



A Database Publication

apple user

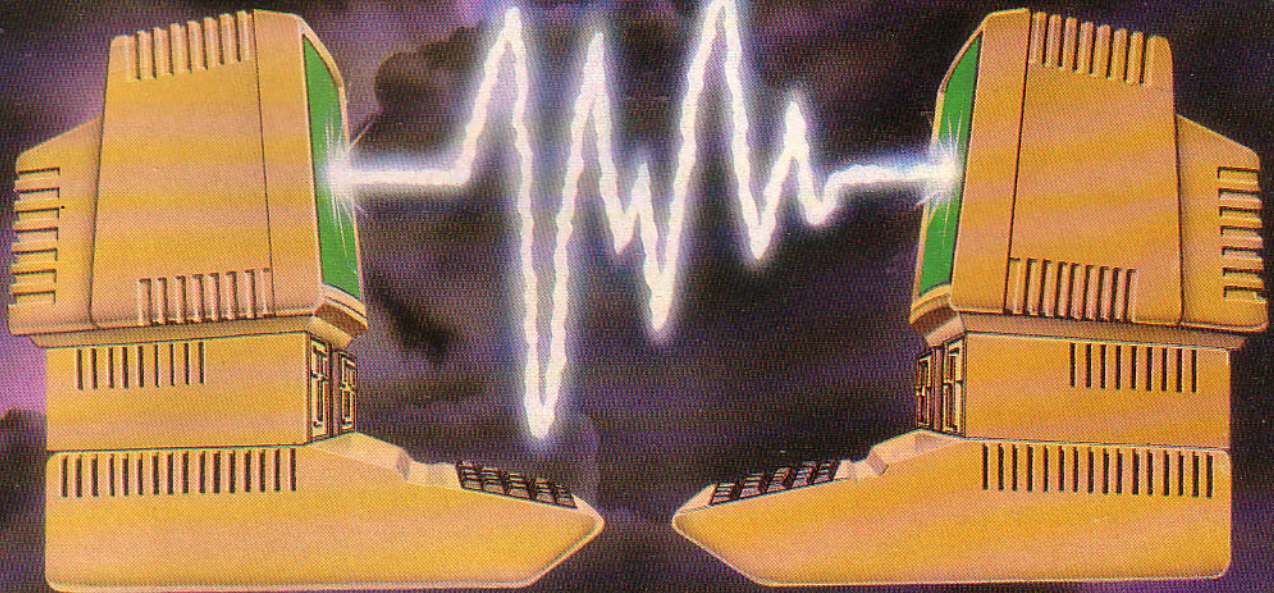
Vol. 5 No. 2

February 1985

£1

**Apple to Apple:
How Qatar police
solved their file
transfer problem**

**EXCLUSIVE: Wozniak
plans a 16 bit Apple II**



**Shape tables: the key to speedy animation
Can an Apple program solve the miners' strike?
Use your spreadsheet to calculate target values**

News

- GEM doesn't sparkle for Apple; after the Big Mac - the giant Mac; Apple blamed for Acorn's US slump; and more. **5**

The Woz

- If you have ever dreamed of a 16 bit Apple II, read what the Wizard of Woz has to say about it. **9**

Survey

- Take time to fill out our survey, and don't forget to send it back to us. You may be one of the lucky 10 to win a year's free subscription. **12**

Quicksort

- It's Forth v Basic in Colin J. Davies' look at the Quicksort method. **17**

Games

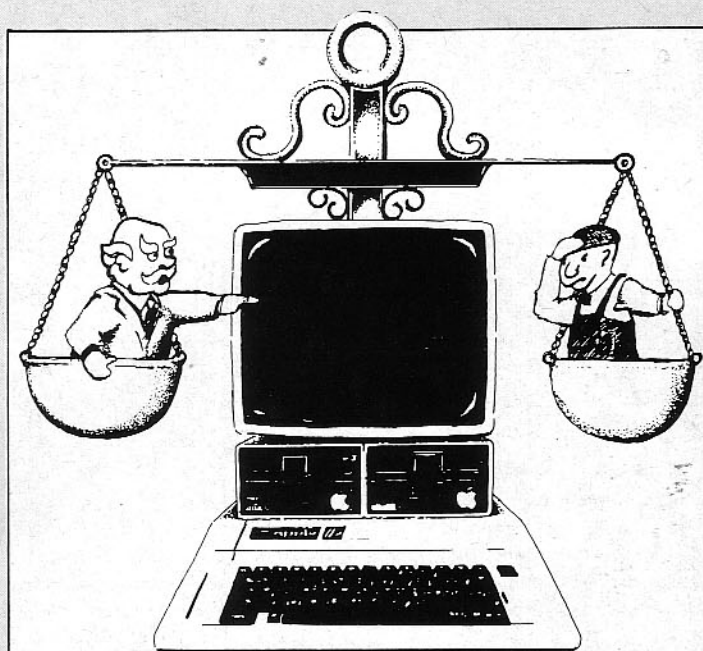
- Adventure games galore with Julian Brewer's Infocom update and Gill Reeve's expedition up the Amazon. **19**

Graphics

- Part XI of Peter Gorry's DIY package takes a first look at shape tables. **23**

apple user

Volume 5
Number 2
February 1985



Conflict resolution?

- Can an Apple II program really resolve the miners' strike? **26**

Spreadsheet

- Patrick O'Beirne tells how to drive your spreadsheet in reverse. **30**

File transfer

- Roy Copping tells how the Qatar police transfer Basic, text and binary files between Apples. **32**

Lisa

- Charles Fairfax and Kathryn Carr describe how complex chemical formulae yield to Lisa's charm. **35**

Macintosh

- Two more of the recent spate of Mac books reviewed. **38**

Review

- Neville Ash tries to psych out the opposition with some Human Edge software. **45**

The 6809

- Cornelis Bongers concludes his series with a look at the OS-9 operating system. **46**

Application

- Two different ways to Apple-assisted book production, as described by Graham Keeler. **51**

Feedback

- Your thoughts on ProDOS, VisiCalc, games software, Apple II series compatibility, and more. **55**

New Products

- From gardening to dentistry, jogging to architectural design, new products to capture your imagination and cash. **61**

Programming

- Give your programs a split personality with P.H.P. Harris's useful routine. **65**

Apple User Managing Editor **Derek Meakin** Features Editor **Cliff McKnight** Technical Editor **Max Parrott**
Production Editor **Peter Glover** Advertisement Manager **John Riding** Sales **Chris Riding**
Design **Heather Sheldrick** News Editor **Mike Cowley** Editor-in-Chief Database Publications **Peter Brameld**

Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising) 061-480 0171 (Subscriptions) Telex: 667664 SHARETG. Prestel: 614568383.

Published by Database Publications Ltd, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

News trade distribution: Europress Sales and Distribution Limited, 11 Brighton Road, Crawley, West Sussex RH10 6AF. Circulation 0293 27053.

Apple and the Apple symbol are the registered trade marks of Apple Computer Inc. Apple User is an independent publication and Apple Computer is not responsible for any of the articles in this magazine, nor for any of the opinions expressed.

Writing for Apple User: Articles and programs relating to the Apple are welcome. Articles should preferably be typed or computer-printed, using double spacing. Unsolicited manuscripts, discs etc, should be accompanied by a self addressed stamped envelope, otherwise their return cannot be guaranteed. Unless agreed, material is accepted on an all rights basis.

© 1985 Database Publications Ltd. No material may be reproduced in whole or in part without written permission. While every care is taken, the publishers cannot be held legally responsible for any errors in articles or listings.

Subscription rates for 12 issues, post free:
£12 UK
£13 Eire (IR £16)
£18 Europe
£15 USA (surface)
£25 USA (airmail)
£15 Rest of world (surface)
£30 Rest of world (airmail)

GEM threat doesn't worry Apple

APPLE UK has turned up its corporate nose in disdain at reports that sales of the Macintosh are threatened by a software breakthrough.

The claim is based on a new "man-machine interface" which is reputed to be able to convert other micros on the market into serious rivals for the Apple machine.

Developed by Digital Research, the package – known as Graphics Environment Manager (GEM) – is said to offer a Macintosh environment complete with windows and icons controlled by mouse, joystick, tracker ball and speech.

"This means that people who want to enjoy the Macintosh-like features will not now be tied

to one machine", Paul Bailey of Digital Research told *Apple User*. "While Macintosh is a nice product, the potential customers will now have a choice".

It has been revealed that several manufacturers have already committed themselves to the GEM software, including Acorn for the ABC 300, ACT for the Apricot and ICL for several new products to be announced this year.

However over at Apple headquarters marketing manager Bob Kissach shrugged off any

possibility of GEM posing a threat to the Macintosh.

"It's very flattering that people want to copy Apple", he said, "but it really is a question of computer generations – 16 bit versus 32 bit.

"GEM is designed for the MS-DOS clones – IBM and its lookalikes. And the 16 bit machine simply doesn't have the horsepower to run it.

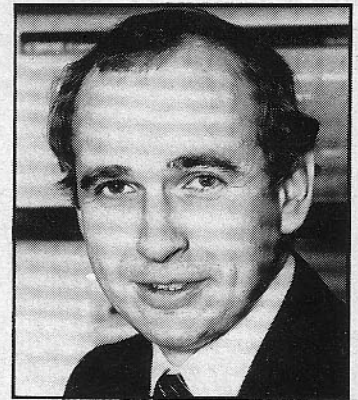
"We designed the Macintosh from the ground up around the Motorola 68000 chip, which has got the necessary power for the job.

Within days of the interview with Bob Kissach the threat to the Macintosh suddenly became a reality.

Atari announced two new 16 bit machines with full Macintosh features at around one third of the price.

Unveiled at the Consumer Electronics Show in Las Vegas, the 128k 130ST and the 512k 520ST both use the full M68000 chip. And each model has 196k built-in ROM which contains GEM and CP/M.

It was also revealed that languages which run on Apple – such as as C, Pascal and Logo – will become immediately available on the new machines.



Paul Bailey... "customers now have a choice"

Test drive goes on

SUCCESS of the nationwide "Test Drive a Macintosh" campaign has led to it being extended until at least the end of March.

The promotion – involving more than 300 Apple dealers throughout the country – was originally intended to run from November 1 to December 31.

"But the offer proved so popular that to finish the campaign on schedule would have meant a lot of people not getting the chance of a test drive at all – and dealers might have lost out on potential sales", a spokesman said.

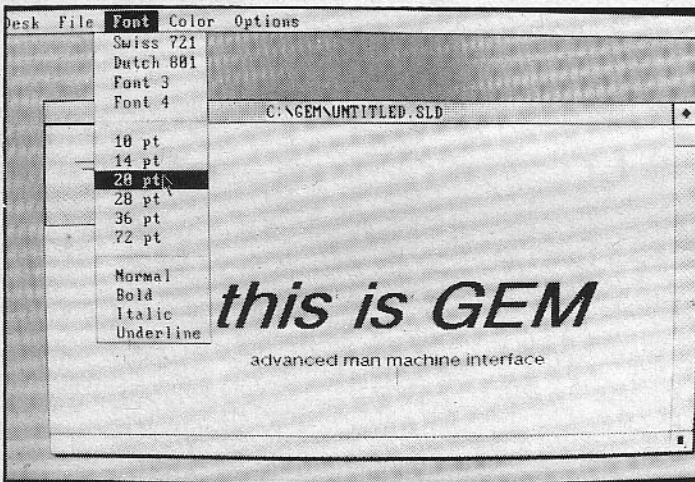
"The extension means everyone who wants to take a Macintosh home should get the opportunity to do so.

"We have also improved the promotion with new features to benefit potential customers and dealers.

"A major improvement is the provision of a much better selection of test drive software, particularly for specific professions and business applications.

"We were finding that many people were going back to dealers saying they liked the machine but weren't sure about its value to them in their trade or profession.

"So we have been busy obtaining a supply of software for the vertical markets so that such potential purchasers can undertake evaluation of the machine in the context of their work".



A screen from GEM Wordchart

Acorn slips on an Apple

APPLE has been blamed for the failure of Acorn's plan to take the lucrative United States education market by storm.

The threatened invasion by a specially-configured version of the BBC Micro ran aground when Apple counter attacked with an aggressive marketing campaign.

"Up until then we had found no real competition", an Acorn spokesman told *Apple User*.

Following "poor sales", Acorn has now slashed its

operation in America by 80 per cent. And the company admits to losing in the region of £6 million in its unsuccessful attempt.

In all Acorn spent some £500,000 promoting the BBC Micro Stateside during 1984. The company also paid more than £300,000 for the US rights to the BBC's educational series linked to learning about the machine, which was shown on American TV.

However an Acorn spokes-

man denied reports that Apple had forced the company to make a total withdrawal.

"We haven't pulled out altogether", he told *Apple User*. "The fact is that we have now tailored the size of the operation out there to meet the present requirements".

Acorn considers that part of its failure to penetrate the US market was due to delays in obtaining approval from the US Federal Communications Commission.

Mac pack hits the headlines

A NEW Macintosh package by Keith Lander and Mike Glover of Icon Technology is arousing considerable interest.

Apple UK's Macintosh software manager, Richard Bradley, calls it "one of the most exciting products yet to come out of the UK".

Mike Glover describes the package, MacAuthor, as "a words and graphics processor" for the Macintosh.

He says it is aimed at technical writers and others who want to produce long documents in standard formats. The package will handle up to 698 pages.

The big advantage of MacAuthor, he says, is that the user can decide in advance what he wants his document to look like.

This chosen style then appears automatically all the way through.

A style can involve multiple paragraphs with various types of headings and sub-headings.

After setting up line length, interline spacing, paragraph separation and justification – left, right or centre – the user can choose the size and style of typeface.

As well as bold, roman or italic there are underlined, outline or shadow styles. These can also be varied in pitch to give normal, condensed or extended lettering.

Style can be edited at any time. So if the writer decides he does not like the look of it, he

MacAuthor combines all the features you would expect to find in a powerful word processor with a unique Album containing up to 18 pages of symbols.

Frames can be opened in the text and symbols can be dragged into them using the mouse.

MacAuthor also has drawing tools so you can draw in the frames giving freedom of expression far beyond that of a conventional word processor.

When words alone are not enough, combine the Album with the drawing tools and text.

Rest sweet nymphs

simply changes the built-in style manual.

Mike's favourite feature is what he calls his "electronic Letraset". This makes available unusual characters – from mathematics, music, etc. The user fits these into the text, which parts to make room for them.

There are also inbuilt drawing tools for producing squares, straight lines and circles, which are also fitted into the text as required.

It should be available shortly, and priced between £150 and £200.

APPLE BARGAINS

APPLE products fetched good prices at the second Crocker Computer Auctions sale in London. A Mac, which failed to reach a reserve at the first auction, was sold for £1,200. An Apple II Europlus made

£230 and assorted Apple drives fetched between £80 and £90 each.

Crocker promises "many more" computers, software and peripherals at its next auction planned for this month.



MORE than 50 Apple dealers saw the latest Macintosh software at a recent demonstration, organised by Softsel.

It was designed to give

them a better understanding of the expanding range of Macintosh software available – Softsel now offers over 120 application packages.

Top of the poll

THE top three places in a survey of Macintosh software have gone to Microsoft products.

The survey, commissioned by Apple and conducted by Mori, puts Microsoft's Chart at number one, followed by Multiplan and Basic.

A delighted Microsoft spokesman said: "The positioning of our products clearly indicates that the Mac user has

a real need for powerful, easy to use business applications.

"With the arrival of Microsoft Word and File early this year we will be in an even stronger position in the Top 20 stakes".

The complete suite will provide Macintosh owners with a comprehensive set of integrated business tools that will include Microsoft Word, File, Chart and Multiplan.

In the news

CANADA'S leading newspaper, the Toronto Globe and Mail, is putting 50 Macintoshes into its newsroom for use by journalists.

The new 512k Big Mac micros will be linked direct to the mainframe that masterminds the newspaper's typesetting.

This means that in future copy will be keyed in once only. Previously, it was keyed in twice – once at the writer's typewriter, and once again by a compositor.

The new Apple micros will have another big advantage. The dumb terminals formerly in use had to refer all typesetting decisions to the mainframe.

The new equipment will be able to handle these in advance, thus taking the load off the mainframe.

A further advantage is that if the mainframe breaks down reporters will be able to go on typing their stories which will then be ready to go as soon as the breakdown is fixed.

Journalists will also be able to use the Mac's graphics to create charts and so on for business pages, and they will be able to call up information from the paper's database.

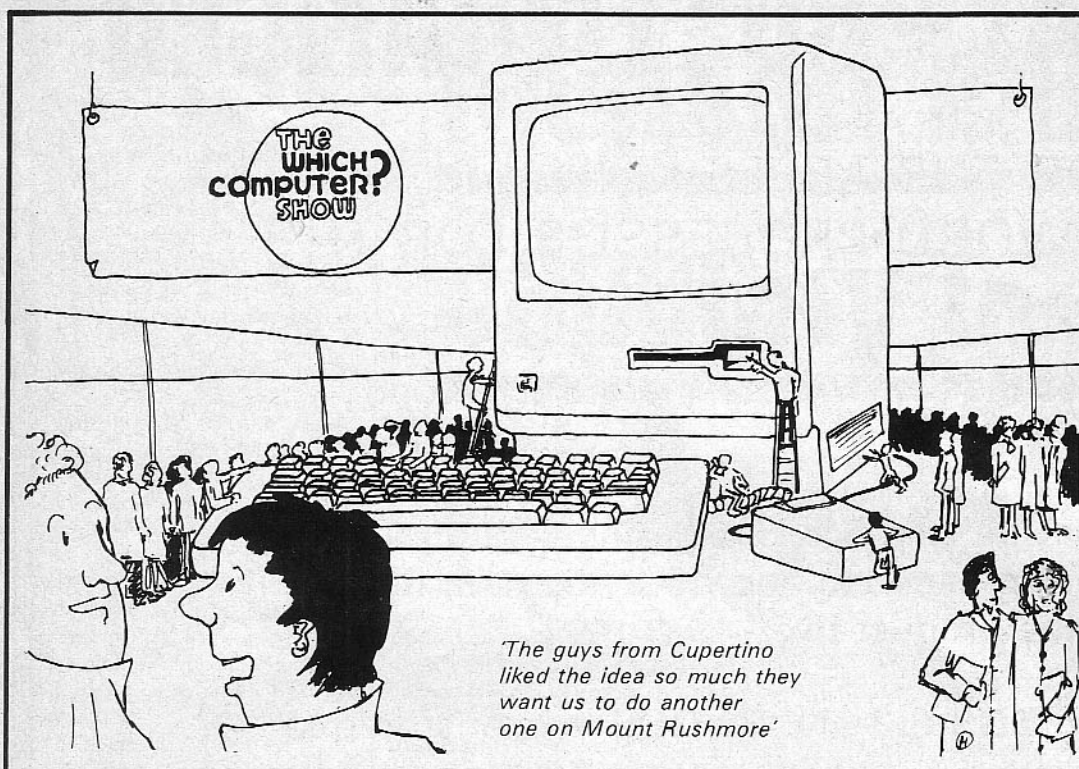
In the wrong

IT seems that those distinguished bowler hatted gentlemen at Dun and Bradstreet can't tell their Apples from their Apricots . . .

One of the company's respected financial publications recently carried a story under the headline – "Low-cost software links Apple with worldwide databases".

However the Credit News article then went on to mention the real subject of the piece – ACT's Apricot – some 12 times.

"This seems to suggest that one Apple is worth a dozen Apricots", observed an Apple spokesman.



Enter a giant Mac

REVERSING the historical trend towards smallness in computers, Apple has come up with a 12 foot high Macintosh.

But you won't be able to buy it even if it would fit your desk top – it is intended only for demonstration purposes at computer shows throughout the country.

Together with an identical twin it was under construction last month at a specialist firm in France before being shipped here to be assembled for the start of its UK tour.

The use of an internal back projection system gives the realistic appearance of a working Macintosh monitor. In reality the image is produced by a presenter operating a standard desktop Macintosh.

Brainy printer to be unveiled

APPLE is likely to launch what is believed to be the world's most intelligent laser printer for the Macintosh to coincide with its AGM.

It is thought that the "hyper intelligence" of the machine may be due to a built in disc drive. As a result all of the Mac fonts will be reproduced as though typeset. The printer is also believed to contain a megabyte of memory for spooling.

Although it is due to go on

sale in the United States at the time of the launch the printer is not expected to be available in the UK until May.

The company's AGM has always been used as a launching pad for major products but this year – according to *Apple User's* source – there will be no new models, not even the long awaited colour Macintosh.

However it is rumoured that Apple will also use the occasion to unveil Applebus, a networking system for the Macintosh.

CHIPS REVOLUTION

THE days when micro manufacturers like Apple based their machines on standard microchips may soon be over.

So says a report by International Resource Development, the American market research company.

Its researchers say it will

soon be just as easy to create a custom built chip as to use a standard one.

This will make life hard for the makers of lookalikes. Also for software developers, who will have to choose very carefully which machines they support.



APPLE is developing a plug-in card based on a new 16 bit processor which would allow existing Apple II micros to compete head-on with the IBM PC in the higher-end personal computer market.

The development was revealed by Apple II inventor and co-founder Steve Wozniak during a visit to New Zealand to speak at the Wellington Apple User Club's champagne breakfast.

Wozniak said the card would contain the new 65816 micro-processor which with its 24 bit address space can address up to 16mbytes of RAM.

In practical terms the card would allow Apple II Plus and IIe micros to be expanded to "many megabytes" of RAM. This would support memory-hungry products similar to Lotus Symphony "... and who knows what else down the track".

Greatly increased memory for the Apple IIe would offset much of the criticism directed at the integrated Appleworks program which suffers from severely limited capacity in both its database and spreadsheet functions.

For instance, Appleworks' spreadsheet in theory contains a large number of cells, but in practice only a small proportion of them can be used - even on a 128k machine.

Wozniak said the 65816 card would act as a co-processor within an Apple II: "You wouldn't notice its presence until you used it", he said.

An advantage of using the

Monster memory capacity on way for Apple II

65816 chip was that an Apple II could still use the existing huge library of Apple II programs.

