Peanut Computer D/S drive, are capable of using forty tracks - 160k per disk side instead of 140k - a feature which can be used by both DOS 3.3 and Pascal with no modifications except to the formatting routines. If time allows, I may write a utility to format disks for 40 track use.

As suggested, Omnis runs under Pascal and can use any disk type which is 'patched' into the I/O system. Omnis will allow a database to be spread over a maximum of four volumes (disks) which must all be available at the same time, i.e. a database with four data disks needs five disk drives. It can however find any record in three seconds

when searching using an index.

An alternative to hard disks may be a "Terradrive". These use a cobalt coated disk (5.25" floppy) and store 1 Mbyte per disk. They come with software for DOS, Pascal and CP/M and as such should work with any program using Pascal (i.e. Omnis) or DOS 3.3 if it uses standard DOS, which most databases don't.

In conclusion, I suggest double sided drives with double the amount of on-line storage, although some programs may not use extra drives. For still larger capacity, Terradrives may be a cost effective solution, though many databases have their own DOS and may not work with these.

I hope these comments are of help and suggest that any reputable dealer will be quite happy to provide advice.

Yours faithfully,

Simon N. Hobson.

Regents Park, London.

Dear Sir,

Dr A. Peter Smith writes in the October 1984 'Hardcore' about difficulties with Databases. In particular speed and capacity are a problem. Our ACCESS database is probably the fastest Apple Database there is. The standard version uses Apple floppies; a version is available for one of the hard disk systems around, but at the original price (£200 rather than the £75 the standard version now costs). Omnis 3 will work with hard disk: we have installed one for a customer who required a purpose-designed system, and he is very happy with the facilities supplied (after several man-days of setting up); it does take over three minutes of file manipulation to produce an invoice.

We don't know for sure that any Apple database will work with a larger capacity of floppy: The trend for now seems to be towards hard disks.

If Dr Smith would like to try Access we can let him have one on three weeks trial if he writes to us. Access is available on 21 day money-back trial to any purchaser (price £75 + VAT).

Mike Salem, Hilderbay Ltd.

Bromham, Bedford.

Dear Editor,

Dr A. Peter Smith asks in the October issue about increased capacity disks for his Apple][Europlus.

Eicon Research of Cambridge produce 1Mb capacity floppy disk drives in high density double sided 5.25 inch and 8 inch d.s./d.d. form and as both single or twin units. We have used these units extensively for both Basic and Pascal and they are also usable for CP/M. Some units are in use with Omnis.

A nice feature of these drives is that both sides of a disk are treated by the drive controller as a single unit so that it is only necessary to address the drive and not drive plus side. They are also much faster in operation than standard drives and can be used in conjunction with standard drives so that the program can be on a standard drive and data on the Eicon drive. The only difficulty we have encountered is that these drives are not compatible with an accelerator card.

A single 5.25 inch 1Mb drive is listed at around £650 which is rather cheaper than the equivalent number of standard Apple drives.

Remember that to use Pascal will require 64K RAM in the machine, to use CP/M will require the Z80 card and that software will also be required if any programming is to be done. The costs of setting up for Pascal or CP/M can be much greater than expected although there are also benefits.

Yours sincerely,

Ronald S. Harrison.

Bebington, Wirral.

Dear Yvette,

In your recent magazine, August '84, you mention something about utilities. I am not sure what kind of response you want, but I seem to have acquired a whole lot of them. I do a lot of programming for other people as well as for my own enjoyment and would not like to do so without utilities to assist Applesoft. For example, a Print Using is essential in my opinion and the best I know came out of a copy of Nibble, allowing both strings and numbers to be printed using the

same mask.

To give you some idea of the utilities I have tried, I own:

1) Ampersoft (very good)
2) Gale (program line editor, very good)
3) Routine Machine
4) & Chart (excellent for graphs, etc.)
5) & Sampleri
6) & Screen

plus assemblers, trace facilities, etc.

I write a lot of machine code routines for speed critical activities, all the commercial ones have drawbacks. For example, Ampersoft is very good - it has the Nibble print using, a machine code sort command and other features, and together with DOS they all fit in the RAM card. What's the drawback? It won't work on a hard disk and I have occasion to use hard disks.

I would be interested in hearing about the most popular utilities and other people's views of them.

Cheers,

Paul Hartley.

Dorking, Surrey.

Dear BASUG,

Thanks for the latest issue of Hardcore (Oct 84). Lots of interest, I am especially fascinated by the Basic program that draws "mountains" that scroll from right to left. Clever!

