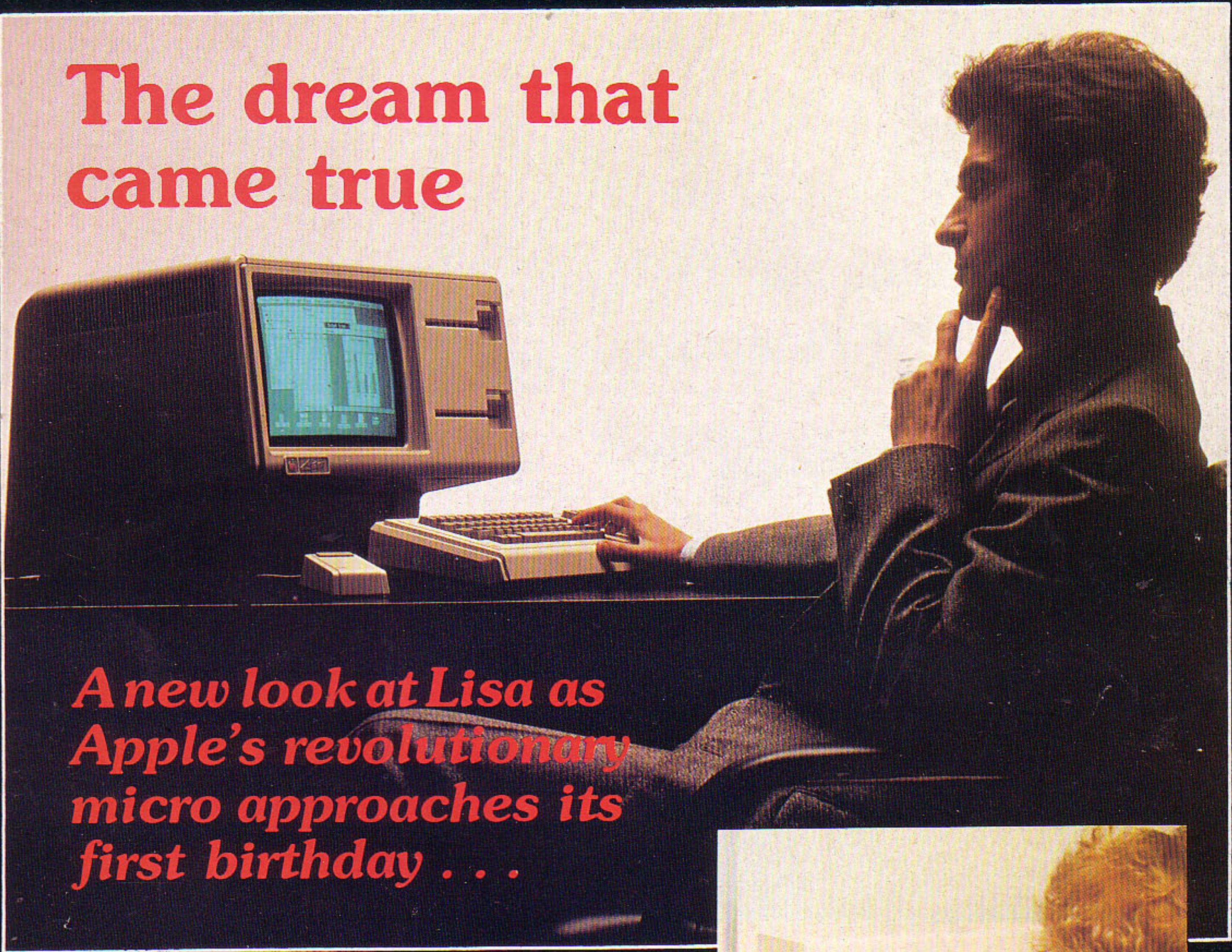
apple user

Vol. 4. No. 1 January 1984 £1

-the new name for Windfall

formerly called:



The dream that came true

A new look at Lisa as Apple's revolutionary micro approaches its first birthday . . .

ANNOUNCING THE INTEC 505
REMOVABLE HARD DISK FOR THE APPLE®

- 10 MB OF ON-LINE STORAGE
- 5 MB IN A REMOVABLE CARTRIDGE
- FULLY APPLE COMPATIBLE
- SOFTWARE SUPPORT FOR DOS, PASCAL, CP/M, FASTDOS & CP/M PLUS
- RANGE OF COMPREHENSIVE UTILITIES FOR FILE TRANSFER & DIAGNOSTICS
- UNLIMITED DRIVE CAPACITY – ADD 5 MB FOR THE PRICE OF A CARTRIDGE
- BOOT FROM EITHER FIXED OR CARTRIDGE
- EMBEDDED SERVO GUARANTEES INTERCHANGEABILITY
- EASY & RELIABLE BACKUP IN MINUTES
- ALL UK MANUFACTURED

interface INTEC technology

INTEC, 41a-45 Knights Hill, West Norwood, London SE27 0HS
Telephone 01-761 5999 Telex 8813271 GECOMS G

Imagine 5 Megabytes in the palm of your hand! Plug in application after application merely for the price of a cartridge.

The INTEC 505 eliminates outright the problems of Winchester backup – your data is stored on the most secure media available – the Winchester itself.

Backups are made quickly and very reliably and may be interrogated and updated without overwriting currently active files.

Cartridges are readable at any time in the future by any drive using the same technology.

At £2700* the INTEC 505 is the only sensible, reliable & economic solution to data storage. How much is your data worth? – Think about it!

The INTEC 505 is the only system capable of expanding to meet your future capacity requirements.

EQUIPPED FOR THE FUTURE
RECORDING THE PAST

*Excluding VAT. Dealer Enquiries welcome.

Ampersand

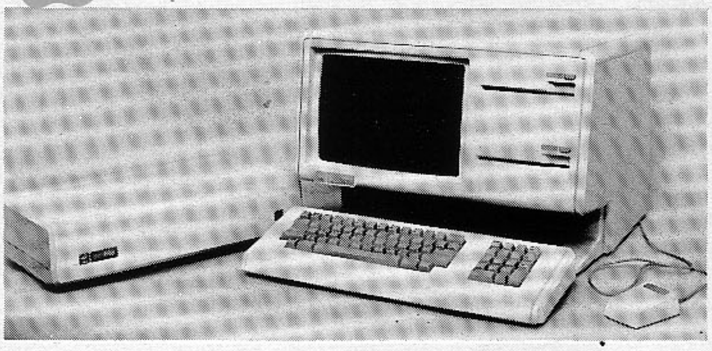
Books

Business

Feedback

Games

apple user
Volume 4
Number 1
January 1984



Graphics

Handicapple

Hi-res text

Lisa

Logo

News

New products

Pascal

Reviews

Technique

WP

Apple User Managing Editor **Derek Meakin** *Production Editor* **Peter Glover** *Features Editor* **David Creasey**
Technical Editors **Cliff McKnight Max Parrott** *Editor-in-Chief of Database Publications* **Peter F. Brameld**
Marketing Manager **Sue Casewell** *Advertisement Manager* **John Riding** *Sales* **Peter Nowell**

Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising) Telex: 667664 SHARET G. Prestel: 614568383.

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

Trade distribution in UK and Ireland by Wells, Gardner, Darton & Co Ltd, Faygate, Horsham, West Sussex RH12 4SU. Tel: Faygate 444.

Apple and the Apple symbol are the registered trade marks of Apple Computer Inc. Windfall 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.

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)

WILDCARD PLUS

COPIES 64K PROGRAMS IN 12 SECONDS

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

FEATURES

□ Hardware copying device...push button operation.

□ Copies 48K programs in 8 seconds, 128K programs in less than 25.

□ Creates autobooting, unprotected disks.

□ Software in ROM, always there...no need to preboot.

□ No language card required, works equally well with or without.

□ Utilities include: compression software-several backups may be stored on a single disk, recover BASIC programs, disassemble machine code software, screen dumps, etc.

WILDCARD PLUS
£99

From your local dealer, or direct from:

Elite Software Company
30 Guildford Street,
Chertsey,
Surrey,
KT16 9PQ
Tel: 09328 67839

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

Dealer enquiries welcome.

Apple is a registered trademark of Apple Computers Inc.

By DAVID CREASEY

Is Apple about to peel off its Mackintosh?

THE launch of Mackintosh, the second in the Lisa family of micros, is imminent, although Apple remains silent about the release date and the machine's specifications.

The announcement of Lisa just under a year ago coincided with a major Apple board meeting.

As a similar meeting is scheduled for later this month and Apple has said a new machine is due "in the first quarter of 1984", an announcement about Mackintosh at the meeting seems likely.

However, under cover of all the industry excitement and speculation about the new micro, some rather special other products were slipped into public view at the Comdex Fall '83 exhibition in Los Angeles last month.

They are particularly significant as they would seem to confirm Apple's commitment to its existing product line.

On show, albeit discreetly, was a mouse control for both the IIe and the Apple III.

Although shown at Comdex, they are not expected to be announced officially until this month. They are already available to software developers and should be on dealers' shelves in the first quarter of this year.

The Mouse IIe features pull-down menus, a multiple window capability and MousePoint, a free form graphics package.

The Mouse III features MouseDraw, which is similar to the LisaDraw package, but has colour capabilities.

First of the new mouse-based software packages, which use the ProDos operating system (see next month's Apple User for a special feature on ProDos) are two integrated three-in-one packages from Apple.

Appleworks (running on a 128k IIe) combines a word processor, database and spreadsheet capabilities in one package and should be available by the end of March.

Its big brother equivalent is III Easy Pieces.

The Database function is described as a bigger, faster and more powerful version of Apple's Quickfile, and the word processor features many of the Applewriter IIe and III capabilities.

The spreadsheet is said to comprise a massive 999 rows by 127 columns. It features variable column widths, a sorting facility and can read both Visicalc and DIF files.

The program commands are consistent throughout the three integrated applications and the program is entirely menu-driven. Release of the two new mice and the new software packages make the message from Apple itself seem clear. It certainly has no intention of dropping products like the Apple II and III.

While its Lisa family of micros will be developed throughout the year – I predict there will be at least three new machines – the company will also have some major releases for its 6502 family.

ON the subject of Apple's not-to-be-spoken-about-just-yet machine, considerable amusement was aroused in journalistic and computing circles by the fact that a senior IBM executive, responsible for the company's "entry systems" (micros) in Europe is Dr Alex MacIntosh.

Apple itself created a laugh in the early days of Lisa by naming a promotional video "Apple Flasher".

This was perhaps slightly unfortunate in the light of the second product in the Lisa family being called Mackintosh.

Touch of the vapours

MANY people have trouble enough comprehending what is meant by the words *hardware* and *software.*

Now you can baffle them further, and enhance your neighbourhood reputation as someone who knows something about computers, by making random reference to "vapourware".

Erik Larson, writing in the *Wall Street Journal* recently, is to be commended for bringing vapourware to public notice.

The word describes that much talked-about, much promised but so far unintroduced machine or program that many manufacturers have in their product inventory.

★ ★ ★

ANOTHER nice touch, particularly in the light of mice proliferating across the Apple product range, is provided by Hewlett-Packard staff, who describe their company's new, touch-screen control micro as a mousetrap.

Over the moon in Cupertino

A SCHOOLBOY from Northern Ireland is set for a dream trip to Silicon Valley to pick up a Lisa from Steve Jobs, Apple Computer's founder-president.

The visit to California and the gift of the computer form the prize package won by 14-year-old Declan McKeown of Downpatrick for coming top in a competition organised jointly by Apple UK and *The Observer* newspaper.

His entry was one of more than 6,500 received, with a panel of judges eventually naming him the overall winner from nine finalists.

Declan, whose parents bought him a micro last Christmas, reached the final by correctly answering that:

- ● Three per cent of British households have already installed a micro.
- ● Britain in 1982 was the fastest growing micro market in the world.
- ● A total of 83 per cent of British residents interviewed could describe a micro without prompting.
- ● The principal business application of micros is accounting/invoicing.
- ● 5.38 per cent of businesses in the UK have already installed micros.

The entries of the nine finalists were judged by Keith Hall, sales and marketing director of Apple UK, Julian Allason of *The Observer* and John Ashworth, chairman of the National Computing Centre.

As a tie breaker, the finalists were asked to devise an acronym for a new computer. Declan's SLAVE – Sophisticated Logical Advanced Versatile Electronics – eventually got the nod.

Drop in profits forecast, yet Apple still has a rosy glow

IF the rest of the world seems to be worried (or gloating, depending on your outlook) at the fact that Apple Inc is only expecting to average $5 million profit a quarter for the next year, Apple itself doesn't seem too perturbed.

In fact after hearing Peter Cobb, Apple UK managing director, give his thoughts on the future to a recent dealer meeting, it seems that it will be a rosy year for Apple.

He predicted that other manufacturers would find it increasingly difficult to cope with technological pressure in the market.

Not only would they find it hard to keep up with the research and development effort put in by Apple and IBM, but they would find it hard to get components – particularly when one considered the volume production planned by the two market leaders.

Companies would also come under pressure to provide adequate quality and support for their products, said Cobb.

"It will be a very competitive world in 1984 all over again", he said.

But Apple's own future was secure, claimed Cobb. "Our position is that of leader and innovator – and that will continue to distinguish us from other major forces in the market place", he said.

"Our mission is the expansion of Lisa technology. Lisa was the first step, and in 1984 there will be other examples".

"But you will also see us develop the technology of our 6502 products", he added. "We will remain a personal computer company – we will sell to people. Customers and not companies are what are important to us".

Taking a dig at IBM, Cobb said that "some people wanting to sell to end users have even given their machine the same name – *personal computer* – but I don't believe that they have the same approach".

He said of IBM and other competitors: "They will keep making obsolescent technology". He noted that the market had been through the 16-bit era and was going out the other end. But Apple hadn't made that mistake.

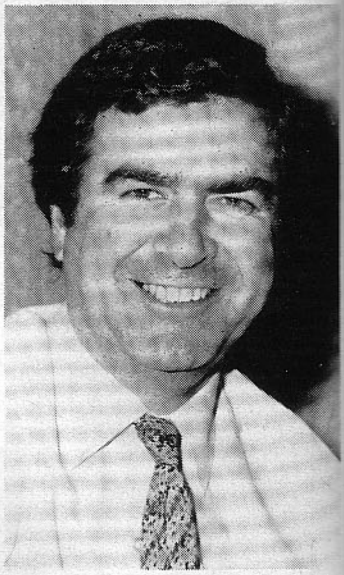
"When we launched Lisa we didn't say that 32-bit technology was here.

"What we call Lisa technology is ease of use, integrated software and style in computing. It just happens to run best on a 32-bit processor".

Cobb predicted volume sales of Apples, but said that mass retailing would increasingly encroach on Apple dealers.

For dealers to survive they would need to "add value to the sale" by offering expert knowledge and continuing support.

He told dealers, "If you try to compete with the high street stores they will eat you alive – they know more about just shifting boxes than you do".


Peter Cobb ... not noticeably concerned about the future

YEAR OF THE APPLE?

A PREDICTION that 1984 will be the "year of the Apple" has come from John Rowland, the buying manager for the W.H. Smith chain of micro stores.

He bases his forecast on the fact that sales of Apple products from his company's outlets have doubled in the few weeks since the introduction of the Professional Home Computer pack.

"A clearer picture will emerge in the New Year when things are not being distorted by the Christmas factor", he told *Apple User*.

"But it looks as though Apple is going to have a much greater impact in 1984".

John Rowland insists that next year Apple will come into its own in the UK micro market as users of "lesser machines" start to trade up.

"A lot of people who have already bought Spectrums and Commodores are likely to want a real one – and that's when they'll turn to Apple", he says.

"For as far as our company is concerned, it represents the top end of the market and will carry on doing so".

The giant retail chain recently increased its store-within-a-store micro shops from six to 30 in major British cities.

IBApple – it's a sort of oriental takeaway

AN Apple lookalike micro in an IBM lookalike box is the latest oddity to surface in the Far East.

Whether it is a brilliant marketing plan or an attempt to bypass eagle-eyed guardians of company copyrights isn't clear – but it seems to be selling well.

The machine was spotted by Peter Fisher, of Pete and Pam Computers, on a recent dealer tour to the Far East.

"Of particular significance – especially with regard to market pressures forcing prices down on world micro markets – is that both Taiwan and Japan are becoming incredibly efficient at manufacturing", said Fisher.

"As a result many of the Americans on the tour are now considering having their products manufactured in the Far East.

"Hong Kong in particular is putting a lot of effort into manufacturing products at low cost".

He added that Apple still dominates that market because of the copycat industry.

"More than 100 stores in Hong Kong's Golden Shopping Arcade concentrate on selling Apple lookalikes, and everywhere you turn there people are prepared to copy discs or manuals on the spot" he said.

"There were copies of just about every interface card and 128k memory card I have ever seen – and just as many lookalike micros available".

In Taiwan Fisher said businessmen are rapidly approaching the point where they will be creative rather than just be good copiers.

It was difficult to gauge how the Japanese market would affect European markets. For example the market leader there with a 35 per cent share was NEC – yet that company had only one per cent of the UK market.

At present the Japanese were concentrating on satisfying their domestic market, although Panasonic was producing non-Japanese machines targeted specifically at the American market.

Lisa – the programmer's dream

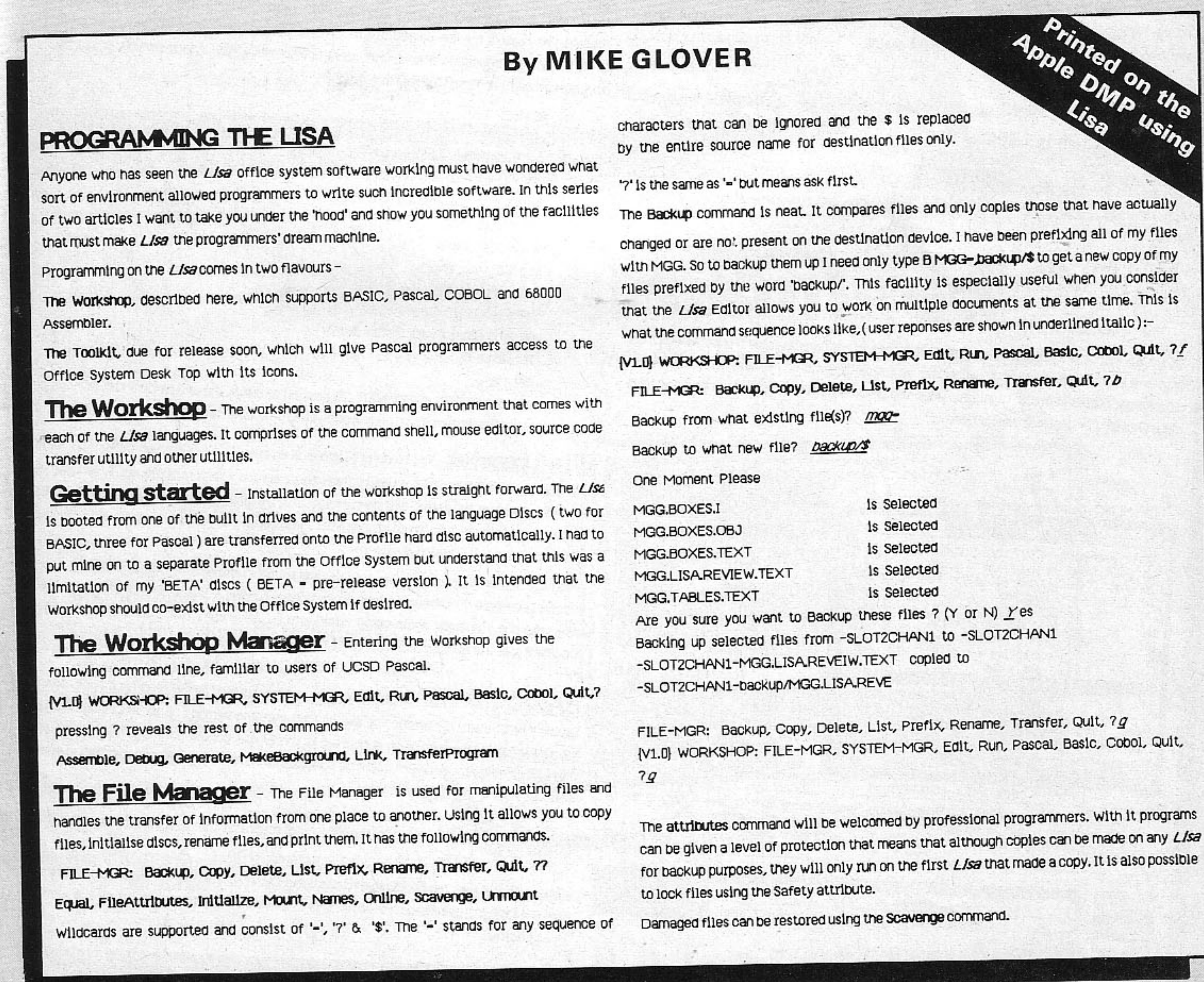
NOBODY ignores Lisa. Apple Inc is consumed with the product. Having bred and nurtured it, it now watches with a mixture of pride and concern to see how the world will salute its darling with sales.