Problems surrounding the development have been largely overcome, Wozniak said, and if a final decision is made to proceed the product could be available in less than a year.

"But we won't go ahead until everything comes together and makes sense", Wozniak said.

"It has to do the job we want, from both a hardware and software point of view, easily and supportably.

"We want to present it in such a way that it will be around for five to 10 years".

Wozniak scotched rumours that Apple was developing an all-new computer based on the 65816 chip and containing expansion slots, code-named the "Iix".

"That project is dead", he said.

But he did confirm that the 65816 update would not be available for the Apple IIc which is not further expandable.

He hinted that the IIc will probably switch to 3.5in Sony disc drives similar to the Macintosh drives.

At least during a changeover period, the smaller, higher capacity drives would be interchangeable with 5.25in drives because of an unpublicised feature of the IIc - it is possible to boot from the external drive by typing PR#7.

He also predicted that the Macintosh would continue for at least 10 years and would become a 4mbyte machine. He said Macintosh would replace the Lisa, which in spite of being an excellent machine had not

caught the public's imagination and was losing "tens of millions of dollars".

Macintosh would take over as Apple's high end machine as soon as its memory and software systems were sufficiently developed.

Wozniak visited New Zealand in a private capacity as a result of an invitation by the Wellington Apple Users' Club.

The club invited him "... on behalf of our 100 plus members, the other 2,199,900 New Zealanders, and 60 million sheep". They got quite a surprise when the invitation was accepted.

"How could I turn down an opportunity to attend a genuine hacker-mode breakfast in a New Zealand pizza hut?" declared Wozniak.

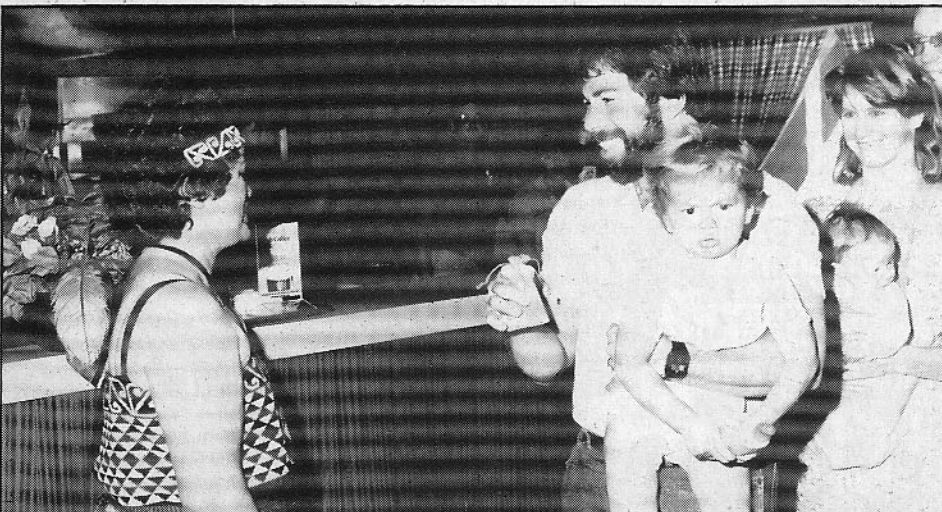
And he brought along his two young children and his wife, former US Olympic kayaker and Apple accountant Candi Clarke.

Also with him was a portable Apple IIc, which Steve donated to the Wellington Apple User Club.

However he didn't go home empty-handed. While in Wellington he bought the New Zealand PAL colour version of the IIc, to add to his collection of computers back home in the Santa Cruz mountains south of San Francisco.

The Kiwi IIc will join an Apple I, several versions of Apple II, Macintosh, Lisa, IBM PC, Grid Compass and the new Data General One.

John MacGibbon



Maori welcome for Steve Wozniak, his wife and children



UCSD Pascal goes Mac!

Now you can penetrate the expanding world of Apple Macintosh while writing applications in the language programmers love:

UCSD Pascal*. The original. The fully-developed industry standard. Complete with extensions for system development and commercial applications programming.

You can write programs which utilise the Mac's graphs, fonts and mouse. Without fuss or complication. And if you've got any really good Apple II Pascal programs, you can run them too on your MAC. All it requires is a simple compilation.

Minimum Macintosh requirements are just 128K RAM and a single floppy disk drive.

It's made possible by TDI Limited, distributors of the microcomputer development environment praised by novice and professional programmers alike. You can capitalise on today's technology as well as tomorrow's emerging standards. Our

completely integrated environment offers you sophistication, coupled with ease of use. And it's modular, expanding to fit your needs.

Start with the basics: UCSD Pascal, a complete set of development utilities; a full-screen, full-function editor and a file manager.

Add another language. Or two. Such as FORTRAN-77 or BASIC. Our compilers generate compact code and are truly integrated with the development environment.

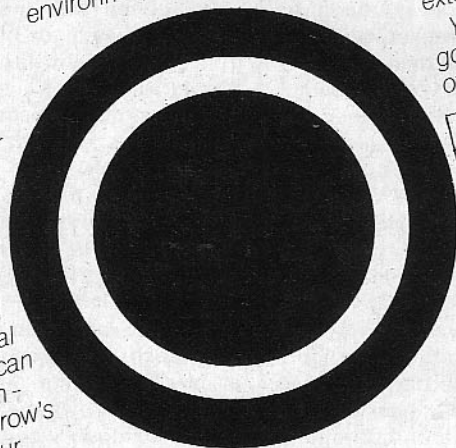
Edit with ease. When an error is detected, the editor may be invoked and positioned at the statement in error.

You can also write in smaller program segments. This simplifies debugging and testing and lets you use common procedures in multiple applications. Even if they were written in one of our other languages.

For professional developers, we also offer a full set of productivity tools such as assemblers and a symbolic debugger.

Clip and post the coupon below. Or call us on 0272 742796. We'll send you information on the extensive list of TDI's products.

You'll find that UCSD Pascal goes Apple Macintosh. And a lot of other places besides.



SPECIAL PRICES

APPLE MACINTOSH P-SYSTEM
Inclusive Pascal or Fortran Compiler,
Editor, Filer, Utilities only **£295**

ADVANCED DEVELOPMENT
TOOLKIT normal selling price **£150**

SPECIAL OFFER PRICE **£75**
JANUARY & FEBRUARY

TDI Pinnacle Limited
29 Alma Vale Road
Bristol BS8 2HL
Tel. (0272) 742796.

*UCSD Pascal is a registered trademark of The Regents of the University of California.

Please send me information on UCSD Pascal for Apple Macintosh

Name _____

Title _____

Company _____

Address _____

Post Code _____

Telephone (_____) _____

- Area of Activity:
- Software developer
 - Retailer/dealer
 - Education
 - Other

TDI Limited
29 Alma Vale Road
Bristol BS8 2HL
Tel: (0272) 742796

AU 2 85

IN addition to demonstrating the Quicksort method, this article by COLIN J. DAVIES provides an interesting comparison of Forth and Basic.

On your marks...

THE Quicksort method is based on the partition algorithm devised by C.A.R. Hoare in the early sixties.

Its advantages are speed and the fact that the sort is done in situ without the need for a second array.

The two programs that accompany this description use the iterative version of Quicksort rather than the recursive version.

In order to understand how Quicksort works, you must first understand how partition works.

The idea is that you take the first element of the array to be sorted as the partition element.

The elements of the array are

then re-arranged so that all those elements that are less than the partition element come before any elements that are greater than the partition element.

The partition element is then positioned between the two.

The best way to illustrate the effect of partition is with an example. See Figure 1.

It should be noted that partition does not of itself sort the array. It re-arranges the array

It's Forth v Basic in this revealing Quicksort comparison

Before partition		After partition	
Element	0 1 2 3 4 5 6 7 8 9	Element	0 1 2 3 4 5 6 7 8 9
Value	4 1 5 8 9 6 0 2 3 7	Value	0 1 3 2 4 6 9 8 5 7

The partition element was 4 - the first element.

Figure 1.

into two sub-arrays.

The Quicksort method works by taking the partitioned array, determining which of the two sub-arrays is the smaller, and partitioning that. The indices of the larger sub-array are pushed onto the stack.

This procedure is repeated until there is nothing left to partition.

At this stage the values on the top of the stack are pulled and the process continued. When the stack is empty the sort is complete.

In order for this method to work, it is necessary for one extra element to be added at the end of the array containing a value greater than any element of the array.

The size of the stack depends on the number of elements to be

sorted and is equal to $2 \log_2 n$ where n is the number of elements to be sorted.

Therefore, to sort 10 elements a stack with a depth of eight would be required, and to sort 1000 elements a depth of 20.

There are two versions of the program shown here, one in Forth and the other in Basic.

They both demonstrate Quicksort by filling an array with 1000 random integers in the range of 1 to 1000, sorting them, and displaying the result.

To facilitate timing of the sort, the bell is sounded at the start and end of the sort.

The Basic version takes three minutes 20 seconds, the Forth version 19 seconds. For comparison, an IBM 370 mainframe takes 0.4 seconds in PL/1.

```

100 REM QUICKSORT DEMO
110 REM
120 LET N = 1000:Q = N + 1:
    DIM A(Q): DIM
    STACK(20):A(Q) = 9999:P = 1
130 HOME : PRINT "STANDBY
    LOADING ARRAY"
140 FOR K = 1 TO N:A(K) =
    INT ( RND (1) * 1000) + 1:
    NEXT
150 HOME : PRINT "SORTING"
160 PRINT CHR$ (7): GOSUB
    260: PRINT CHR$ (7): CHR$
    (7): HOME
170 FOR K = 1 TO N
180 LET AK$ = STR$ (A(K))
190 IF LEN (AK$) < 5 THEN
    AK$ = " " + AK$: GOTO 190
200 PRINT AK$: IF K / 12 =
    INT (K / 12) THEN PRINT
210 IF K / 240 = INT (K /
    240) THEN GET A$: HOME
220 NEXT : END
230 REM
240 REM QUICKSORT
250 REM
260 LET T = 0: REM STACK
    POINTER
270 IF NOT (P < Q) GOTO 320
280 LET L = Q + 1: GOSUB 390:
    IF NOT (L - P < Q - L)
    GOTO 300
290 LET STACK(T + 1) = L +
    1:STACK(T + 2) = Q:Q = L -
    1: GOTO 310
300 LET STACK(T + 1) =
    P:STACK(T + 2) = L - 1:P =
    L + 1
310 LET T = T + 2: GOTO 270
320 IF T = 0 THEN 350
330 LET P = STACK(T - 1):Q =
    STACK(T):T = T - 2
340 GOTO 270
350 RETURN
360 REM
370 REM PARTITION
380 REM
390 LET V = A(P):K = P
400 LET K = K + 1: IF A(K) <
    V THEN 400
410 LET L = L - 1: IF A(L) >
    V THEN 410
420 IF NOT (K < L) THEN 450
430 LET TEMP = A(K):A(K) =
    A(L):A(L) = TEMP
440 GOTO 400
450 LET A(P) = A(L):A(L) = V
460 RETURN
    
```

```

SCREEN #120
0 ( PARTITION )
1
2 1000 CONSTANT N 1 N 1+ ARRAY A
3
4 VARIABLE P VARIABLE L
5
6 : SET-P-VAL P @ DUP A @ SWAP : ( ---V,K )
7
8 : POSN-P-VAL L @ A @ P @ A ! L @ A ! : ( V--- )
9
10 : INTERCHANGE DUP A @ L @ A @
11 3 PICK A ! L @ A ! : ( V,K---V,K )
12
13
14
15 --->

SCREEN #121
0 ( PARTITION CONTD )
1
2 : UP-LOOP BEGIN ( V,K---K,V )
3 1+ DUP A @
4 3 PICK >=
5 UNTIL
6 SWAP :
7
8 : DOWN-LOOP BEGIN ( K,V,L---K,V,L )
9 1- DUP A @
    
```


QUICKSORT

```

10          3 PICK <=
11          UNTIL
12          DUP L ! ;
13
14 -->
15
SCREEN #122
0 ( PARTITION CONTD )
1
2 : MAIN-LOOP BEGIN ( V,K---V )
3   UP-LOOP
4   L @
5   DOWN-LOOP
6   3 PICK >=
7   WHILE
8     SWAP
9     INTERCHANGE
10    REPEAT SWAP DROP ;
11
12 : PARTITION SET-P-VAL ( --- )
13   MAIN-LOOP
14   POSN-P-VAL ;
15 -->
SCREEN #123
0 ( QUICKSORT )
1
2   VARIABLE Q VARIABLE D
3
4 : NEAR-BOTTOM L @ P @ - Q @ L @ - < ; ( --- f )
5
6 : REMEMBER-TOP L @ 1+ Q @ L @ 1- Q ! ;
7
8 : REMEMBER-BOTTOM P @ L @ 1- L @ 1+ P ! ;
9
10 : UPDATE-STACK NEAR-BOTTOM ( --- S1,S0 )
11   IF
12     REMEMBER-TOP
13     ELSE
14     REMEMBER-BOTTOM
15   THEN ; -->
SCREEN #124
0 ( QUICKSORT CONTD )
1 : INNER-LOOP BEGIN
2   P @ Q @ <
3
4   WHILE
5     Q @ 1+ L !
6     PARTITION
7     UPDATE-STACK
8     REPEAT ;
9 : QUICKSORT Q ! P ! DEPTH D ! ( P,Q--- )
10  BEGIN
11  INNER-LOOP
12  DEPTH D @ < >
13  WHILE
14  Q ! P !
15  REPEAT ; -->
SCREEN #125
0 ( QUICKSORT DEMO )
1
2   VARIABLE RND HERE RND !
3
4 : RANDOM RND @ 31421 * 6927 + DUP RND ! ;
5
6 : CHOOSE RANDOM U * SWAP DROP ; ( U1---U2 )
7
8 : FILL-ARRAY 32767 N 1+ A !
9   N 1+ 1 DO
10  1000 CHOOSE 1+ I A !
11  LOOP ;
12
13 --
14
15
SCREEN #126
0 ( QUICKSORT DEMO CONTD. )
1
2 : DISPLAY N 1+ 1 DO
3   I DUP DUP
4   A @ 4 .R SPACE
5   12 MOD Q= IF CR THEN
6   240 MOD Q= IF PAUSE PAGE THEN
7   LOOP ;
8
9 : QUICKSORT-DEMO PAGE . " STANBY LOADING ARRAY "
10  FILL-ARRAY PAGE
11  " SORTING " BELL
12  1 N 1+ QUICKSORT
13  BELL BELL PAGE
14  DISPLAY ;
15

```

TIME IS MONEY.

Introducing Project Leader, the total programme specially written for the professional practice.

Architects, Accountants, Engineers, Quantity Surveyors, Solicitors and Consultants all agree.

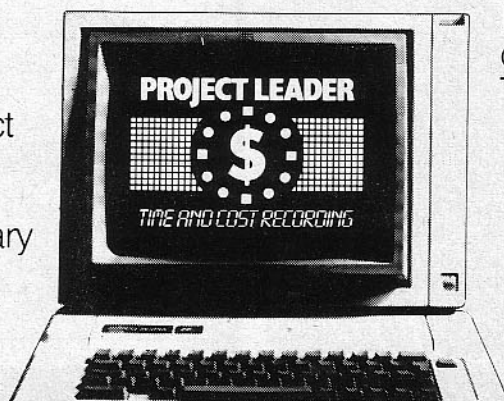
Project Leader is the ideal software package for their business. And Project Leader was written with Apple computers in mind—though it's compatible with all major micros.

Easy to use with clear concise documentation, Project Leader fulfills all time, cost and client billing analysis, work in progress listings, client summary broken down by period, full audit trail, expenses analysed by client or by employee, time

sheet listings, client ledger, fast posting procedure, everything to help the professional practice run smoothly and efficiently. So if you want to make your practice pay, contact;

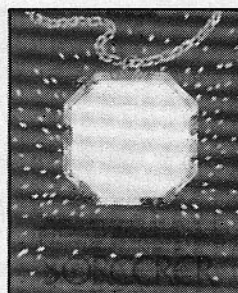
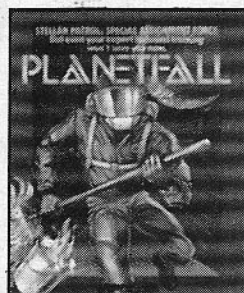
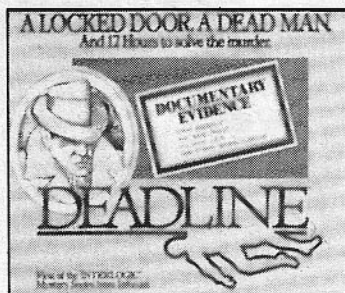
Datacode (Systems) International Ltd., 1-2 Leeson Close, Dublin 2, Ireland.

Phone 0001-761242
or 688481.
Telex 24309 SHAF EI.



TIME AND COST RECORDING

The perfect business partner for your Apple.



FIVE years ago Infocom released their first product. It was called Zork I and it was a pretty revolutionary slice of software, a new form of text adventure.

Zork, an underground adventure, changed the dull tradition of verb-noun inputs into complex sentences. People used to type "Kill troll". Now they could "Jump on the troll then stab him with the dagger". Not much fun for the troll, but a whole lot more fun for the player.

Towering over old games was a vocabulary of more than 500 words. And with this freedom of input came a vast and complex world to explore. A real chart-topper.

Since then Infocom have released 13 text-only adventures. Each has maintained a tradition of high quality. This has always been reflected in the superb packaging and documentation.

Each instruction manual is plastered with the wit of a genius, and splashed with paints that spare no detail. The descriptions and responses within the game are equally lavish.

I have been looking at five of their games recently – Planetfall, Enchanter, Sorcerer, Deadline and Witness.

Planetfall is the latest of their SF games. Often its problems are of a technical nature. Being more of a fantasy man, I found this one very tricky. In fact, it reminded me of that old classic "Mission Impossible".

To me, "Mission Impossible" was impossible. Let's just say I haven't quite finished Planetfall yet!

As an Ensign 7th Class of the Stellar Patrol, your abilities are

Firmly in the footsteps of Zork...

stretched to the full. Your awesome task – scrubbing the deck of the starship Feinsteint!

But very soon Ensign 1st Class Blather reminds you how terribly real Infocom characters are.

"You call this polishing, Ensign Seventh Class? We have a position for an Ensign Ninth Class in the toilet-scrubbing division, you know. Thirty demerits!"

Then it's bye-bye to Blather as disaster hits the Feinsteint, leaving you marooned alone on a deserted planet.

Exploring the buildings of the twin complexes on the planet can get lonely, until you discover Floyd. This little guy is a robot, and he's the solution to several problems in the game. His masterpiece of a character

is typical of Infocom.

There seems to be no aim to the game at first other than to explore and survive. The one problem playing Planetfall is that you are never quite sure why you are doing just what you are.

While perhaps this makes the adventure more exciting, frustratingly tossing a puzzle to the heap of other unsolvables often results. Why waste time solving a problem of no apparent benefit?

Positively, Planetfall is a very funny comedy. Responses are very often hilarious, and Floyd is particularly amusing. Descriptions are brief yet graphic.

The package spares no expenses – a folder opens to reveal pockets containing an instruction manual, an identifi-

cation card, interplanetary postcards ("Wiggle on over to Nebulon, home of the Worm People"), and a very funny diary recounting your exploits in the Stellar Patrol up to date. I only wish I could win!

In **Deadline** and **Witness** you become your favourite detective. Complex characters replace the vast number of rooms found in adventures, and the hunt for objects is replaced by the hunt for clues.

In the other games it's points you are after. In these two, it's the time limit that sets the challenge.

Witness has the 1930s perfected. The game and documentation together create an atmosphere that Deadline definitely lacked. What is more, improvements have been made in the interpretation routines. As a result it is altogether easier to question suspects.

The descriptions and characters are often brilliant. I think the guy who wrote the story could be writing best-selling novels. The lowdown is that a high class society dame has been murdered.

Her old man (sorry folks, the lingo's straight from the game!) worries that he'll be the next stiff. What can you do, detective? The client's life is at stake, so make no mistakes.

Your case starts outside his house where a red matchbook turns up. Could this be your first clue?

The literature in the package



Part of the packaging of Witness... 'plastered with the wit of a genius'

Arguments prove this expedition is a winner

I THINK I would enjoy **Expedition Amazon** if I ever got a chance to play it – most of the time I was looking over the shoulders of my teenage offspring watching their progress.

This is because only two expeditions can be in progress at any time, that is, saved on the disc to be resumed later.

It means that if I were to sneak a quick game while they are at school, and try to start a new expedition of my own, one of the existing parties would be wiped out.

This is a serious drawback in a good game – what a pity there is no facility to initialise another disc for storing lots of expeditions. Potentially dangerous situations could then be backed up and replayed several times.

The aim is to discover the fabled lost city of Ka, which lies buried beneath the jungle in Peru.

The scene is set by the prologue in the leaflet accompanying the game which describes how Professor Arrowhead – a part-time academic of somewhat dubious reputation – became convinced that Inca is really spelled In Ka – hence the lost city.

He also spells Indian as Injun

– does this mean a possible follow-up set in North America?

There is lots of humour in the game. The fun starts with the pictures on the back of the disc envelope. As well as the usual

warnings. Penguin advise us, for example, not to put the disc in the pop-up toaster, not to use it as part of a shish-kebab, nor to line the floor of the budgie's cage. But seriously after recruit-

ing the expedition members field assistant, medic, radio operator and guard – one goes predictably to the trading post to equip them.

After an appropriate mess-



Spot the sub-aqua Penguin in the title page

In the footsteps of Zork – from previous page

is outstanding. There's a "National Detective Gazette" which among the '30s memorabilia includes hints on how to play the game. A newspaper outlines the stories of the day, including the murder that introduced the case. Even a telegram and suicide note are included, and the matchbook found outside the house completes the package in style.

Deadline launches you into the Robner estate, where Mr Robner has died of an overdose. Suicide probably. But why was he about to change his will?

Sergeant Duffy will fingerprint anything for you, or take objects to the lab for analysis. With his help, you must uncover any foul play. Having gathered enough clues, you may arrest your suspect and await the

outcome of the trial – so they tell me!