Can you tell me if there is any advantage in using a program such as GPLE for program entry as against a word processing program (I have Applewriter II). Using Applewriter II gives you all the cursor movement/insert/delete commands that are most necessary and the resulting Text file can be Exec'd into a Basic program which in turn can be saved. It is a little more long-winded to ammend both versions but not excessively so.

This brings me to a point I haven't seen commented on elsewhere, regarding the lack of a syntax checker in Applesoft. When Exec'ing a Text file there is a Control-G "beep" whenever an illegal instruction is encountered. Would it be possible to incorporate this function into the ordinary program entry routine? I do not have the necessary knowledge of the inner workings to experiment with this myself but I would be very interested to know if it is a possibility.

One last tip, I have found it very helpful to dictate machine code programs onto a tape recorder and then play it back to myself when entering the code. This saves all of the twitching of the eyes from page to keyboard to screen to page etc. Just be sure not to dictate too quickly!

Many thanks again for your efforts,

Yours faithfully,

Michael J. Owner.

Geneva, Switzerland.

Dear Sirs,

I am a former owner of an Apple //e that I have recently traded in for an Apple //c. However, since this change I have discovered that there are some problems with the software that I had for the //e that I am now using on the //c. The major problems have been with the programs Visicalc, Applewriter and PFS:File. I am now writing to find out if, to your knowledge, this is a common problem. If that is the case, I would be very grateful if you could inform me of what could be done about this. I am in fairly urgent need of these programs.

I thank you in advance for your help and consideration of my problems.

I remain,

Henrik Kjellqvist.

Stockport, Cheshire.

Dear Sirs,

Macintosh Multiplan.

There is a simple answer to the problem desribed by P. Knight on p. 13 of October Hardcore. The procedure is as follows:

1. Make a copy of your Multiplan disk. This copy will not boot fully but will ask you to put the master disk into the drive. Better still, ask your friendly dealer to make a copy of your Multiplan disk using the copy utility that needs two disk drives. This copy will boot without the need for the master disk and save you paying Microsoft for a copy.

2. Boot up your System disk, display the icons, then eject the disk.

3. Insert the copy of Multiplan then copy the Font Mover from the System disk to the Multiplan disk.

4. Open the Font Mover on Multiplan and copy all the fonts into a font file.

5. Put the System Folder into the wastebasket.

6. Copy the System Folder off the System disk on to the Multiplan disk.

7. Open the Font Mover and delete all the fonts (except those asterisked) from the System File on the left hand side.

8. Move fonts (particularly Seattle 10 and 20) from the font file to the System file.

9. Put the Font File and the Font Mover in the Wastebasket.

Your new copy of Multiplan should now show a Wastebasket instead of Trash and, more important, will operate off the UK keyboard. You can type in the £ sign but if you choose $ format the numbers will have a dollar sign in front. However, you can choose decimal format (the default is two decimal places) and put a £ sign at the top of each column of figures.

Not having two disk drives I cannot say whether this amended version of Multiplan recognises the second disk drive.

Yours truly,

E. G. Wood.

Riyadh, Saudi Arabia.

Dear Sirs,

I'd like to enquire whether you are able to help a few Apple //c users to set up our systems with the Brother HR15 printer. Unfortunately, being stuck in this neck of the woods leaves us rather more self-reliant than we'd care to be, and with a local Apple 'dealer' who is as much use as the proverbial chocolate fireguard. (All our equipment has had to be purchased out of this country due to the local 200 to 300% mark up! The computers were bought in the UK, and our particular printer in the USA).

Although, as can be determined by the fact that you have this letter, we have achieved some measure of success, we are still having trouble with the system. If I try to outline

the problems, it's possible that you may have some experience which could help remove the bugs.

1. When we try to print a letter of more than about one page, the tail end of the text starts to become garbled. (The end of this letter will probably contain a sample of it!) It gives the impression that the printer buffer fills up with some 3k and then gets 'confused' as the computer continues to send more data.

2. Attempts to try and control the printer from the computer, such as 'characters per inch' succeed in changing the characters, but loose the bi-directional printing format, this despite adding in the codes for 'auto backward print'. Is this to be expected?

3. In order to recover the computer when requesting the printout, it is necessary to clear the printer buffer and allow the computer to finish sending. This operation must be repeated if necessary. Is this normal?

4. How do you manage to print directly from the keyboard?

Some friends here, who are involved with computer installations, had a look at our set up. The conclusion reached was that the 'little black box' supplied as the serial to parallel interface is at the heart of the problem. It doesn't appear to be communicating the printer signals back to the //c. It is the opinion that a 'Busy' signal when the buffer is full does not reach the computer to tell it to stop sending, hence the garbage at the end of about 3k.