Cliche-lovers describe it as the best thing since sliced bread. Non-Apple dealers and other manufacturers describe it as an expensive mistake that could bring the parent company down.

Here are two viewpoints from people who have used Lisa with a vengeance, having first cut their teeth on years of working with Apple IIs.

In the first of a two-part feature, MIKE GLOVER – who is a dealer, a programmer and a utility/fixit/Epson printer specialist – describes why he thinks Lisa is the programmer's dream machine. And we reproduce the Lisa-produced printout of his article.

And in our second Lisa article CHARLES FAIRFAX and JO WALSWORTH-BELL explain how Lisa is creating a breath of fresh air at the homes of patients in the North West Regional Health Authority area.



By MIKE GLOVER

Printed on the Apple DMP using Lisa

PROGRAMMING THE LISA

Anyone who has seen the *Lisa* office system software working must have wondered what sort of environment allowed programmers to write such incredible software. In this series of two articles I want to take you under the 'hood' and show you something of the facilities that must make *Lisa* the programmers' dream machine.

Programming on the *Lisa* comes in two flavours –

The Workshop, described here, which supports BASIC, Pascal, COBOL and 68000 Assembler.

The Toolkit, due for release soon, which will give Pascal programmers access to the Office System Desk Top with its icons.

The Workshop – The workshop is a programming environment that comes with each of the *Lisa* languages. It comprises of the command shell, mouse editor, source code transfer utility and other utilities.

Getting started – Installation of the workshop is straight forward. The *Lisa* is booted from one of the built in drives and the contents of the language Discs (two for BASIC, three for Pascal) are transferred onto the Profile hard disc automatically. I had to put mine on to a separate Profile from the Office System but understand that this was a limitation of my 'BETA' discs (BETA = pre-release version). It is intended that the Workshop should co-exist with the Office System if desired.

The Workshop Manager – Entering the Workshop gives the following command line, familiar to users of UCSD Pascal.

```
{V1.0} WORKSHOP: FILE-MGR, SYSTEM-MGR, Edit, Run, Pascal, Basic, Cobol, Quit,?
```

pressing ? reveals the rest of the commands

```
Assemble, Debug, Generate, MakeBackground, Link, TransferProgram
```

The File Manager – The File Manager is used for manipulating files and handles the transfer of information from one place to another. Using it allows you to copy files, initialise discs, rename files, and print them. It has the following commands.

```
FILE-MGR: Backup, Copy, Delete, List, Prefix, Rename, Transfer, Quit, ??
Equal, FileAttributes, Initialize, Mount, Names, Online, Scavenge, Unmount
```

Wildcards are supported and consist of '=', '?' & '$'. The '=' stands for any sequence of characters that can be ignored and the $ is replaced by the entire source name for destination files only.

'?' is the same as '=' but means ask first.

The **Backup** command is neat. It compares files and only copies those that have actually changed or are no: present on the destination device. I have been prefixing all of my files with MGG. So to backup them up I need only type B **MGG=,backup/$** to get a new copy of my files prefixed by the word 'backup/'. This facility is especially useful when you consider that the *Lisa* Editor allows you to work on multiple documents at the same time. This is what the command sequence looks like,(user reponses are shown in underlined italic):–

```
{V1.0} WORKSHOP: FILE-MGR, SYSTEM-MGR, Edit, Run, Pascal, Basic, Cobol, Quit, ?f
  FILE-MGR:  Backup, Copy, Delete, List, Prefix, Rename, Transfer, Quit, ?b
  Backup from what existing file(s)?  mgg=
  Backup to what new file?  backup/$
  One Moment Please
  MGG.BOXES.I                        is Selected
  MGG.BOXES.OBJ                      is Selected
  MGG.BOXES.TEXT                     is Selected
  MGG.LISA.REVIEW.TEXT               is Selected
  MGG.TABLES.TEXT                    is Selected
  Are you sure you want to Backup these files ? (Y or N) Yes
  Backing up selected files from -SLOT2CHAN1 to -SLOT2CHAN1
  -SLOT2CHAN1-MGG.LISA.REVEIW.TEXT  copied to
  -SLOT2CHAN1-backup/MGG.LISA.REVE

  FILE-MGR:  Backup, Copy, Delete, List, Prefix, Rename, Transfer, Quit, ?q
  {V1.0} WORKSHOP: FILE-MGR, SYSTEM-MGR, Edit, Run, Pascal, Basic, Cobol, Quit,
  ?q
```

The **attributes** command will be welcomed by professional programmers. With it programs can be given a level of protection that means that although copies can be made on any *Lisa* for backup purposes, they will only run on the first *Lisa* that made a copy. It is also possible to lock files using the Safety attribute.

Damaged files can be restored using the **Scavenge** command.

The System Manager

- The System Manager has the following commands:-

SYSTEM-MGR: ManageProcess, OutputRedirect, Preferences, Time, Quit, ??

Console, FilesPrivate, Validate, DefaultPrinter

ManageProcess - This command enables the performance of the Workshop to be made even faster by nominating processes to be resident in memory rather than having to be loaded in from the Profile when required. With both BASIC and the EDITOR resident I was able to move from typing this text in the Editor, to the BASIC ready prompt, in 11 seconds and that included saving the file! The move back was even quicker. From typing 'bye' in BASIC to editing again took just three seconds.

OutputRedirect - Using this command it is possible to have a copy of all output that is displayed on the console sent to another device such as the printer or a file on disc.

All of the command lines displayed in this text were taken from text files created using this facility. I hurt my brain quite badly trying to figure out why this command appeared to only work part of the time, until I realised that the file would not exist on the disc until output was directed away from it to some other source such as the console.

Obvious when you know, but a note in the documentation would not go amiss.

Preferences - is the same as in the Office System and enables such things as screen brightness, repeat key speed and speaker volume to be configured.

Other commands let you see and set the date and time (*Lisa* has a built in clock with battery backup), see system files (normally hidden) and redirect screen output to a secondary screen (*Lisa* has two) or to a remote device.

The Editor

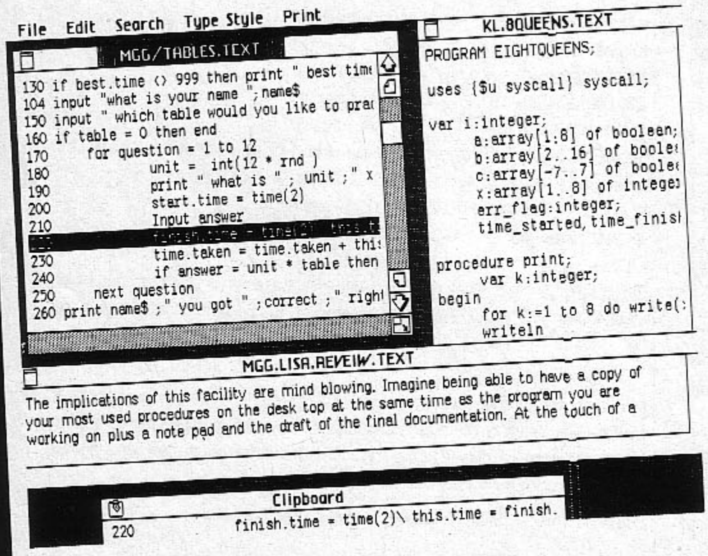
- The Workshop Editor works in the same way as The Office System by providing multiple pieces of paper (windows on files) on the Desk Top (screen) at the same time.

Each may be reshaped and repositioned by means of the Mouse and you may jump from one to the other by simply pointing to the desired destination document and clicking the mouse button.

The Editor uses the same 'Pulldown menus' as does the Office System. A menu bar across the top of the screen contains the menu items -

File Edit Search TypeStyle Print

These are selected by pointing the mouse at them and holding down the mouse button. This causes a menu to drop down showing the available selections. Movement of the mouse downwards causes items to highlight, when the desired item is selected, letting go of the mouse button completes the selection.



```
130 if best.time <> 999 then print " best tim
104 input "what is your name ";name$
150 input " which table would you like to prac
160 if table = 0 then end
170      for question = 1 to 12
180              unit =  int(12 * rnd )
190              print " what is " ; unit ;" x
200              start.time = time(2)
210              Input answer
220              finish.time = time(2)\ this.t
230              time.taken = time.taken + thi
240              if answer = unit * table then
250      next question
260 print name$ ;" you got " ;correct ;" righ
```

```
PROGRAM EIGHTQUEENS;

uses {$u syscall} syscall;

var i:integer;
     a:array[1:8] of boolean;
     b:array[2..16] of boole
     c:array[-7..7] of boole
     x:array[1..8] of intege
     err_flag:integer;
     time_started, time_finis

procedure print;
     var k:integer;
begin
     for k:=1 to 8 do write(
     writeln
```

The implications of this facility are mind blowing. Imagine being able to have a copy of your most used procedures on the desk top at the same time es the program you are working on plus a note pad and the draft of the final documentation. At the touch of a

```
220          finish.time = time(2)\ this.time = finish.
```

Information may be moved or copied from one to the other by means of the 'cut and paste' facility which is at the heart of the *Lisa* mouse technology.

To copy information from one worksheet to another you simply wipe the mouse over the desired text whilst holding down the mouse button. The text is then copied on to the clipboard (scratchpad) by selecting copy from the pulldown edit menu. A click in the destination window at the insertion point and a second visit to the pull down menu - this time for Paste - are all that are required to complete the operation.

The implications of this facility for multi-tasking are mind blowing. Imagine being able to have a copy of your most used procedures on the desk top at the same time as the program you are working on, plus a note pad and the draft of the final documentation! At the touch of a button you can jump from one to the other transferring information visibly on the way. The potential increase in productivity is staggering.

Eight different type styles are available, two of which are proportional fonts. These type styles range from 20 Pitch Gothic to 10 Pitch Courier giving a maximum of 136 characters per line.

One of the **print** commands is particularly helpful to a programmer as, upon command, keywords can be underlined and each page date stamped in the footer along with the file name.

Exec Files

- EXEC files are a very useful tool in the programmer's armoury taking over the job of typing commands at the keyboard. Instead the commands are typed into a file which is then EXEC'ed. A typical use for an Exec file would be controlling a large compilation.

Lisa EXEC files are very powerful and have a number of commands that make them very flexible. Parameters may be passed to Exec files at the time of invocation, or requested at run time by the Request command. Conditional statements may be included and it is even possible to nest Exec files! Commands include:-

```
$EXEC
$ENDEXEC
$SET <%n> TO <str Expr>
$DEFAULT <%n> TO <str Expr>
$REQUEST <%n> WITH <str Expr>
$READLN <%n>
$READCH <%n>
$WRITE <str Expr>
$WRITELN <str Expr>
$IF <Bool Expr> THEN stuff
$ELSEIF <Bool Expr> THEN stuff
$ELSE stuff
$ENDIF
```

Boolean functions **EXISTS** and **NEWER** can be used to check the existence and relative age of a file, preventing for example, compilation of a file that already exists.

File Transfer

- The transfer program is an easy-to-use communications package that enables text files to be transferred between *Lisa* and a remote computer. It can also be used as a terminal emulator. This means that *Lisa* could be used as an incredibly powerful development tool by mainframe programmers who could have all the benefits of the mouse editor and the multi-tasking environment.

Using the **Transfer** program I was able to send this document to my Apple //e using ASCII Express (an Apple // communications package), switch the *Lisa* into the Office System and then take the file back again using *Lisa* Term (*Lisa*'s communications package). The document was then pasted into *Lisa*Write for final editing.

BASIC

- The BASIC on *Lisa* is a powerful Interpreted BASIC and is compatible with DEC's BASIC-Plus. It permits the use of long variable names (up to 30 Characters) and multiple line statements. Also supported is the 64-bit double precision part of the IEEE numerics. Some programmers might be disappointed to see no graphics capability in BASIC. You will have to use Pascal for this.

Syntax diagrams are similar to the 'Railroad' diagrams that are used to describe Pascal Syntax.

Editing in the BASIC enviroment is primitive being limited to the destructive backspace and the DELETE command. However this is more than compensated for by the Workshop Mouse editor.

File handling commands include:- **Save, Unsave, Replace, Append, Cat.**

Save remembers the name of the file and makes sure you are aware that the previous version is going to be overwritten.

System commands include :- **New, Old, Renumber, List & Listnh, Length, Run, Cont, Trace, Variables.**

Most of these are well known to BASIC programmers. **Listnh** lists the program without headers (program name). **Variables** lists the program variables with their currently asigned values and length tells you how much memory is used and available. I had the Editor, BASIC, Pascal and a code file all resident and still had 541K to spare!

Specifications

Variables supported are:- Integer : -32768 to 32767

Real : +/- 4.9 * 10^-324 to +/- 1.7 * 10 ^ 308

String : up to 32767 characters!

Arithmetic operators :- + , - , * , / , ^ or ** , Remainder (but not MOD)

Logical Operators : **AND, OR, NOT, XOR,** and the less familiar, **EQV & IMP**

x EQV y is true if x & y are both true or both false

x IMP y is true unless x is true and y is false

Mathematical Functions include:-

Sin, Cos, Tan, Arctan, Log, Log10, e^X, Abs, Pi, Int Fix, Rnd, Sgn, Intpart, Compound, Annuity, Time

String Functions include:-

Len, Left, Right, Mid, Instr, + , Space$, Chr$, String$, Xlate, Cvt$$, Sum$, Dif$, Prod$, Quo$, Place$, Comp%, Val, Num$, Ascii, Rad$, Date$, Time$

*Lisa*BASIC also supports matrix arithmetic:-

Dim, Mat, idn, zer, con, Mat Read, Mat Input, Mat Print, + - *, Trn, Inv, Det, Linsys, Cond

Virtual Arrays - The power of *Lisa* BASIC is considerably increased by means of Virtual Arrays. Dimensioning an array with a file name causes that array to live on disc. The following program fragment illustrates the method of use.

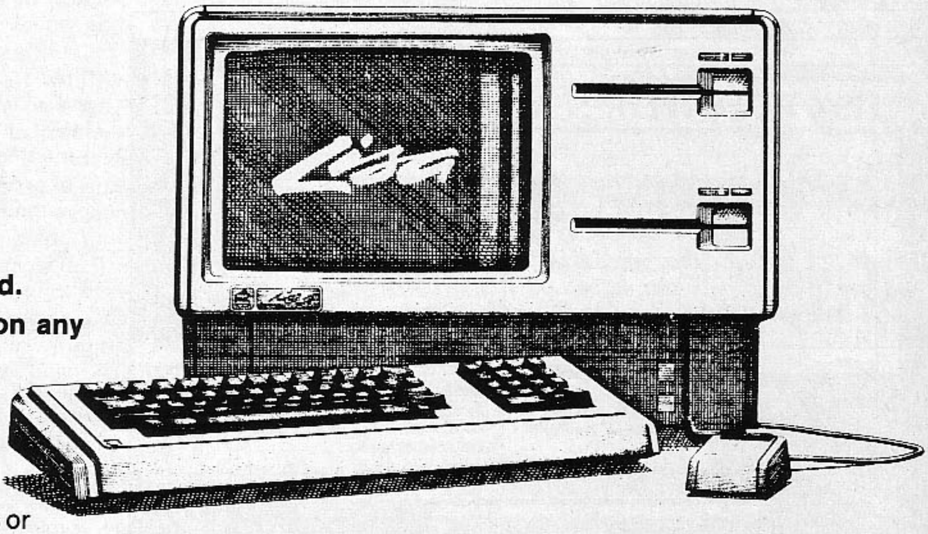
```
10 open  "-upper-mybigarray" as file #3
20 dim #1, test% (1000,1000)
30      for row = 1 to 1000
40         test% (row,column) = row * column  for column = 1 to 1000
50      next row
60 end
```

Next month we will take a look at Pascal and the fantastic graphics capabilities of the Quickdraw Package that are part of it. Built into the *Lisa* software are some incredible routines that include **Three dimensional graphics** and **colour** - just waiting for the right hardware to come along and let them out!

Lisa BASIC Plus costs £240

Lisa Pascal costs £480

Both systems include The Workshop and are available from authorised *Lisa* dealers.



Lisa takes NHS research in its stride

CHARLES FAIRFAX and JO WALSWORTH-BELL illustrate the beneficial effects of using Lisa to process research data

THE arrival of Lisa has revealed a new dimension in the processing of research data.

Until now it has been the custom, at the stage where accumulated information is ready for analysis, to sort, tabulate and print the results by computer.

The printed papers, which can be quite bulky, need to be looked at and physically handled on the research worker's desk or table top.

When the appropriate sections have been identified, further investigations might be carried out and the process repeated until the material is ready for final presentation.

At this stage, a typist armed only with a humble typewriter takes over the arduous task of preparing the tables. Graphs are often drawn by hand. Amendments to texts, tables and illustrations can cause many headaches.

Even with small-scale investigations involving analysis of limited numbers of results, this series of steps is tedious. As a consequence, a lot of effort has to be put into systems of sequential analysis to avoid collecting too much data.

In the field of Health Service studies it is of particular importance to reduce expense and, in some instances, the exposure of volunteers to inconvenience.

Often the relationships between sets of data remain unexplored for lack of data processing time or from the daunting prospect of facing a mountain of paper.

Modern microprocessor technology has given us Lisa to overcome these problems. We can now look at our data much more efficiently and very much more effectively than was previously possible.

The use of inbuilt multiple programming provides fast input and transfer of information, say between LisaCalc and LisaGraph.

It also permits rapid viewing of a series of different graphs alongside their related data to aid selection of material for further analysis or printing.

Processing can be restricted to those results showing promising relationships; the rest may be discarded.

Should it be required, Lisa's calculating facility may be used to enhance the visual demonstration by computing significance levels.

Scatter, line and bar graphs in both single and multiple form can be produced at a touch of the mouse button, providing a feeling of control not unlike on-line operation.

The selected diagrams appear on half the screen as fast as the data is fed on the other half. The graphs grow with the data. Alternatively, the data may be pasted in, say from LisaCalc.

The input can be terminated as soon as it becomes apparent that the relationship between the parameters is random. Conversely, any obvious relationship or trend can be singled out. Thus a minimum of paper is produced. Only useful work need be done.

Too ruthless trimming need cause no fears, since all the data can be stored.

These methods can be illustrated by an investigation into the use of oxygen concentrators in patients' homes. This is being conducted for the North Western Regional Health Authority.

The size of the enquiry is very limited in terms of Lisa capability, but the principles involved can be applied readily to a wide range of comparable applications which may encompass many times and scale of the one chosen.

The oxygen concentrator is a piece of apparatus that can be used in the patient's home as a substitute for oxygen in cylinders. It does not supply pure oxygen, but air containing an enhanced level of over 90 per cent.

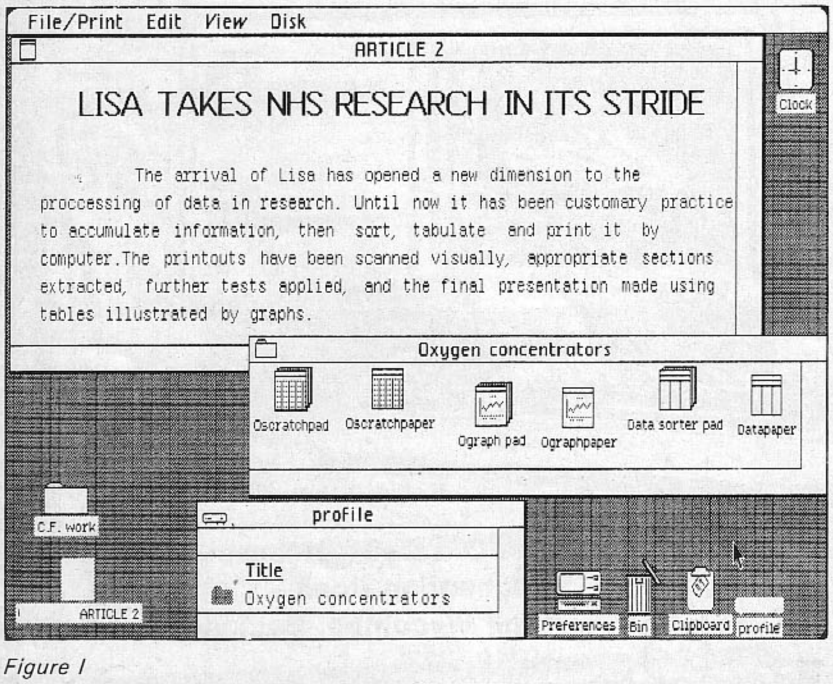
Considerable interest has focused upon its use, as a recent addition to the physician's armamentarium.

The trial has set out to investigate the therapeutic value and the cost-effectiveness of this new introduction.

During the course of the enquiry, large volumes of data have been collected on its many aspects. Much of this lends itself to processing by the method described.

For the purpose of illustration, the examination of that section of data comparing the number of persons needing to use certain drugs before and after treatment by the oxygen concentrator, demonstrates the speed and simplicity of the method.

There are many ways in which this can be approached



LISA TAKES NHS RESEARCH IN ITS STRIDE

The arrival of Lisa has opened a new dimension to the proccessing of data in research. Until now it has been customary practice to accumulate information, then sort, tabulate and print it by computer.The printouts have been scanned visually, appropriate sections extracted, further tests applied, and the final presentation made using tables illustrated by graphs.