The potpourri of supplementary evidence in the package includes notes from interviews with all the main characters, a coroner's examination, a photograph from the scene of the death, a lab report, and even three tablets from the suicide bottle! Official letters outline your task, complementing an informative inspector's casebook.

Be careful Watson, these cases are not elementary!

Enchanter and **Sorcerer** kick off a series of magical quests by Infocom. The nicknames, Zork IV and V, reflect the similarity of style with the Zork trilogy.

Their hallmark is the spell book. Throughout the games

you discover more and more spells to add to your collection. They may be cast on almost anything, a terrific accomplishment by the programmers.

Some spells found on scrolls are too powerful to learn, and may only be cast once. Often, such a spell could solve many problems, but to which one is it the unique solution? Using it unnecessarily will thwart the quest.

In **Enchanter**, your adventures start when the Circle of Enchanters summons you. Dark times have arrived, you are told. Krill's power is strengthening and its influence is stretching.

An ancient document speaks of such times, and declares that a young novice enchanter like yourself must go forth and discover Krill's secrets. To send

a more powerful enchanter would too soon reveal the power of the circle. You must explore Krill's castle, and bring back his dark secrets.

This is the introduction to a superb game, my favourite of the five. Delve deep into the castle, discover magic, uncover sinister secrets, dream strange dreams.

Dreams? That's right, you actually *live* Infocom games. Forget to sleep and you are too tired to remember spells. Eat, or starve to death. Drink, or die. No two ways about it.

The characters in **Enchanter** are all beautifully developed. Owners of the Zork trilogy will especially enjoy the familiar adventurer who is found along the way.

Just as well developed is the

age on the screen during the disc access – un momento por favor – one meets a wonderful character called Pedro who wiggles his moustache, winks at you, and tells the 'Gringo' appalling jokes until you can bear it no longer and press a key to get on with the game.

Once in the jungle the expedition begins to map the various sectors. The treasure they find can vary from the valuable – such as a raw diamond – to the worthless – a mere clay pot – or even the ridiculous – a pet rock, a Dr Spock book, or, would you believe, a Speak-and-Spell.

The hazards are equally bizarre – rabies, yellow fever, fleas which cause plague, and even crocodiles which can give a character 'big toe injuries'. In this game D & D stands for disease and disability.

Treasures can be taken back to the trading post and sold to our old amigo, Pedro. He's something of a crook though, and bargaining can push up the prices.

When the team members have acquired enough equipment and built up their strength, they can begin to explore the underground tunnels trying to



Pedro greets you bilingually in Expedition Amazon

avoid being drowned in flooded crypts or wiped out by volcanic eruptions.

It is here with animated action they eventually find the maze and the door to the lost city. It takes a very long time.

If characters are killed, new ones can be recruited by finding the way back to base camp and though their expertise will be low, the expedition retains its equipment and money.

There is a choice of nine

difficulty levels. Even level one, which is supposed to be very easy, is really quite hard so this should give many hours of entertainment.

A minor quibble is that the instructions consist of nine pages of print on the boot side of the disc so once the game is in progress and the disc turned over, they cannot be referred to.

On first playing the game it is a good idea to make notes of key instructions. Why not

provide these pages on paper and include them in the package?

Niggles apart, if a game is judged by the number of arguments over whose turn it is to play then this looks a winner.

Gill Reeve

*Title: Expedition Amazon
Author: Willard Phillips
Publisher: Penguin Software
Requirements: 48k*

mysterious atmosphere. Playing the game is like reading a fairy story. Just don't expect a happy ending all of the time. Krill isn't smiling because he likes you!

Equally magical is the packaging. Enclosed is the sealed document that defines your task. It's on parchment paper, and I'm still not sure if the ink is dry. Prepared by the "Printer's Guild" is an instruction manual, a fine work of calligraphy.

In **Sorcerer** you are the renowned enchanter who defeated Krill. But a new task is at hand. Belboz, the leader of the Circle of Enchanters, has disappeared. His abandoned cryptic diary hints that he has fallen victim to evil sorcery. You must find Belboz and rescue him from... from what?

An engrossing magical

kingdom opens itself for exploration. The kingdom is large and diverse. Problems from magical minefields to a menacing hawk in an amusement park face you. A worthy sequel to Enchanter.

The packaging is once again out of this world. Especially peculiar is the "infotater", a windowed envelope containing a rotating disc which reveals information about monsters when turned. The solution to a problem perhaps?

Both adventures have fascinating puzzles. There is no messing around with nonsensical solutions. Each problem is charming yet has a logical end.

I think these games come closer to the non-computer role-playing games such as Dungeons and Dragons than do

adventures like Wizardry. Highly recommended for fantasy fanatics.

Each Infocom game is very convenient to play. Saving the play to disc is possible at any point, and there are single letter abbreviations for several common commands.

If you have been in a room more than once descriptions are shortened. In fact, a whole list of features cater for every fuss. The play may even be recorded on a printer.

Excellent hint books and maps are available from Infocom if you get desperate. Even if you have to resort to such devices, you should still be able to enjoy the games because care has been taken not to give too much away.

There is no doubt that

Infocom games are immensely popular. Why? Because of the full-sentence input? The latest games have 1,000 plus word vocabularies. Because of the superb realism? Playing a game is like acting out a book. Perhaps because of the fine attention to detail? The packages are ever more spectacular.

Whatever it is, people will rave about these games until the cows come home.

Julian Brewer

*Titles: Planetfall, Enchanter, Sorcerer, Deadline, Witness.
Authors: Not credited.
Publisher: Infocom.
Requirements: 32k minimum for any Apple II or for the Mac.*

ONE feature of Applesoft's graphics that causes more trouble and confusion than any other is the use of shape tables. But once mastered they are truly indispensable little creatures.

Shape tables provide a means of putting text on the hi-res screen, plotting symbols on graphs and they hold the key to speedy animation from within Basic.

Unfortunately there are a number of difficulties in using them that often drive the beginner to near suicide, not the least of which is that the section on shape tables in the Applesoft manual reads as if it has been written by a demented cypher clerk!

This month we start a two part series on shape tables in which I hope to convince even the newest novice that mastering these mysterious objects is really quite easy.

First I want to look at how we use shape tables, where we put them and how they are organised.

Next month I'll describe in detail how to construct a table and provide a simple shape table editor, written in Basic, to take all the tedium out of the job.

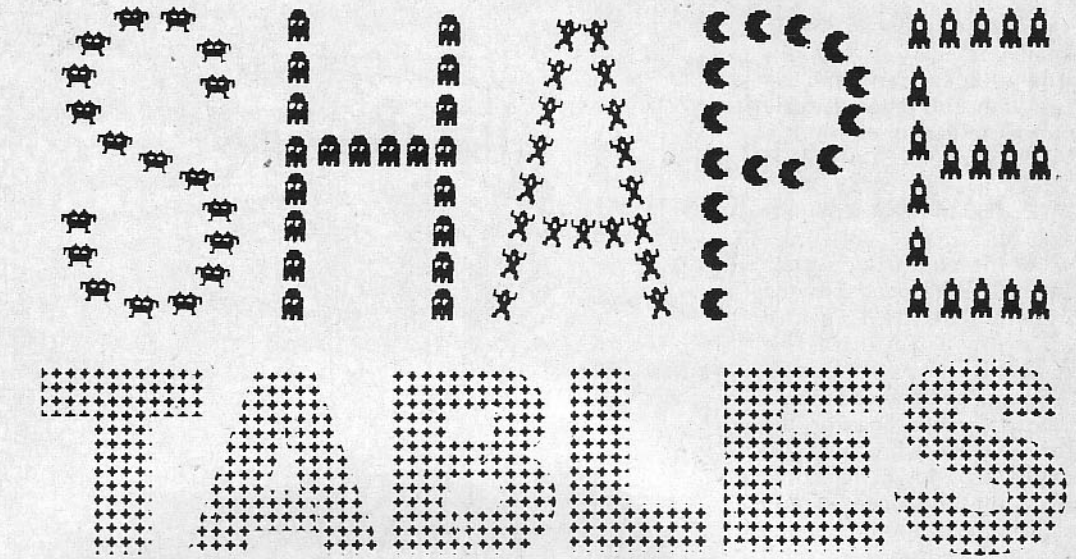
A shape table is a self contained little library of shapes. It has an index at the beginning to say where each shape starts and then a set of plotting instructions for each shape, as shown schematically in Figure 1.

Unfortunately the plotting instructions are coded in a special way which makes them very compact but quite unreadable.

It is the coding part that we shall leave till next time. This month I shall provide ready made shape tables when we need them.

Applesoft supplies several commands to manipulate shapes once you have a shape table installed. These are DRAW, XDRAW, SCALE and ROT.

DRAW does what its name suggests - it draws the shape. Since a shape table generally has more than one shape we have to tell it which one, and



Part XI of the Apple User Graphics Library shows how to achieve speedy animation from Basic

where on the screen it is to be drawn. The general form of DRAW is thus:

DRAW N AT X,Y

where N is the shape number, X is the horizontal screen position (0-279), and Y the vertical screen position (0-191).

The DRAW command will attempt to draw the shape in whatever colour you have set using HCOLOR.

I say attempt since the poor way the Apple handles colour often produces a multicoloured version instead. The best colour to use is white (HCOLOR=3).

XDRAW performs a similar task to DRAW except that it changes the colour of every screen point that the shape affects to its complementary colour.

For instance, if you XDRAW a shape onto a black background it will come out white. If the background is green it will come out blue (approximately).

The advantage in this is that if you XDRAW a shape and then XDRAW it again at the same place, you will erase the shape and restore the background the second time around. This tech-

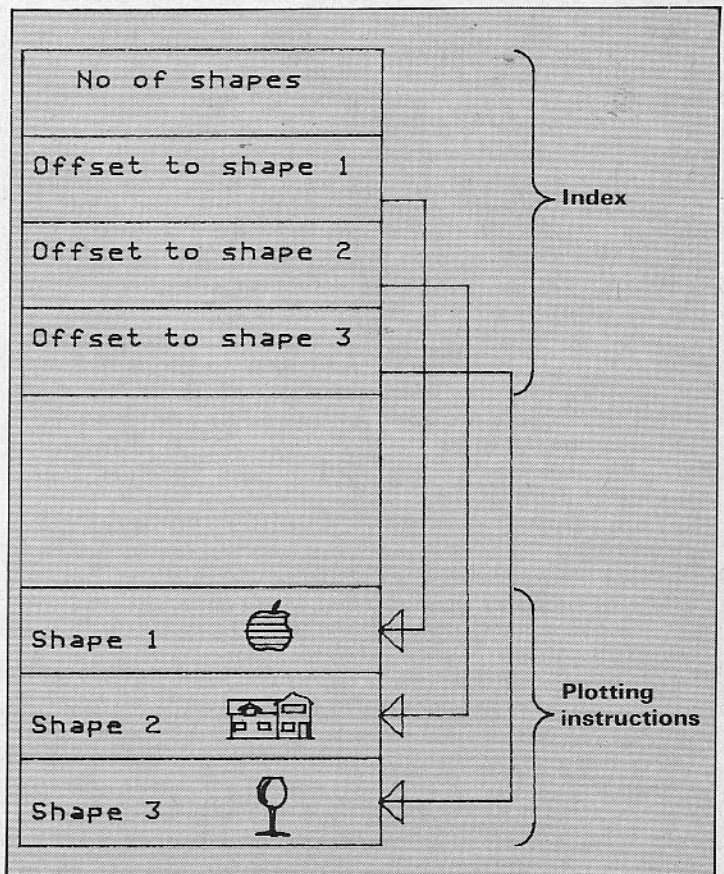


Figure 1: Shape table layout

nique is used to create simple animation effects.

The shapes can also be drawn with 256 different scale factors, selected using the SCALE command. SCALE=1 is the smallest size and SCALE=255 the largest permitted value.

A point to note is that SCALE=0 corresponds to the 256th value and is the maximum size, not the minimum.

If your shape comes out as a series of gigantic lines on the screen you have probably forgotten to set SCALE. In fact the scaling method used is very simple and often the shape ceases to resemble the original for large magnifications. More of that next month.

The shape can also be rotated using the ROT command. This rotates the shape clockwise by set amounts. ROT=0 is non-rotated, ROT=16 represents 90 degrees, ROT=32 is 180 degrees and ROT=48 is 270 degrees. By the time we reach ROT=64 we have come a full revolution.

You can't use negative values to change the direction however. If the shape is rather small it isn't possible to draw a shape with small rotations since there isn't a fine enough grid of screen points to do it. As a rough guide only four values are available for SCALE=1 (ROT=0,16,32,48), eight for SCALE=2 and so on.

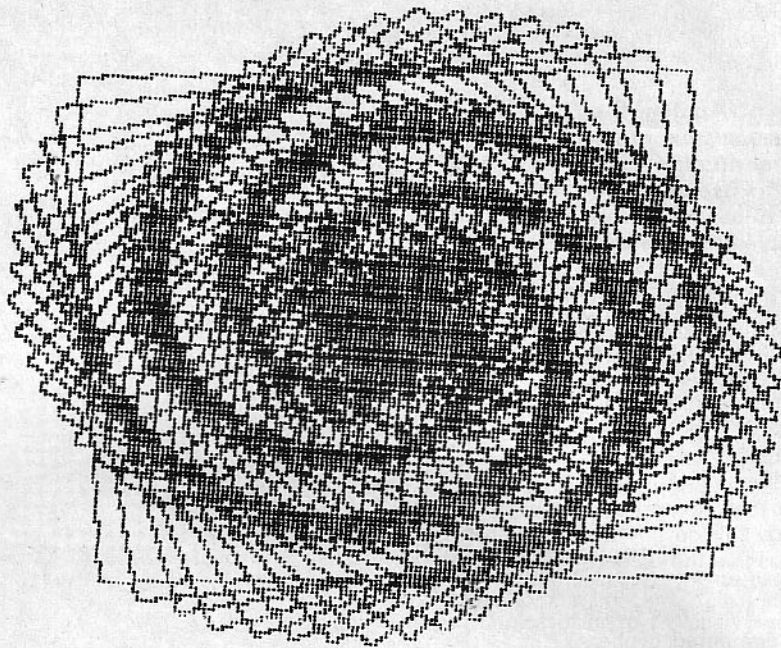
Listing 1 demonstrates how to use all of these commands. The routine at line 1000 creates a one-shape shape table and stores it in memory. Line 1010 is the table index and line 1020 is the plotting instructions - I told you they were unreadable!

The shape is a small square with a horizontal and vertical bar through the middle.

The program uses various keys to set the position, size and rotation of the shape. These are - I up, M down, J left, K right, E bigger, X smaller, D clockwise, S anticlockwise and Q quit.

Using these you can easily see how each of the commands work. The XDRAW at line 130 puts the shape on the screen and the one at line 180 erases it.

As a bonus, the Quit command creates a spiral



The special pattern produced by Listing 1

pattern by drawing the shape at various sizes and rotations. The pattern can be changed by altering the loop values in line 340, and changing DRAW to XDRAW in line 350.

We now come to an awkward point that always arises with shape tables - where to put them? The problem arises because the shape table is not a piece of Basic code like PRINT or POKE, it's just a list of numbers that only DRAW and XDRAW understand.

Because of this they can't just be typed into a program - we must find a place in memory for them. Unfortunately this isn't very straightforward since we must use a piece of memory that isn't going to be overwritten by your program or its variables.

The first 2048 memory locations of the Apple are forbidden territory and normally you would be ill advised to try and POKE things into this area. Fortunately there is one bit of this territory which lies unused, locations 768 to 975, and we can utilise it for small shape tables.

Providing the table is shorter than 207 numbers it will fit neatly into this space. The variable AD, line 1030 in Listing 1, sets the addresses to the start of this space and line

1040 POKEs the table in there.

Another solution is just to put the table a long way up in memory (AD=28000 say) and hope that the program doesn't get up that far. This isn't very satisfactory, especially if you use a lot of string variables.

A more satisfactory procedure is to set aside a safe area of memory and to tuck the shape table into there.

The simplest way to do this is to alter the Applesoft HIMEM pointer. This tells the Apple the value of the highest memory location it can use.

We can change this to a lower value than normal and put our table in the free space thus generated. This generally works well, although the DOS command MAXFILES can be deadly. The following piece of code added to Listing 1 will do the trick:

```
1030 TL = 13 :REM put the table length here
1031 AD = PEEK(116)*256+PEEK(115):REM himem
1032 AD = AD - (TL+5)
.
. line 1040 now
. put the table in
. from AD upwards
.
1045 HIMEM: AD-1
```

Those of you who have been following the *Apple User* graphics package will know that we

use a third method there.

An Applesoft program usually starts at location 2048, but we can alter this to start a bit further up in memory and put the shape table underneath the program.

A fuller description of this method was given in the March 1984 *Apple User*.

There are still other methods of creating safe hiding places for shape tables but I won't pursue them further here. What is absolutely indispensable is that we tell the Apple where we have put the table.

Memory locations 232 and 233 are reserved for that purpose and lines 1050 to 1070 in Listing 1 perform this task. The address is split up into two parts since, in general, a memory value can't be squeezed into a single location.

The routine starting at line

1000, or its modified form using HIMEM:, can be used to incorporate a shape table into

your own programs very simply.

As a reward for all the mental effort expended in reaching this point I have provided a very simple "duck shoot" program in Listing II for recreation.

It uses a two-shape shape table, again stored down at 768. The first shape is a simple disc and the second provides the debris. The "duck" is set off using game button 0 (line 150) and the shot fired using button 1 (line 190).

The speed of the duck is set randomly in line 270 using the variable TIME. A simple explosion effect is created by XDRAWing the debris at different SCALEs (lines 500-580).

The game was inspired by misspent youth and a (now defunct alas) old fashioned, relay driven version at the Grafton Arms in Manchester. I'm sorry to all those who know it well, but space doesn't allow me to do it justice and I had to cut out the multiple directions, bonus points and trophies.

Mine's a pint - and pass me the game controller.

```

100 REM SQUARE DEMO
110 HOME : GOSUB 1000: REM SHAP
    E TABLE
120 XC = 100:YC = 100:SC = 1:RO =
    0: HGR : HCOLOR= 3: POKE -
    16302,0
130 SCALE= SC: ROT= RO: XDRAW 1 AT
    XC,YC
140 X = PEEK ( - 16384): REM LO
    OK AT KEYBOARD
150 IF X < 127 THEN GOTO 140: REM
    NO KEY PRESSED
160 POKE - 16368,0: REM RESET
    KEYBOARD
170 K$ = CHR$ (X - 128)
180 XDRAW 1 AT XC,YC
190 IF K$ = "I" THEN YC = YC - 5
200 IF K$ = "M" THEN YC = YC + 5
210 IF K$ = "K" THEN XC = XC + 5
220 IF K$ = "J" THEN XC = XC - 5
230 IF K$ = "S" THEN RO = RO - 1
240 IF K$ = "D" THEN RO = RO + 1
250 IF RO < 0 THEN RO = 63
260 IF RO > 63 THEN RO = 0
270 IF K$ = "E" THEN SC = SC + 1
280 IF K$ = "X" THEN SC = SC - 1
290 IF SC > 255 THEN SC = 255
300 IF SC < 1 THEN SC = 1
310 IF K$ = "Q" THEN GOTO 330
320 GOTO 130: REM DRAW AND CARR
    Y ON
330 XC = 140:YC = 100
340 FOR I = 1 TO 63 STEP 1: ROT=
    I: SCALE= I
350 DRAW 1 AT XC,YC: NEXT
360 END
1000 REM SHAPE TABLE
1010 DATA 1,0,4,0
1020 DATA 62,36,45,54,7,160,29,
    15,0
1030 AD = 768:TL = 13: REM TABLE
    ADDRESS AND LENGTH
1040 FOR I = 1 TO TL: READ SD: POKE
    AD - 1 + I,SD: NEXT
1050 AH = INT (AD / 256)
1060 AL = AD - AH * 256
1070 POKE 232,AL: POKE 233,AH
1080 RETURN