In fact doubts were expressed as to what could be communicated back to the computer via the 5-pin DIN connector, considering the complexity of the commands that the Brother is configured to send when supplied with the RS-232C Serial interface. (The interface is made by Micro Peripherals Ltd. and called a 'Microface IIc', supplied with the computer by Micro Computer Consultants Ltd., of Manchester. I have also written to this outfit but have not received a response yet).

The software being used is AppleWorks.

If you have any experience of the type of problems described, your assistance would be appreciated.

Yours sincerely,

Allan Crawford.

/Ed. -Samples showed odd characters missing here and there with no apparent pattern to it./

Corbridge, Northumberland.

Dear Sir,

Could you please advise me on the purchase of a lower case chip for my Revision 6 Apple I+. I would prefer a chip that displays true descenders, a point which many of the adverts for such chips don't mention.

Yours faithfully,

David Steward.

/Bob Raikes replies: For pre-Rev. 7 Apple Is, the problem is addressing the full space in the display Eprom, and a small piggy-back board needs to be made. This picks up two address lines from elsewhere so that the full display is available. Such a board used to be available from Microsource (0727 72917); I don't know whether it still is. P & P Micros still list a Videx converter (£24.95). The board is only small and we could probably get one made if a number of members are interested. Let us know.

Most lower case chips get a 1 dot descender by redefining the characters to sit 1 dot higher than normal. This does not usually cause any problems./

hardcore

Three Years of Hardcore - Quick Reference.

Articles and Authors 1982/1983/1984.
Reconstructed by Peter Blair, Tony Williams and Yvette Raikes.

February 1982 Volume 2 Issue 1.

February 1984 Volume 4 Issue 1.

April 1984 Volume 4 Issue 2.

June 1984 Volume 4 Issue 3.

August 1984 Volume 4 Issue 4.

October 1984 Volume 4 Issue 5.

DIARY

December

4th	Herts Group. 8pm. Communications - A look at ASCII Express and Prestel editing in colour on the Apple.
5th	Essex Group. 8pm.
6th	Central London Group. 6pm.
10th	Hants & Berks Group. 7.30pm.
14th	Mid-Apple (Birmingham). 8pm.
17th	Croydon Group - Adventure Games: Philosophy of design. 7pm.
19th	Essex Group. 8pm.

January

1st	Herts Group. 8pm.
2nd	Essex Group. 8pm.
3rd	Central London Group. 6pm.
10th	Mid-Apple (Birmingham). 8pm.
14th	Hants & Berks Group. 7.30pm.
16th	Essex Group. 8pm.
21st	Croydon Group. 7pm.

February

5th	Herts Group. 8pm.
6th	Essex Group. 8pm
7th	Central London Group. 6pm.
8th	Mid-Apple (Birmingham). 8pm
11th	Hants & Berks Group. 7.30pm.
18th	Croydon Group. 7pm.
20th	Essex Group. 8pm.

If you would like your events in the diary, please write in and tell us about them.

Advertisers

Advertising Rates

Full page	£50.00
Half page	£27.50
1/4 page	£17.50
Flysheets	£75.00

Preparation of artwork from £ 5.00

Copy Dates

Date	Edition
January 4th	February
March 1st	April
May 3rd	June
July 5th	August
September 2nd	October

Please send complete camera-ready artwork in monochrome. If the original is in A4, then the typeface must stand photographic reduction to A5. We can undertake minor alterations to copy.

ELITE SOFTWARE COMPANY.

Special Offer see page 15

S-C Macro Assembler: Full macro assembler with 100 page manual. Operates in any II or //e. Source files may be in core or on disc. Object files may be assembled directly to memory, or to disc. Can read or write standard text files. Editor and assembler are co-resident, all Apple DOS and monitor commands are available.
Price £60.00 + VAT.

Appli-Kit: Applesoft utility. **Appli-Kit** is a package designed to aid the writing and debugging of Applesoft Basic programs. All commands are typed in as if they are part of the standard Applesoft set.
Price £25.00 + VAT.

Wildcard: plugs into any slot and copies the program in core onto a standard DOS disc. Copies 48K core resident programs. No parameter list needed. Copies most programs requiring RAM card. Copies made with the **Wildcard** can be copied with standard copy programs. Copies even the bit copiers. Works with any RAM card - plugs into any slot.
Price £80.00 + VAT. (Now works with //e.)

Wild-Word: a 40 column word processor for the II or //e. True SHIFT key operation. Built-in mailing list and merge. Core-resident.
Price £35.00 + VAT.

Format-80: 80 column word processor for II and //e. What you see is what you get. Editing and formatting achieved with single keystrokes. Supports all Apple compatible printers. Mailing list and merge feature. Core-resident.
Price £129.00 + VAT.