Figure I

on Lisa and individual operators will have their own preferences.

From our initial experimentation with the Desktop Manager System it appears to us that a useful step, first having opened the Profile (i.e. the 5mb hard disc), is the designation of a working area on it.

This is followed by the creation within this area of duplicate workpads and papers, renamed according to their new subject matter. These are made up from the appropriate tools of the Profile.

These duplicated pads and papers may be created on the desktop, renamed and then transferred to the new workbox or folder, as the working area is called, ready for subsequent use.

They can be switched to and fro at will. It is much better for ease of subsequent handling to give this folder and contents new names rather than to work all the time with those of the built-in Desktop Manager master system tools and their pads (that is, untitled).

The illustration of a screen dump made at this stage shows the renamed pads and papers we used. (Figure I).

To previous Apple users this might appear a little excessive, but the large size of the central processor memory and Profile backup permits a degree of sophistication that otherwise would seem extravagant.

The name chosen to identify within the Profile the special folder used in this study was "Oxygen concentrators". Simple descriptive words are used in place of special file names.

The prepared folder was mounted from the File/Print menu and a blank sheet of graph paper – which we called Ographpaper, was opened. This was chosen to indicate which file (O for oxygen) the paper was taken from.

More precise naming of the new document at this stage would normally be carried out if one adhered to the straightforward Lisa procedure.

The preparation of the special folder renders this superfluous at this stage. Only the selected documents will need to be identified with more specific names.

Opening this document brought up the familiar LisaGraph screen format on the VDU. Column A was widened and a list of drugs entered, using their names in word form.

Lisa is capable of handling text in this area as readily as the handling of numbers. In column B the number of patients on each drug before oxygen concentrator treatment was entered and in column C the number after.

The pie chart showed the distribution of numbers of patients using the various drugs. The double bar chart showed the differences that obtained before and after treatment.

Several graphs can be visualised together on the screen along with the tables. They can be shrunk or enlarged and moved about to suit convenience. (See Figure II.)

For the first inspection, only rough unedited display quality was used. The words on the x axis took up more space than numbers and consequently overlapped when they appeared on the bar chart.

On the pie chart, as might be expected, small numbers were not worth doing, since their subtended angles were so small.

The ability to adjust column widths on the display is a great asset since it obviates the need to use excessive abreviations.

On simple inspection it was very clear which drugs showed "before" and "after" differences of an order that indicated they were worth pursuing.

On a second run these few were transferred to a new duplicated Ograph paper, then processed, renamed and printed. (See Figure III.)

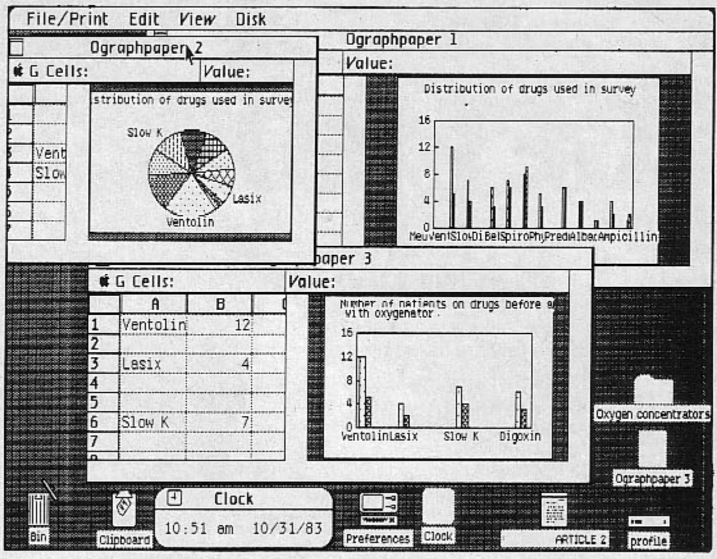
A small point worth watching, which can be seen in the illustration, is to create spaces between the line entries of the table to avoid excessive overlap of the printed names where they appear on the graph.

The use of this method enables a good print of the bar graph to be made very quickly.

This shows only the relevant data which has been selected, leaving out the unwanted information.

Tables and graphs selected on subsequent runs can be added together by collecting them on the desktop, that is, using the Set Aside selection from the File/Print menu.

When the final assembly of the material to be printed is reached, the clipboard provides a facility for viewing the Set Aside items required.



	A	B
1	Ventolin	12
2		
3	Lasix	4
4		
5		
6	Slow K	7
7		

Figure II

A pleasing feature of Lisa is the versatility of formatting and page layout, both on the VDU and on the printout. The whole of this article has been produced without use of pen or paper, except for scribbled notes on the printouts. The text was prepared on LisaWrite and the rest simply on LisaGraph.

Pasting was left out, since only enough data has been presented to illustrate the fundamental principles.

As the screen dumps indicate, it is possible to display multiple pieces of work simultaneously.

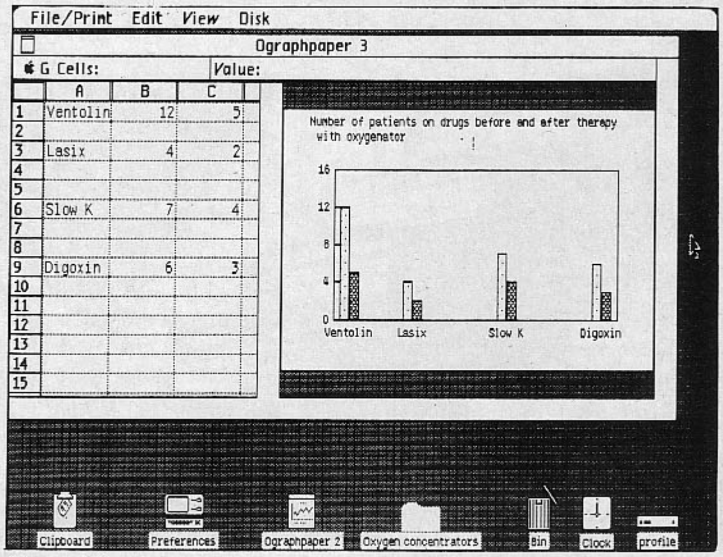
When we come to printing it is possible to print table alone, graph alone or both together. The graphs can be rescaled readily or plotted within preset limits. They can be done in a range of sizes and shapes and even automatically tailored to fit.

Should texts be drawn from sources with differing type styles, they can be converted all to the same style automatically. A wide range of type styles is available.

The system was tried out on a complete beginner: an arts graduate with no scientific background and no previous experience of computers of any type at all.

She experienced no difficulties in understanding the system and operating it without anxiety. This must be a test indeed of the high level of friendliness of Lisa!

** Dr Charles Fairfax and Dr Jo Walsworth-Bell are community medicine consultants working for the North Western Regional Health Authority.*

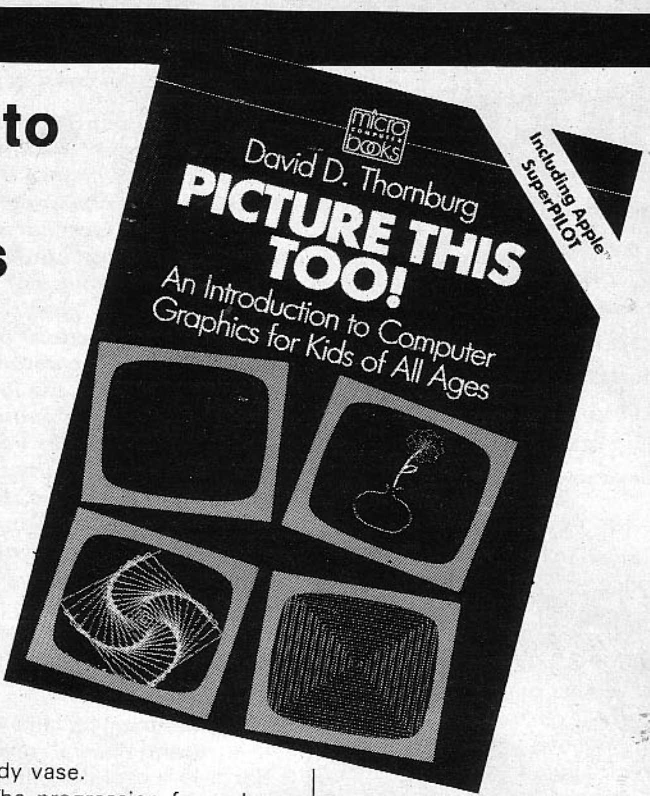


	A	B	C
1	Ventolin	12	5
2			
3	Lasix	4	2
4			
5			
6	Slow K	7	4
7			
8			
9	Digoxin	6	3
10			
11			
12			
13			
14			
15			

Figure III

Painless way to get to know turtlegraphics

Picture This Too! An introduction to Computer Graphics for Kids of All Ages, by David D. Thornburg *(Addison-Wesley).*


micro books
David D. Thornburg
PICTURE THIS TOO!
An Introduction to Computer Graphics for Kids of All Ages
Including Apple SuperPILOT

PICTURE This Too! is a book designed as an introduction to Turtlegraphics for children. It is one of those American computer books which actually thinks it's a computer program.

Somehow language which seems acceptable from a computer in response to one's tentative typings looks totally out of place when written down in a book. Perhaps it is because monitor screens have the decency to wipe themselves of these semi-secret banalities whereas the printed word will be on open display for ever.

This book includes such purple prose as, "Congratulations! You just made the turtle draw a square", "Wow! We're all done with our first picture. Not bad for your first session" and "Now that's pretty encouraging" and so on. Although it is by no means the worst of its type, I suspect that most adults with a double-digit IQ will find the style off-putting to say the least.

As you will have gathered this is not a book to be read on the bus going to school in the morning but one intended to be held in one hand while typing away with the other.

At that level it is not at all bad. It takes the user through some elementary SuperPilot commands and encourages the interested novice to play with the Turtlegraphics commands in SuperPilot's immediate mode. Some of the shapes are quite interesting and I quickly found myself modelling snail shells with a spiral drawing program.

The text describes the construction of simple shapes such as squares, polygons and curves and builds up to the multiple presentation of such shapes to make pictures. It culminates in producing an oddly geometric flower in pretty colours ("Ta Daa! We have finished our flower.") and sticking it in a handy vase.

The progression from shape to shape is logical and simple and the intricacies of the Pilot language are not allowed to intrude too much on the drawing of pictures.

By working through the examples and ignoring the over-enthusiastic asides, this book is a relatively painless way of becoming familiar with Turtlegraphics in SuperPilot.

Buyers should be wary of the front cover. To look at it one would expect the book to be about computer graphics in a variety of languages including Apple SuperPilot. The whole of the book however is in SuperPilot – no other languages are used.

So if you have an Apple with two disc drives, have Apple SuperPilot, have (or are) computer literate kids who want to learn to play with turtlegraphics and can spare $14.95 then this is the book for you.

Tony Cook

Pascal takes a bit longer

Pascal for the Apple, by Iain MacCallum

IF you want to learn all about Pascal, it will take only 160 hours, according to Iain MacCallum's book. This consists of 80 hours spent reading the book, and the remainder using the computer.

A disc of worked examples and explanations complements the book itself, which is said to teach an absolute novice all about operating the Pascal system as well as learning the language.

The book has its good points and several bad ones. It is in my opinion, one that many people will like and many others loathe.

On the positive side, the author has taken the attitude that the only way to learn constructively about computers is to sit at a keyboard and use one.

As a consequence, there were several programs with explicit instructions on compiling and running on the supplied disc.

There were also programs with slight errors, and accompanying instructions on the necessary corrections.

This arrangement allows the beginner a sense of achievement and participation very early on in the course.

As a further aid to producing a visible sign of success, a large amount of the course work revolves around producing drawings of windmills – you can actually see that your programs are working.

A feature I found extremely annoying, but other people may find positively helpful, was the fragmentary style in the first few chapters.

One example of this was a single page which went through the routine:

□ Let's now get rid of the last program.

□ This is what bugs are, there are some in the next program.

□ First let's change the system date.

□ Now get the next program and compile it . . . by the way . . .

This may not seem too devastating a flaw, but the rapid switch of directions certainly contravenes one of the standard rules of good teaching style that I learnt.

Apart from objections to Iain MacCallum's use of the Pascal language throughout the book, I also felt that the content was not suitably chosen or presented.

The foreword indicated that you would have learned all you needed to know about the Apple Pascal operating system, and the implementation of the language after your 160 hours of study. This is definitely not the case.

While one would not expect anyone to cram into one book a complete description and tutorial on both system and language, I felt that much was omitted that needed to be said.

(For example, you can get a printout of any text file by using the FILER to transfer it to PRINTER: or 6:.)

At the other end of the scale, I felt that the change from fairly straightforward programming in the beginning to the final sections on POINTERS as a means to LINKED LISTS, and GRAPHS etc. was an unnecessarily large step forward; especially as FILES and disc storage received only a very brief mention.

A reasonable number of people could find the book useful as a basis for getting to grips with the simpler use of Apple Pascal.

I feel, however, that the book's purpose and its target audience have not been considered sufficiently carefully, with the result that the content, pace and presentation are not well-balanced enough to make it a good one.

J.P. Lewis

COMPILED THOUGHTS

MAX PARROTT introduces his regular column on programming with a look at NANO 6502–an assembly language package

Well! What an end to the year – *Windfall* no longer on the book shelves. But "The King is dead, long live the King" and here we are again, and what am I doing here in Compiled Thoughts?

Simple, this is a column dedicated to nothing in particular but aspects of programming and programs on the Apple (in whatever flavour).

I hope to wander, with your help, over assembly language, Basic, maybe Forth – but won't dip into Pascal as we'll leave that to Jonathan Lewis.

I thought I would start with some assembly language for no reason other than A comes before B and *Windfall* happened to have received quite a few letters about the text on hi-res screens program in the November issue.

Now if you don't think much of assembly language because you don't know it, bear with me rather than turn the page – because in a paragraph or two you'll be reading something to your advantage.

A letter from Peter Trinder of Berkshire (see Feedback) refers to assembler incompatibilities – and I have received others on the same subject.

The drawback with assemblers for the 6502 is that there appear to be no real standards for the opcode mnemonics, especially for the "pseudo opcodes".

For this reason *Windfall* always publishes the assembly listing together with the assembled machine code, if only as a hexadecimal dump, so that the true meaning could be worked out.

Most listings have appeared as Apple Toolkit assembler listings, but Lisa and more recently Merlin (or Big Mac) listings have become more popular.

I'll try to indicate which assembler is being used for a listing.

Another letter, from John Blaiklock of Norwich, points out that the routines used in the November program to calculate the hi-res base address for the y co-ordinate is not the best way to go about it.

His own routine (published in the May 1983 issue of Windfall) is both shorter and swifter.

He is quite right, and I would recommend anyone about to type in the program to incorporate his routine.

I would have used it myself if I had had it to hand when writing my original version (which was well over a year ago) but I'm reluctant to change a working program although always keen to learn new tricks.

By the way, did anyone notice the first line of that program? It was an attempt at a joke which has backfired on me.

I had thought that it would appear in the April 1983 issue, but the vagaries of publishing ensured that it wasn't. Which reminds me, if the gems you send to *The Apple User* aren't published immediately, don't worry.

There are many pressures in the publishing world and timetables often become upset.

My reluctance to play with working programs is especially marked in the case of machine code routines because of the amount of effort required.

This explains my liking for Forth as a programming language.

Unfortunately I'm still very much a beginner here, and hence have a lot of sympathy with those beginning assembly language.

When I began to learn it there was no such package as NANO 6502 which I'm convinced would have helped me greatly.

NANO 6502 REVIEWED

'There comes a time when most Basic programmers become frustrated with the language and recognise the fact that it was not really intended as a serious programming tool. Whether this realisation comes about because of the lack of speed, structure, or the fact that you are locked into the set of commands implemented in the version you use is really immaterial, the cure is the same. Either write part of your code in machine code and add it to your Basic program, or change languages.'

THIS statement by Mike Glover in a recent *Windfall* review of Forth is absolutely true and probably explains why possibly the most common question in our postbag is: "What book would you recommend for learning assembly language?"

This question, usually followed by "I need one which is not too deep" not only supports Mike's statement, but also indicates the fear many people have of assembly language.

The vast amount of symbolism (absolutely meaningless at first) which stares the reader in the face as soon as such a book is opened is a probable reason.

This same symbolism presents an obstacle for me in writing this review because I have to start using it although I should write so as to appeal to the man, woman or child who wants to take up assembly language and hopes that NANO 6502 will help the learning process.

I will say immediately that it will, without doubt! Having said that, what can I say without beginning the jargon? Well, NANO 6502 will not spare you that but it will allow you the luxury of stepping through your programs, observing the effects (expected or otherwise) on memory, registers, stack – there you are I've started on the jargon – without fear of crashing or wandering off into the darkest depths of your Apple's memory.

Let me explain why this is necessary. One of the biggest problems facing someone new to programming is how to go about the techniques of "debugging". With assembly language the sheer speed of program execution, and the ease with which the program can end up in a part of the memory for which it was never intended, makes the job awesome.

If you don't have an old, non-autostart ROM Apple with the STEP & TRACE facility then you have to fall back on a relocated version of this facility (see *Windfall,* December 1981, page 10 if you're interested) or to a hardware solution.

STEP & TRACE via software doesn't solve all problems by any means, as anyone who has tried stepping through COUT ($FDED, a useful routine in the Apple's monitor) or tries using the same zero page locations as STEP & TRACE will have found.

NANO 6502 overcomes these problems by giving you a simulated 6502 with eight pages of memory. These eight pages stretch from $000 to $7FF and are yours, not the Apple's; you can use them as you wish.

The stack is, as usual, in page 1, so you have to be careful there. But then again you have to be with a real 6502.

The memory may be programmed via a simple "memory set" command or by a "mini-assembler", very similar in operation to the mini-assembler contained in Apple's Integer Basic.

Your program may be disassembled, a line at a time, or more quickly if desired whereupon a listing similar to the Apple monitor's but with the Ascii equivalent as well, is obtained.

The memory may also be saved to disc as a work-file and recalled as required, and any binary disc file which begins at an address in the required range can be loaded and worked within the memory range allowed.

After setting the memory you can step through your program a line at a time, or more quickly if desired, until a BRK instruction is encountered or until the Program Counter goes out of range. (That is, outside the eight page memory map.)

While tracing the program the registers are clearly displayed in both binary and hexadecimal formats. The display is the best of its kind I have

Learn to love assembly language with NANO 6502

seen. The values appear in small boxes, broken so as to clearly indicate individual bits. The registers may be altered and program execution continued at will by single keystroke commands under three main menus.

The first menu obtained after booting the front side of the disc offers four options:

- *Trace a program*
- *Modify and view memory*
- *DOS options*
- *Return to Basic.*

The appropriate letter followed by RETURN brings up a sub-menu. Return from each sub-menu to the main one is done by pressing R followed by RETURN. Other command letters bring about the required action.

The DOS menu allows the saving and loading of the work area (which can be saved as many times as you want under different names), the altering of slot and active drive number, CATALOGing data discs, deleting work files on disc and initialising new data discs.

These are initialised as standard DOS 3.3 (unlike the program disc which comes protected) but they do not carry a copy of DOS itself and so are unbootable. This is done to extend the backup memory store available.

The Trace menu allows tracing, examination and changing of the registers and a quick reference guide. The Memory Modifying menu allows assembly of a program, dis-assembly of a program, modification of memory and clearing of the memory. This is well explained in the manual which comes with the program disc.

In particular there is a diagram/flow chart of the program which does much to aid the user in the early days of its use – although I found I quickly assimilated the commands and did not have to refer too often to the manual.

Generally the user is allowed to move directly from one activity within a sub-menu to another without going via the menu.

But he cannot jump from one sub-menu to another. He has to return to the main menu by pressing the R key. I found this a little irksome.

For example, it would be useful to have the ability to stop tracing and immediately re-assemble a piece of code by pressing just the A key; or when assembling it, be able to immediately flip through the quick reference guide.

This, by the way, is a short guide to the nmemonics for the 6502 instruction set with an indication of the possibilities of each instruction (as regards memory addressing etc) and a diagram of the processor status register, indicating which flags will be altered by the instruction.

I tried all sorts of programming tricks with the assembler and tracer. I wrote self-modifying programs, I manipulated the stack to move through memory, I used indexed addressing across page boundaries (proper wrap-round was obtained with zero page usage) and I tried out some binary coded decimal techniques to see how well the simulator behaved. In all my tests it behaved well.

One facility of NANO 6502 is the ability to flip from "tracing" to "examining registers" and back again, so that the program counter, the general registers and stack pointer (which are displayed in different formats on different pages) may be changed and the flow of the program changed.

While tracing, the next line (to be executed) is displayed at the bottom of the screen and the registers are displayed as set by the immediately preceding line. Pressing a numerical key then executes the line at the bottom of the screen, displays the registers as appropriate, and writes the next line of your program at the bottom of the screen.

If a BRK is executed or an error is met, a message is displayed. Pressing the space bar returns you to the sub-menu. If the program counter goes out of the eight page range an error message is also given.