```

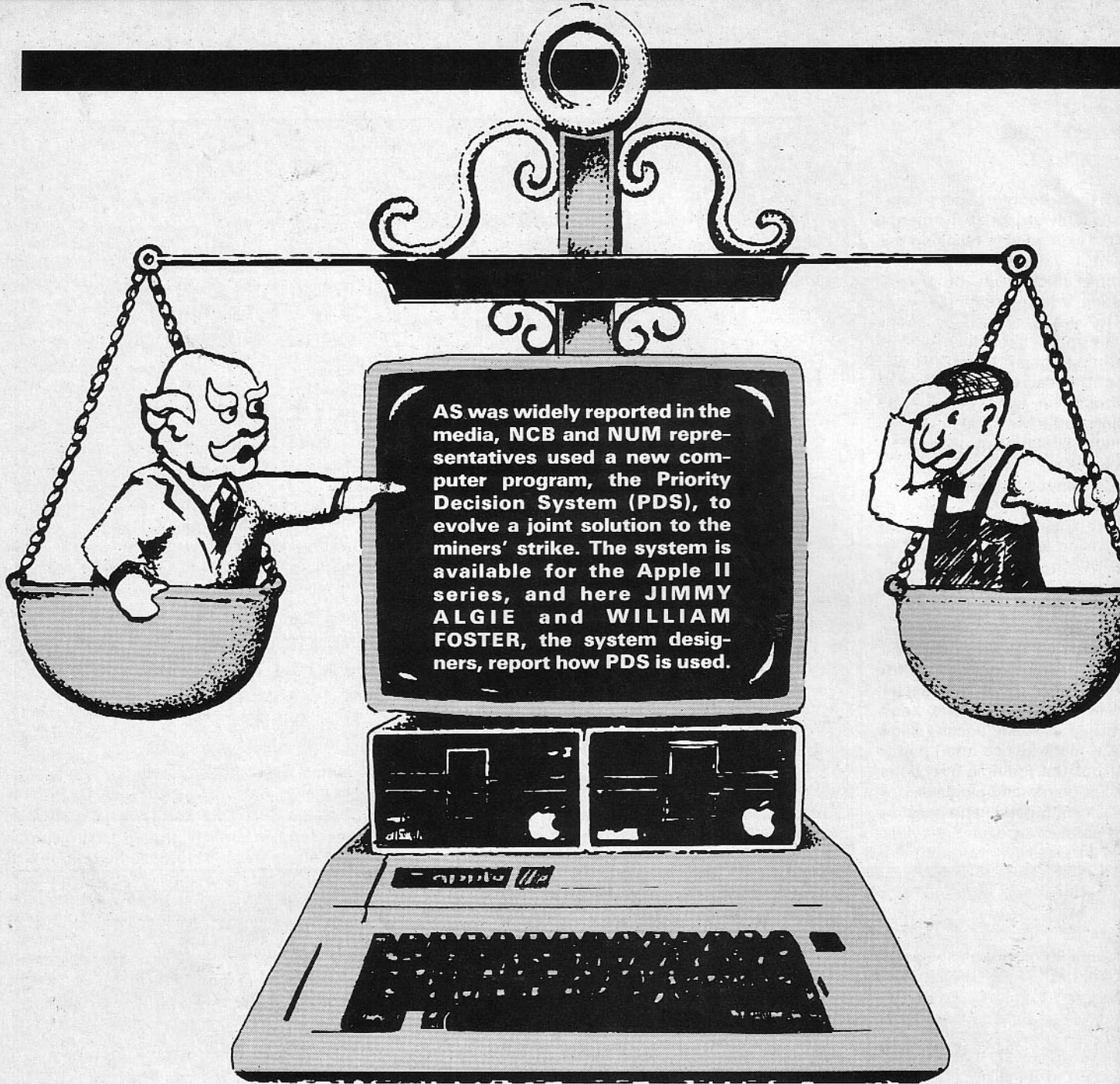
Listing I: Shape manipulation program

```

100 GOSUB 2000
110 GOSUB 1000: REM SET UP SCRE
    EN
120 REM START GAME
130 NS = NS + 1: IF NS = 11 THEN
    280
140 HTAB 20: VTAB 22: PRINT "SHO
    T NO ";NS
150 P1 = PEEK ( - 16287): IF P1 <
    128 THEN 150: REM BUTTON 0
160 REM BEGIN FLIGHT
170 FOR XP = 3 TO 276 STEP TIME
180 C = 140 - XP:YP = B + A * C *
    C: XDRAW 1 AT XP,YP
190 P2 = PEEK ( - 16286): IF P2 >
    127 THEN FF = 1: REM FIRE F
    LAG
200 XB = 79 + GC:YB = 160 - GC: HPL0T
    XB,YB
210 IF GC > 149 THEN FF = 0:SS =
    1:GC = 0
220 FOR K = 1 TO 10: NEXT
230 XDRAW 1 AT XP,YP: HCOLOR= 0:
    HPL0T XB,YB: HCOLOR= 3
240 IF YB > 25 OR YB < 11 THEN 2
    60
250 IF ABS (XB - XP) < 10 THEN
    GOSUB 500:XP = 276: GOTO 27
    0
260 IF FF = 1 AND SS = 0 THEN GC
    = GC + 10
270 NEXT XP:GC = 0:TIME = RND (
    1) * 3 + 5:SS = 0: GOTO 130
280 REM END OF GAME
290 VTAB 24: HTAB 10
300 INPUT "ANOTHER GAME ?":M$
310 IF M$ = "Y" THEN NS = 0:SC =
    0: HOME : GOTO 120
320 HOME : TEXT
330 END
500 REM A HIT
510 FOR I = 1 TO 10: SCALE= I
520 IF I = 2 OR I = 4 OR I = 6 THEN
    CALL - 19B
530 XDRAW 2 AT XP,YP: XDRAW 2 AT
    XP,YP
540 NEXT
550 SCALE= 1:FF = 0:GC = 0: HCOLOR=
    3:SC = SC + NT
560 HTAB 10: VTAB 22: PRINT "SCO
    RE=";SC
570 XB = 80:YB = 159
580 RETURN
1000 REM GRAFTON DUCK SHOOT
1010 A = 0,002:B = 10:GC = 0:D =
    21 / 36:E = 159 / 279:FF = 0
    :NT = 1:NS = 0
1020 TIME = 5: SCALE= 1: ROT= 0
1030 HOME : HGR : HCOLOR= 3:SC = 0
1040 HPL0T 0,0 TO 279,0 TO 279,1
    59 TO 0,159 TO 0,0
1050 HPL0T 80,159 TO 90,149
1055 HCOLOR= 1
1060 FOR I = 100 TO 279: HPL0T I
    ,159 TO I,159 - ABS ( SIN (
    (100 - I) / 18)) * I / 8: NEXT
1070 HCOLOR= 3: RETURN
2000 REM SHAPE TABLE
2010 DATA 2,0,6,0,16,0,18,63,32
    ,100,45,21,54,30,7,0
2020 DATA 32,198,40,159,125,42,
    207,155,125,50
2030 DATA 84,42,79,32,70,125,56
    ,29,24,40,7,0
2040 FOR I = 0 TO 37: READ D
2050 POKE 768 + I,D: NEXT
2060 POKE 232,0: POKE 233,3
2070 RETURN

```

Listing II: The duck shoot game



AS was widely reported in the media, NCB and NUM representatives used a new computer program, the Priority Decision System (PDS), to evolve a joint solution to the miners' strike. The system is available for the Apple II series, and here JIMMY ALGIE and WILLIAM FOSTER, the system designers, report how PDS is used.

Apple weighs up the decision makers

THE coal dispute appears intractable because neither side can agree on any common ground. If the new generation of computer-aided decision support systems are to live up to the claims that they can resolve virtually any problem, then they have to prove themselves on such intractable issues.

One of the new aids, the Priority Decision System has been put to such a test.

PDS, or "the Decision-maker" as users call it, is a decision support system that

does not just analyse — it actually produces decisions on industrial disputes which prove intractable to customary negotiating procedures.

Several moderate and hard-line representatives of the National Coal Board and the National Union of Miners tried out PDS. They were probably more interested in evolving their own negotiating strategies than in finding the best achievable solution. Nevertheless, they evolved through PDS a practical joint solution of the strike

without even meeting.

Messrs. Macgregor and Scargill did not participate. They could in principle just ratify the PDS solution package, or reformulate it by replacing PDS with their own views, or use it as a basis for a settlement by customary processes of negotiation.

Although the representatives claimed to be fully cognizant of the positions of the principal negotiators on every aspect of the dispute, there is a limit as to how far one person can fully

articulate the detailed views of another in PDS as in any substantial negotiations.

Both sides state all the options and views for achieving their objectives. PDS elicits each individual negotiator's priorities and arguments about the joint list of options thus produced.

They do this by means of a 10-minute judgement analysis, either interactively on program or as in this case, on forms which are then fed into the PDS.

Any options utterly una-

ceptable to both sides are eliminated—for example, “close all disputed pits immediately” and “close no disputed pit ever”. The remaining options constitute the joint solution package which is probably the least mutually unacceptable to both parties, given their views.

If the solution is rejected the process is iterated more systematically in about 30 minutes. The options are extended to include both sides’ interests and potential offers.

Each option and criterion is further spelt out so that PDS can elicit more specific priorities.

For example, PDS calculated what relative powers NCB and NUM might exercise when participating on future plans as well as a joint analysis of what could count as a “beneficial” as opposed to an “uneconomic” pit. For PDS can be used as a tool for participative decision-making as well as conflict resolution. The solution package elicited by PDS included:

- A one year moratorium on pit

closures.

- An external review of disputed pits followed by secondary arbitration.

And an in-principle agreement that:

- Some disputed pits might be closed on stricter safety/geological criteria (with job relocations).
- Some might be subject to phased closure.
- Some might be mothballed (to reopen later).
- Some might be transferred to another organisation (for example an EEC financed miners’ co-op).

New plans would be negotiated for alternative NCB investment in coal, in miners, and in mining communities, backed by Government and/or EEC as necessary.

Why can PDS succeed where old-style negotiations fail? The disputants need not meet face to face, so problems of maintaining face, of maintaining unity of your own side, and of thinking on your feet across the negotiating table do not arise.

Every aspect of each negotiator’s views is winkled out and fed into the decision algorithm. In fact their most reliable rational views are put in, since each person’s judgements are measured for reliability and logic so that inconsistencies are pinpointed.

The areas of common ground are often obscured in old-style negotiations, each side concentrating on winning the absolute position on the presenting overt issue. Whereas PDS breaks deadlocks by teasing out the component and underlying elements of each position to calculate the most mutually acceptable (or least mutually unacceptable) package which provides space for negotiation and potential agreement.

The solution which emerges is no watery compromise based on the lowest common denominator, but a genuinely joint resolution which scargillises Macgregor just enough to macgregorise Scargill.

Although PDS has suc-

cessfully been used to resolve several industrial disputes, it is proving most beneficial as a systematic general decision-making aid, applied to major organisational issues as they arise.

PDS can so improve decision-making that consequences as disastrous as the miners’ strike are prevented before they begin.

As one ICI manager who uses the system reflected: “PDS combines the insight of the insider with the objectivity of the outsider”.

Whether Messrs Macgregor and Scargill adopt the PDS package to end the strike or not, the system has proved it can tackle and resolve the major national conundrum of recent months.

Jimmy Algie and William Foster are co-directors of Brunel University’s Management Decision Programme and of Work Science Associates.

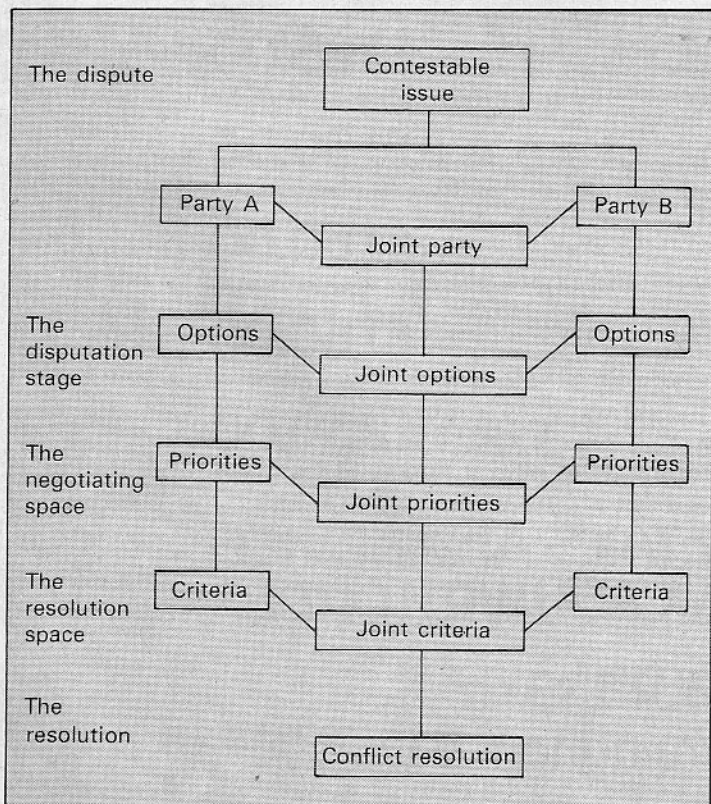


Table I: Conflict resolution, straightforward version

Issue: Resolving the miners’ strike – options.		
Criterion: All things considered.		
Method: Priority Decision System.		
Decision-makers: NCB and NUM “representatives”.		
OPTIONS		JOINT PRIORITIES
1	Additional/alternative NCB investment in coal	12.6%
2	Additional/alternative NCB investment in miners	9.5%
3	Closures now for moratorium on future closures	7.8%
4	Close some disputed pits (with job relocations)	7.7%
5	Stricter safety criteria for closures	6.9%
6	Phased closure of disputed pits over 2/5 years	6.7%
7	‘Mothball’ disputed pits	6.1%
8=	Disputed pits closures to external arbitration	6.0%
8=	1 year moratorium of pit closures	6.0%
10=	No closure now for closure agreement in principle	5.5%
10=	Messrs MacG & S delegate negotiators	5.5%
12	Joint NCB-NUM analysis of disputed pits	5.3%
13	Regional decentralisation of negotiations	5.1%
14	Transfer disputed pits to another organisation	4.8%
15	NCB diversification into other industries	4.4%
All options		100.0%
Consistency measure		0.228
Decision standard		Acceptable
Agreement measure		-0.28
Agreement standard		Significant disagreement

Table II: A joint package (the solution space)

PATRICK O'BEIRNE describes how to use the spreadsheet to calculate target values

YOU will often have heard it said that you cannot do targeting – also known as goal seeking or even backward iteration – on a spreadsheet. Well, although it is not easy with a full modelling language system, it is perfectly possible.

The usual way of using a spreadsheet is to ask: "What-if?" That is, *What* is the effect on my output figure, such as total net profit, *if* the input, such as projected sales volume, changes.

This is achieved by typing in a new input value and observing the change on the output.

Targeting refers to finding what input value will give a target output value. For example, what must our market share be to meet a 20 per cent return on investment?

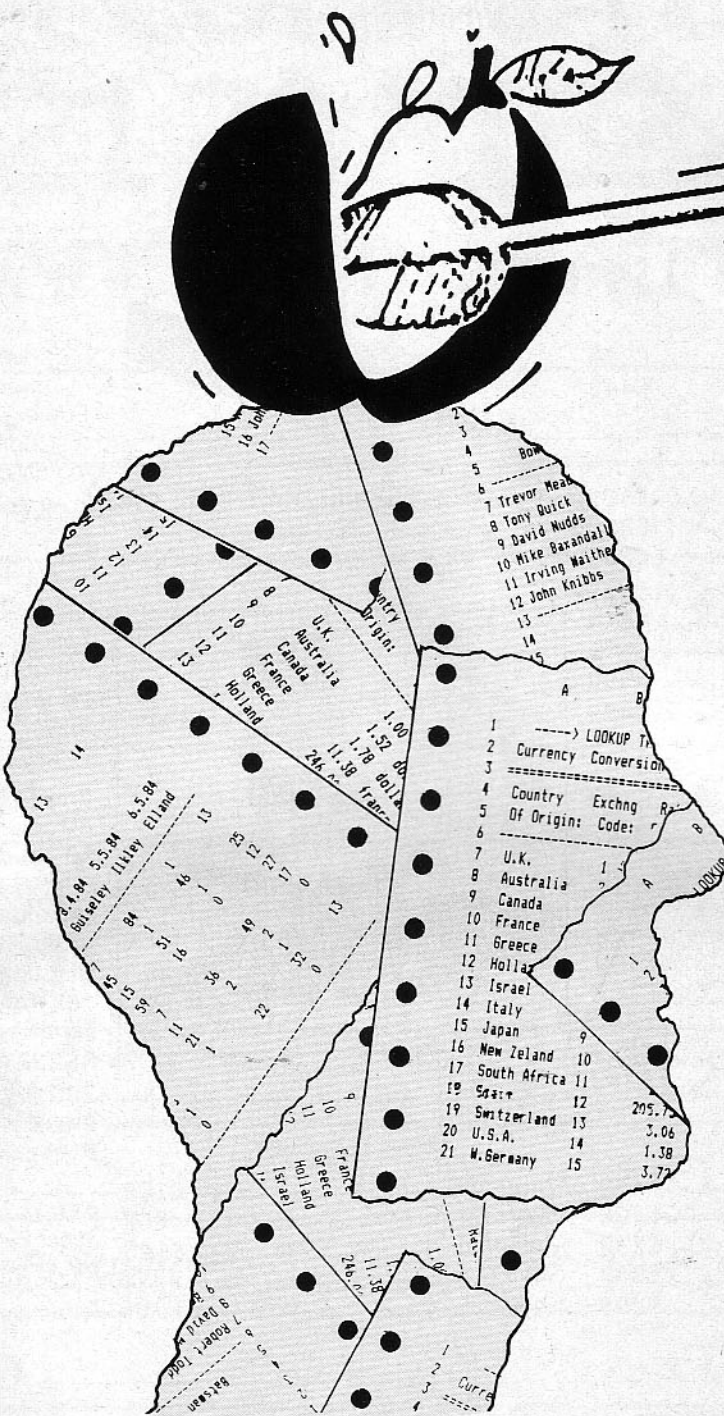
One way is to make a series of guesses and see how close you can get to the answer. The other way is to use the calculating power of the spreadsheet to work out what the input should be and so save you having to guess.

The point about doing it in keystroke terms is that it makes it independent of where you are in the worksheet. If you follow these steps, you can add targeting to any of your worksheets. Figure I shows the symbols.

Experienced users will of course recognise the Visicalc advanced version symbols. Dif-

Symbol	Meaning
^U	Up arrow
^D	Down arrow
^>	Right arrow
^<	Left arrow
^^	Caret (^)
^R	Return

Figure I: Keystroke symbols



ferent systems have different symbols.

Now, save the keystrokes I describe as keystroke sequences – or macros in 1-2-3 terms – and you can call them up any time to add a targeting calculation section into any spare column of your worksheet.

Suppose your spreadsheet doesn't have this feature. No problem. Just create a file using your favoured word processor or text editor with the appropriate control keys embedded, such as Ctrl-J for the down arrow key, and then load that up using the

/SL command. The commands on the file will be acted upon as if you had typed them in from the keyboard.

Incidentally, have you considered using your spreadsheet as a simple text editor? At least it can calculate, which very few word processing packages can. I set the column width to 19, which gives me four large columns to type in.

Look at Figure II, a simple budget projection in columns A to H. Column I is another window and was created using the method described here.

Suppose we want to find out what the direct costs percentage should be to achieve a £50,000 profit annual total.

Think of the calculation method first.

To find out what input would give a desired output, we need an equation relating output to input. As this would be very difficult in any normal spreadsheet, we can make a simple assumption that output is directly related to input and a straight line to two sets of figures. Then use that to make a prediction as to what input value should use.

In any case, a linear calculation usually finds the answer within four to 10 trials – iterations as the professional modellers call them.

We start by establishing a base case by doing one calculation. Then we change the input by a small amount – say increase it by one per cent – and observe the change in the output.

For example, we may get two sets of figures like:

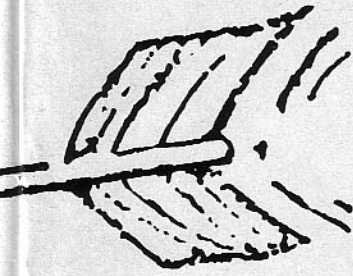
Input (cost %)	Output (profit)
Case 1 50.00	72.05
Case 2 50.50	69.72
Change 0.50	-2.32

Then calculate the equation: $Output = base + slope * Input$

The slope is the sensitivity of the output to the input. It is obtained by dividing the difference between the two output figures by the difference between the two input figures. Here, slope = $-2.321/0.50 = -4.642$.

The base figure is what the output would be if the input were zero. It is obtained by multiplying the input by the slope and subtracting that from the output. Here, the base = $69.729 - (50.50 * -4.642) = 304.15$.

So if we had zero cost



ON TARGET

percentage, we would make a £304,000 profit.

Now we can calculate what input we need to meet any particular target by subtracting the base output from the target and dividing by the slope.

Thus for a target of £50,000, we would calculate: $(50 - 304.15) / -4.642 = 54.75$. Trying an input of 54.75 does indeed give an output of £50,000 profit.

So the equation is:

$$\text{Profit} = 304.15 - 4.642 * \text{cost}\%$$

If the answer is still not close enough, just repeat the above step again. Only five keystrokes are needed to perform each iteration.

Now I know that all this looks horrendously complicated to type in but it is easier if you just step through it thinking about what each step actually means.

And, of course, if you save what you are doing as a keystroke sequence, all you have to do after testing it out is to clear the sheet by /CY— which does not clear the keystroke sequences — /SSave the worksheet, and there you have a nice macro you can call back any time to use with any worksheet.

To get this into our spreadsheet, put the cursor at cell I2 and divide the screen into two windows. This makes further iterations very easy as you will see.

I suggest /GRM to allow calculations to be commanded by the ! key, rather than every entry causing a recalculation.

The techniques used are:

□ To get the spreadsheet to show input and output from two runs of the same worksheet it is

1	A	B	C	D	E	F	G	H	I
2				Qtr 1	Qtr 2	Qtr 3	Qtr 4	Total	TARGET
3	Units '000s			100	110	121	133	464	Prev.out
4	Sales								4 0
5	Cost %	50.5							5 Prev.inp
6	Costs			51	56	61	67	234	6 0
7	O/Heads			40	40	40	40	160	7 OUTPUT
8									8 69.7295
9	Profit			10	14	20	26	70	9 INPUT
10									10 50.5
11									11 Slope
12									12 1.380782
13									13 Zero pt.
14									14 9E 10
15									15 TARGET
16									16 50
17									17 try:

Figure 11

necessary to use cells to hold previous as well as current values.

Visicalc works in from the top left corner to the bottom left corner so any cell which refers to a cell after itself in sequence will always show the last value

which the other cell had.

So by putting +I8 into I4, we get I4 to show what was the previous value in I8.

□ To get around problems caused if the worksheet is saved with these forward-referencing formulas in them and subsequently re-loaded, we never use +I8, but @IF(@ISERROR(I8),0,I8). This will show a zero if referring to I8 returns an @ERROR result.

A similar test may be done where there are self-references as in I12, or to prevent @NA values creeping in.

□ We use a window to quickly flip between the input cell of the worksheet and the cell in the targeting column that contains the value calculated to be the next guess at what value will produce the desired result.

Now there is still more extension that could be done. For example, suppose the worksheet requires more than one ! to recalculate it?

Then we will not want the previous value to be overwritten again as we will have both previous and current inputs the same and no calculation of slope will be possible.

The answer there is to add yet another test for the input values.