Ramview: 80 column board for the //e, upgradeable to 64K.
Price £60.00 + VAT.

All Elite Software Company products are <u>not</u> copy protected.

Elite Software Company, 93 Eastworth Road, Chertsey, Surrey, KT16 8DX. Tel: 09328 67839

P&P MICRO DISTRIBUTORS P&P MICRO DISTRIBUTORS P&P MICRO DISTRIBUTORS

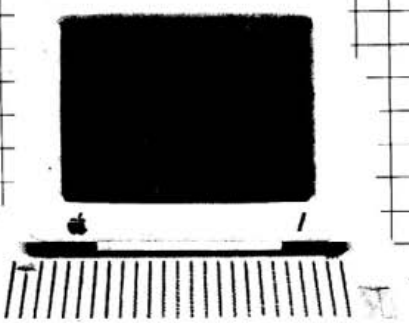
TRADE IN - TRADE UP!

TO AN APPLE IIe, APPLE IIc OR A MACINTOSH.

Apple's Macintosh

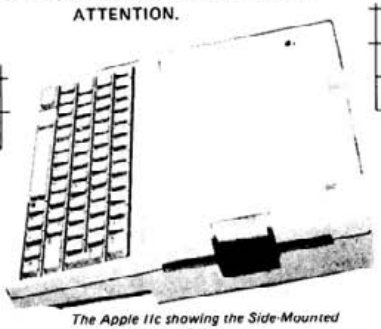
OK, SO YOU LIKE YOUR PRESENT APPLE A LOT. BUT, THERE CAN BE MANY ADVANTAGES IN MOVING UP TO ONE OF APPLE'S NEW, POWERFUL AND PRACTICAL MACHINES – IF ONLY THE PROBLEM OF WHAT TO DO WITH YOUR PRESENT MACHINE CAN BE SOLVED. **WE HAVE THE ANSWER.**

WE NEED SUPPLIES OF APPLE II+ AND IIe WE WILL LOVINGLY CLEAN, POLISH AND RESTORE THEM TO LIKE NEW, BEFORE FINDING THEM NEW HOMES. **SO WE WILL MAKE ALL YOU APPLE OWNERS AN OFFER.**



WE WILL TAKE YOUR TRUSTY AIDE IN PART EXCHANGE FOR A NEW APPLE IIe, APPLE IIc, OR A MACINTOSH. THE TRADE WE CAN GIVE WILL BE DEPENDENT ON THE CONFIGURATION AND AGE OF YOUR PRESENT MACHINE.
INTERESTED? **CALL STEVE McLEAN TODAY ON ROSSENDALE (0706) 217744 FOR IMMEDIATE ATTENTION.**

The Apple IIc with IIc Monitor & Stand.



The Apple IIc showing the Side-Mounted Half-Height Disk Drive.

NEW APPLE PRICES

Apple IIc Computer–£925 External Disk Apple IIc–£230
Monitor Apple IIc–£140 Monitor Stand Apple IIc–£27
Mouse Apple IIc (inc. Mouse Paint)–£70
Carrying Case Apple IIc–£27 Additional Power Supply Apple IIc–£27
Imagewriter Acc. Kit Apple IIc–£38
Appleworks–£175 Logo Apple II 128K–£75
Access Apple II–£57
Apple IIe 64K Computer–£587
Apple IIe 64K Computer (Disk Drive with Controller)–£795
Apple IIe 64K Computer (Duodisk with Acc. Kit IIe, Monitor IIe)–£1095
Macintosh Computer with free Macwrite/Macpaint–£1795
Macintosh External Drive–£349 Macintosh Numeric Keypad–£69
Macintosh Security Kit–£34 Macintosh Carry Case Delux–£69
Macintosh Carry Case Standard–£39
Imagewriter Printer 10"–£385
Imagewriter Printer 15"–£525
I/Writer Acc. Kit Macintosh–£38

Head Office:
NEW HALL HEY RD., ROSSENDALE, LANCS., BB4 6JG Tel: (0706) 217744 Telex: 635740 PETPAM G
1 GLENEAGLE RD., LONDON, SW16 6AY Tel: 01-677 7631/01-769 1022 Telex: 923070 PPCOMP G
DALE ST., BILSTON, WEST MIDLANDS, WV14 7JY Tel: 0902 43913
Norwegian Agent: Programvare Huset, Okernveien 145, N-OSLO 5. Tel: 47 2 64 55 77

Lines open for orders
8 a.m. - 6 p.m. Mon - Fri (Lancs)
9 a.m. - 6 p.m. Mon - Fri (London)
9.30 a.m. - 3 p.m. Sat (both offices)