There are three exceptions to this. The Apple monitor routines – COUT, KEYIN and PREAD, which respectively write a character to the screen, read a character from the keyboard, and read a paddle value – are usable. They do not behave as other lines of program code in that you cannot trace through them.

What happens is that they execute immediately. Consequently you do not see the stack or the stack pointer change. But you will see the effect of the appropriate routine.

There is a "bug" in the simulator here, in that a JSR to one of these routines is allowed but a JMP is not. For example the code

```
$200  LDA   $C1
$202  JSR  $300
$205  LDA   $C2
          ,
          ,
          ,
$300  JMP  $FDED
```

will not allow you to access COUT (which is the name of the routine at &FDED) but the alternate structure at $300, namely

```
$300  JSR  $FDED
$303  RTS
```

will function quite correctly.

To sum up, I believe NANO 6502 will be useful to everyone who wants to undertake the task (and joy) of learning assembly language.

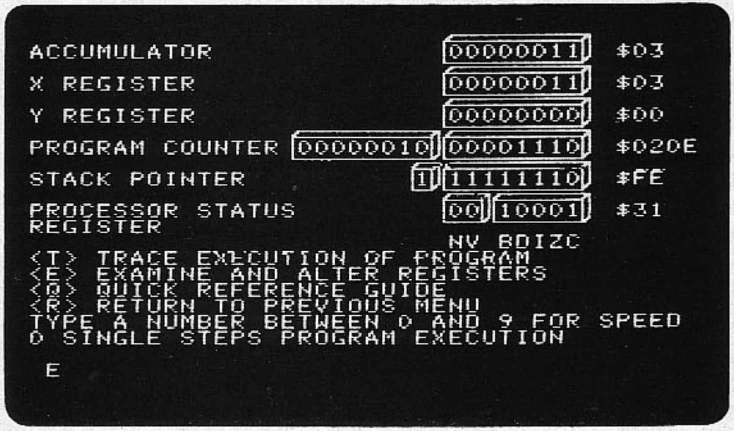
Together with the program there are seven demonstrations of assembly language routines, each showing some facet of programming and a tutorial, contained within the manual.

I have only seen a pre-production copy of this, without all of its parts, but it should be more than adequate for the newcomer, judging from the version I have.

You will also need a good text book, and perhaps the hardware or software to make use of the Apple's mini-assembler such as either an Integer card, a language card and disc system or a relocated version of the mini-assembler.

This is because NANO 6502 will naturally lead you on to use it, which is probably a good thing to do before you rush out to buy a full, symbolic assembler.

Having now appreciated the power of a tool such as NANO 6502, I rather hope that Malcolm Whapshott, the author of NANO 6502, will write a Z-80 simulator for the Apple – something which I for one could readily make use of.



```
ACCUMULATOR                      00000011   $03
X REGISTER                       00000011   $03
Y REGISTER                       00000000   $00
PROGRAM COUNTER        00000010  00001110   $020E
STACK POINTER                 1  11111110   $FE
PROCESSOR STATUS             00  10001      $31
REGISTER
                             NV  BDIZC
<T> TRACE EXECUTION OF PROGRAM
<E> EXAMINE AND ALTER REGISTERS
<Q> QUICK REFERENCE GUIDE
<R> RETURN TO PREVIOUS MENU
TYPE A NUMBER BETWEEN 0 AND 9 FOR SPEED
0 SINGLE STEPS PROGRAM EXECUTION

 E
```

The NANO 6502 trace menu

Help for those who aren't micro minded

PETER BRAMELD begins a regular advice column with first time business Apple users in mind.

MICROS, and Apples in particular, are wonderful *for* business and *in* business. Everyone says so . . .

However, if you are business minded but not micro minded the whole matter can be particularly confusing and frustrating.

In this column we intend to present the power of micros in a fashion which can be readily understood by those coming from a non-technical background.

If you find it difficult to relate your commercial problems to specifications (or jargon) such as 64k, RAM, twin floppy discs, input, output or CP/M, and if booting and Basic are boring and baffling, then you are probably an average business user – and please read on.

We are going to try to relate computers to actual business activities in a way you will understand.

If you are a businessman baffled by science, please let us know. Briefcase will try to come to your aid.

If you have bought, or are looking for a computer system, you will almost certainly have encountered the typical salesman who tends to stress seemingly meaningless (to you) technical specifications.

Most car drivers are interested only in driving – not in how to put together or repair an engine.

Similarly, most businessmen simply want to "drive" a computer – not program it or understand its inner workings.

However, as you are not going to be able to avoid technical jargon, let's look behind some of the "magical numbers" and see how, without fully comprehending what they mean, you can relate them to your own activity.

Computers are often measured in terms of their memory capacities.

A standard Apple IIe for example, has a 64k memory, and a standard Apple III 256k – with the k standing for kilobyte and each k representing 1,024 storage locations (think of it as 1,024 unique pigeon holes).

A 64k machine therefore has a memory which will accomodate just over 65,500 units of information.

Some of the micro chips fitted to the green "motherboard" inside your Apple case make up this memory space, or temporary storage area.

It is important to make a distinction between two types of memory which you, as the business user, will be concerned with.

Some of the memory on the motherboard which we have just described can be thought of as a work area – but one which can be wiped clean (like a school blackboard) simply by switching off the power supply to your Apple.

Semi-permanent memory (or volatile memory, another salesman's favourite) involves disc drives, either hard or floppy, which are the electronic/magnetic equivalent of the traditional filing cabinet or the music cassette tape. More about this later.

First of all – what does 64k or 256k of memory mean in day-to-day terms, and is it enough for you?

There is a direct relationship between the size of a computer's memory and the size of a document or final schedule which it can accommodate.

In deciding how much memory you need, you should first look at what type and what volume of information your business deals with.

Long legal documents, thousands of address labels, extensive customer records and accounts and large chunks of financial information can be processed efficiently using a computer.

They can be processed more quickly and more efficiently if the computer's memory capacity is in proportion to the size of the information.

Spreadsheet users working with large models will soon find that they need more than the basic 64k on an Apple IIe.

But if you only use the machine to produce short letters, to handle only several hundred addresses or manipulate small amounts of information, 64k is more than adequate.

You can increase the memory capacity of your machine using memory expansion cards (known as RAM cards). But beware, if you do so. Make doubly sure that the software or program you are using – for spreadsheet – is actually able to spreadsheet) is actually able to

If you have the older Apple II or II Plus, these cards fit into slot 0 (one of the seven connector slots inside your Apple).

If you have a IIe you have the choice of buying either a RAM card or an extended memory 80 column card. The latter also gives you 80 characters per line on the computer screen, which is a great advantage for database and spreadsheet users.

In addition to memory inside the computer, permanent storage of information is required.

Most businessmen begin their computing experiences using flexible discs 5¼in in diameter (hence the term 5¼in floppy).

Floppy discs are also available in 8in diameter (offering a storage capability of up to two mbytes) and more recently, 3½in sizes.

Whatever the size, you need a disc drive connected to the Apple to handle the disc. This performs a similar function as a cassette recorder does for cheaper home micros – but far more quickly.

In the case of an Apple IIe, each 5¼in disc will accommodate between 128k and 143k of information.

To visualise this try relating each disc to an office filing system. 128k will enable you to store between 800 and 1,000 names and addresses or around 50 pages of A4 text – a far more digestible statistic than the ubiquitous k!

If you are a serious business user you will have started with two floppy disc drive units.

Not only does the possession of two units double the available storage, but it also enables back-up copies of information to be made speedily and reliably.

The importance of duplicating information which is stored on any semi-permanent storage device (a magnetic disc) cannot be stressed too highly.

The cost of an ordinary floppy disc varies between £2 and £3.

But when it contains between 800 and 1,000 customer addresses it is far more valuable to your business – hence the wisdom of making regular backup copies!

Other, larger types of disc storage which can handle millions of characters are also available.

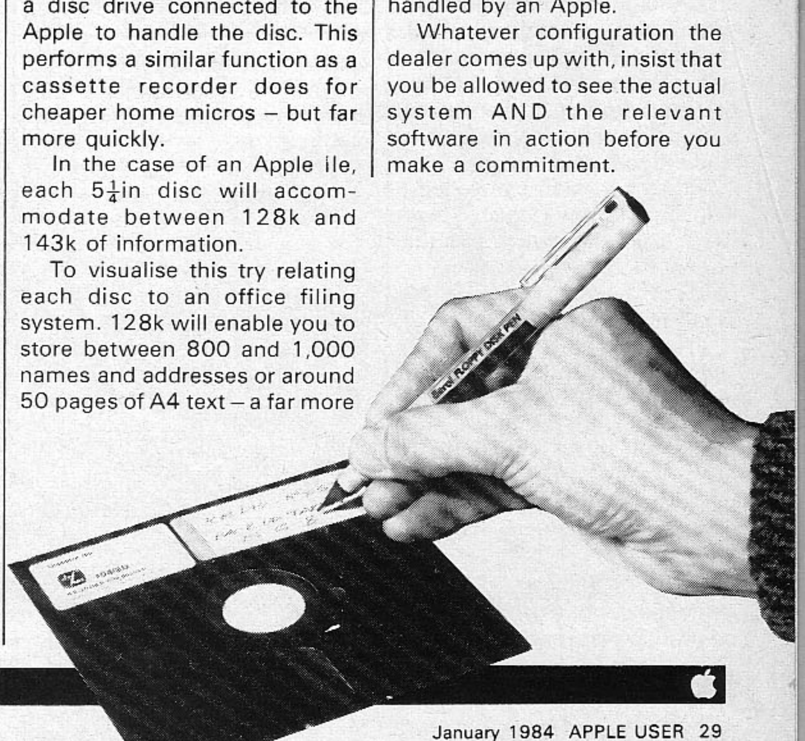
These are known as hard discs. Unfortunately some of these are fixed in place and so cannot be interchanged. This makes floppy disc back-up or tape back-up desirable.

This has been a simple explanation for a first time user of micros.

The fundamental requirement in matching a computer system to your business is to first decide, very carefully, what it is you want the machine to do.

With this established, go to a dealer – most are reputable – and ask him to demonstrate how your needs, particularly in terms of the amount of information you will process, can be handled by an Apple.

Whatever configuration the dealer comes up with, insist that you be allowed to see the actual system AND the relevant software in action before you make a commitment.



Oriental peripherals

MAX PARROTT casts a critical eye over four Apple add-ons from the East, all now available in Britain.

THE cost of peripheral equipment for computers has always been relatively high but there are now signs of prices dropping for disc drives, printers and joysticks. Leading the way are Japanese and Taiwanese manufacturers – especially the latter.

Recently *Apple User* received a new Taiwanese slimline 5¼in disc drive for review, the Aviette FDD 820.

This is styled in Apple colours with a beige shell, black front and black-filled slots along the sides.

There is a lever to move the head and lock the disc in place – and which ejects the disc when moved upwards to the open position.

The back of the drive sports a small three-way slider switch which allows the drive to act as a typical, normal read/write drive with write protection being offered by the notch on the disc when it is covered.

In the second position the drive will read only, which may be useful in demonstration and teaching environments.

In the third position the write-protect offered by the notch on the disc is over-ridden so that files can easily be written to the backs of discs.

There is a small, simple lock, held by a screw, which effectively locks this switch into the normal mode or the non-write-protect mode but not the read-only mode.

I coupled the drive to an Apple disc control card both as the boot drive and as second drive and used it together with a full-height Apple drive for a period of time. I used DOS 3.3, Pascal, Fortran (running under the Pascal system) and CP/M. All worked well.

The notes (I hesitate to call it a manual – but proper documentation has been promised by the distributor) which accompanied the drive suggested that the Apple should not be switched on with a disc in the drive as corruption of data could occur.

Knowing full well that many computers do offer this little extra whereas the Apple drives always seem safe I tested several discs in this way until I was confident that there was in fact no problem.

After some time I forgot about it and just used it as a normal Apple drive. Only time will tell how reliable it is but up to now I have confidence in the drive, which should be a good buy at £192.

Product: 5.25in disc drive.
Name: Aviette FDD 820.
Price: £192.
Distributor: Tashkl Computer Systems, 24 Logan Rd, Wembley, Middlesex HA9 8PX. Tel: 01-904 4467.
Comment: Seems a good buy.

While on the subject of cheaper peripherals, now on the market is a 40-column, Apple compatible, dot matrix printer – the KGP-40 – also made in Taiwan.

It costs £110 including the interface card and cables. It uses standard typewriter two-colour ribbons and 2.7in tally roll paper, so is only really suitable for listings, not for word-processing. Unless, that is, you want to send letters of complaint to your local bus company.

It uses a 7 x 5 matrix with seven needles on the print head which has a claimed lifetime of 40 million characters and is replaceable. There is only one size of character available (1.25 x 2.75 mm) but the full Ascii set is supported. The lower case characters do not have true descenders.

The ribbon position may be toggled between upper and lower by a software switch (Ascii 1, CHR$(1) and the interface card carries on board the firmware necessary to do graphics dumps of either hi-res page, in inverse if required, in either red or black.

The two pages may also be logically EORed, ORed and ANDed together when printed.

All of this is accomplished by a POKE to one location to set the modes and the sending of CHR$(17) to activate the dump.

There is a problem, however. The image is printed sideways and the printer can only accommodate 150 dots across the page so only the top of the hi-res screens are printed. See the figure which adequately demonstrates the problem .

The aspect ratio is good and circles really do look like circles.

The printer also offers bit image printing through individual control of the seven needles but it is not the easiest in the world to programme.

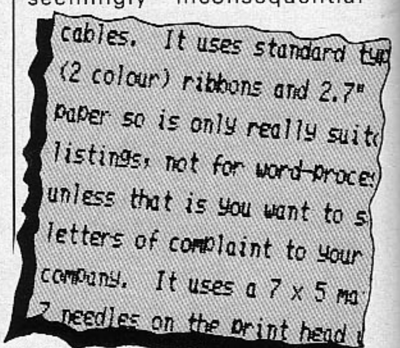
The needles can only be fired or not fired, the line spacing is fixed so it is probably a little easier to control than the Epson printers. The bit image mode is switched in by the sequence ESC,K,n where n is the number of bytes of data following, sent as the Ascii code.

The manual doesn't admit to it but there is a similar mode invoked by the sequence ESC,L,n but it seems to produce the same results.

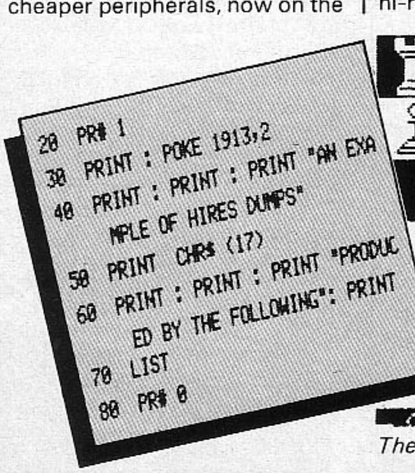
Both of these image modes require 128 CHR$(0)s to be sent after the line of data for some inexplicable reason.

The printer behaved well with many kinds of programs running under DOS 3.3 but hung under the Pascal and CP/M systems, although Metacraft's Forth drove it well.

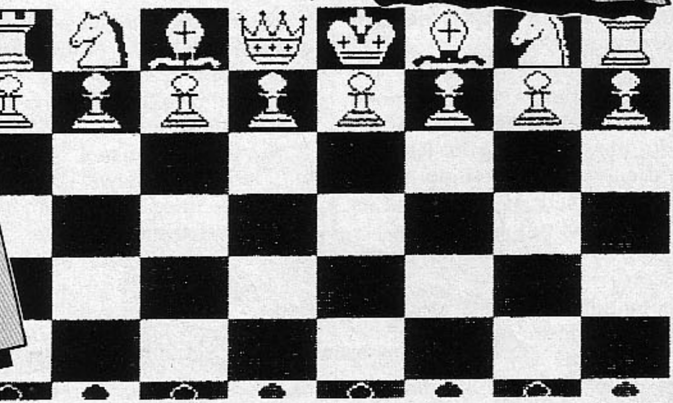
The manual refers to patches being necessary to the Pascal and CP/M system discs and describes how to apply them to the discs using the ZAP utility from "Bag of Tricks". I followed the instructions and the given example which contained a few seemingly inconsequential



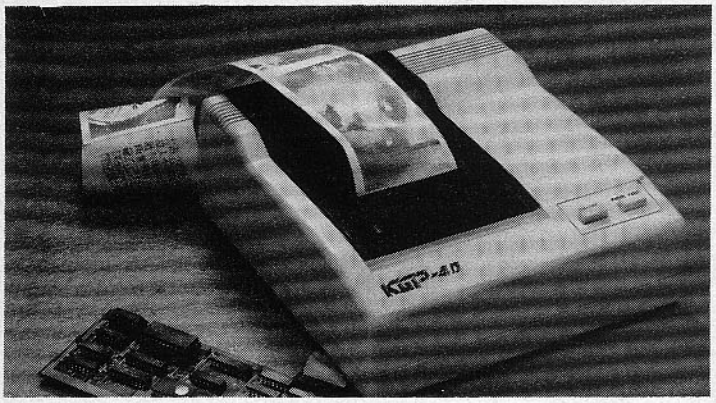
```
cables. It uses standard ty
(2 colour) ribbons and 2.7"
paper so is only really suit
listings, not for word-proce
unless that is you want to s
letters of complaint to your
company. It uses a 7 x 5 ma
7 needles on the print head
```



```
20 PR# 1
30 PRINT : POKE 1913,2
40 PRINT : PRINT : PRINT "AN EXA
   MPLE OF HIRES DUMPS"
50 PRINT CHR$ (17)
60 PRINT : PRINT : PRINT "PRODUC
   ED BY THE FOLLOWING": PRINT
70 LIST
80 PR# 0
```



The hi-res dump above was produced by the program on the left



The KGP-40 dot matrix printer

errors but we could not make the Pascal Filer transfer either a directory listing nor a TEXT file to the printer.

The worm-drive which moves the head went into overdrive and wouldn't switch off without the power switch being invoked. Not wishing to ruin the printer I gave up, although I did check the patched disc using "CIA files" (see Page 40) and it did appear correct according to the printer manual.

These problems aside, the printer seems to give excellent value for use in the home where its main application will be for listings of Basic or assembly language programs or for dumping output from programs such as CIA.

Teaching establishments could also benefit – it should do very well when linked to machines such as the BBC or Spectrum with expansion box.

I hope for the sake of Apple users that this will happen since the more a peripheral is bought the better it is for everybody.

Product: 40-column dot matrix printer.
Name: KGP-40.
Price: £110 (including interface card and cables.
Distributor: Peanut Computer, Unit 22A, Low Mill, Dewsbury, W. Yorks WF13 3LX. Tel: 0924-499366 ext. A.
Comment: Excellent value for home or schools use.

Taiwan has also turned its attention to cheaper peripherals such as joysticks. We have recently seen two which were both good value for money.

The first, the Pride, made under the authority of Pride Computer Products of Canada, features auto-centralising with electrical zeroing in both planes.

There are two fire buttons mounted in the now fashionable back and top left positions for easy one handed control. These are of a decent size – small ones hurt the finger and thumb after a while – and have a travel of about 0.5mm which feels comfortable.

The 10ft-long cord is coiled and sports a plastic anchor which slides into the rear slots of the Apple II. Whether an adaptor, or another version, is available for the Apple IIe I don't know at the moment.

I took the unit apart to see how well made it was and it appeared robust enough.

There are two potentiometers each driven by a plastic piece with a slot cut in it through which the end of the actual joystick passes. That is, doesn't use the gimbal mechanism which the better known Kraft joystick has, although the unit as a whole looks similar superficially.

The Pride certainly gave a very stable performance and always returned to the centre when let go. The one feature lacking was the ability to turn off the auto-centralising and have a free mode which is sometimes useful for drawing type programs.

The name of the second joystick was printed in Chinese, although I'm told it is the AK-GC-3.

It sported four fire buttons which looked very promising but they turned out to be linked in pairs so that the one handed or the two handed approach could be used.

The lead was a substantial 6ft long and anchored into the Apple slots just like the Pride.

Opening it up I discovered a mechanism similar to the Pride's but made of metal. It was not as thick as the plastic version but probably strong enough.

One slightly disturbing thing was that the front two fire buttons were mounted on a small PCB within the case, fixed only by a couple of screws.

When the buttons were pressed the board tended to flex a little although in the time I had it they gave no problem.

The joystick again was very stable and very good at centralising although there are no provisions for electrically zeroing the potentiometers.

With all this talk of foreign manufacturers it was good to see a British product – one made by a new, enterprising group from Cheltenham.

Product: Joysticks.
Name: (a) The Pride AK-GC-5. (b) AK-GC-3.
Price: (a) £25. (b) £13.50.
Distributor: Tashkl Computer Systems (address as earlier).
Comment: Both good value with stable performance and good centering capabilities.

Their first product is a £25 light pen which fits in the games port of the Apple II (again I don't know if there is an Apple IIe version or adaptor).

It gives its input to the Apple through the third – usually unused – paddle button PB2.