Now type in the following keystrokes:

```
/GRM/WV;
TARGET^D
Prev.out^D
@IF(@ISERROR(^D^D^D),0,^D^D^D)^D
Prev.inp^D
@IF(@ISERROR(^D^D^D),0,^D^D^D)^D
OUTPUT^D
/FG +; (point to Output cell) ^R
^R^DINPUT^D
/FG +; (point to Input cell) ^R
^R^DSlope^D
/FG@IF(^U^U=^U^U^U^U^U,@IF(@OR(
  @ISNA(^D^U),@ISERROR(^D^U)),0,^D^U),
  (^U^U^U-^U^U^U^U^U)/
  (^U^U-^U^U^U^U^U))^R
^R^DZero pt.^D
/FG+^U^U^U^U^U-(^U^U^U*^U^U)^R
^DTARGET^D
(Enter Target value, say 50)^R
^DTry:^D
/FG@IF(^U^U^U^U^U=0,^U^U^U^U^U^U^U
  ,(^U^U-^U^U^U^U)/(^U^U^U^U^U))^R
;f*1.01^R!
```

If the resultant output is still not on target, all you have to do is:

+;f^R!

and you get this result:

```
Cell:
Cursor is in the righthand window.
I2:TARGET
I3:Prev.out
I4:@IF(@ISERROR(I8),0,I8)
I5:Prev.inp
I6:@IF(@ISERROR(I10),0,I10)
I7:OUTPUT
I8:/FG+H9
I9:INPUT
I10:/FG+B5
I11:Slope
I12:/FG@IF(I10=I6,@IF(@OR(
  @ISNA(I12),@ISERROR(I12)),0,I12),
  (I8-I4)/
  (I10-I6))
I13:Zero pt.
I14:/FG+I8-(I10*I12)
I15:TARGET
I16:50
I17:Try:
I18:@IF(I12=0,I10,(I16-I14)/I12)
Move to left window, increase
input value by 1% and recalculate.
+ starts a value entry
; switches to the new trial value
f^R enters it and ! recalculates
again.
```




THIS protocol was developed to enable a remote Apple to access another Apple and receive and store files from it. The scheme does not require the source Apple – the one with the files on it – to know any specifications regarding the file under transfer.

The source must merely run the Hello program from drive number one and select Option 1 – send mode. The disc containing the file to be transferred should be in drive number two.

The destination Apple – the one receiving the file – runs the Hello program from its drive number one and enters Option 2 – receive mode.

The file type – Applesoft, text or binary – is then selected and

THE Qatar State Police Force uses a number of Apple IIs in their communications section to assist in planning and maintenance.

Here ROY COPPING describes a software interface they have developed which enables Applesoft, text and binary files to be transferred between Apples.

The suite of programs is too large for listings to be printed, but readers who would like them should send a blank disc and return postage to Apple User.

the required file name entered. The disc for storing the requested file should be in drive number two.

The two Apples used for testing were connected using Apple II serial interface cards

A2L0008 and RS232 cable operating at 1200 baud.

Applesoft program files

The source through Option 1 waits to receive the file type and

name from the destination. The destination, through Option 2, sends the file type (A) and name and enters the receive state to await the sender's response.

The source, via an exec file PROGXFER runs Loader which gets the file name and loads the file from drive number two.

In case the file is not found the error flag location \$342 is set to the error code value.

PROGXFER now runs BRUN RELOC.OBJO which relocates the Applesoft file from the normal start of program buffer (\$801) to a new buffer starting at \$1001.

Now the main buffer is free to take the next controlling program, PXFER. This sends the

FILE TRANSFER

end of program pointers of the requested file to the destination.

In case the error flag \$342 is set to error code 6, a file absent code (NN) is sent instead. PXFER now enters a delay loop to enable the destination to ready itself to receive the main program.

Through the interface program the destination receives the end of program pointers and computes the program size to be received. Interface then enters the batch input mode waiting for the Applesoft file.

If file absent flag is received from the source then the destination exits from the transfer mode.

The source now transfers the file via a batch output routine and exits from the transfer mode.

The destination receives the file from the source in locations \$1000 onwards. This program runs an exec file 'PROGEXEC'. PROGEXEC Bruns XFER which transfers the received file to the normal program buffer area starting at \$801.

XFER also initialises the locations \$69 to \$6E and \$AF to \$B0 to the program buffer end address. PROGEXEC then executes the save command to store the received file on drive number two as GUEST FILE.

Finally PROGEXEC Bruns the change routine to change the file name to the original name, available in locations \$3A0 to \$3BD. The transfer is now complete with the Applesoft file residing on disc number two of the destination Apple.

Text file transfer

As in the case of program files, the source receives the file type (T) and the name from the destination. The source now runs the TEXTXFER program which initially Bloods TXFER.OBJO.

The latter searches for the file name and reads the first 32 data sectors – or less if a file is smaller. In case the file is absent, the error flag \$302 is set to Ascii(N).

A flag indicating the number of sectors or file absent is sent to the destination which is waiting after having opened the

file on its drive number two. If the file is absent, the source exits from the transfer mode and the destination follows suit after deleting the opened file.

Through the text program, the destination computes the block size and enters the batch input mode. The source, after a timing delay to enable the destination to ready itself, outputs the data block and waits for a ready flag from the destination.

The destination receives the data block and scans through it, grouping the data into fields written into the opened file. The FREEZE routine is called periodically to free the redefined strings.

If the file is complete – known from the flag from source – the destination closes the file and exits from the transfer mode. Otherwise, a ready flag is sent to the source and the transfer process is repeated until the file is complete.

Binary file transfer

As mentioned earlier, the source receives the file type(B) and name from the destination and runs the BINXFER program. BXFER.OBJO, Blooded by BINXFER, searches for the requested file and reads the first data sector which contains the RAM image start address and length of the file.

If the file is absent, error flag location \$302 is set to Ascii(N). A flag indicating the start address and length of file or file absent is sent to the waiting destination.

If the file is absent, the source exits from the transfer mode, otherwise the binary file is Blooded at locations \$1000 onwards.

The destination receives the flag from the source and exits

from the transfer mode if the file is absent. Otherwise the destination computes the binary file size and waits for the file transfer from the source. The source now outputs the file and exits from the transfer mode.

The destination B Saves the file at \$1000 onwards in its original name on drive number two and Bruns BFRAM.OBJO.

This routine searches for the file – file name available at locations \$3A0 to \$3BD – and rewrites the first two bytes of the first data sector to show the original RAM image start address thereby completing the transfer process.

In the case of text files, sequential or random access files are both written as sequential files in the destination Apple. The integrity of the file is nonetheless maintained since the DOS relies only on the L (length of record) parameter to define random access files.

This scheme delivers the files to the destination as a true copy of the files in the source.

The transfer of program and binary files is of course limited by the size of the available user memory in the Apples. However, a text file may be of any length since the transfer is in blocks of 8k.

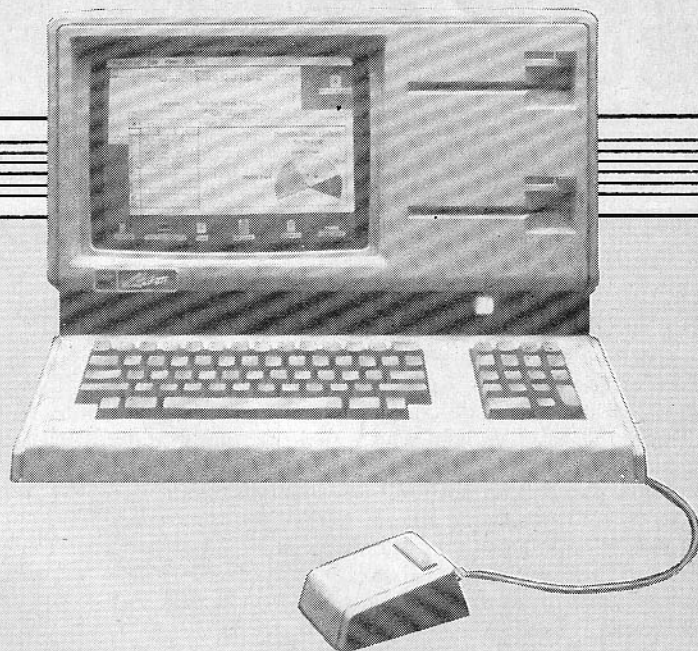
At present this scheme requires the source to run the transfer utility prior to start of the process.

The system can run as a limited Star network with one standard interface card for each other micro connected – for four micros, each would require three interface cards fitted.

Finally, I must place the main credit for this work on two of our staff, W/O Ramandahan and Sgt. Roy Misra, who produced these programs under my direction.

File Type	Number of Sectors	Time for Transfer (seconds)
Applesoft program files	28	105
Text files	48	360
Binary files	34	115

Table 1: Results of tests conducted at 1200 baud with various types of files



THE use of powerful micros in the preparation of complex chemical formulae and their handling by cut and paste techniques is being developed as a spin-off from the work of the Emergency Planning Department of the North West Regional Health Authority. CHARLES FAIRFAX and his colleague KATHRYN CARR demonstrate how Lisa has already achieved this and – how its stablemate Macintosh has yet even more to offer.

No inaccuracy, no frustration, no gimmicks!

By CHARLES FAIRFAX
and KATHRYN CARR

AS our experience with Lisa has matured, the preparation in advance of blocks of information for subsequent transfer from one document to another using cut and paste has increased. It has rapidly become one of the most useful methods employed.

It has many varied and interesting applications not only in emergency planning but also in allied fields.

Colleagues in other professions, seeing the method in operation, have been stimulated to discuss its application in their own work. This in turn has led to several worthwhile spin-offs.

Such a development resulted from discussions which entailed the use of the chemical formulae of a range of related steroids. These are substances of considerable importance in medicine including cholesterol, cortisone, sex hormones and the pill, to name but a few.

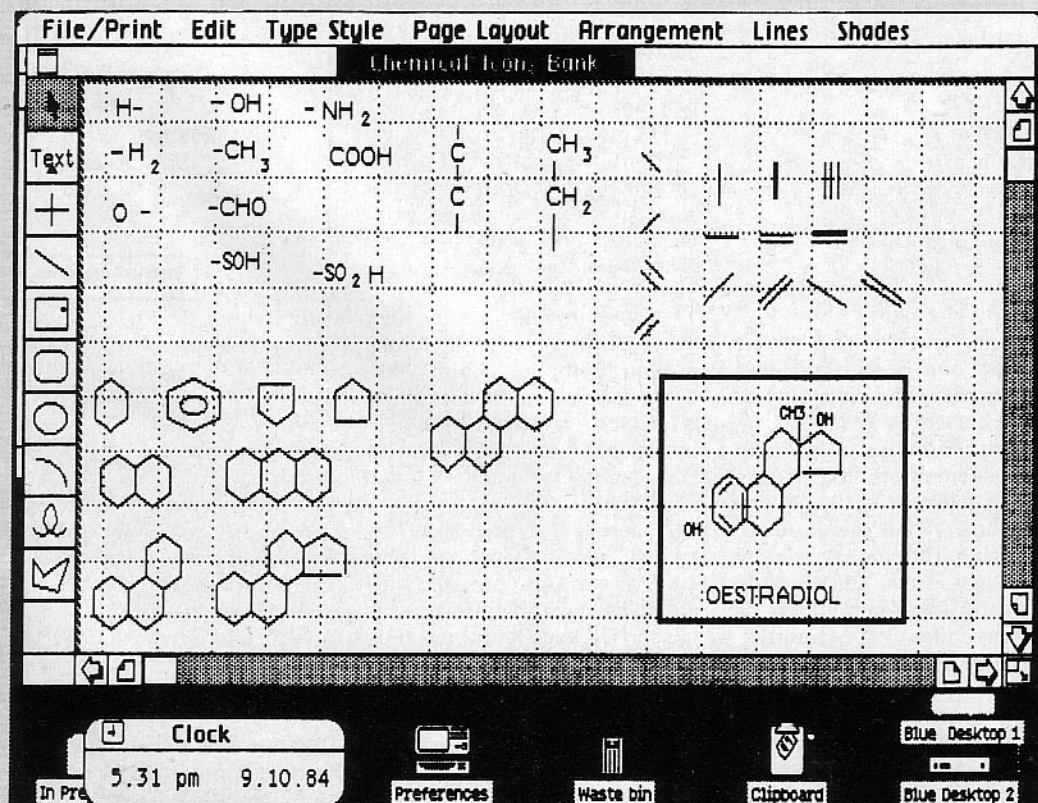
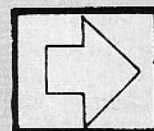
One of the co-authors, Kathryn Carr, had been doing research into the use of the element palladium in the synthesis of various steroids and in their subsequent chemical reactions. Having obtained results of wide interest and importance, next came the task of writing this up and publishing it.

Although one is almost ashamed to have to say this, speed is of the essence since the flagrant poaching of the work of others is not the isolated prerogative of the computer world.

Anyone who has had to write many organic chemical formulae or watched students at work on their theses can hardly have failed to notice the tedium of tracing awkward shapes with fine point pens and tracing tools, or the painstaking work of the stenographer who has to construct a presentable transcript.

Those who have to read the finished work will now appreciate how, with the best will in the world, inadvertent errors can still creep in from faults in the accuracy of transcription.

This is where the computer

comes in. With its total accuracy and speed of operation, especially when programmed in user friendly form, it can transform the scene.

It is the tired worker who makes mistakes. Micros ease the workload and do not tire.

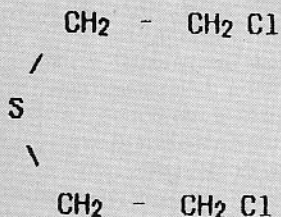
Once cut or copied, a stored formula is reproduced time and time again with infallible accuracy.

Those formulae required often are best stored on a dedicated document – in effect a miniature purpose designed icon bank. With a little thoughtful planning, complex formulae can be built up from a selection of smaller modules.

One can put into the store the symbols for individual elements, standard groups, chemical bonds and even some finished formulae that, because of their particular structure, lend themselves as a starting point from which by alteration, addition and subtraction further formulae can emerge.

A chemical icon bank should be a dynamic tool to be used to build up structures, not just a static library of finished formulae ready for use.

A surprisingly wide range can be achieved just using LisaWrite. The formula to which we first applied this method, using LisaWrite alone, was dichlorethyl sulphide, commonly known as mustard gas.

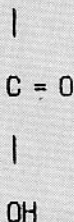


It is a simple but delightful pleasure to see a formula like this flip up at will amidst a page of normal type at a mere touch on Apple-V. Perhaps even more delight may be experienced when one comes to edit a straightforward LisaWrite document and insert chemical formulae.

One uses the Search facility. Apple-W opens the search dialogue box and enables a space to be identified where the

insertion is to go, for example: **carboxyl group** which is then entered under Find.

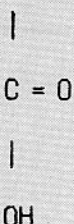
The additional material, in this case the formula COOH is entered under Change To. The expanded form:



can be included in the icon bank.

Apple-F finds the next occurrence highlighting it on the screen. COOH may be entered using Change This Occurrence from the menu box. If one decides at this stage to go ahead with more amendments Apple-N takes care of that, inserting the change when next due.

The menu box allows single changes or the same change to be made throughout the document. Complex expanded formulae can be put in using cut and paste. For example our COOH can become:



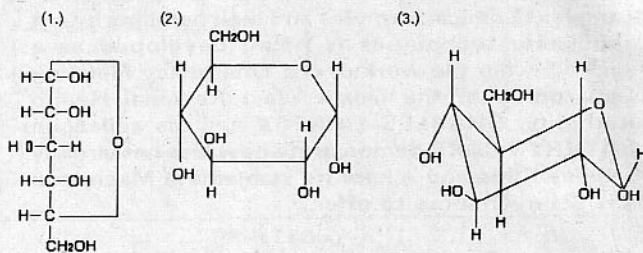
pasted in from the icon bank.

When one is satisfied with the alterations the dialogue box can be closed by Apple-H.

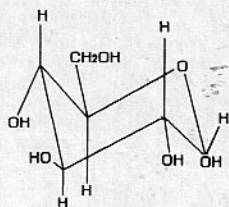
As an alternative to the Apple Keypress method, the mouse can be used directly on the menu box. When one is tuned into using the mouse almost exclusively, it can seem a little strange to issue commands by keypress codes. It does however ease the flow when one is typing.

It is not necessary to memorise all the key codes – they are included for reference in the menu boxes. This technique

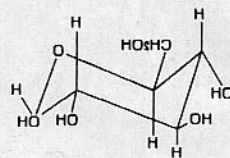
Alpha-D-glucopyranose



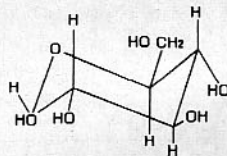
Three different ways of showing the chemical structural formula



Stage 1 - Grouped and copied



Stage 2 - Flipped horizontally



Stage 3 - Ungrouped
letters flipped horizontally
whole regrouped

Construction stages to produce an enantiomer

allows a great deal of symbol handling to be worked out with LisaWrite alone.

For more complex work – say that entailing polycyclic hydrocarbons – LisaDraw is better suited. One then uses both tools, pasting both text and formulae onto a LisaDraw document.

In the middle of the work being done on Lisa, Macintosh arrived complete with MacWrite and MacPaint in their issued form and a good range of pre-release American system software tools which included MacDraw.

These had, to coin a phrase, to be "UKised" using the Mac UK System. Left untreated they would cause problems since the US and UK keyboards have their characters set out in different ways.

Who could resist the temptation to produce personalised versions? MacWar and WarPaint emerged. We leave

1-2 mg. per day
given by mouth
1-5 mg. by i.m.l.

IMPLANT;
25-50 mg.

Test of wrap round insertion

Example of composite document

readers to guess who thought of those.

Life became easier at once. Formulae could now be turned round sideways, upside down and into mirror images with complete abandon.

All we regretted was that subscripts and superscripts were not quite so easy as on LisaWrite. The flip over and rotation software took care of most of the difficult bits that had previously been a little tedious

on Lisa.

What might appear to the uninformed to be a graphics software sales gimmick suited only to computer games, in experienced professional hands revealed itself in true perspective as a powerful tool for use in a field where previous methods were fraught with difficulty, tedium, inaccuracy and frustration.

We can now hardly wait for the consequent Lisa software upgrade.

Quite unexpectedly the rub-out facility was to prove its worth in tailoring difficult shapes. Instead of freehand drawing of ring forms, boxes could be made and their ends simply erased.

Inverted Vs for lids and

As we became more and more adventurous with our new technique we tackled more and more exotic applications. Stereo-isomers seemed obvious, folded rings yielded with a little coaxing and thoughts were given to the use of Duplicate to manufacture long chains.

Even quite complicated structures such as protein chains no longer seemed such a mammoth task. However we did not lose sight of the fact that its best use was to do all the simple things well.

Thinking now of the wide practical use to which this method may be put, it offers something for all. In a simple elementary form it has an obvious place in the chemistry classroom. At the other end of the scale, the professional chemist may wish to construct his own set of symbols in his own personalised icon bank.

Should these prove too complicated and a little beyond him, help could initially be sought to set this up. The difficult part over, its use is then simplicity itself.

● To keep abreast of software developments is rather like running to jump onto a moving train. As this article drew ready for publication our new Lisa 7.7 software arrived.

Anticipating our high hopes the new LisaDraw incorporates Flip Horizontal and Flip Vertical and lots of other goodies.

The complicated carbohydrate formulae yielded even more readily to Lisa than it had done to Macintosh. If required, with 2,000 blocks allocated to MacWorks the Macintosh software could be used.

The spatial aspects of molecules could now be demonstrated in a flash at the touch of the mouse or a touch on the keyboard, with extended pasting for good measure.

Dr Charles Fairfax is a consultant with the North Western Regional Health Authority. Miss Kathryn Carr is a post-graduate research worker in the University of Manchester Chemistry Department.

OESTRADIOL

Symptoms, primary amenorrhoea
Ethinylloestradiol

Course

21 days with 7 day interval
before the next course

cut onto a cut and paste of graphic

these techniques

flipped over lids as bottoms could be slipped into place and the final form grouped. Very soon it was realised that it is a distinct advantage to work on a scaled grid. It is then so much easier to get the sizes to match.

Another little tip that may not be known too well is to select None from the shade menu box which facilitates the drawing of double and triple bonds by avoiding the merging of close parallel lines.

CHEMICAL ICONS BANK

H- -OH - NH₂
- H₂ -CH₃ COOH
O - -CHO
-SOH -SO₂H

$\begin{array}{c} | \\ -C- \\ | \end{array}$
 $\begin{array}{c} CH_3 \\ | \\ -CH_2- \end{array}$

ESTRADIOL

ESTRONE

3-METHOXY-ESTRONE

Lisawrite: chemical icon bank - principal groups

GROUP	Suffix	Prefix
-COOH	carboxylic acid	carboxyl
$\begin{array}{c} O \\ // \\ (-C) \\ \backslash \\ OH \end{array}$	oic acid	
-CO.OAlk	alkyl...carboxylate	alkoxycarbonyl
$\begin{array}{c} O \\ // \\ (-C) \\ \backslash \\ OAlk \end{array}$	alkyl...oate	
-SO ₃ H	sulphonic acid	sulpho
-SO ₂ H	sulphinic acid	sulphino
-S.OH	sulphenic acid	
-CO.Hal	carbonyl halide	halogenocarbonyl
$\begin{array}{c} O \\ // \\ (-C) \\ \backslash \\ Hal \end{array}$	oyl halide	
-CO.NH ₂	carboxamide	carbamoyl

The race is on to read all about it

THE race is on! "What race?" do I hear you cry. Why, the race to fill your bookshelves with books about the Macintosh of course.

The problem with rushing to publish books on new machines is that it's usually difficult to find anything interesting to say. After all, it's how the machines are *used* that's really interesting, and nobody's had much chance to do anything particularly wonderful yet.

Undeterred, the publishers continue to bombard editorial offices with their own offerings. Of the two I'm about to describe, one is a US import and the other is a home-grown product.

The import is *The Apple Macintosh Primer* by William B. Sanders. Its purpose is twofold - to acquaint Mac owners with their machine, and to describe



the machine for those considering a purchase.

If you fall into the first category, forget it. You won't learn anything from this book that you can't get from the manuals which came with your machine. From this point of view, it's a typical attempt to jump on a bandwagon.

If you're considering a purchase then the book has a little more going for it. You could read it in an evening and get a reasonable introduction to the

desktop, MacPaint and MacWrite.