This may be a problem if you are using certain 80-column cards and/or lower case adaptors because they may also use PB2 as a shift indicator.

Product: Light Pen.
Price: £25 (mail order only).
Distributor: Young Enterprise, Boyne House, College Rd, Cheltenham, Gloucestershire.
Comment: Only suitable for menu selection or simple games such as solitaire.

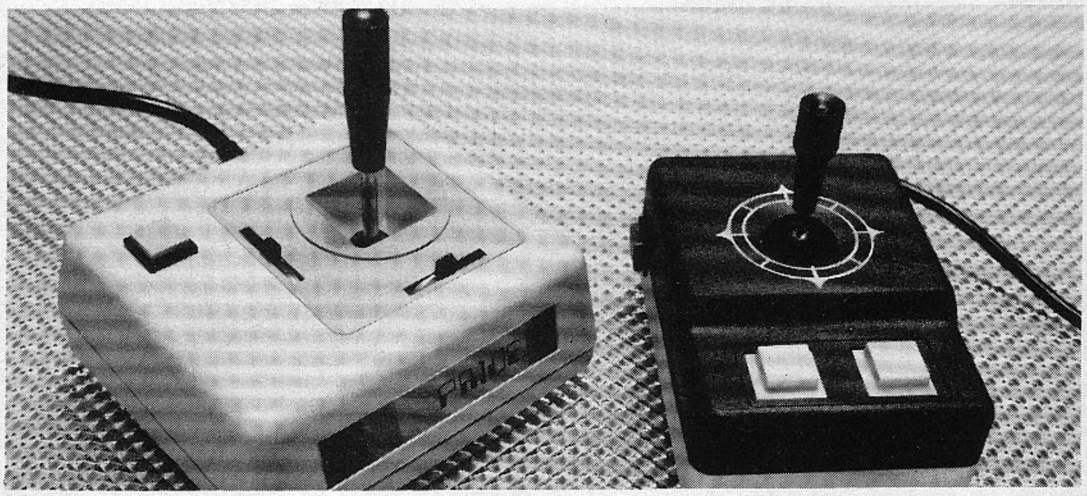
The light pen doesn't preclude their use. It connects to the Apple mother board by a small PCB with quite lengthy pins for clearance and which has a socket on top to take joystick, paddles or adaptors.

Presumably you could use the light pen as a shift lock.

It looks for squares of light on the screen so is really only suitable for menu selection or simple games such as solitaire.

At first it proved too sensitive for my screen but the PCB has an adjustable resistor on it which solved that problem.

I found that screen coverings of the nylon mesh type, designed to cut down on glare, also interfered with the pen's use.



Joysticks distributed by Tashkl Computer Systems

Word counting with Applewriter II

Kevin Farrell shares this quick routine

THIS small utility routine enables you to count the words in any article, such as this one, which has been written using Applewriter II.

It exists on my Applewriter disc as text file "COUNTW" and is simply called by typing CTRL P followed by "DO COUNTW".

The listing in Figure I should be entered using Applewriter to write it and to save it to disc.

It should be entered exactly as given. Note carefully that every line begins with a space character as the first letter.

Assuming that you have Applewriter II up and running, call up the program by typing "CTRL P DO COUNTW".

Reply to the prompt with the name of the text file for which you want to count the words.

In a few seconds you should be presented with a screenful of "˜" signs.

The "LEN" notice in the Applewriter top-line display will display the number of words in your file.

The program actually counts the number of spaces after words and the number of carriage returns, since this is a fairly good representation of the total number of words.

The first line beginning with "p" is a comment only. The second line "ny' is "new yes", clearing the screen.

The third line "pnd" is "no display" which speeds the program up. If this line is deleted the program can be watched working . . . but at a five times slower rate.

The fourth line "pin file name=$a" asks you to input a file name and assigns the reply you give to string a.

The fifth line "1$a" is "load a file with name as just given". The sixth line "b" places the cursor at the beginning.

The seventh line "f<? <?x̃<a" says find any character followed by a space and replace it by the same character and a "˜" do all occurrences, that is replace all spaces by "˜" signs.

The eighth line "?" is a dummy to allow for the find command reaching its end and the ninth line "f<><˜<a" says replace all carriage returns by ˜ signs. ">" stands in for carriage return here.

The tenth line is dummy "?" again. The eleventh "f<=˜<˜<a" says find any piece of text ending in ˜ and replace it by a ˜ sign. That is delete all other than ˜.

The twelth line is a dummy "?" again. The thirteenth "e" puts the cursor at the end of the text, and the fourteenth "pqt" says quit.

In this form a string of spaces is also counted, at the rate of one word for every two spaces. If this is important one way to avoid it is to insert the section in Figure II after the sixth line "b".

Each of these lines says find three spaces and replace with one.

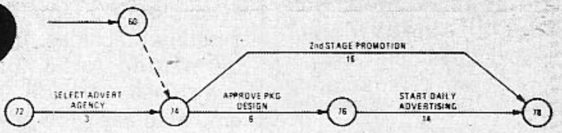
I leave it as an exercise for the reader to decide how many spaces in a row doing this thrice eliminates and why three spaces not two.

```
p  counter of words
ny
pnd
pin file name=$a
1$a
b
f<? <?~<a
?
f<><~<a
?
f<=~<~<a
?
e
pqt
```

Figure I

```
f<   < <a
?
f<   < <a
?
f<   < <a
?
```

Figure II



BEGINNING PASCAL

"Pascal can be described in over-academic terms, so I will keep things pretty informal". That is how GORDON FINDLAY introduces his new tutorial series. It is written for the newcomer to Pascal, although it assumes that the reader has some experience in Basic.

It should get you well on the road towards appreciating some of the more advanced Pascal pointers that Jonathan Lewis will be introducing in forthcoming issues of Apple User.

BLAISE Pascal was a French mathematician and philosopher, born in 1623 at Clermont. He was one of the most important philosophers in an age of great intellectual achievement.

He is remembered for his work in geometry, algebra and physical science as well as philosophy. His greatest work, Pensées, was written to examine his religious beliefs. Pascal died in 1662, aged 39.

What does this man have to do with a sophisticated modern programming language?

Pascal invented the first mechanical calculator in 1641. He was then 19 years old. His aim was to help his father in his work as tax collector – surely a portent of things to come! It is to commemorate this invention that Professor Niklaus Wirth named his language "Pascal" when he introduced it to the world in 1971.

Pascal has often been characterised in the following terms:

It is a structured language. That is, the language reflects the fundamental processes involved in a program in a natural way.

It is a sophisticated, rich language. There are a variety of control structures, and also a variety of ways of structuring data.

Using Pascal, one may handle data in ways which reflect the data, rather than ways which reflect the machine.

It is a consistent language. By this is meant that the language is free of ad hoc constructions and special cases.

It is a language which does not impose its own detail on the programmer. Probably the classic example here relates to arrays.

In Fortran arrays, subscripts must start at 1. In Basic, they must start at zero, even if most programmers ignore the zero element.

In Pascal, the programmer can make array subscripts start and finish at any value he wishes.

It is a powerful language. But it is nevertheless easy to learn.

The standard definition of Pascal was given by Jensen and Wirth in the Pascal Users Manual and Report, 1975.

UCSD Pascal is an implementation of the Pascal language and a powerful operating system developed at UCSD (University of California at San Diego) under Dr Kenneth Bowles.

It is intended to be a portable implementation – that is, to be available for a large number of different machines. This is done by writing for a 'universal' machine, which may be emulated easily on other machines.

Using UCSD Pascal on several machines should not be difficult. It has only minor differences from standard Pascal.

Apple Pascal has a number of extensions added to UCSD Pascal for the Apple computer, such as graphics and sound capabilities and extensions to file handling for interactive programs.

As a beginning to our study of Pascal, here is a trivial, but complete Pascal program which we can use to introduce some important ideas.

```
PROGRAM SIMPLEST;
BEGIN
  WRITE (3+4);
END.
```

When run, this program prints the number 7 – hardly of earth-shattering significance! Let us examine it in more detail.

Pascal is a language which takes nothing for granted, and which expects everything to be declared before it is used. In some languages, Basic and Fortran for example, a variable may be defined simply by using it. In Pascal, this is not so.

A Basic program will detect that a new variable is called for, and create it, making certain assumptions about it. The Pascal programmer must define everything he is going to use.

At first this seems a backward step – but it is done for three reasons:

● The programmer gets complete control over his program. As an example, the programmer may place limits on the legal size of variables.

● Programs become much easier to debug. The Pascal programmer will never (should never!) be caught out by the system making assumptions different from his own.

● Explicit declaration makes it possible to define many different types of data.

In the program, called SIMPLEST, only one thing is declared – the program itself. Programs need a name, which may be any legal identifier (defined below).

The part of the program which does the work is found between the BEGIN-END pair. Here it is simply a WRITE statement, which will print the value on the Apple screen.

Should more than one value be required, they may be simply listed with commas separating them – for example WRITE (3+4, 7+2).

Notice the punctuation. A semicolon after the program name, and a full stop as the last character of all. Punctuation is something we will return to many times!

Now it's time to get a little more daring. Let's look at this program,

```
PROGRAM SIMPLER;
VAR
  I: INTEGER;
BEGIN
  I:= 3+4;
   WRITE(I)
END.
```

This program performs exactly the same task as SIMPLEST, outputting the value of 3+4. Program SIMPLER uses a variable, I.

Variables in Pascal are just like the variables in other languages – Basic included. They act as pigeonholes, storing a value for later use. Variables in Pascal have three important attributes – their name, value and type.

The variable in program SIMPLER has *name* I. This is an example of an *identifier*. A legal identifier is any string of letters or digits, as long as the first character is a letter. Legal examples are INCOME, PLAYER, FYR1982 and so on, but not 3D, or F2.8.

Although the system allows identifiers to be any length at all, it only looks at the first eight characters. This means that INCOMETAXOWED and INCOMETAXPAID would look the same. Be careful!

The variable in program SIMPLER is an INTEGER. This means it can only have a whole number value (positive or negative). The only alternative to INTEGER at this stage is REAL, which permits fractional values.

Every variable must be declared (introduced to the system) in a VAR clause. If you have several integer variables, they may be declared in one list:

```
VAR
   I,J,LIMIT: INTEGER;
```

If you have both real and

integer variables, declare them in this way:

```
VAR
   X,Y,LIMIT: INTEGER;
    WEIGHT, HEIGHT : REAL;
  (note the punctuation!)
```

Pascal is a totally free-format language. The compiler pays no attention to spaces (generally), or to lines. A statement can be spread over several lines, or several statements may occupy one line.

It is to separate statements that the semi-colons (;) are used. The declaration above could be written:

```
VAR X,Y,LIMIT: INTEGER; WEIGHT,
HEIGHT: REAL;
```

but it would be hard to read!

Note that a space is required to isolate keywords (like VAR above).

When they are declared, variables are set up, but have no value. Most definitely they are not set at zero. In Standard Pascal, variables are set to a special 'undefined' value, but in Apple Pascal they will (sometimes) be given a random value. But don't rely on it.

The assignment operator, := is used to give values to variables. The example in the program SIMPLER gives I the value 7. There should not be a blank between the : and the =. Reading the symbol is a problem. Try reading "A:=3" as "A gets 3".

Arithmetic operators are fairly standard in most programming languages: +,–,* and /. Pascal has no operator for powers in general – stay tuned for a way round this – but for the special case of squares, use SQR(X).

For example, SQR(3)=9. (Basic programmers beware!) The order of arithmetic operations is standard algebraic, but parentheses may be used to modify this as desired.

Some sample assignment statements:

```
X:= X+1
X:= X
X:= B*B-4*A*C
AREA:= 0.5*BASE*HEIGHT
```

Suppose X is a real variable and I is an integer variable, and we perform the following statements:

```
I:= 7;
X:= I;
```

Now I has the value 7 (an integer) and X has the value 7.000000 (a real, remember, has a fractional part). No problem here.

But what happens in this case?

```
X:= 2.718
I:= X
```

It cannot be 2.718 – an integer cannot have a fractional part. So what happens? In Basic, and most versions of Fortran, the system decides. But in Pascal, all the control is left in the hands of the programmer. You must decide!

The statement as it stands is in error. No assignment of the form <integer>:=<real> is permitted. The programmer has options – see the functions TRUNC and ROUND, covered later.

The restriction mentioned above leads to an interesting problem with division. Suppose we have two integers I (with value 7) and J (with value 5). Then I/J is a real number 1.4. Combining two integers gives another integer, except in this case of division.

To give a facility for integer division there is another arithmetic operator provided, DIV.

DIV is an integer division operator – it takes two integers only and returns an integer result – 7 DIV 5 = 1; 9 DIV 10 = 0. (The fractional part is thrown away, not rounded.)

This is a skeleton of a simple Pascal program. We will use it as a framework to build on.

```
PROGRAM name ;
VAR               )
  ....  :INTEGER; ) declarations
  ....  :REAL;    )
BEGIN
    .             )
    .             ) statements
    .             )
    .             )
END
```

A program isn't much use unless it can communicate with the user, but input-output is a major topic, with many options available to the Pascal programmer.

At this stage, we will use just four simple 'procedures' (Pascal's name for a built in routine).

READ (A,B,C . . .) gets values for A,B,C in turn from the keyboard. Each value typed in should be separated by a space. After all the numbers are entered, press RETURN.

READLN (A,B,C . . .) gets values in just the same way as READ (A,B,C . . .) but at the conclusion, terminates the line.

WRITE (A,B,C . .) outputs the values of A,B,C . . on the current screen line (ie, at the right of whatever is already there).

WRITELN (A,B,C . .) acts just like WRITE, but then terminates the screen line – the next output will be on a new line.

There are a few traps in these procedures:

● No spaces are printed between integers. If a program contains the lines

```
A:= 1;  B:= 2; C:= 3;
WRITE(A,B,C)
```

then the output produced is 123. This could be confusing!

● Real numbers are spaced, but not always as you would wish. Reals which are either too large or too small are printed in scientific form. To see this, input and run this program:

```
PROGRAM TEST;
VAR
  A: REAL;
  I: INTEGER;
BEGIN
  A:= 1000;
  FOR I:= 1 TO 20 DO
    BEGIN A:= A/2;
          WRITELN(A)
    END
END
```

(Don't worry about the parts of the program we haven't covered yet – just type them in!) The first few, and last several, numbers output are in scientific form. To interpret them, recall that "E" means "times ten to the power of", so that:

1.000E3 = 1.000 x 10_3 **= 1000**
and
5.000E^{-4} = 5.000 x 10^{-4} = 0.0005.

● Any spaces you want between output numbers can be

included between single quotes, like WRITE (A, '   ', B)

Of course, anything else can go between the quotes instead of blanks; WRITE ('PLAYER', I) will give (if I has the value 4) PLAYER 4.

● To 'finish off' a screen line use WRITELN; (no arguments required). If the screenline had already been terminated by another WRITELN (perhaps with arguments) then this will produce a blank line.

The input-output procedures need careful study. Try some simple test programs to see how they work.

Comments may be included in Pascal programs by surrounding them with (*   *). Anything within these "comment brackets" is ignored by the compiler.

This has been a fairly heavy introduction. Before we go further on our tour of Pascal you should try simple calculations and input-output using the parts of Pascal we have visited.

TO run Pascal on the Apple IIe requires at least one, but preferably two disc drives, and the Apple Pascal software which is obtainable from dealers for £149 (UCSD Pascal version 2.1).

It's use on the older Apple II machines requires a language card – the Apple card costs £106 – in Slot 0.

The Pascal software for the Apple III costs £175. But note that you cannot run Apple II Pascal on the III in emulation mode.

More and more commercial application packages, particularly for business use, are being written in Pascal. However these should be "self-contained" in that they do not require the Pascal software as an extra expense.

FREE HI-RES TEXT GENERATOR!

No, don't write in – M. BOWYER gives some helpful advice on how to use your existing character set

I HAVE read many articles about hi-res text generators. How to get one? Can I write one?

No one seems to have realised that they can get a perfectly adequate text generator for free.

In fact, everyone who owns a DOS 3.3 System Master – supplied with every Apple system sold – already has one!

It is simplicity itself to pull it out of the Applevision program.

First boot the System Master and wait for IntBasic to load (if you have a II Plus you will need a language card for this).

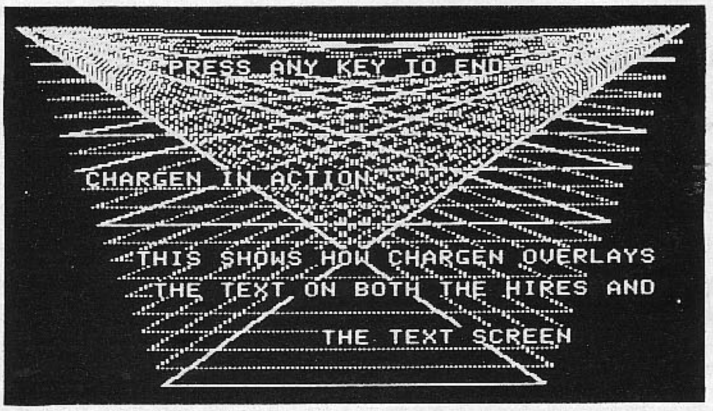
RUN Applevision and when the graphics start, break in by pressing CTRL-C.

Type:

BSAVE CHARGEN,A$847, L$346 (Return)
FP (Return)

and then LOAD and RUN the second program after re-setting the program pointers (103, 104 decimal) to by-pass the hi-res area as in the last few lines of Program II.

This allows the characters to be viewed and re-made in their internal order (as listed). It also creates an EXEC file to change the Basic pointers, load your generator, and start Basic.



```
PRESS ANY KEY TO END
CHARGEN IN ACTION
THIS SHOWS HOW CHARGEN OVERLAYS
THE TEXT ON BOTH THE HIRES AND
THE TEXT SCREEN
```

This is necessary as the generator starts so low in memory that even a small program cannot be fitted in.

This part of the program (from line 560) should be run first to set up an EXEC file to load it properly.

The generator writes to HGR and this can easily be extended to full-screen by POKE –16302,0.

To set it up POKE 54,0:POKE 55,11.

The characters are stored between 2304-2816 decimal.

Each character takes up eight bytes with the bits a mirror image of the screen bits (and the two most significant bits unused).

This is allowed for in my program while re-making the characters.

The short first program is a demonstration of how to use your character set.