Then again, with the test-drive promotion, you could actually be trying the machine itself rather than reading about it. You'd get more software to look at that way, too.

The book is written in laid back, "easy to understand" language which I must confess to finding a little irritating. It's the sort of style that you can imagine somebody saying, but it doesn't quite convert to print.

The complete lack of an index suggests it's only meant to be read once and then discarded. Maybe this is a little unfair because the contents pages are sufficiently detailed for the page numbers to increment in steps of one most of the time. Short sections are the order of the day.

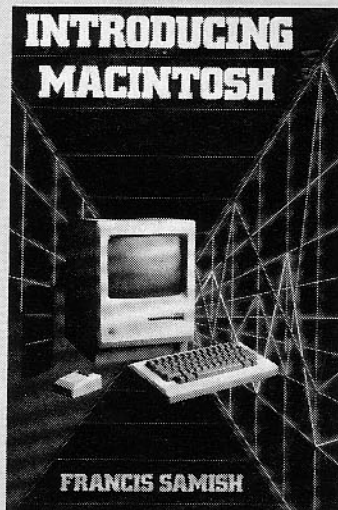
The home-grown offering is *Introducing Macintosh* by Francis Samish. It scores over the import in several ways.

For a start, it's got an index, it has more pages and it's cheaper. The content is a little better too.

For example, as well as the obligatory MacPaint and MacWrite sections there are chapters devoted to Microsoft's Multiplan and File, and Apple's own MacProject and MacTerminal.

This provides a much better view of the potential uses to which a businessman might put the machine than the simple tour around Paint/Write.

There's also an appendix on "Useful software to consider" which briefly describes Microsoft's Chart and Word, Apple's own MacDraw, Peachtree's MacAccounting, and Rodent Software's Personal Appointments Diary.



This is all very well, but the rate at which Mac software is appearing combined with the lead time for publishing means that such sections are out of date before they appear in print.

The description of the software is still valid, but it can't give a comprehensive view of what can be done now.

Although Francis Samish obviously had a fair bit of help from Apple UK, even they didn't know at the time that Phil Peters would be sent to America charged with the task of bringing back Mac software.

So what do I expect from a book on the Macintosh?

Given that the machine is a closed box there's not much

scope for the "let's stick bits in" brigade - the "hobbiests" as Sanders calls them on more than one occasion.

Hence, there's very little alternative to talking about the software. However, I would find it more useful to read a review of software rather than just a description.

Alternatively, a comprehensive catalogue of available software would be useful, but it's too soon to attempt one of those.

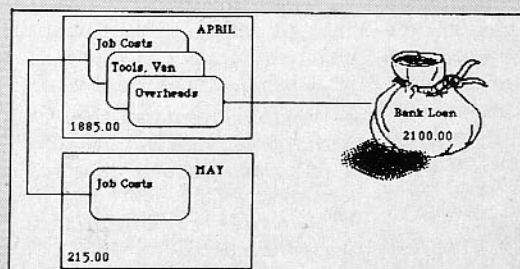
Of the books I've seen on the Mac so far, I'd choose Samish's *Introducing Macintosh* if I was forced to pick one. However, I have to say that I wish publishers weren't so quick off the mark.

I suspect that many of them will find the bandwagon piled high with remaindered descriptions of MacWrite/Paint, and that's one of the factors pushing the price of books up out of all proportion to their worth.

Graham Price

Title: The Apple Macintosh Primer
Author: William B. Sanders
Publisher: Datamost/Reston
Price: £9.65

Title: Introducing Macintosh
Author: Francis Samish
Publisher: Granada/Collins
Price: £7.95



Where the loan goes... an illustration from Introducing Macintosh

WE'VE all accepted spreadsheets, word processors and databases — now the latest buzzword is "mindware".

Human Edge Software has produced a range of packages in this field claimed to improve management performance. Two were supplied for review, Sales Edge and Management Edge.

Although I tested Macintosh versions, they are also available for the Apple II range.

Each consists of four basic sections — the program which enables the user to produce the right strategy with data disc; the user manual for running the program; a fundamentals book of general, management or sales strategies and a set of worksheets for keeping records of the contacts involved.

While I like to get on with producing a practical review of a program starting by loading a disc and switching on the micro, these two packages needed a far more serious and detailed approach.

Both programs recommend the user start not with inserting a disc into their Mac, but instead by reading the book of generalised strategies — sales in one case and management in the other.

Only when that has been read — and presumably understood — is the second stage started.

Again no contact with the program directly but instead in the case of the Sales Edge a worksheet is produced on one of the prospects or customers.

Only on completion of the worksheet is the program used.

The Human Edge package has two discs — one for the program and the other for data, known as the output disc.

The program discs cannot be copied but back-ups are available from the supplier.

After switching on the Mac and inserting the Sales Edge program, Open is selected from the Finder Menu. A reversed-out icon showing two people shaking hands together indicates that this is the Sales Edge.

The first screen tells you to assess yourself from the Self menu. At this stage instead of keeping to the manual, just take the About The Sales Edge

It's all in the mindware

NEVILLE ASH tests two practical programs for improving sales and management performance

option and information is available describing what has to be done. It's a form of on-line Help facility.

Under the heading of assessing yourself there are 86 different statements to which you have to give the answer Agree or Disagree.

And when you have ploughed through this lot, it's the turn of the customer. First create the file with the customer's initials and whether they are male or female.

Then go through 50 different adjectives which may be used to describe this customer. Name the file, remove the program disc, insert the data disc and save this information.

This isn't a once-only operation. At any time this information can be updated. Load the program, select the file by the name, load the data disc and off you go.

The product of all these questions and adjectives is a report which can be used for action when you meet that particular person.

In the case of the Management Edge, instead of customers the people involved are likely to be your subordinates or even the boss.

Here there are even more statements in the beginning — 92 in all. When it comes to assessing a subordinate there are 89 different adjectives or

statements. And then the file is stored in the same way as the Sales Edge.

In using any of these packages the most important thing of all isn't the program but the user. Enter false information and there will be a completely useless result.

But try to answer with some form of honesty as far as possible and the results could be far more useful.

But these programs have one major difference from most packages used in Apple computers.

Use a spreadsheet and figures are entered, straight data is used in a database and words in a word processor.

But in this case we are going into the realms of opinions, an area where virtually everything is completely subjective.

So the value of any form of package can only be as effective as the people who treat the software seriously.

Although these packages have sold extremely well in the United States, there they tend to take on anything new with great enthusiasm.

Here we tend to be conservative with a small c, and fairly resistant to any form of dramatic change. So we have a case of attitude rather than quality and usefulness of the program being more important.

From a personal point of view

I tried the package as a freelance writer having to deal with an editor. My problem wasn't in entering the information, but in knowing with any degree of certainty enough to be able to produce sensible Agree or Disagree responses to the adjectives.

This highlighted the importance of gathering as detailed information as possible, as the programs recommend, before trying anything out.

My initial results weren't too informative, but when I spent some time compiling information, then the result was quite good.

I feel that the intended user — a company person on the sales side or management, would be interested if this type of approach was already used to some degree in the company.

Similar types of system, usually with fancy names, have been in operation manually in a number of companies, with a similar type of end result.

All the micro does is process them far quicker and make it possible for more people to take advantage of the technology. Nothing else.

There are four Human Edge programs. The two I didn't try were Negotiation Edge and Communications Edge.

From the end user's point of view these packages can bring their Mac into a far wider range of work activities.

I certainly found that the Mac adaptation had been done efficiently, so that using the mouse appeared to be the only way to use the program.

Overall for the company which is into evaluations, the Human Edge range of programs offers another way of creating a more logical approach to management, sales, communications and negotiations.

In areas where opinions are used instead of facts and figures, the end result should make face-to-face contact far more effective.

Title: *The Sales Edge, The Management Edge.*
Price: £195 each.
Supplier: *Thorn EMI Software.*



WE have already examined the Rehaflex board and the Flex operating system so now let's look into the OS-9 operating system and the Pascal Speedup Kit. Each must be bought separately.

OS-9 from Microware Systems is completely different from Flex or DOS 3.3.

While working under Flex is so similar to working under DOS 3.3 that reading the manual can be postponed, you will not get away with this under OS-9.

Since an attempt to describe OS-9 in detail in one article is doomed, discussion here will be restricted by and large to those OS-9 features not shared by DOS 3.3 and Flex.

OS-9 is a Unix-like operating system, obviously developed with implementations on large systems in mind as its capabilities are very impressive.

On a small system like the Apple one is bound to meet memory restrictions, but nevertheless many features work flawlessly.

The OS-9 (version V1) disc I received for testing the Rehaflex board was clearly meant for use on an Apple. However, I had to make several patches in the 6809/6502 communications software to adapt the cursor setting to the 80 column Basis 108.

Furthermore, another patch was needed to prevent a second set of head retracks upon booting and I had to move my Integer Basic firmware card from slot 5 to another slot since the 6502 startup routine enabled this card for some reason.

These modifications are not difficult, but they indicate that the OS-9 communications software runs less smoothly than Flex's.

When the OS-9 disc is booted, the memory size of the system is first determined. If you have a 16k RAM card 147 pages (37k) of free memory are available, otherwise you have 107 pages.

Working under OS-9 requires a quite different way of thinking. In the "normal" Apple environment you usually load and execute programs one at a time. This is not so under OS-9, where it is no problem to have

OS-9...THE system for going beyond conventional software

CORNELIS BONGERS concludes his exploration of the 6809

several programs in memory at the same time executing concurrently.

Supervision is by an advanced management system which allocates memory, CPU time and I/O resources to each of the programs.

I/O is highly unified, meaning that data transfers to all I/O devices are done in the same way. Each I/O device has a unique name. For example, Term is the console terminal, P is the printer and D0 is drive 0.

This implies that OS-9 commands can be used in many ways. For instance, the COPY command can be used to copy files to the console, to disc and to the printer with:

respectively.

A filename with the corresponding drive parameter - /D1/GREETINGS - is called a pathname in OS-9.

The reason for formalising this aspect is that OS-9 supports multiple hierarchical directories, so filenames alone don't necessarily define a file - a single filename may occur in several directories.

A new directory can be created with the MAKDIR command. After creating a new directory, files can be stored in it by mentioning the directory in the pathname. For example, MAKDIR /D1/BASDIR creates a new directory BASDIR on the disc in drive 1 and SAVE

COPY GREETINGS /TERM
COPY GREETINGS /D1/GROT
COPY GREETINGS /P

Displays file GREETINGS on screen
Copies file GREETINGS to the file GROT
Prints GREETINGS (without line feeds)

The route taken by the data between a program and a file or printer is called an I/O path. The parameters of the COPY command above are thus two I/O paths designating the source and the destination of the data

/D1/BASDIR/MARJO saves the file MARJO in BASDIR.

Note that on the DIR command only the directory file BASDIR is shown and not the file MARJO. If you want to see the files of BASDIR you have to

specify DIR BASDIR.

You may also log into a certain directory with the CHD command. For example, CHD /D1/BASDIR makes BASDIR the working directory so that when Basic is active all Basic programs are automatically saved in BASDIR.

Although the advantages of hierarchical directories may not be apparent at first sight I find it very convenient to suppress - on the DIR command - many pages of filenames I am not interested in by creating multiple directories with mutually related files.

Input and output may also be redirected to other files or devices. The implementation of redirected I/O is very general. For example, if a program normally sends its output to the screen the output can be redirected to the printer by simply specifying >/P behind the pathname.

An OS-9 concept more difficult to grasp is the SHELL - a command interpreter that translates user input to tasks for the internal OS-9 routines.

Organisation of the SHELL is far more sophisticated than the command interpreter part of DOS 3.3. Apart from driving the loading and execution of files SHELL also allocates working RAM memory to the file.

Default values of working memory, specified in the file's header, may be overruled by the user. For example, COPY#10K /D1/ABC /D0/PQR allocates 10k to the copy process thereby overruling the default specification.

As already mentioned under OS-9 it is no problem to have several files in memory at the same time. With the LOAD command, for instance, you can load BASIC09 and some frequently used commands, such

as DIR and COPY, as with Flex all operating system command drivers reside on disc.

The difference between loading and executing a program or routine is that in the latter case the memory allocated is freed

upon termination.

When a file is loaded first and executed next it stays in memory, so subsequent invocations involve no disc access.

One of the reasons the feature of co-resident files is hard to realise in a 6502 environment is that it is difficult to write 6502 position independent and reentrant code.

Code written for use with OS-9 is position independent and reentrant, so the SHELL can put it anywhere in memory.

The SHELL itself is also reentrant, which implies that it can be used simultaneously by several programs and this in turn opens the way to multitasking, which I consider one of the most fascinating features of OS-9.

Multitasking is initiated by appending an & to a command line - when you enter LIST /D1/TEXT >/P&, the SHELL starts the LIST command and assigns the process an ID number.

Next the prompt reappears and you can enter other commands while the file TEXT is printed. In principle you can execute as many programs as you want simultaneously but on the Apple the practical limit will be three or four.

It is no problem to let a software clock display the time, print a file and run two Basic programs simultaneously, but scheduling more jobs with one keyboard and one screen becomes confusing.

The printer spooling mechanism under OS-9 works much

better than under Flex. When entering text at the keyboard you never lose characters and the printer keeps going all the time.

Even when a concurrent job requires disc access you don't have to stop typing because OS-9 maintains a keyboard buffer that is hard to overrun.

Some OS-9 commands with a short description are contained in Table 1.

Unlike Flex, OS-9 or the communications software does not support any 6502 interaction. The only way to enter the 6502 monitor is to hit Reset.

The documentation didn't mention an OS-9 warmstart address so reentering OS-9 is only possible by partially rebooting the system with 801G.

This implies that exiting a hang-up program with Reset (which happened several times) means a total loss of everything entered between the last SAVE operation and the system hangup.

Unfortunately the quality of OS-9 software I presently have available is far more impressive than the quantity.

On the OS-9 system disc there are only a few utilities such as EPSON1 to install the printer driver, NOCLK to install a software clock, a copy program like COPYA and some others which are undocumented.

What *is* present, however, is BASIC09, the best Basic interpreter I have seen so far. It combines the best features of Pascal and Basic while many of

ATTR	Gives the attributes of a file.
BUILD	To create a textfile.
COPY	To copy files.
DEL	To delete files.
DUMP	To dump a file in hex/Ascii.
EX	Executes a program and discards SHELL.
FREE	Gives free space on disc.
KILL	Aborts process with specified ID number.
MDIR	Displays modules currently resident in memory.
MFREE	Displays free memory areas.
PRINTERR	Initiates output of error messages (rather than numbers) if an error occurs.
PROCS	Displays a list of existing processes.
SETIME	Initiates clock.
SLEEP	Suspends execution of a program for a certain time.
SETPR	Set priority for a process.
UNLINK	Free memory occupied by a module.

Table 1. Description of OS-9 commands.

```

PROCEDURE PERMNEW
0000 REM < GENERATE PERMUTATIONS FROM 1 TO N >
0026
0027 BASE 0
0029 DIM PERM(20),I,J,T,K,N,COUNT:INTEGER
004D
004E INPUT "ENTER NUMBER OF OBJECTS ".N\PRINT
0070
0071 REM INITIALIZATION
0082
0083 FOR I=0 TO N
0094     PERM(I)=I
00A0 NEXT I
00AB COUNT=0
00B2
00B3 REPEAT
00B5
00B6     REM PRINT PERMUTATION
00CA
00CB     COUNT=COUNT+1
00D6     PRINT USING "15>".COUNT;\ PRINT " ";
00EC     FOR I=1 TO N
00FD         PRINT USING "13>"; PERM(I);
010D     NEXT I
0118     PRINT
011A
011B     REM GENERATE NEXT PERMUTATION
0136
0137     I=N-1
0142     WHILE PERM(I)>=PERM(I+1) DO
0158         I=I-1
0163     ENDWHILE
0167     J=N
016F     WHILE PERM(J)<=PERM(I) DO
0182         J=J-1
018D     ENDWHILE
0191     T=PERM(I) \ PERM(I)=PERM(J) \ PERM(J)=T
01B7
01B8     REM SORT LAST PART OF PERMUTATION ARRAY
01DE
01DF     J=N \ K=I+1
01F2     WHILE K<J DO
01FF         T=PERM(K) \ PERM(K)=PERM(J) \ PERM(J)=T
0225         J=J-1 \ K=K+1
023B     ENDWHILE
023F
0240 UNTIL I=0
024B
024C END
    
```

```

ENTER NUMBER OF OBJECTS 3
1      1  2  3
2      1  3  2
3      2  1  3
4      2  3  1
5      3  1  2
6      3  2  1
    
```

Figure 1.

the shortcomings of both languages are absent.

Most of the control structures of Pascal are implemented in BASIC09. Available are IF/THEN/ELSE/ENDIF, REPEAT/UNTIL, WHILE/DO/ENDWHILE and LOOP/ENDLOOP.

The latter structure generates a continuous loop from which you can exit with the EXITIF statement. An example is presented in Figure 1.

This program displays the permutations of the numbers 1 to N in ascending order and uses several of the control structures mentioned above.

As can be seen, the listing is nicely formatted, which makes reading BASIC09 programs a far more pleasant task than reading Applesoft ones.

Note also that the program contains no line numbers. These are optional in BASIC09 and can be omitted if you manage to write a GOTO-less program.

BASIC09 supports five different datatypes – byte, integer, real, string and boolean. The default type is real (four byte mantissa and one byte exponent). Byte, integer and boolean variables must be explicitly dimensioned with a DIM statement – for example DIM FLAG :BOOLEAN.

String variables on the other hand, may be either dimensioned with type string or be implicitly declared to be of type string by appending a "\$" to the name of the (string)variable.

All this is nice enough, but the real surprise is that all the data types mentioned above can be combined in new user defined (complex) data types.

With the TYPE instruction, PL/1-like structures can be created and be handled as an entity. For example, the instruction TYPE CUSTOMERS = NAME:STRING[20]; ADDRESS :STRING[20]; IDNO :INTEGER; AMOUNT:REAL defines a structure where several elements have a different type.

After defining the TYPE CUSTOMERS, we may create an array of customers and a working structure with DIM CUSTARRAY(250), CURCUS:CUSTOMERS.

The elements of the array CUSTARRAY can be accessed by specifying the array element

```
PROCEDURE SETUP
0000 REM PROCEDURE SETUP: GET PARAMETER VALUES AND CALL HANOI
0037 DIM N; SOURCE; SPARE; DESTINATION: INTEGER
004A N=3 \REM NUMBER OF DISKS
0064 SOURCE=1 \REM SOURCE ROD
0079 DEST=3 \REM DESTINATION ROD
0093 SPARE=6-SOURCE-DEST
00A3 RUN HANOI (N,SOURCE,DEST,SPARE)
00BC END

PROCEDURE HANOI
0000 REM PROCEDURE HANOI:MOVE N DISKS FROM SOURCE TO DEST ROD
0037 PARAM N; SOURCE; DEST; SPARE: INTEGER
004A IF N>1 THEN RUN HANOI (N-1,SOURCE,SPARE,DEST) \ ENDIF
0074 PRINT "MOVE DISK FROM ROD "; SOURCE; " TO ROD "; DEST
009E IF N>1 THEN RUN HANOI (N-1,SPARE,DEST,SOURCE) \ ENDIF
00C8 END

MOVE DISK FROM ROD 1. TO ROD 3.
MOVE DISK FROM ROD 1. TO ROD 2.
MOVE DISK FROM ROD 3. TO ROD 2.
MOVE DISK FROM ROD 1. TO ROD 3.
MOVE DISK FROM ROD 2. TO ROD 1.
MOVE DISK FROM ROD 2. TO ROD 3.
MOVE DISK FROM ROD 1. TO ROD 3.
```

Figure 11

followed by the structure element. For example, CUSTARRAY(13).NAME="EVA" assigns the name EVA to customer 13.

An entire array of structures or a single structure may be read from or written to disc with one instruction. The instruction PUT #OUT, CUSTARRAY for instance writes the entire CUSTARRAY to disc.

There is also a convenient statement to determine the length in bytes of a structure. This is the SIZE statement and in combination with the SEEK statement structures may easily be retrieved from disc.

For example, to read the N th structure of CUSTARRAY, we may specify SEEK #OUT, (N-1)*SIZE(CURCUS) which sets the pointer of the next record to be read in to the N th structure.

Execution of GET #OUT, CURCUS then loads the structure in CURCUS. Apart from structure orientated disc instructions, BASIC09 supports structure assignments. The statement CURCUS=CUSTARRAY(13) assigns the structure

associated with customer 13 to CURCUS.

A limitation in using structures is that they cannot be printed just by specifying the name of the structure. Instead all structure elements must be specified individually in the PRINT statement.

An additional limitation is that arrays of structures, as well as normal arrays, cannot be dynamically dimensioned.

So, as with Pascal, you have to guess what the maximum required dimension at run time will be and dimension the array accordingly.

Another matter well worth mentioning is that BASIC09 is, unlike Applesoft, a procedure-oriented language. This means that you may split up your Basic programs in logical parts and call these from a main program. Setting up programs in this way usually improves readability and makes updating easier.

With the LOAD command you can load a number of BASIC09 programs in memory. Each of the loaded programs

can be accessed with the editor (see below) or be executed independently.

When one of the programs is executed, control can be passed to another program with the RUN instruction.

Variables – and arrays or data structures – can be passed to the called procedure by reference by specifying them in a list after the RUN keyword.

The receiving procedure links the addresses of the passed variables to the variable names mentioned in the PARAM statement. If variables are passed by reference their values may be changed by the receiving procedure so results of computations and so on can be passed back to the main program.

Constants and expressions may also be specified in the RUN instruction, but for obvious reasons these are passed by value rather than by reference. Since passing by value is supported, recursion is possible in BASIC09.

The program in Figure 11 illustrates this by printing the

solution to the Towers of Hanoi problem.

The environment in which BASIC09 programs are entered, edited and executed is also different from Applesoft. Rather than two operation modes – immediate and deferred – there are four – system, edit, execution and debug modes.