Program I

```
5  IF  PEEK (8 * 256) = 56 THEN 2
     0
6  POKE (8 * 256),56
10  PRINT  CHR$ (4)"BLOAD CHARGEN
     "
20  HGR : POKE  - 16302,0
25  HOME
26  GOSUB 90
30  POKE 54,0: POKE 55,11
40  VTAB 10: HTAB 5: PRINT "CHARG
     EN IN ACTION"
50  VTAB 20: HTAB 19: PRINT "THE
     TEXT SCREEN"
60  VTAB 15: HTAB 8: PRINT "THIS
     SHOWS HOW CHARGEN OVERLAYS"
65  VTAB 3: HTAB 10: PRINT "PRESS
      ANY KEY TO END"
70  VTAB 17: HTAB 9: PRINT "THE T
     EXT ON BOTH THE HIRES AND"
80  VTAB 23
84  FOR A = 1 TO 1000: NEXT
87  GOTO 140
90  FOR A = 1 TO 20
100 F =  INT ( RND (1) * 7 + 1): IF
     F = 0 OR F = 4 THEN 100
105  HCOLOR= F
110  HPLOT 0,0 TO 279 - (3 * A),A
      * 9 TO 3 * A,A * 9 TO 279,0
120  NEXT
130  RETURN
140  TEXT
150  FOR A = 1 TO 1000: NEXT
160  POKE  - 16297,0
165  POKE  - 16304,0
166  IF  PEEK ( - 16384) > 127 THEN
      GOTO 180
170  GOTO 84
180  POKE  - 16303,0: POKE  - 163
     68,0
190  TEXT
195  VTAB 20
200  PRINT "TYPE 'TEXT'"
```

Program II

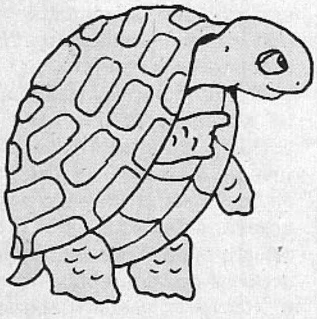
```
4  GOSUB 400
5  IF  PEEK (8 * 256) = 56 THEN 50
6  POKE (8 * 256),56
7  PRINT "     LOADING CHARGEN"
8 CH$ = "BRICHARSET"
10  PRINT  CHR$ (4)"BLOAD "CH$
15  GOTO 50
20  HGR : POKE  - 16302,0
25  HOME
30  POKE 54,0: POKE 55,11
40  RETURN
50  GOSUB 300: HOME
53  PRINT " 1 = ON, 0 = OFF, <- =
      RESTART LINE.": PRINT : PRINT
     "RETURN FILLS LINE WITH LAST
     CHARACTER"
54  VTAB 14: HTAB 16: PRINT "NO C
     HAR "
55  FOR A = 6 TO 13
60  VTAB A: HTAB 16: PRINT "*****
     ****"
70  NEXT
80  FOR A = 6 TO 13
90  FOR B = 16 TO 23
95  VTAB A: HTAB B
100  GET A$: IF A$ <  > "1" AND A
     $ <  > "0" AND A$ <  >  CHR$
     (13) AND A$ <  >  CHR$ (8) THEN
     95
110  IF A$ = "1" THEN  INVERSE : PRINT
     " ";:A(A - 6) = A(A - 6) + 2
      ^ ( - 16 + B)
120  IF A$ = "0" THEN  NORMAL : PRINT
     " ";
130  IF A$ =  CHR$ (13) THEN  PRINT
      SPC( 23 - B):Q = B:B = 23: IF
     B$ = "1" THEN  NEXT B: FOR B
      = Q TO 23:A(A - 6) = A(A -
     6) + 2 ^ ( - 16 + B)
140  IF A$ <  >  CHR$ (13) THEN B
     $ = A$
150  IF A$ =  CHR$ (8) THEN B = 1
     5:A(A - 6) = 0
200  NEXT B,A
210  NORMAL : PRINT
```

GRAPHICS HI-RES TEXT

```
214  PRINT : PRINT : PRINT
215  PRINT "NUMBERS TO POKE IN LO
     CATION ";LC: PRINT "ONWARDS.
     ": PRINT
220  FOR A = 0 TO 7: PRINT A(A)",
     ";: NEXT
223  TEXT : PRINT
224  PRINT "PRESS ANY KEY"
225  GET A$
230  FOR A = LC TO LC + 7: POKE A
     ,A(A - LC):A(A - LC) = 0: NEXT
240  GOSUB 20
250  FOR A = 64 TO 95: PRINT  CHR$
     (A);: NEXT
260  FOR A = 32 TO 63: PRINT  CHR$
     (A);: NEXT
270  PRINT : PRINT : PRINT "ANY K
     EY:";: GET A$
280  TEXT : HOME
290  PRINT "ANOTHER CHARACTER? (Y
     /N):";: GET A$: PRINT : IF A
     $ = "N" THEN 500
295  GOTO 50
300  PRINT : PRINT : PRINT "WHAT
     IS THE NUMBER OF THE CHARACT
     ER YOU WANT TO CHANGE?"
310  INPUT ":";NM
315 NN = NM
320  IF NM < 32 THEN NN = NM + 64
330  PRINT " USED TO BE "; CHR$ (
     NN);" ? (Y/N)";: GET A$: IF
     A$ <  > "Y" THEN  PRINT : PRINT
     "TRY AGAIN";: GOTO 310
340 LC = 2304 + (8 * NM)
350  RETURN
400  HOME : TEXT : HOME
405  INVERSE
410  PRINT  SPC( 52);"CHARACTER M
     AKER"; SPC( 52): NORMAL : PRINT
     : PRINT : PRINT
420  RETURN
500  PRINT : PRINT : PRINT "SAVE
     THIS CHARGEN? (Y/N):";: GET
     A$: IF A$ <  > "Y" THEN  END
510  PRINT : INPUT "FILENAME FOR
     CHARGEN: ";A$
515  POKE 43698,192: REM  CHANGES
      ^D TO @
520  PRINT : PRINT "@BSAVE "A$",A
     $847,L$390"
530  PRINT "@VERIFY "A$
540  PRINT "     SAVED & VERIFIED
550  PRINT : PRINT "MAKE SET-UP E
     XEC FILE? (Y/N):";: GET A$: IF
     A$ <  > "Y" THEN  POKE 43698
     ,132: END
560  PRINT : INPUT "FILENAME FOR
     EXEC FILE: ";A$
564  PRINT : INPUT "FILENAME OF B
     ASIC PROGRAM: ";B$
565  POKE 43698,192
570  PRINT "@OPEN "A$
575  PRINT "@DELETE "A$
576  PRINT "@OPEN "A$
580  PRINT "@WRITE "A$
590  PRINT "FP": PRINT "HOME:POKE
     50,128"
600  PRINT "POKE103,1"
610  PRINT "POKE104,96"
620  PRINT "POKE24576,0"
630  PRINT "LOAD ";B$
640  PRINT "CALL54514": PRINT "PO
     KE50,255"
650  PRINT "RUN": PRINT "@CLOSE"
660  PRINT "     EXEC "A$" MADE F
     OR "B$"."
670  POKE 43698,132: END
```

When the future arrives, be ready for it

Christopher Roper reports from the USA on the lively Logo scene



LOGOLAND is alive and jumping with exciting new ideas and techniques – and the language promises an adventurous intellectual ride to anyone who decides to learn it now.

That is the main message I have heard on both the East and West coasts of America.

My three week long visit there recently kept me away from my Apple II – and as a result I was unable to develop my next Logo project. That must wait until next month.

Meanwhile, since I have been visiting MIT (Massachussetts Institute of Technology) where Apple Logo was born, and Silicon Valley, where the next generation is already being conceived, I thought it would be worth talking about what can, and should be done next.

Here are some of the ideas I have noted:

Boxer: The Boxer Project at MIT is led by Hal Abelson and Andy DiSessa, co-authors of a famous (and difficult) work on Turtle Geometry.

Abelson himself wrote an excellent introduction to the Apple Logos for McGraw Hill.

The core idea of Boxer is that the programmer creates boxes (they look like Lisa's windows) which then become objects for simple manipulation.

The boxes may contain programs, data, pictures, or templates.

The idea is to extend the power of Logo by providing the user with a powerful visual metaphor, analagous to the original turtle, but extending beyond the domain of graphics.

It is notorious that Logo becomes a lot more difficult when you leave graphics and begin conventional list processing.

Boxer may not exist as a product for two or three years. Abelson and DiSessa are academics not entrepreneurs.

But Apple Logo came from MIT and the Boxer project addresses well-known and serious problems. It will undoubtedly influence future versions of the language.

Wordprocessing: A number of different people (my lips are sealed) are working to include wordprocessing as a part of the language environment.

This should allow the user to mix text and graphics in a document, and to create new primitives for editing and formatting documents.

Several household-name companies think this may be the road to a truly integrated and user-friendly computing environment.

It clearly won't be possible on machines with memories as small as the Apple IIe, but one thing we can be sure of is that 1mbyte main memories are going to be commonplace on personal computers (including Apples) within four to five years.

Animation: Animation is really not possible on the Apple II Logo at present, but if you want to see what can be done, have a look at the Atari Logo from the same software house (Logo Computer Systems Inc) that produced the Apple and IBM PC versions of the language.

I hasten to add that I am not just feeling envious frustration, as many of the Atari animation features will soon be available on an Apple II, using a special board.

This does mean further expense, but it is cheaper than buying an Atari, which I was tempted to do when I first saw their Logo.

Robotics – Robots are currently all the rage in America. But the robots are either horribly hard to program, or respond only to a limited set of commands.

Logo began its life 16 years ago as the command language for a robot with very limited capabilities – the floor turtle.

Attention is now turning to using Logo as a flexible and simple language for programming robots.

The list could be extended almost indefinitely as Logo provides a perfect front end to complex systems and procedures, perhaps written in machine code.

But it isn't just Logo itself. It is the ideas which flow from Logo. Seymour Papert has long been speaking of the Logo-like languages, or the Logo family of languages.

Alan Kay, now at Atari, who played an important part in developing the language Smalltalk (perhaps the greatest single influence on the Lisa), has a team in Cambridge working on QLogo, at present a wonderful collection of ideas rather than a programming language.

Charles Moore, the inventor of Forth, recognises that his fast and ingenious language needs a populariser. "Someone should do for Forth what Seymour Papert did for Lisp", he told me recently.

By the way, don't let my ravings about the future deter you from buying one of the Apple Logo systems today, if you have an Apple II or a IIe.

They provide excellent environments for learning the language, and the ideas behind it, even though they won't solve all your programming needs. When the future arrives, you will be ready for it.

Visiting the America of today reminded me of my first visit almost 30 years ago, when the country was reeling from the shock of the Sputnik.

It was an affront to the American system of education and apple pie, and a number of other things, which are truly Good and American.

I tried to console my American hosts at the time by saying that all it showed was that the Russians' German scientists were better than the Americans' German scientists – but that was considered bad taste.

Today, the same sense of panic has been aroused by Japan's Fifth Generation computer project.

It is creating a furious debate over education and the place of computers in the class room. Artificial Intelligence research is suddenly a hot topic.

But to the despair of many leaders of the artificial intelligence community, whence Logo came and first sprouted wings, the reaction from the schools is to bury children's noses in a Basic primer.

One educationalist, who had recently visited Britain, said to me at a conference in New York: "Really, you are doing it much better in England. Your teachers are so much more open to new ideas".

Well, it was music to my ears, but I really liked some of the software I have seen in the US – especially the advanced Logos mentioned above.

Many people buy an Apple simply to do a job for them – and remain ignorant, uninterested or afraid of how the system performs its task.

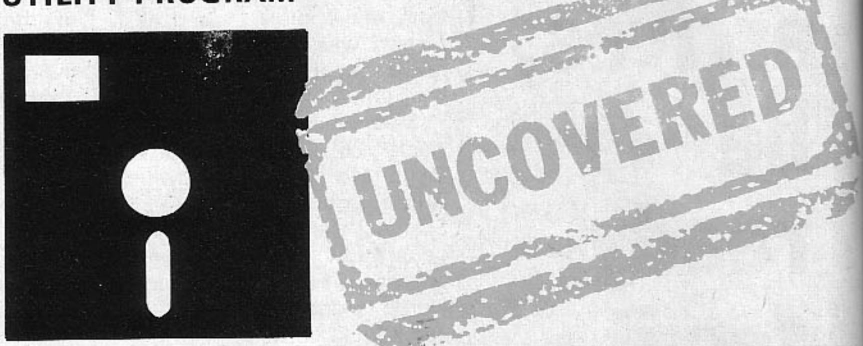
For the more adventurous, delving into the world of floppy discs and the ways that information is stored on them will prove fascinating and extremely useful.

Here we uncover The CIA – a remarkable British software package which in turn allows users to delve into the mysteries of the Apple disc.

We rate it alongside Bag of Tricks as an essential part of an Apple user's repertoire.

The CIA

UTILITY PROGRAM



THE Confidential Information Advisors Package is a tremendous aid for anyone who wants to know more about discs, DOS and that most shadowy of areas – disc protection methods. The name summons up images of espionage agents and secret codes, this connection being obviously intended.

However in some ways this is unfortunate since CIA is also a powerful disc editing/inspection package.

When working on normal discs the commercial program that has set the standard by which all others are judged is Bag of Tricks – from the authors of Beneath Apple DOS.

CIA has a great deal in common with Bag of Tricks, although the latter does still have the edge with the vast number of facilities it offers.

However, if you want to investigate the world of protected discs there is no comparison – CIA is the one for the job.

Until now if you had wanted to inspect, repair or patch protected discs the task was formidable and could be achieved by only a few cogniscenti.

Now, with a little perseverence, the fascinating world of protection schemes can be revealed to anyone who wants to know – much to the consternation of the experts.

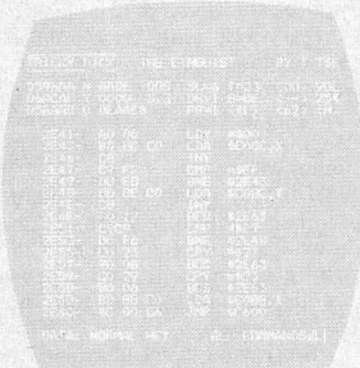
This review is not the place to describe the layout of a typical disc and I will assume a certain degree of familiarity with them.

If the terms track, sector, nibble and VTOC are meaningless to you then a quick glance at Appendix C (or at least its first page) in the Apple DOS manual supplied with your system should help.

The CIA package consists of a disc and manual of over 60,000 words.

The disc is not protected, although it does contain one protected and one half track for tutorial and practice purposes.

All but these two can be copied using FID on the System Master disc.



The manual adopts a tutorial approach. Work through it in the set order if you want to make sense of everything.

The text is well written with several amusing asides.

In principle the manual is self contained but in practice the novice would find some of it hard going.

I personally found it essential to have Beneath Apple DOS handy for more thorough descriptions of the basic disc layout and file structures.

The real worth of the CIA manual is twofold. Firstly it provides comprehensive instructions for using CIA and secondly it reveals a lot about the techniques currently used to protect discs – much of which you won't find written down anywhere else.

The disc contains five utility programs: Tricky Dick, Linguist, Code Breaker, Tracer and Tracker, which can be selected from a menu which comes on-screen.

All the utilities except Tracker must be used in conjunction with Tricky Dick.

Tricky Dick is the heart of the system and is really a master driving program.

Its function is to read a single sector from the disc and to display its contents.

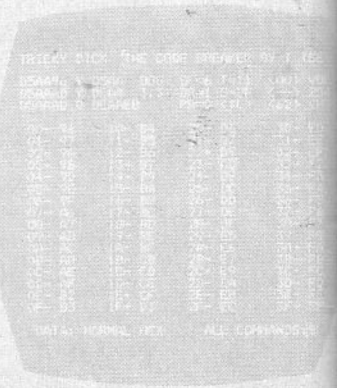
These contents can then be altered and written out again if desired.

If you're dealing with a normal disc (DOS 3.3 or 3.2) the default parameters are all that's needed. The fun begins with protected discs.

Tricky Dick allows you to alter completely address and data field headers, trailers, checksums and encoding format.

It also permits "don't care" values to be substituted for each of these. This means that any disc which uses normal length sectors (just about every disc I've ever seen) but a modified DOS, can be read in easily.

Once the disc has been read in the hex values of the sector contents, along with their Ascii equivalents, are screened.



The sector can also be listec as a machine code dissassembly, Applesoft Basic o Integer Basic program.

It is thus possible to pick up a sector from an Applesoft program, list it to the screen, see what it does and, if you know enough about Applesoft tokens edit it directly and write it back out.

Printer dumps of all these car be obtained at any time anc proved invaluable for long investigations.

Each program has a help screen, selected by the /? key which speed up the learning process considerably.

However, what happens i

you don't have the slightest idea what the sector marks (headers, trailers etc.) are? In this case you need the Linguist.

Linguist can be selected at booting or BLOADed at any subsequent time. It is entered from Tricky Dick (CTRL-E) and provides a raw track dump of any specified track (or half-track).

In essence this allows any track from any disc to be grabbed into memory for inspection.

The content of the track in raw nibble form is displayed on the screen and several comands can be used to wander about the track looking for useful bits of information.

When you think you understand the changes required a CTRL-C puts you back in Tricky Dick ready to alter the sector marks accordingly. If that doesn't work it's back to the Linguist for a more thorough look. This going to and fro can take some time.

I found the Linguist extremely easy to use and the chapter on it very informative.

The only addition I would like is a command to move the cursor by a sector's worth of nibbles at one go.

The Code Breaker is accessed in the same manner as the Linguist but only one module, other than Tricky Dick, can be active at any one time. However swapping between them is fairly easy.

This module allows you to change the translate tables used by the Read-Write-Track-Sector (RWTS) routine in DOS (3.3 and 3.2).

The tables arise from hardware limitations in writing to discs, which necessitate "translating" the 256 possible values of a memory byte into a form using only 64 special bytes.

Some software protection schemes involve swapping the table values around so that a normal DOS can make no sense of the information on the disc.

This may sound very arcane, but the manual explains the procedure very simply and a short exercise shows how to set up your own protected disc using just such a method.

The Code Breaker plays no active role in breaking such a scheme, it merely allows the user to change the values. However, since usually only a couple of changes are made this is very quick.

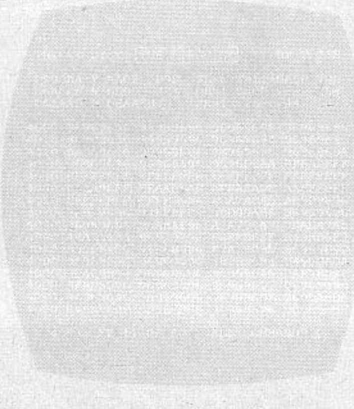
A particularly common method (used by Bag of Tricks, for instance) is to swap D5 and D6, since this allows the sector marks to be altered too.

The Tracer is a particularly useful utility which lets you search selected areas, or the complete disc, for various types of information.

It is not unusual to find more than one set of sector marks on a protected disc and one can VERIFY which track/sectors conform to a given set of values.

The disc can also be scanned for VTOC, CATALOG and TRACK/SECTOR lists when these have been moved from their usual locations.

Finally up to six strings, in hex, low Ascii or high Ascii, can be searched for simultaneously.



On finding a string the Tracer returns to Tricky Dick with the cursor over the located string ready for editing – a CTRL-E returns to searching.

The Tracer is indispensible for patching discs where it can be used to find all occurrences of particular messages, text or code.

It's just a shame that a global editing facility isn't linked to this.

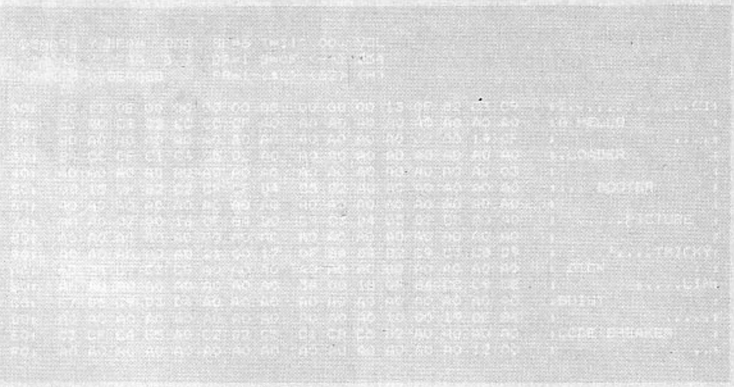
The last of the utilities, the Tracker, is perhaps the least useful in that it only works with a slightly modified DOS 3.3.

Its job is quite different to the other programs since it isn't used to inspect the disc.

When it is installed it provides a record on the screen of the tracks and sectors visited by the disc head every time the disc is accessed.

In principle this could be very useful, but in practice one often can't install the Tracker and then run a protected program.

This is either because of the difficulty of interrupting the boot process or because the DOS is too different for the Tracker to intercept it properly.



A CATALOG sector from CIA

This sums up the major features of CIA, but what about its uses?

I've already used mine to patch protected discs, facilitate backing up important software, recover a corrupted (and very long) Word-Handler file for a friend, create my own protected discs and to transfer a Basic program from a protected disc to a normal one to enable several changes to be made to it.

This last operation is rather tedious since CIA is ***not*** a copy program and although it can read and write with different sector marks it will only do it sector by sector. Transferring a long file requires a lot of loading and saving.

One of CIA's uses is undoubtedly to provide information for backing up protected discs and for this a bit copier with comprehensive parameter changing options is essential.

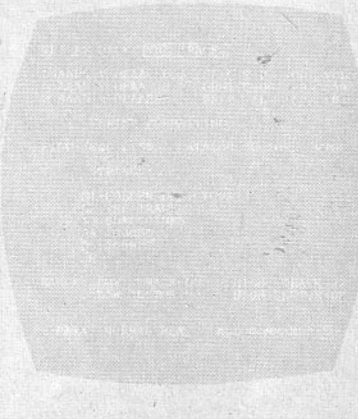
CIA does make the task of choosing those parameters very much easier.

The manual deals exclusively with Apple DOS, although CIA will also work fine on Pascal and CP/M discs – yet there is no information about the layout for these.

It offers several tips for creating your own protection schemes. Some reside on the protected track and are the reward for breaking the protection used.

There is, however, almost no discussion of protection methods that rely on timing dependencies, which is a shame since so many of the new schemes opt for this method.

Although there are a few additions I would like to see to CIA – automatic file transfer, a hex calculator, easier half track access – overall I think it one of the best packages I have bought.



It has earned its place in my box of essential discs which travel around with me.

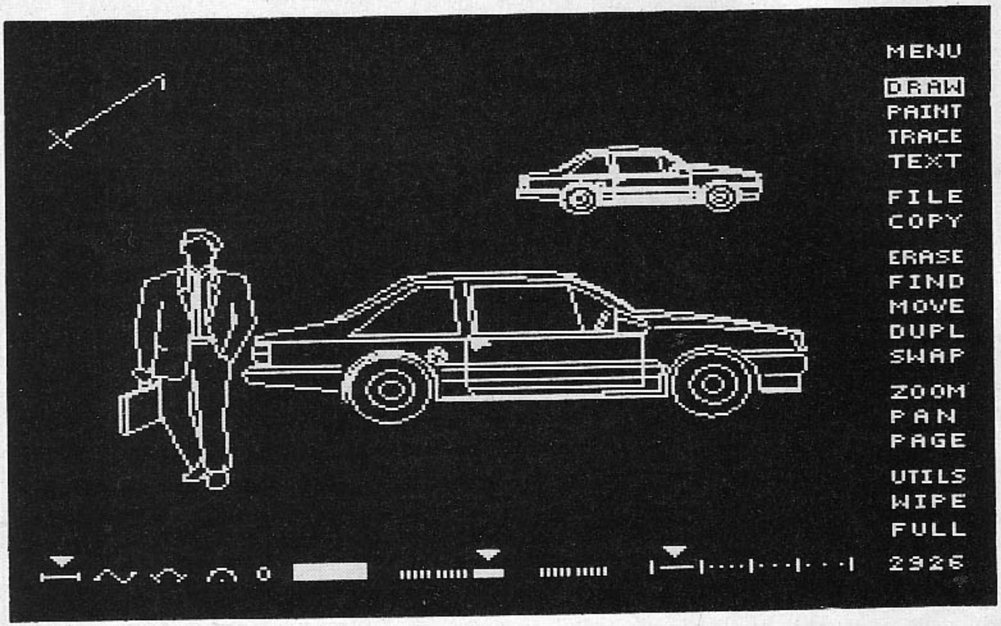
The professional will find it an indispensible tool and with it the newcomer will soon be doing things he never thought possible.