When BASIC09 is invoked you find yourself in system mode. Here OS-9 commands may be issued and Basic programs loaded or deleted. To enter a Basic program the letter E must be typed, which invokes the editor. Fortunately this involves no disc access, as the editor forms an integral part of BASIC09. The editor is not screen-orientated and is therefore less easy to operate than the Applesoft line editor.

A nice feature is that some syntax checking is done when a statement is entered. If a keyword is wrongly spelled you are informed immediately.

When you quit edit mode, an overall check of the program is carried out. At this check, GOTOs to non-existing line numbers and multiple DIM statements of the same variable are detected.

When you are back in system mode after leaving the editor the program may be executed by entering RUN. The system then goes in execution mode. If a STOP or END statement is encountered, control goes back to system mode.

If an error occurs in execution mode, or a PAUSE statement is executed or Ctrl-C is pressed, you enter debug mode.

In this mode you may print the values of variables, assign new values to variables, enable or disable tracing or display the status of the program, such as the nesting order of all active procedures.

Execution can be continued at the next statement by entering CONT, but it is also possible to STEP a specified number of steps (lines) through the program.

As with Applesoft, real numbers occupy five bytes, but in processing reals BASIC09 is much faster than Applesoft.

For example, the loop FOR I=1 TO 10000/NEXT I takes only 8.7sec (10.5sec in

Applesoft). If an integer loop variable is used, execution time reduces to 1.8sec.

The time required to initialise a large array to unity with FOR I=1 TO 2000/A(I)=1/NEXT I (I is integer) is 2.3 seconds, about four times faster than Applesoft.

The permutation program in Figure 1 requires 8 seconds to generate 6! permutations (without printing), whereas a similar program in Applesoft takes 49 seconds. Multiplying real variables is also extremely fast in BASIC09. For example, the program:

```
DIM I:INTEGER
B=123456789
FOR I=1 TO 1000
/A=B*B/NEXT I
```

takes 5.3 seconds in Applesoft, 10.5 seconds in XBasic but only 1.4 seconds in BASIC09.

A program which ran slower in BASIC09 was the computation of 1000 SIN functions in a loop. This took 33 seconds whereas Applesoft needed only 27 seconds.

According to the BASIC09 manual execution speed of programs can be increased by PACKing them. The process removes, among other things, names and line numbers.

Packed programs cannot be edited however, so before a program is packed it must be saved on disc in source form. I tested the PACK feature with a program to calculate 400 primes (supplied on the OS-9 disc) and noticed a 10 per cent reduction in the code but no significant speed improvement. The permutation program reduced 30 per cent in size but didn't run faster either.

I suppose packing a BASIC09 program has about the same effect as optimising an Applesoft program with the Optimiser – little time gain but usually considerable memory savings.

The precision of BASIC09 is definitely less than that of Applesoft. For example, computation of 2^X gives an error in the final decimal for $X=9$ and $X>11$, that is, $2^9 = 512.000001$. Furthermore, the SIN function and the multiply operation sometimes give an incorrect final decimal.

Another less attractive point of BASIC09 is that it is not

bug-free. For example, on execution of the statements INPUT A/PRINT A, you get the output :.0E36 when 1E37 is entered.

Less innocent however is the irrecoverable system hangup that occurs if you enter 4942348234 on the INPUT A statement.

The default amount of free memory allocated to BASIC09 is 2.8k. This can be increased to slightly more than 14k if you invoke BASIC09 with BASIC09 #15k.

When you are prepared to discard the SHELL an additional 1k can be squeezed out of the system by entering EX BASIC09 #16k. However, in that case you cannot specify OS-9 commands anymore from BASIC09 system mode.

I would not recommend OS-9 to readers who have just started with computers, but I am sure that anyone who has a reasonable knowledge of the Apple and its software will appreciate OS-9 as much as I did.

There is a considerable difference between Flex and OS-9. While Flex provides an excellent environment for developing software, OS-9 is the operating system for those who want to take a look beyond the conventional software.

Relative to Flex, OS-9 is generally less easy to work with and the probability of a system crash is definitely higher.

Nevertheless, despite these critical observations, I find OS-9 the more powerful, and more fascinating, operating system and I deeply regret that my OS-9 software is limited to BASIC09.

The Pascal Speedup Kit from Stellation II is a software package designed to speed up your Pascal programs with the help of the 6809.

Before it can be used a new SYSTEM.APPLE must be generated. When this file is substituted for the SYSTEM.APPLE file on the APPLE1:disc, the 6809 will do all the P-code interpretation.

The Speedup Kit runs without problems with the Rehaflex board, except for the fact that the proper memory map must be selected before the new APPLE1: is booted.

This inconvenience can be easily eliminated by patching the code on track \$0, sector \$0 of APPLE1:. It is sufficient to replace the JMP (\$FFF8) instruction at \$A9 (relative) by LDA #20/STA \$C300/JMP (\$FFF8). To my knowledge this has no side effects.

Programs involving FP calculations will not benefit from the Speedup Kit since for real arithmetic the 6502 is used. A separate FP package is available which speeds up FP arithmetic.

In the case of integer arithmetic there is a substantial improvement. For example, multiplication of two integers is more than twice as fast with the Speedup Kit. However the speed improvement in the average program will be much less. For example the Pascal version of the permutation program – see Figure 1 – needed 7.6 seconds to generate 61 permutations without the Speedup Kit installed and 5.3 seconds with it, implying a speed improvement of 30 per cent.

Those readers who on the basis of this figure are tempted to conclude that Pascal is a fast language because it realised the best time with the permutation program should keep in mind that they are comparing a Pascal program with uncompiled Basic programs.

A compiled version of the permutation program, written in MBasic, due to lack of a BASIC09 compiler, generated the 61 permutations in less than a second.

If you do a lot of work in Pascal and are prepared to buy an 6809 board I would surely recommend the Speedup Kit. You don't have to worry about compatibility problems with programs written under the 6502 for after installation everything works the same as before – except some things go faster.

OS-9: Microware Systems Corporation, 5835 Grand Avenue, Des Moines, Iowa 50312. Tel: (515) 279-8844.
Pascal Speedup Kit: Stellation II, The Lobero Bldg., P.O. Box 2342, Santa Barbara, CA 93120.



IN 1982 a group of colleagues and I at Salford University decided to use our experience with the Apple II to run a series of short courses on the machine.

After two or three of these, it seemed a natural extension to use the extensive manuals we had written for the courses as the basis for two books on the Apple.

The courses are "The Apple for Beginners", "Getting More from your Apple" and "Machine Code Programming on the Apple". Addison-Wesley agreed to publish a book combining the manuals from the first two courses under the title "Getting the Most from your Apple II/IIe".

They decided that the machine code manual clashed with a book they were already publishing but Prentice-Hall International were happy to accept it for publication as "Machine Level Programming on the Apple II/IIe".

A key feature making the task of adapting the manuals into books feasible was that they were originally prepared on the Applewriter word processor. In both cases this feature was utilised by the publishers, though in rather different ways.

Prentice-Hall wanted us to provide camera-ready copy which I could print directly from Applewriter textfiles using a daisy wheel printer.

Addison-Wesley's proposal was even more interesting. They wanted to use the word processor textfiles to be interfaced directly to a computer operated typesetter.

This involved incorporating a lot of typesetter codes for such purposes as change of type style, italics, new lines, indents and setting up columns for tables and so on.

This made me realise how different printer output is from typesetting, where letters and characters have widely varying spacing so you can't just line up indents or columns by counting characters and spaces.

Figure I shows a section of the original manuscript, and Figure II shows the same material after coding.

The machine code book was the easier of the two since the manual was already fairly close to the form needed for a book. Some changes were needed of course. Frequent references to 'manual' had to be changed to

The book starts here

By GRAHAM KEELER

'book', and 'section' to 'chapter'.

The references to course demonstrators needed some substantial changes. Where on the course we could help with any points which gave difficulty, in the book such topics needed explaining very carefully.

Other changes needed were mainly by way of extra material – particularly solutions to the

problems – and an index.

A further problem specific to the machine code book was the question of assemblers and mini-assemblers. On the course we had permission from Apple to use their Toolkit assembler, but in the book we had to assume that some readers would have other assemblers such as the more recent and

powerful Merlin.

The solution we used was to make the course material on using the Toolkit assembler a much more general discussion on assembly language.

It needed much more re-organisation and modification to graft the other two manuals together and remove the overlapping material to produce a single, logically structured book.

For instance, together the two manuals dealt with disc drives three times – this had to be reduced to one introductory and one advanced chapter. The graphics sections – at somewhat different levels in the two books – also had to be combined.

In both cases, the preparation took a lot more effort than I expected.

It was necessary to check very carefully for complete

Figure I

```
[h3](d)[en]SCALE = [it]a[ro][3x]
[t1]Restriction:[i1]0 <= [it]a[ro] <= 255[ix]]
]]
[sk]Action:[em][cc17,l]a = 2[cc17,l][em][em][cc18,l]two points drawn for
each plotting vector[cc18,l][kx]
[sk]Action:[em]a = 2[em][em]two points drawn for[cc19,r].[cc19,r][kx]
Action:[c17][it]a[ro] = 1[c18]one point drawn for each plotting vector[[
+6]
[c17][it]a[ro] = 2[c18]two points drawn for each plotting vector[[
+6]
[c19].[[
[c19].[[
+6]
[c17][it]a[ro] = [it]n[ro][c18]n points drawn for each plotting vector[[
[cr]
]]
N.B. SCALE=0 gives the maximum size, as if [it]a[ro] had the value 256.
Since 0 is the default value for SCALE and this is unlikely to be what
you want, SCALE should always be set before using DRAW.
[A further practical restriction on the value given to SCALE is that
the shape must be smaller than the screen size. Note also that the
action of SCALE is not, as might be expected, simply to plot the shape
on a larger scale. The effect is much more complicated, as explained
below, and makes the SCALE command of limited use.]]
```

Figure II

```
(d) SCALE = a

Restriction:  0 <= a <= 255

Action:  a = 1    one point drawn for each plotting vector
        a = 2    two points drawn for each plotting vector
        .
        .
        a = n    n points drawn for each plotting vector

N.B.  SCALE=0 gives the maximum size, as if a had the value 256. Since
0 is the default value for SCALE and this is unlikely to be what you
want, SCALE should always be set before using DRAW.
A further practical restriction on the value given to SCALE is that
the shape must be smaller than the screen size. Note also that the
action of SCALE is not, as might be expected, simply to plot the shape
on a larger scale. The effect is much more complicated, as explained
below, and makes the SCALE command of limited use.
```


APPLICATION

accuracy, adequate and clear explanations, and to try and incorporate every important point and tip as the manuals were written for short courses where only the main points had to be stressed.

Solutions had to be written for all the problems, and for the machine code book some utility programs had to be written.

A program was needed to read the assembly language textfiles for readers without an assembler, and another to extract the mini-assembler from the Integer Basic file on the System Master disc.

The curse of using a word processor is that you can never say the job is finished because when you think of something else that ought to be included it is so simple to add it in.

This was probably the most time consuming feature of all but hopefully it will be reflected in the completeness of the finished books.

Associated with each book is a supplementary disc contain-

ing about 100 programs and text files, and again each program had to be checked for accuracy and satisfactory operation in all circumstances, and finally generally polished up.

Many of the programs are listed in the books and the fact that Applewriter II uses textfiles made this simple. Each program

other hand produced a really professional appearance. For example, a different type face looking rather like a dot matrix output but very sharp and clear gave an attractive appearance to the program listings.

The general programming book for Addison-Wesley was prepared a little later than the

supplied by the publishers suggest a complicated process involving entry of every item on a separate index card, then sorting the hundreds of resulting cards and collecting common references onto a new set of cards.

But with the aid of the invaluable word processor I entered every item directly, moving the cursor each time to the correct place on the index.

At least proof reading was simple since the computer typesetter faithfully reproduced every character as written. The camera ready copy of course needed no proof reading at all.

In spite of my experience, I have been unable to resist the temptation to have another go. I am now writing a book for Prentice-Hall on ProDOS, Apple's new disc operating system.

I had previously discussed a book on disc and file handling with them and the arrival of ProDOS made the book too good an opportunity to miss.

Proof reading was simple since the computer typesetter faithfully reproduced every character as written

was 'captured' into a textfile using the techniques explained in the DOS manual and then simply incorporated into the word processor text.

The contrast between the final production of the two books was interesting. Camera ready copy had the advantage that I had complete and exact control over the layout of pages and could decide exactly where to split each page.

Computer typesetting on the

other, and during the last stages the Apple IIc was released.

It seemed an attractive possibility to incorporate it in the book and so a rapid extra effort was needed to check out the machine, add all the differences and machine description into the book, and find out whether any of the programs would give any difficulties - none did.

Just when all the work seems to be completed an index is needed. The author's guides

512 x 512 GRAPHICS

DIGISOLVE's Apple II High Resolution Vector Graphics Processor card quadruples the screen resolution and saves using up your Apple's memory. The enhanced Applesoft support disc supplied with our card provides the powerful graphic functions and simplifies the conversion of existing software.

512 x 512 Pixels resolution **FAST** 1,500,000 pixels per second **MEMORY** An extra 64K on the card **TEXT** To 85 characters by 57 lines **CURSOR** Drawing features **SOFTWARE** 18 new functions are added to Applesoft and extra utilities too, all with source listing **OUTPUT** To a dot matrix printer or save images on a disc **PASCAL** and **TASC** Compilers supported too.

£399.00 + P&P & VAT

SOFTWARE PACKAGES: Art and Design, Business graphics, Slide generation, printing packages, Architectural design and modelling and Visicalc preboot.

COLOUR VGP64

DIGISOLVE offer you high resolution colour graphics too. Designed to plug into any compiler with a bus host adapter. The VGP64 gives you 512 x 512 pixels in 64 colours. With its own vector processor and 384K bytes of memory, your computer can become a sophisticated graphics system.

64 COLOURS or 8 if you insist, not many do! **4096 COLOURS** now you can really paint a picture **384K BYTES OF RAM** or 768KBytes with 4096 colours displayed simultaneously **2 PICTURE BUFFERS** we give not 1 but 2 pictures stored to help with animation **GREY SCALE OUTPUT** **EXT VIDEO SYNC LOCK** available for making video tapes **APPLE, SAGE, PET, IBM, S100, VME BUS, MULTIBUS** all have interfaces available to make use of our fast hardware. New ones are coming along all the time so give us a ring if your requirement is not listed.

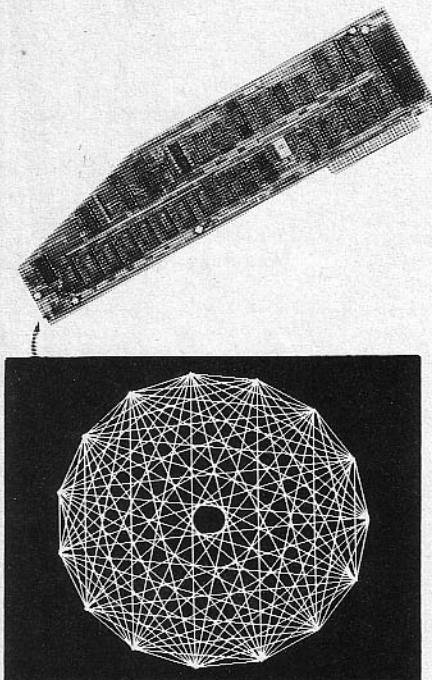
NEW UNIVERSAL RS232C/PARALLEL INTELLIGENT CONTROLLER CARD

8 COLOURS £899.00, 64 COLOURS £999.00

4096 COLOURS £2000.00

(Prices do not include interface)

SOFTWARE PACKAGES: Painting and Slide generation, Business graphics, Architectural 3D design.



Digisolve

AIRE & CALDER WORKS
CINDER LANE, CASTLEFORD
WEST YORKSHIRE
TELEPHONE: (0977) 513141 or 513382
TELEX: 557661 AGRAM
A MEMBER OF THE LAMBSON GROUP

ProDOS has lost a useful command

HAVING recently received the *Basic Programming with ProDOS manual I* have come across a glaring omission in both the ProDOS itself and the manual which may be of interest to your readers.

The offending item is the loss of the DOS 3.3 VERIFY command. ProDOS has no equivalent command for the verification of a file's integrity or existence on disc.

Although it was a little-used command as a check of a file's integrity, it was exceptionally useful in determining a file's existence on disc.

If used in conjunction with the ONERR command a programmer could check if a file was already on disc if it was required by the main program or alternatively, if writing out a data file, then the check could be made to avoid overwriting a previous file of the same name.

The manual does not mention VERIFY either in the improved DOS command list or deleted command list — or anywhere else for that matter.

The problem of checking for a file on disc only really becomes important with text files as the OPEN command opens one if it does not exist.

Why Apple has left this function of the OPEN command in the system is a mystery when it also gives you a CREATE command which could ade-

quately cope with opening a new text file and leave the OPEN command to try and open an existing file.

One way around this would be to try and execute a type-specific command on the text file so that if it was not on disc then a PATH NOT FOUND message would result.

If it did exist then a FILE TYPE MISMATCH message would be output. For example, an attempt to LOAD a text file would result in one of the above errors, although it hardly makes for good understandable programming. — **Allan Ogg, University of Glasgow.**

Replicate is the answer

I WAS glad to read your article about *Advanced Version Visicalc bugs* in the September *Apple User*.

I currently store 17 columns by 90 rows using a DIF format and use Visicalc for management consolidations.

I have only experienced your

problem when trying to save a DIF file when an @t label formula is contained within the saved block. This appears to corrupt the data saved in the same format you suggested.

This can be bypassed by using the replicate command, as in my case the labels are the same for all files stored. — **D.K. Major, Thorn EMI, Dover, Kent.**

Strategic thoughts

I READ the November *Cliff's Column* with some interest. For what it may be worth *Ye Woodland Outlaw* would like to throw in his two pence worth.

The price is a factor but not a prohibitive one if the game in question is worth the money. Today people want more for the money that they now have to spend on each title, at least as far as the Apple is concerned.

Computerised jigsaw puzzles — oh sorry, adventure games as most people prefer to call them — are probably worth the money in that it takes time to complete

them, which is five or six months on average.

Though having said that I do know of a couple of people who can complete an Infocom title in less than seven days.

The biggest snag with these games is that once completed they are hardly likely to be played again. I have tried several, but they are not my cup of tea.

Arcade games are a ripoff at the prices now asked for them. I don't cotton to them and never have, but some people like them enough to waste money on them.

The number we have sold in the last six months can be counted on the finger of one hand which is a good indication that we should scrap them. But we won't do so, at least not yet.

Some of the fantasy role playing games are under priced when one considers the amount of playing time and fun one gets out of them. I am thinking in particular of *Wizardry*.

It took me a good six months to complete the first scenario and it didn't end there, as you can play the game over and over again with the same characters or new ones, unlike computerised jigsaw puzzles.

The *Ultima* series is about right as far as price goes. There are a couple of dozey titles, but you get good and bad in everything.

The sport and simulation games are a mixed bag. Some like *Flight Simulator II*, *Computer Quarterback* (if you follow American football) and even *Hi-Res Computer Golf II* are worth looking at. As far as the latter is concerned I detest the game itself, but the simulation is excellent.

Which brings me onto the war and strategy titles. Again a very mixed bag. Some are downright awful and not worth a light, while yet others are superb.

The best are the most expensive, but unlike some other games the war and strategy ones are played over

A doctor in the house?

I HAVE just renewed my subscription to the *Apple User* for 1985 after having thoroughly enjoyed reading the journal for the past year.

I am the happy owner of an Apple II+ computer which in this country hasn't met the success it deserves, apparently because of a high and unreasonable price — at least compared to British and US prices.

Hence my attempts to get in touch with other Apple users from my own profession and speciality hasn't met any success here in Denmark.

I wonder whether you are aware of any Apple User Group of physicians, especially

anaesthesia and intensive therapy physicians or whether you will possibly put in a note in a forthcoming issue of the journal asking such persons or groups to get in touch with me, mainly for exchange of software, but also exchange of ideas.

I also wonder if you can refer me to a dealer in the United Kingdom who is able to supply Comal-80 for my Apple.

This is the computer language now standard in all Danish schools, but so far none of the Apple dealers here with whom I have been in contact has been able to obtain any information regarding this lan-

guage.

My Apple is provided with a CP/M card and I foresee no problems using the language should it be available. — **Mogens Djernes M.D., Department of Anesthesiology, Vejle Hospital, DK-7100 Vejle, Denmark.**

● There is an Apples in Medicine Interest Group that you can contact via Dr Alick Elithorn, Dept. of Psychological Medicine, Royal Free Hospital, Pond Street, London NW3.

We don't know of a source for Comal-80.

Perhaps one of our readers can help.

and over again and you don't get bored with them.

You may well begin a strategy game with a set opening gambit as in chess, but like chess no two games will be quite the same.

They make you think too – some of them are good history lessons. In fact one title that springs to mind is nothing but a history lesson.

Our current top range of titles are without doubt war and strategy followed by fantasy games – at one time thanks to Wizardry they were in the top spot. Then come the simulations and adventures, with the arcade games at the bottom.

Incidentally Woodland Software is run by games players for games players. We don't just sell them. We play them or at least most of them. – **Robin A. Hood, (Ye Woodland Outlaw), Woodland Software, London.**

Data on Data

IN the September 1984 issue of Apple User there was a small article headed "Data on tap". It dealt with a system called KI which would allow an Apple with a modem to retrieve information from the American mainframe of Knowledge Index.

There was no mention as to who is supplying or marketing this service. I would very much appreciate it if you could supply me with any details of the company I should contact to get more information. – **Elaine Bragg, Mill Hill, London.**

● The company concerned is Learned Information (Europe) Ltd, PO Box 8, Abingdon, Oxford.

Vote for Apple Plot

OF all the graphics packages now available I keep coming back, on my Apple II+ to good old Apple Plot for honest-to-God, easy to use, no frills business graphics.

Combined with Beagle Bros Alpha Plot or Flex Text for adding further text (including lower case), plus Phoenix Software's Zoom Graphix for printing options it satisfies the

vast majority of my needs, without having to BSAVE hi-res screens (heavy on disc space).

Even the standard print option from Apple Plot is sufficient in many cases, and puts three graphs very neatly on one A4 size page. (Note the need to modify the Apple Plot Hello program to accommodate, say, an Epson printer.) – **M.T. Muir, Muscat, Sultanate of Oman.**

Exit Theseus

IN the January 1984 issue of Apple User (great mag!), Julian Brewer reviewed Theseus and the Minotaur, from TSR.

The Lord of the Rings was mentioned, but search as I might I could not find a supplier's list with this on. Also mentioned was an unnamed better trilogy. Could you please send information on both these games (price, supplier etc). – **Jonathan Robinson, Great Yarmouth.**

● As far as we know, Julian was talking about the book of Lord of the Rings and we've never come across a micro version.

Also we suspect the other trilogy he mentioned was the Donaldson series of books, and they haven't appeared in game form either. We gather TSR has stopped distributing Theseus too, which is a pity.

Apple IIe compatability

WOULD you please advise me, before placing an order for back issues of Apple User and Windfall, whether listings in previous years will readily run on the Apple IIe? – **David Ross, Skelmersdale, Lancs.**

● On the whole listings in the magazine will work on both the II+ and IIe machines because of the high degree of compatibility between them.

Most problems are solved by keeping the IIe Caps Lock on because II+ input routines typically only expect upper case.

Because of some hardware differences, some Appletips

may not work on a IIe because they are, for example, POKEing a II+ specific location. However these are usually described as being for the II+ only.

The main worry about compatibility between the two machines is expressed by II+ owners because an increasing amount of commercial software is being written specifically for the IIe.

Of mice and maps

I PURCHASED a copy of the May 1984 issue of your magazine while on a European trip. I read, with great interest, the article by Charles Fairfax on the Fledermaus.

I also am interested in using a mouse and a IIc for digitising maps and recorder charts. I am just getting started so I cannot

Better not always bigger

IT would appear from the response to my letter of June 20 by Ms Clarke (Feedback, January 1985) that she finds the excuse of having to meet internally generated deadlines as sufficient justification for the publication of a program containing multiple errors.

I trust that she will display the same degree of sympathy and understanding in the event of misfortune selecting her as the victim of a "computer error".

The lady ought to feel highly honoured that her correspondence not only received editorial approval but was promptly published in the first edition following her contribution – waiting for acceptance "without comment" apparently requires an extra three months or more.

While I agree that "perfect programs" (an endangered if not already extinct species) are a luxury which few – if any – can afford, it can not be disputed that the constant aim of any professional programmer must be to create an end product which fulfils the system requirements.

My statistical summary of the revised Casino demonstrates my contention that, in many instances, better pro-

say how it will turn out.

The way that I intend to solve the problem of rotation of the mouse is to use small drafting tools, either the parallel rule or the small drafting machine. I have a large professional drafting machine but it costs nearly as much as the IIc so I elected to go with the smaller, cheaper version.

You should be able to find these in any well-stocked stationery store or engineering supply house.

I hope to use this also in laboratory instruction for students. The drafting machine will prevent inexperienced users from rotating the mouse even though movement of the mouse will not be as flexible as your set-up.

I hope you find this information useful. By the way, I think that Fledermaus is a great name. – **Professor Myron Molnau, University of Idaho.**

grams are not of necessity bigger and does not support Ms Clarke's assertion that I have engineered a "marvellously streamlined and efficient program".

In my opinion the revised program does not have a distributive value, and I therefore suggest that any and all of your readers who seek an "Aunt Sally" – fresh eggs only, please – should concentrate their attention on my program of Package Deals in Windfall, October 1983 (anonymity was and has continued to be an editorial decision).

This publication of an original game is a transcription of a printout submitted for consideration which was rushed into production (who failed to meet their deadline?) with a major variation from the original highlighting an imperfection on my part – I admit, without reservation, to one tiny clanger.

Contrary to any confusion which might arise from the editorial introduction, the program clearly requires spatial perception from the players and may be run on a non-disc system. – **G. Mathieson, Strood, Kent.**

Plantin' package

PLANTIN' Pal is a new gardening-by-computer package that has been prepared with the help of some of America's leading gardening experts.

Any or all of 40 vegetables can be selected and taking account of the size and shape of the garden, Plantin' Pal will tell when to plant and when to harvest, proper care and probable yield.

It even gives a detailed map of the garden with plant-by-plant and row-by-row suggestions about layout, taking into consideration plant height, compatibility with other plants and so on.

Plantin' Pal lays out the garden in wide rows, which is claimed to save space and reduce weeding and watering.

● P&P Micro Distributors, Todd Hall Road, Carrs Industrial Estate, Haslingden, Rossendale, Lancs. BB4 5HU. Tel: 0706 217744.

Keep one step ahead

A NEW "weapon", with potential in battles from the boardroom to the bedroom, is now available from Thorn EMI Computer Software Distributors.

Mind Prober for the Apple II claims to provide users with an uncannily accurate insight into the minds of anyone they care to



Plantin' Pal... gardener's friend

probe. By providing a complete character assessment and analysis of the other party, Mind Prober is said to give users the edge in all their relationships, whether business, social or personal.

The company insists it works even on subjects known only slightly to the user.

It asks the user to decide whether or not 100 characteristics – for example, ambitious, charitable, sensitive, meticulous, wary – apply or do not apply to the subject.

The information is then processed and the system quickly returns an overall character assessment, and an extensive analysis of his/her attitudes to relationships, work, stress, sex and, finally, what makes the subject tick.

On the strength of this insight, the user is alleged to have "an enormous advantage in his relationships, which can be exploited to the full in facing specific situations.

"For instance, understanding

the boss is a big step towards dealing with him in job appraisal or promotion interviews; likewise penetrating the minds of business colleagues will help in keeping ahead of the game; so will understanding friends and family better and even the person you have long admired from afar and to whom you would like to get closer...!"

Thorn says Mind Prober is "astonishingly accurate", whether used with serious intent or for pleasure.

Mind Prober costs £29.95.

● Thorn EMI Computer Software Distributors, Thomson House, 296 Farnborough Road, Farnborough, Hants. Tel: 0252 543333.

Visicalc optimiser

VC/KIT will sort a Visicalc spreadsheet, it will display – and let the user print out – all formulae in their correct grid

locations, and will allow Visicalc to load datafiles from many word processors and databases – even remote mainframes.

It adds help screens to the program, and also provides a library of command files.

VC/Kit is a productivity companion for Apple II Visicalc – or any DOS 3.3 file-compatible workalike, such as Magicalc. Minimum requirements are a 48k II+, IIe or IIc with one drive.

Features such as the help screens and command files are available when the spreadsheet is active, while others are accessed through a friendly spreadsheet-like menu system.

First-time users will have results within minutes while more demanding users will discover flexibility without complexity.

● Proform Development, PO Box 1602 Edmonton, Alta, Canada T5J 2N9.

Mouth mouse

DENTISTS are now being offered mouse technology to computerise their practices.

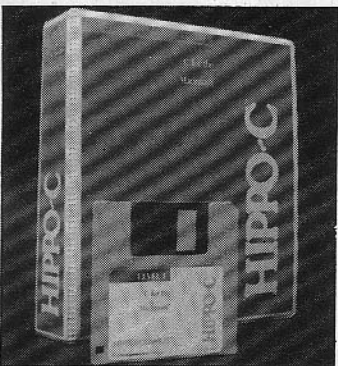
A dental supply chain is marketing a range of packages for dentists, based on the Apple IIe and IIc.

The entry-level package is FeeCalc, which works out NHS fees and mimics a Form FP17 on screen.

Extra software packages cover patient records, appointments and dental laboratory work, plus missed appointments, reminders and overdue account letters.

● Claudius Ash, Casco House, Moon Lane, Barnet, Herts. EN5 5UA. Tel: 01-440 8100.

Hippo runs C programs on the Macintosh



HIPPO-C is an interactive C environment which allows the user to edit, compile, link, and run C programs on the Macintosh.

The Hippo-C compiler follows the standards of Kernighan and Ritchie. Advanced C features such as structure passing and assignment have also been implemented.

The Macintosh Toolbox and Quickdraw are convenient to access from Hippo-C. They are

called as any other C routine.

A partial list of Toolbox and Quickdraw routines are included in the Hippo-C manual, but the publishers recommend a copy of Apple's 'Inside Macintosh' if any serious software development is planned.

The tutorial menu can be used by the novice C programmer as a tutorial, or by the experienced as a reference guide.

The debug feature allows

viewing of the program execution. It is useful for locating bugs, and also as a tool for learning C program flow and logic.

The print function allows a file to be printed as a background task.

This allows the Macintosh to be used while the program is printing out.

● P & P Micro Distributors, Todd Hall Road, Carrs Industrial Estate, Haslingden, Rossendale, Lancs., BB4 5HU. Tel: 0706 217744.

SPLIT SCREEN TACTICS

P.H.P. HARRIS helps you to
develop schizoid tendencies

IT is often useful to partition the video screen into two or more segments. Within a program, for example, half a screen may be used for entry and output of data while the second half displays instructions to the user.

Alternatively, running totals for January to May may be displayed on one half while the user deals on the other half with June.

In programming mode one half may be used for listing and writing programs while a disc catalogue or a second program listing may be called up on the second.

The programming of the screen for this purpose is simply a matter of POKEing appropriate values into the special locations \$20-23 (decimal 36-39) as described in the Apple II Reference Manual on page 31.

However, switching rapidly between two halves of the screen is tedious, particularly using direct commands during a programming session.

The Setscreen facility (see Listing I) makes light work of this problem.

Screens are defined by the &S command, with five parameters separated by commas. The syntax is &S,n,l,w,t,b where

n,l,w,t,b are variables determining the reference number of the screen, the left edge, the width, the top edge, and the bottom edge of the screen window respectively.

The parameters entered must be sensible, as the routine does not perform error checking. Variables not having integer values, or arithmetic expressions evaluating to non-integral values, may also be entered as parameters, as they are automatically rounded down to integral values.

Table II on page 31 of the Reference Manual gives the maximum and minimum usable values, but unfortunately contains several misprints. The correct values are shown here in Table I.

Having established the limits of one or more screens by means of the &S command, the &V,n command may be used. This sets the screen limits to the appropriate parameters for the nth screen, and positions the cursor at the top left of the new screen.

Nothing is deleted from either the old or the new screen, but the cursor can only be moved with the new screen limits until either the TEXT command or another &V com-

mand is issued.

The command &V,0 is legal, and sets the screen to the limits defined by the most recent &S command. Setting the n parameter to a value higher than the highest value set by any previous &S command results in a "? SYNTAX ERROR" message.

The &L command is not strictly necessary and may be omitted (lines 122-174 of the listing, addresses \$40A1-\$40F7 of code). It is then necessary to omit lines 73-74 of the listing, or to put:

```
403D: EA EA
```

```
403F: EA EA
```

in the code listing. This routine is intended to enable text to be written on the bottom line of a screen right to the lower right hand corner, something that Basic does not permit without scrolling, and is of particular value when making use of very small screens.

The syntax is &L,s,<var(name)> where <var(name)> is either a string literal or arithmetic expression, or the name of a string or arithmetic variable.

S is a variable (or arithmetic expression) used as a parameter in a similar way to the Basic SPC command to establish the point on the line at which to start printing the string. It does not overprint existing text with blank spaces. The cursor position is not affected by the &L command.

This routine checks whether the sum of the s parameter and the length of the string is less than or equal to the width of the

window, and truncates the string appropriately if it is not.

In the context of working with small screens in direct (programming) mode, it is worth noting that the key sequence <esc>@ is a better way of clearing the screen than the command HOME.

The buffers used in the &S and &V routines are located at the end of the code, and are arranged to provide sufficient space to store the parameters for nine screens. This should be sufficient for most purposes!

If the routine is saved to disc with the command BSAVE SETSCREEN, A\$4000,L\$11B any screen parameters which have been established up to that point will be saved at the same time. This means that the next time the program is BRUN the screens are immediately available via the &V command.

```
&S,1,0,20,0,24
&S,2,20,0,24
&S,3,0,40,0,12
&S,4,0,40,12,24
&S,5,0,33,0,24
```

Table II: Useful sets
of screen parameters

Sets 1 and 2 partition the screen into two equal parts vertically, while sets 3 and 4 partition it horizontally. Set 5 enables a quick &V,5 command to achieve the same effect as the irritating POKE 33,33. The quickest way of getting back to the full screen is still the TEXT command.

Function	Location		Minimum/normal/maximum value	
	Dec	Hex	Decimal	Hex
Left edge	32	\$20	0/0/39	\$0/\$0/\$27
Width	33	\$21	1/40/40	\$1/\$28/\$28
Top edge	34	\$22	0/0/23	\$0/\$0/\$17
Bottom edge	35	\$23	0/0/24	\$0/\$18/\$18

Table I: Text window special locations

SOURCE FILE: SETSCR2

```

1 *****
0000: 2 *
0000: 3 * SETSCREEN (MARK II)
0000: 4 *
0000: 5 * &V,SCREEN NO.,LEFT,WIDTH,TOP,BOTTOM
0000: 6 *
0000: 7 * &V,SCREEN NO.
0000: 8 *
0000: 9 * &L,SPACES,TEXT
0000: 10 *
11 *****

```

```

0000: 13 PTR EQU 0
0002: 14 TEMP EQU PTR+2
0004: 15 SPC EQU TEMP+2
0005: 16 LEN EQU SPC+1
0011: 17 VALTYP EQU #11
0020: 18 WNDLFT EQU #20
0021: 19 WNDWTH EQU #21
0023: 20 WNDPTH EQU #23
0024: 21 CH EQU #24
0025: 22 CV EQU #25
0028: 23 BASL EQU #28
00A0: 24 FACHO EQU #A0
00B1: 25 CHRGET EQU #B1
03F6: 26 AMPER EQU #3F6
DD7B: 27 FRMEVL EQU #D7B
DEBE: 28 CHKCOH EQU #DEBE
DEC9: 29 ERR EQU #DEC9
E9E7: 30 STRLIT EQU #E9E7
E74C: 31 COMBYTE EQU #E74C
ED34: 32 FOUT EQU #ED34
FB4B: 33 SETWHD EQU #FB4B
FBC1: 34 BASCALC EQU #FBC1
FC22: 35 UTAB EQU #FC22

```

```

----- NEXT OBJECT FILE NAME IS SETSCR2.OBJO
4000: 39 ORG #4000

4000:A9 35 40 INIT LDA #>PARSE
4002:8D F6 03 41 STA AMPER
4005:A9 40 42 LDA #<PARSE
4007:8D F7 03 43 STA AMPER+1
400A:A9 00 44 LDA #0
400C:8D 20 41 45 STA NBUF
400F:60 46 RTS
4010:A9 1C 48 SETPTR LDA #>NBUF-4
4012:85 00 49 STA PTR
4014:A9 41 50 LDA #<NBUF-4
4016:85 01 51 STA PTR+1
4018:CA 52 SP2 DEX
4019:30 0F 53 BMI RET
401B:38 54 SEC
401C:A5 00 55 LDA PTR
401E:EA 04 56 SBC #4
4020:85 00 57 STA PTR
4022:A5 01 58 LDA PTR+1
4024:EA 00 59 SBC #0
4026:85 01 60 STA PTR+1
4028:DD EE 61 BNE SP2
402A:60 62 RET RTS
402B:20 81 00 63 PARM1 JSR CHRGET
402E:20 4C E7 64 JSR COMBYTE
4031:EC 20 41 65 CFX NBUF
4034:60 66 RTS
4035: 68 HSB OFF
4035:1C9 53 69 PARSE CMP #/S
4037:1F0 05 70 BEQ SETSCRN
4039:1C9 56 71 CMP #V
403B:F0 39 72 BEQ VIEWSCRN
403D:1C9 4C 73 CMP #L
403F:F0 40 74 BEQ LASTLINE
4041:4C C9 DE 75 JMP ERR
4044:20 2B 40 77 SETSCRN JSR PARM1
4047:30 03 78 BMI S1
4049:BE 20 41 79 STX NBUF ;CURRENT MAX NO. OF SCREENS
404C:BA 80 S1 80 TXA
404D:4B 81 PHA
404E:20 4C E7 82 JSR COMBYTE ;LEFT )
4051:8E 1C 41 83 STX NBUF-4 ; )
4054:20 4C E7 84 JSR COMBYTE ;WIDTH ) TO TEMPORARY
4057:8E 1D 41 85 STX NBUF-3 ; )
405A:20 4C E7 86 JSR COMBYTE ;TOP ) BUFFER
405D:8E 1E 41 87 STX NBUF-2 ; )
4060:20 4C E7 88 JSR COMBYTE ;BOTTOM)
4063:8E 1F 41 89 STX NBUF-1
4066:6B 90 PLA
4067:AA 91 TAX
4068:20 10 40 92 JSR SETPTR ;PARAMETERS
406B:A0 03 93 LDY #3
406D:EA 1C 41 94 S2 LDA NBUF-1,Y ; TO
4070:91 00 95 STA (PTR),Y
4072:88 96 DEY ; DEFINITIVE
4073:10 FB 97 BPL S2
4075:60 98 RTS ; BUFFER
4076:20 2B 40 100 VIEWSCRN JSR PARM1
4079:30 05 101 BMI V1
407B:F0 03 102 BEQ V1
407D:4C C9 DE 103 JMP ERR
4080:A9 00 104 V1 LDA #0
4082:A8 105 TAY
4083:20 4B FB 106 JSR SETWHD
4086:20 10 40 107 JSR SETPTR
4089:81 00 108 LDA (PTR),Y ;SET CURSOR
408B:85 24 109 STA CH
408D:C8 110 INY
408E:C8 111 INY ; TO NEW
408F:81 00 112 LDA (PTR),Y
4091:85 25 113 STA CV
4093:20 22 FC 114 JSR UTAB ; TOP LEFT
4096:A0 03 115 LDY #3
409B:81 00 116 V2 LDA (PTR),Y ;
409A:99 20 00 117 STA WNDLFT,Y ;SET NEW
409D:88 118 DEY ; SCREEN LIMITS
409E:10 FB 119 BPL V2 ;
40A0:60 120 RTS
40A1:20 2B 40 122 LASTLINE JSR PARM1

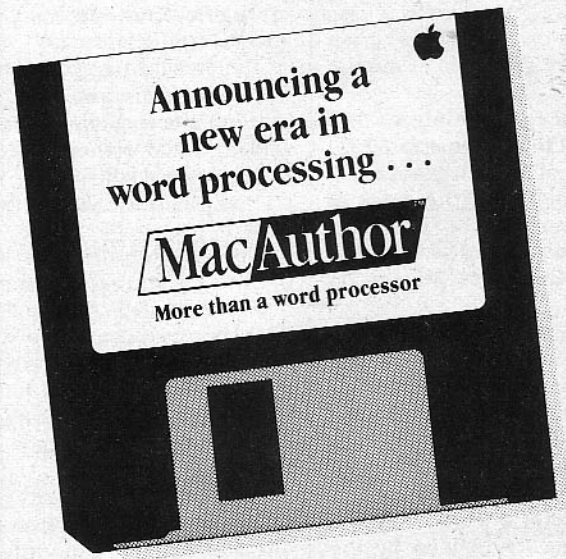
```

```

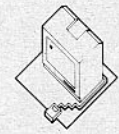
40A4:86 04 123 STX SPC
40A6:20 BE DE 124 JSR CHKCOH
-----
40A9: 126 *-----
40A9: 127 * USEFUL ROUTINE TO EVALUATE FORMULA POINTED AT BY TTXPTR.
40A9: 128 * CONVERTING ARITHMETIC RESULTS TO STRING. POINTER
40A9: 129 * TO STRING RESULT LEFT IN FAC MD,LO.
-----
40A9:20 7B DD 131 JSR FRMEVL
40AC:24 11 132 BIT VALTYP ;BRANCH IF STRING NAME
40AE:130 06 133 BMI L3 ; OR LITERAL (IN QUOTES)
40B0:20 34 ED 134 JSR FOUT ;CONVERT NUMBER TO STRING
40B3:20 E7 E3 135 JSR STRLIT ; AND ESTABLISH END OF STRING
-----
40B6: 136 *-----
40B6:A5 23 138 L3 LDA WNDPTH
40B8:38 139 SEC
40B9:EA 01 140 SBC #1
40BB:20 C1 FB 141 JSR BASCALC
40BE:18 142 CLC
40BF:A5 28 143 LDA BASL
40C1:A5 20 144 ADC WNDLFT
40C3:A5 04 145 ADC SPC
40C5:85 28 146 STA BASL
40C7:90 02 147 BCC L1
40C9:E6 29 148 INC BASL+1
40CB:A0 00 149 LDY #0
40CD:81 A0 150 LDA (FACHO),Y ;STRING LENGTH
40CF:85 05 151 STA LEN
40D1:C8 152 INY
40D2:81 A0 153 LDA (FACHO),Y ;SAVE
40D4:85 02 154 STA TEMP ; POINTER
40D6:C8 155 INY ; TO
40D7:81 A0 156 LDA (FACHO),Y ; ASCII CODE
40D9:85 03 157 STA TEMP+1 ; OF STRING
40DB:38 158 SEC
40DC:A5 21 159 LDA WNDWTH
40DE:E5 04 160 SEC SPC
40E0:90 15 161 BCC RTS
40E2:E5 05 162 SEC LEN
40E4:80 05 163 BCS L4
40E6:18 164 CLC
40E7:A5 05 165 ADC LEN
40E9:85 05 166 STA LEN
40EB:A4 05 167 L4 LDY LEN
40ED:88 168 DEY
40EE:81 02 169 L2 LDA (TEMP),Y
40F0:09 80 170 ORA #80
40F2:91 28 171 STA (BASL),Y
40F4:88 172 DEY
40F5:10 F7 173 BPL L2
40F7:60 174 RTS RTS
40F8: 175 BUFFERS DS 10*4
4120: 176 NBUF DS 1

```

*** SUCCESSFUL ASSEMBLY; NO ERRORS



A powerful word and graphics processor for Apple Macintosh.
For further information contact:



Icon Technology Limited
9 Jarrom Street
Leicester LE2 7DH
Telephone: 0533 546225