Peter Gorry

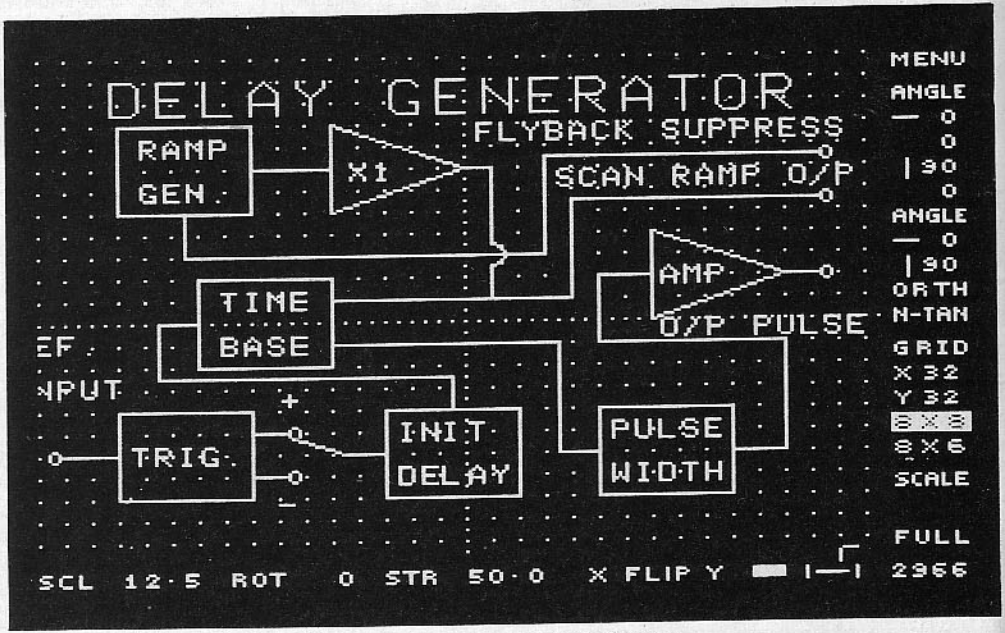
Title: The Confidential Information Advisor (CIA)
Price: £35 (mail order only)
Author: James Martin
Publisher: Golden Delicious Software, 7 Sloane Square, Chelsea, London SW3 3JD.
Requirements: Apple II using either DOS 3.2 or 3.3
Description: Five utility programs that teach you about discs and the Apple Disc Operating System and enable you to examine and modify programs on normal and protected discs.

Now Bit Stik comes up to CAD standards

PETER GORRY starts a new regular column by looking at what is probably the most advanced Computer Aided Design package available for the Apple.



Typical screen during DRAW



SCL 12·5 ROT 0 STR 50·0 X FLIP Y

Here the GRID mode is used to produce an electrical diagram

IN the November 1982 issue of Windfall, I reviewed the Bit Stik graphics system from Robocom and while enthusiastic was unsure about who its potential users were.

It had a number of crucial limitations which prevented it from fully coming up to the level required of a professional computer aided design (CAD) system – although it certainly came the closest to one that I had seen on any micro.

At first glance, version 2 seems very similar to the original. But once in operation it rapidly becomes apparent just what a major overhaul Robocom have carried out.

The whole system has a professional feel about it that the first one lacked. Almost all the earlier objections have been impressively dealt with.

This brings it into the useful CAD market proper, and I shall be applying the somewhat tougher criteria which that implies when discussing it in detail.

The system still requires a 64k Apple and two disc drives as a minimum. Plus, of course, a printer and plotter for serious work.

The most immediately obvious change is that the Bit Stik is now protected by a hardware dongle which plugs into the game socket, and the master discs are copy-protected.

An accelerator card was supplied for the review and, although one isn't crucial, it makes a real difference to the response time.

I became aware of just how much better an accelerator makes it when the card died on me and I had to go back to normal speed.

The basic philosophy of the system is still the same, the keyboard is hardly used and drawing options are chosen from menus along the bottom and side of the screen.

The earlier review went into most of the options in some detail, so I won't repeat them here. Rather I will concentrate on changes to the system.

The first change is that the menu options are not exactly the same. This is especially obvious in the wider variety of line types now available.

Robocom have also made menu selection easier, particularly in "lock" mode, which in the earlier version seemed to require the manual dexterity of a master pianist.

The line types now available are: straight lines, circles, tangent arcs and compass arcs. Of the last two, the former blends arcs into straight lines, while the latter draws sections of circles as a pair of compasses might.

There is still a choice of six colours and various dotted options as well as solid.

The most striking new feature of all comes the moment one starts drawing. The new software adopts a technique called "rubber banding".

The name comes from the similarity to having a rubber band attached to the end of a pen nib (with the other end fixed to the origin of the line). As the pen is moved around the paper the rubber band "shows" the line that would be drawn between the two points.

The great advantage of this technique is that it allows one to see the result of a line before planting it on the drawing.

The technique is used on all line types and is an almost magical improvement when drawing arcs. (I effectively gave up on arcs in the old version.)

The arc/circle algorithm is the fastest one I've seen on the Apple and is another example of the effort that has been put into the new version.

The NIB mode is also much updated, allowing a better control of nib types, and they are now lockable to a grid, which improves their usefulness considerably.

The facilities for putting text on the screen were somewhat elementary before, and they too have undergone improvements.

It is now possible to put text directly onto the screen in four possible orientations (any angle is possible via library units) with the text size constrained to sensible fixed increments.

I found previously that, with any size available, a final drawing often ended up with text that was noticeably disparate in size.

Unfortunately Robocom have not tackled an important weakness – that is the inability to edit text effectively.

Text, as a block, can be removed from the drawing and then retyped, but there is nothing more annoying than to plant a block of text only to notice a simple spelling error – requiring the whole block to be repeated.

This is especially frustrating if the Bit Stik is being used to produce complex data sheets or illustrated instructions – a task it is otherwise well suited to.

The problem can be lessened by breaking the text into smaller units so that only parts need to be replaced, but this is fighting a system rather than having it work for you.

The most impressive part of the Bit Stik system is its library facility.

This allows one to build up a pictorial data base of drawings, parts of drawings, useful components, etc, for incorporation into later work at any scale or orientation.

This part of the system has been considerably strengthened and is now a really powerful CAD feature, so it's worth going into the new options in more detail.

Firstly, the library filing facilities themselves are improved with the option to label units if desired – a big advantage if you have a lot of similar components with only small changes of detail.

Secondly, anything can now be saved as a library unit. The old system only allowed "new data" to be saved. This meant that any drawing that incorporated a library unit couldn't simply be popped back into the library.

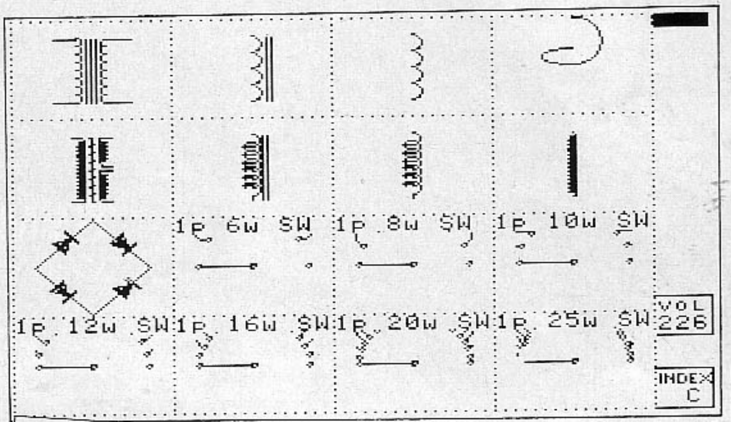
The removal of this restriction much improves the convenience to the user, since it provides for easy editing of library units and allows development versions of a complex drawing to be kept as security.

There is, however, a potential danger in saving a hierarchy of drawings building up to a final one – deleting or overwriting an early unit can be disastrous.

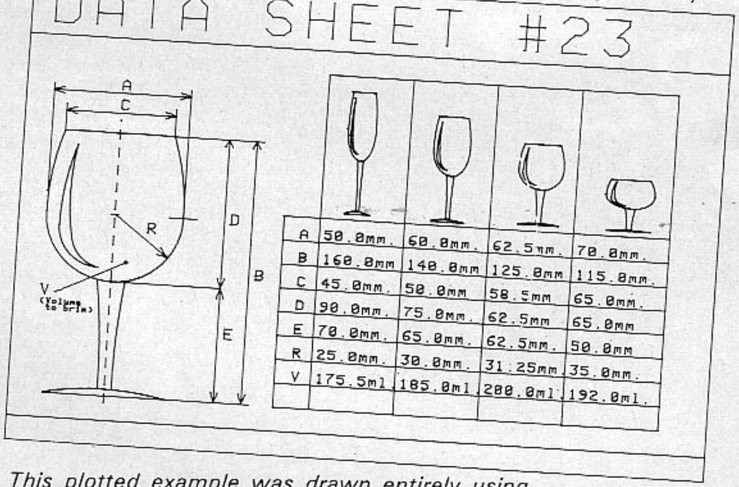
This is because the database structure on the disc, in order to minimise storage requirements, only stores references to earlier units and picks them up at drawing time when required.

So if you overwrite a unit that used to be a transistor with a rubber duck all your circuit diagrams will look like bird sanctuaries!

Such a system has strengths and weaknesses. It does mean that "global" replacement of units is automatic, so that updating drawings is very simple. But if you only want an isolated change it becomes more difficult.



A library file



DATA SHEET #23

A	50.0mm.	60.0mm.	62.5mm.	70.0mm.
B	160.0mm	140.0mm	125.0mm	115.0mm.
C	45.0mm.	50.0mm	58.5mm	65.0mm.
D	90.0mm.	75.0mm.	62.5mm	65.0mm
E	70.0mm.	65.0mm.	62.5mm.	50.0mm
R	25.0mm.	30.0mm.	31.25mm	35.0mm.
V	175.5ml	185.0ml	200.0ml	192.0ml.

This plotted example was drawn entirely using the Robo 1000 computer aided drafting system

There are two methods of making such one-only changes, depending on what you require. The simplest uses the SWAP command (see later) but if the unit is deeply nested the process becomes a real horror story.

Obviously some extra help in this area would be a big improvement.

There are several new features when manipulating library units on their way to the drawing.

The first is that the COPY command now gives information on the scale, rotation and stretch of the unit, each of which can be locked to some chosen value very simply.

There is also a FLIP option which allows the x or y coordinates to be flipped in direction, instantly producing horizontal or vertical mirror images of the unit. This might not sound too useful at first but is essential in, say, circuit board design.

This is because, having drawn one side of the board, it is necessary to ensure perfect registration of the images when drawing the other side. Now this is much easier, since turning the board over is just a FLIP command.

Other features are:

- ● MOVE, which allows one to move a library unit if it's been planted in the wrong place,
- ● SWAP, which enables a library unit to be swapped for another one in the library, and
- ● DUPLICATE, which copies a unit that is already on the screen instead of having to go back to the disc for it (which is much slower).

This last option can be particularly useful if one creates a "local" library of frequently used units at one side of the screen at the start and erases them at the end.

The ZOOM feature has now been augmented by PAN which allows wandering over the picture at the current zoom magnification and makes working on complex pictures much easier, although I did once manage to lock the system up in pan mode. But since I couldn't repeat the trick I don't know how I did it.

We now come to the area which has undergone the most thorough revision and is the heart of transforming Bit Stik to a true CAD tool – the LOCK and GRID facilities.

In the review of the earlier version I wrote, "I developed a love/hate relationship with the DIGIT/LOCK mode. Its uses are enormous but I found more often than not that I ended up fighting it".

I can say that the new version overcomes all the previous complaints and offers a lot more besides.

I found it very easy to use and well thought-out, although even now there are some additions which would improve its flexibility.

The idea is still the same. The motion of the cursor is confined to set angles or grid points to enable very accurate positioning and line drawing to be achieved.

In this way drawings of technical drafting quality can be constructed which are not limited by the resolution of the Apple screen.

The constraints fall into two main types, **angle locks** and **grid locks.**

There are four angle lock features which define the two directions the cursor is free to move along.

The first one allows x and y to be defined in degrees and minutes for very accurate work, the second provides degrees only.

The third sets default orthogonal (0 + 90 degree) directions while the fourth provides a normal-tangent lock.

This last one generates a lock which is tangential and normal to the last drawn vector (line or arc) and is particularly useful for blending curves and straight lines.

The grid locks fall into three types. There are two default orthogonal grids of 8x8 and 8x6 screen points (pixels), which are frequently used values.

There is also a user-defined grid where the x and y values can be set in the 4-32 pixel range. Non-orthogonal grids are also easy to generate, since the x and y directions are set by the angle lock facilities.

In this way, virtually any imaginable grid can be generated. When using grids the zoom, pan and copy cursors are all locked onto the grid to ensure that library units are positioned accurately every time.

The origin of a grid can be moved to other locations.

This means that different grids can be used at different places on a complex drawing, giving an immense flexibility to their use.

The manual provides a number of ingenious examples to illustrate this, although one or two are complicated enough to put the ancient Greeks off geometry, let alone mere mortals.

The final mode to consider is SCALE. This is completely new, but was the mode I most hoped would appear in version 2 of the software.

In some ways, I am a little disappointed. I feel it still offers the most scope for further improvement.

This mode allows the user to set an absolute size to the drawing page, ranging from 1mm to 1km. The software then generates a grid at sensible step sizes.

The size of the page under view and the grid size is displayed at the top of the screen and updated under zoom and pan operations.

Zoom also selects sensible grid sizes, and any library units that were created under scale mode are automatically brought across at the correct size.

Scale is obviously indispensable for any task where correct physical size is important, ranging from circuit board layout to architectural plans – but it's a little clumsy to use at the moment.

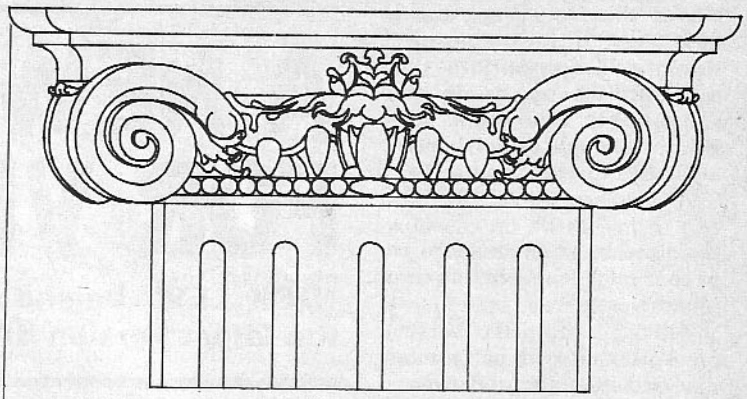
The main problem is that one has to count grid points in order to set the length of a line correctly. There is nothing more irritating after a while (and error-prone) than counting little dots on a screen.

The process can be made easier by imposing a mesh on the page every 10 grid points say – but all that takes time. The problem becomes even more severe on complex drawings working under zoom and pan.

Two additional pieces of information would simplify life so much.

The first would be to know the x, y coordinate value of the cursor relative to some origin (the grid origin for instance).

The second would be to be given the length of the current line, or its x and y lengths, preferably both. These two sets of numbers would do away with nearly all the grid counting.



Both illustrations on this page were drawn on a plotter using the Robo 1000 computer aided drafting system.

While on the subject of weaknesses, there are a couple of further additions I'd like to see.

The first is that compass arcs should display the angle of the segment that has been selected. The present method of using locks to set the angle is very time consuming.

A similar criticism applies to the FIND option, which allows one to set the origin of a line at some point that is already on the drawing.

Unfortunately, lines have two end points, so if you want the other end of the line to meet something on the drawing as well, it's back to grids and locks. If the find mode allowed both ends to be set it would save so much time.

I don't want to give the impression that these are serious problems. They would just be improvements to what is now a truly impressive piece of software which puts real CAD applications within the grasp of many users, at a tiny portion of the price normally required.

The system is easy to learn. For instance, at the Kent Apple Village I had 4 and 6 year olds colouring houses, putting parrots in windows and designing multi-coloured cars for Dad within a couple of minutes.

One problem which surfaces in proper use is that of the limited resolution of the Apple screen.

One needs as many screen points as possible to display large portions of complex drawings and indeed most large CAD systems offer at least 512x512 resolution, and more usually it is in the 1028x1028 range.

Such vector graphics displays have recently become available as add-ons for the Apple and I've now tried out two graphics systems that use the Digisolve 512x512 board.

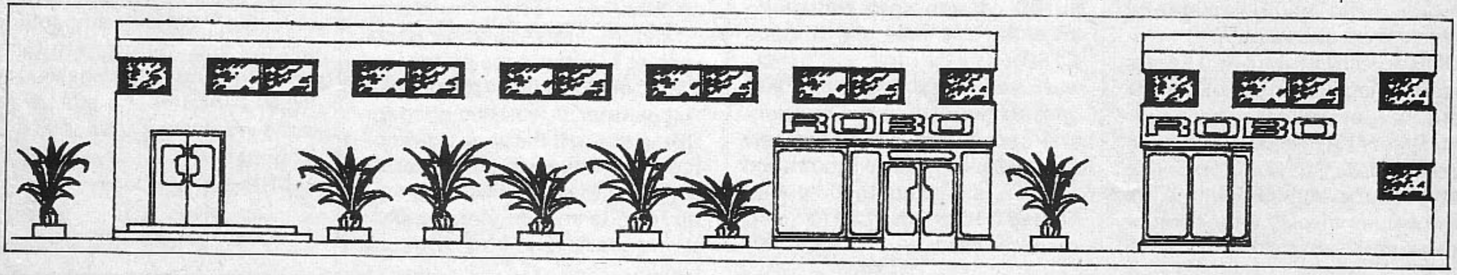
It would be extremely attractive if the Bit Stik could be upgraded to this resolution.

The question arises again of who the potential users are?

This time the answer is much clearer: anyone with "modular" design requirements – be it for hydraulic systems, electrical circuits, kitchen designs, data sheet layouts, architectural plans, schematic diagrams and indeed most creative graphics designs and presentations, all can make use of Bit Stik.

I think it still doesn't have a lot to offer the draughtsman doing complicated one-off designs. These certainly can be managed, but probably with greater difficulty than by old fashioned paper and pencil.

As with all CAD systems, it is important to try out the software to see if it fits your requirements, but I'm sure that for very many applications the Bit Stik offers a genuine tool that will earn its keep.



A NEW LOOK FOR VISICALC

NICK LEVY begins the year by assessing the latest version of Visicalc for the IIe

WITH the New Year and a new look to the magazine I *thought it appropriate this* month to discuss briefly the remarkable new-look version of Visicalc which is now available for the Apple IIe.

(If you are waiting for part two of my article on cashflow promised last month please be patient – I'll deal with it in next month's issue.)

Visicalc Advanced Version presents the kind of features that had frustrated most users of the "ordinary" Visicalc by their absence.

And in the right hands it could be one of the most powerful tools available to business executives.

However it has to be in the right hands, because even the best electronic spreadsheets are only as good as the people who use them.

To get the best from this package a user must be good at the use and application of quantitative techniques, as well as good at manipulating Visicalc.

This doesn't mean that you should learn business statistics and discounted cashflow techniques etc., before learning to use Visicalc!

On the contrary, just by learning to use Visicalc you will greatly enhance your ability to understand and apply mathematical techniques used in everyday business management.

That in turn will give you more confidence in applying the full power of Visicalc Advanced Version to the solution of the more intricate and complex business problems.

Furthermore, I would have no hesitation in recommending that first time Visicalc users on the Apple IIe or Apple III should, from the outset, start using the Advanced Version.

The program retains all the features of the ordinary Visicalc as well as allowing you to:

- Produce worksheets with columns of different width.
- Protect cells containing formulas from being accidently blanked or overwritten.
- Replicate whole blocks instead of just one column or one row at a time.
- Present negative numbers in brackets.
- Produce models where all the values display either one or three (or any other desired number of) decimal places.
- Display and print formulas in cells.

The above are just some of the features which users of ordinary Visicalc often wish it could perform.

Advanced Version can now fulfil all these wishes and many others. One command starts with /A (A standing for attributes) and contains no less than 40 sub commands (/AH, /AMP, /AV. etc).

However a number of these sub commands are not really new attributes since all they do is to cancel a previous attribute and restore the situation to what it was before the cancelled attribute was implemented.

If you are feeling depressed at this stage because you own ordinary Visicalc, note that many of the features incorporated as attributes in the advanced version can also be achieved by using Vergcourt's Ramex 128 expansion board when used with the Super Expander disc and a 16 sector Visicalc program.

The Expansion Board just slots into the Apple and gives your computer an extra 128k memory. The function of the disc is to perform all those tricks like protecting formulas etc.

The cost of implementing such a configuration would be in the order of £375.

Now for the bad news. First of all, use of Advanced Version Visicalc requires that you replace the ordinary 80 column card in slot three of your IIe with an 80 column card containing an extra 64k memory. It costs £180.

If you already use a RAM card to give you extra memory you cannot use it with the new program. Visicalc Advanced Version is only compatible with the 80 column extended memory card.

Furthermore, having two memory boards in your IIe at the same time confuses the VC Advanced program. It will not function properly unless the "alien" second board is removed.

Once you have installed the extra memory you will only be able to use 70k of it to create spreadsheet financial models. The program itself uses 56k RAM.

Data files produced by the ordinary VC program can be accessed by VC Advanced provided they do not take up more than 70k memory.

You need two disc drives to operate the VC Advanced whereas the ordinary VC requires only one drive, with the second optional.

This is unlikely to be a problem however, as most Apple IIes used in business probably already have two disc drives.

Being larger and more sophisticated than the ordinary VC, the Advanced Version performs more slowly.

For example, loading a particular VC model using the older program took two and a half minutes including pressing the exclamation mark three times (because of foreward calculations).

The same model took three minutes 40 seconds to load using the Advanced Version.

However when I tried loading the model with the Accelerator II card installed in my IIe, the whole process took less than a minute.

Before any Apple IIe users rush to buy the Accelerator II, a word of warning. The original Accelerator II was designed for use with the II Plus and does not function well with all Apple IIe software – so make sure you buy the IIe version of the board.

Incidentally, having an Accelerator II on an Apple II can have its funny moments.

First of all you can play musi through your Apple and hear th tunes played three and a ha times faster!

Just think what a time savin exercise this is.

And if you participate in th Decathlon (one of my favourit games on the Apple) then you running speed when approaching a high jump or a long jum suddenly increases from abou 6 metres per second to 20.

What a pity that you mis your jump when approaching a such speeds.

Finally, if you want t upgrade your ordinary VC to th Advanced Version you can do s at a special price of £132 (ha the normal purchase price).

Such transactions can onl be carried out through a deale and you will have to return you earlier VC program.

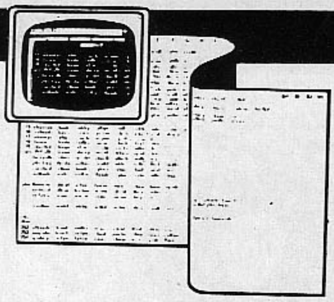
JUSTIN D. Underwood, of Ampex, writes following Dr M.R. Kibby's letter re/X in November Feedback:

THE /X Visicalc datagram dat functions determine the position of the "window" the screen provides within the total pag and the cursor position withi the screen.

It should be remembere that "split" screens can b saved and that different colum widths and format attribute can apply in the two halves. Al these attributes are stored an retrieved using /X functions.

Examples:

1) Single window
Top left AI
Cursor at A1



Cursor move horizontal
/X-/X>A1:>A1:

2) **Single window**
Top left H27
Cursor at H30
Cursor move mode vertical
/X!/X>H27:>H30:

A datagram (see Windfall, October 1982) is a technique which produces automatically up-dating spreadsheets. It is the Visicalc equivalent of an EXECUTE file.

3) **Horizontally split screen**
Window split at 13th row from top of screen

Upper window	**Lower window**
Top left F15	Top left J23
Cursor G21 (Horiz move)	Cursor M26 (vert move)
/GF$	/GFI
/GC9	GC6
Cursor reqd in upper half:	
last part of datagram	
/W1	
/GOC	Calc down columns
/GRM	Manual re-calc
/XH13	Horiz window at row 13
/GF$	Global format $
/GC9	Global column width 9
/X-/X>F15:>G21:;/GFI	
/GC6	Second window column width 6
/X!/X>J23:>M26:;	

If the cursor is required to be in the lower window (at M26) omit the semicolon at the end of the last line.

4) **Vertically split screen and sync scrolling**
Vertical window bar. Right edge of bar 27 character spaces from left hand side of screen.

Right hand window	**Left hand window**
Top left C10	Top left F10
Cursor E21 (Horiz move)	Cursor G21 (Vert move)
/GF$	/GFD
/GC6	/GC6
Last part of datagram:	
/W1	
/GOC	
/GRM	
/XV27	Vert window at character 27
/GF$	Global format $
/GC6	Global column width 6
/X-/X>C10:>E21:;/GF$	
/GC5	Second window column width 5
/X!/X>F10:>G21:/WS	

The "!" or "-" following the first X show the cursor movement axis in response to the left or right arrow keys.

For IIE Visicalc a "-" should always be used.

The /GF$ or GFI etc at the end of the first X line determines the format in the second window.

If the cursor is required in the first window, the final colon in the second X line should be followed by a semicolon.

If synchronised scrolling is required WS should be added to the end of the line.

The earlier table entry /W1 remains unchanged whether 1 or 2 windows are required.

/X is, therefore, much more than a pretty postscript as suggested by Nick Levy.

T.W. Creighton, of Cardiff, describes how to increase your datagrams beyond 254 lines.

ON reading R. Gear-Evans' article on datagrams in Think Tank (*Windfall,* May 1983) I was interested to find he had encountered the very same problem I had in writing datagrams of more than 254 lines.

Like Mr Gear-Evans I used a word processor in conjunction with Visicalc (in my case Zardax).

I have found it very useful for creating, and more importantly, editing datagrams. For example by using its FIND and REPLACE function I can change any references to column A to column S easily.

Despite its usefulness I soon found it tedious to switch between the word processor and Visicalc, so I devised a simple method of creating datagrams of greater than 254 lines.

The method is as follows: First write your file as normal in column A.

When you have reached the bottom cell A254 leave this blank and continue in column B. When you have finished you must save your file in two parts.

In cell A254 write the datagram (as a label, remember) /SLPFFILE2. Now save column A (using /PF) with the file name PFFILE1 and save column B as PFFILE2.

When you then load your PF file the instructions in column A will be carried out first, and the final instruction in A254 will then load in the instructions in column B.

As a result, you have two files on disc named PFFILE1 and PFFILE2 with the first file CHAINing the second.

I am grateful to Nick Levy for his articles on datagrams which have saved me no end of time.

However there is one use for them which I employ to good effect. If you write the following datagrams:

```
/IR
/IR
/IR
```

and save the file under, for example, IR3, when loaded it will insert 3 rows automatically.

This technique can be easily extended for as many rows as you like (and columns for that matter).

You will find that inserting rows on a large model takes a long time, and the program doesn't let you type ahead.

Use this method and you can insert 50 rows with one command (but you'll probably have to go and have lunch while it does it).

Finally the mysterious /X command.

This undocumented command doesn't seem to have much use except when used with GOTO, that is /X>D5 will move to co-ordinate D5 and move the cursor to the top left of the worksheet.

But it does have some peculiar effects when used in certain combinations. Clear your worksheet with /CY and type in the following series of commands, pressing RETURN at the end of each line.

```
>A3
/WH;/X>A1
/XV100000
/X>A1
```

I cannot see any use whatsoever for this, but someone might find it handy to confuse their colleagues.

T.W. Creighton, Cardiff.

A reader from South Africa, J.D. Eve, challenges Nick Levy to solve this problem . . .

YOURS certainly is a very useful and informative magazine. I read it from cover to cover every month.

The first thing I turn to is Nick Levy's article on Visicalc which has been of tremendous value to me and others who use this package.

Perhaps Nick would like to deal with the following problem?

I refer to the "round-off formula" "@ INT (value or formula) * 100 + .5)/100". If the value or formula being rounded off is a minus quantity, the +.5 rounds the figure down, not up. It has to be changed to –.5 to round-up correctly.

When one cannot determine in advance if the value will be plus or minus, can the formula be adapted to do this?

● *THE following formula should do the trick:*

*@ INT (@ABS(value or formula)*100+.5)/100 × (value or formula/@ABS(value or formula))*

Nick Levy

Space & Time

No, it's not another games listing. But when the Apple adds spaces to a listing it can be very frustrating when editing, says A.J. JAMESON. So here is a method of saving those spaces – and help to save your editing time as well.

THE Apple's main design philosophy appears to be if in doubt put in another space.

Of course POKE33,33 has been mentioned in numerous articles as an easy remedy, but is it the ideal solution? Spaces can still creep into text strings and data statements while editing.

They are still inserted after colons and command names and can pose a problem if the listed line plus spaces is longer than the keyboard buffer, which seems to occur all too often.

Many program listings contain control characters which remain invisible and are lost when edited.

Finally how many of us have tried to list when displaying a hi-res or lo-res graphics page, only to watch the program line appear briefly at the bottom of the screen as it scrolls behind the graphics window.

This ampersand list utility solves all the above problems. It:

- Removes all unnecessary spaces.
- Reveals control characters as inverse text.
- Automatically sets TEXT display mode.
- Compresses 'PRINT' to the abbreviation '?'.
- Prints the number of characters within the listed line.
- Allows the line number to be assigned a variable or an expression.
- Moves cursor to the beginning of the listed line.

It should be noted that there is a common misconception that the maximum line length is 249 characters – when the Apple beeps.

However, at this point 10 characters will be totally lost as the Apple uses the buffer to tokenise the entered line.

Consequently, the maximum Basic program line length that may be entered is 239 characters long.

The ampersand list program may be BRUN at any location, as the program is self locating.

To list a particular line enter &L(N) where N = line number, that is &L10 will list line 10.

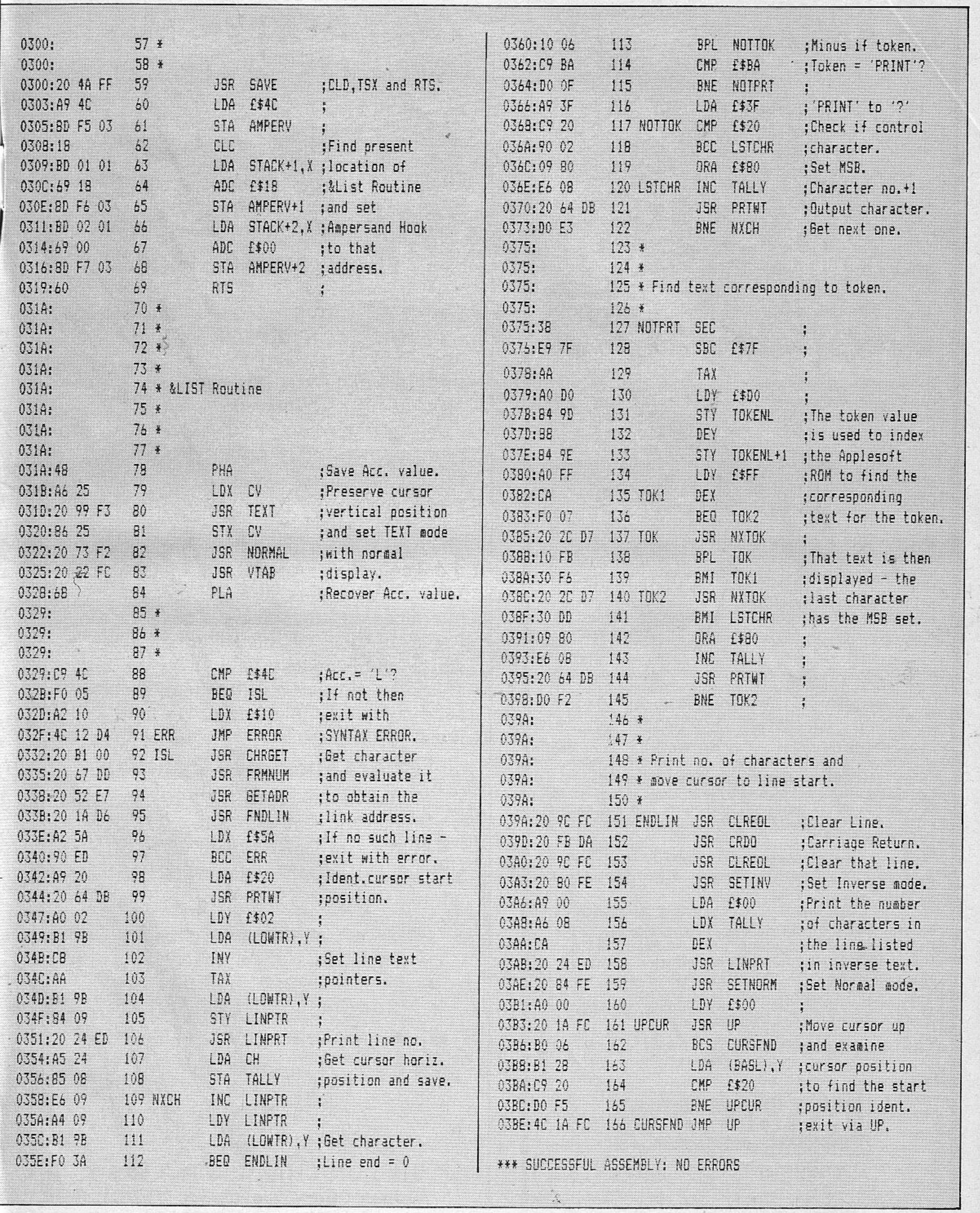
Originally the ampersand list embodied all the features of LIST.

However as the text is compressed, it does become difficult to read if many lines are listed at once.

Therefore &L will only allow a single line to be listed.

The ability to allow a line number to be assigned a variable might not seem noteworthy, but by adding a few extra lines to a program under development, any errors present will automatically list the offending line.

```
0300:           ORG  $300
0000:           OBJ  $300
0300: *
0300: *
0300: *
0300: ******************
0300: *                *
0300: * Ampersand List *
0300: * -------------- *
0300: *                *
0300: * by A.J.Jameson *
0300: *                *
0300: *                *
0300: *                *
0300: ******************
0300: *
0300: *
0300: *
0300: * The 'Ampersand List' routine is
0300: * relocatable and may be BRUN at
0300: * any convenient memory location.
0300: *
0300: *
0300: *
0008: TALLY   EQU  $08
0009: LINPTR  EQU  $09
0024: CH      EQU  $24
0025: CV      EQU  $25
0028: BASL    EQU  $28
009B: LOWTR   EQU  $9B
009D: TOKENL  EQU  $9D
00B1: CHRGET  EQU  $B1
0100: STACK   EQU  $100
03F5: AMPERV  EQU  $3F5
D412: ERROR   EQU  $D412
D61A: FNDLIN  EQU  $D61A
D72C: NXTOK   EQU  $D72C
DAFB: CRDO    EQU  $DAFB
DB64: PRTWT   EQU  $DB64
DD67: FRMNUM  EQU  $DD67
E752: GETADR  EQU  $E752
ED24: LINPRT  EQU  $ED24
F273: NORMAL  EQU  $F273
F399: TEXT    EQU  $F399
FC1A: UP      EQU  $FC1A
FC22: VTAB    EQU  $FC22
FC9C: CLREOL  EQU  $FC9C
FE80: SETINV  EQU  $FE80
FE84: SETNORM EQU  $FE84
FF4A: SAVE    EQU  $FF4A
0300: *
0300: *
0300: *
0300: * Locate Program Location
0300: * and set Ampersand hook.
0300: *
```



```
0300:          *
0300:          *
0300:20 4A FF          JSR  SAVE       ;CLD,TSX and RTS.
0303:A9 4C             LDA  £$4C       ;
0305:8D F5 03          STA  AMPERV     ;
0308:18                CLC             ;Find present
0309:BD 01 01          LDA  STACK+1,X  ;location of
030C:69 18             ADC  £$18       ;&List Routine
030E:8D F6 03          STA  AMPERV+1   ;and set
0311:BD 02 01          LDA  STACK+2,X  ;Ampersand Hook
0314:69 00             ADC  £$00       ;to that
0316:8D F7 03          STA  AMPERV+2   ;address.
0319:60                RTS             ;
031A:          *
031A:          *
031A:          *
031A:          *
031A:          * &LIST Routine
031A:          *
031A:          *
031A:          *
031A:48                PHA             ;Save Acc. value.
031B:A6 25             LDX  CV         ;Preserve cursor
031D:20 99 F3          JSR  TEXT       ;vertical position
0320:86 25             STX  CV         ;and set TEXT mode
0322:20 73 F2          JSR  NORMAL     ;with normal
0325:20 22 FC          JSR  VTAB       ;display.
0328:68                PLA             ;Recover Acc. value.
0329:          *
0329:          *
0329:          *
0329:C9 4C             CMP  £$4C       ;Acc.= 'L'?
032B:F0 05             BEQ  ISL        ;If not then
032D:A2 10             LDX  £$10       ;exit with
032F:4C 12 D4  ERR     JMP  ERROR      ;SYNTAX ERROR.
0332:20 B1 00  ISL     JSR  CHRGET     ;Get character
0335:20 67 DD          JSR  FRMNUM     ;and evaluate it
0338:20 52 E7          JSR  GETADR     ;to obtain the
033B:20 1A D6          JSR  FNDLIN     ;link address.
033E:A2 5A             LDX  £$5A       ;If no such line -
0340:90 ED             BCC  ERR        ;exit with error.
0342:A9 20             LDA  £$20       ;Ident.cursor start
0344:20 64 DB          JSR  PRTWT      ;position.
0347:A0 02             LDY  £$02       ;
0349:B1 9B             LDA  (LOWTR),Y  ;
034B:C8                INY             ;Set line text
034C:AA                TAX             ;pointers.
034D:B1 9B             LDA  (LOWTR),Y  ;
034F:84 09             STY  LINPTR     ;
0351:20 24 ED          JSR  LINPRT     ;Print line no.
0354:A5 24             LDA  CH         ;Get cursor horiz.
0356:85 08             STA  TALLY      ;position and save.
0358:E6 09     NXCH    INC  LINPTR     ;
035A:A4 09             LDY  LINPTR     ;
035C:B1 9B             LDA  (LOWTR),Y  ;Get character.
035E:F0 3A             BEQ  ENDLIN     ;Line end = 0
```

```
0360:10 06             BPL  NOTTOK     ;Minus if token.
0362:C9 BA             CMP  £$BA       ;Token = 'PRINT'?
0364:D0 0F             BNE  NOTPRT     ;
0366:A9 3F             LDA  £$3F       ;'PRINT' to '?'
0368:C9 20     NOTTOK  CMP  £$20       ;Check if control
036A:90 02             BCC  LSTCHR     ;character.
036C:09 80             ORA  £$80       ;Set MSB.
036E:E6 08     LSTCHR  INC  TALLY      ;Character no.+1
0370:20 64 DB          JSR  PRTWT      ;Output character.
0373:D0 E3             BNE  NXCH       ;Get next one.
0375:          *
0375:          *
0375:          * Find text corresponding to token.
0375:          *
0375:38        NOTPRT  SEC             ;
0376:E9 7F             SBC  £$7F       ;
0378:AA                TAX             ;
0379:A0 D0             LDY  £$D0       ;
037B:84 9D             STY  TOKENL     ;The token value
037D:88                DEY             ;is used to index
037E:84 9E             STY  TOKENL+1   ;the Applesoft
0380:A0 FF             LDY  £$FF       ;ROM to find the
0382:CA        TOK1    DEX             ;corresponding
0383:F0 07             BEQ  TOK2       ;text for the token.
0385:20 2C D7  TOK     JSR  NXTOK      ;
0388:10 FB             BPL  TOK        ;That text is then
038A:30 F6             BMI  TOK1       ;displayed - the
038C:20 2C D7  TOK2    JSR  NXTOK      ;last character
038F:30 DD             BMI  LSTCHR     ;has the MSB set.
0391:09 80             ORA  £$80       ;
0393:E6 08             INC  TALLY      ;
0395:20 64 DB          JSR  PRTWT      ;
0398:D0 F2             BNE  TOK2       ;
039A:          *
039A:          *
039A:          * Print no. of characters and
039A:          * move cursor to line start.
039A:          *
039A:20 9C FC  ENDLIN  JSR  CLREOL     ;Clear Line.
039D:20 FB DA          JSR  CRDO       ;Carriage Return.
03A0:20 9C FC          JSR  CLREOL     ;Clear that line.
03A3:20 80 FE          JSR  SETINV     ;Set Inverse mode.
03A6:A9 00             LDA  £$00       ;Print the number
03A8:A6 08             LDX  TALLY      ;of characters in
03AA:CA                DEX             ;the line listed
03AB:20 24 ED          JSR  LINPRT     ;in inverse text.
03AE:20 84 FE          JSR  SETNORM    ;Set Normal mode.
03B1:A0 00             LDY  £$00       ;
03B3:20 1A FC  UPCUR   JSR  UP         ;Move cursor up
03B6:B0 06             BCS  CURSFND    ;and examine
03B8:B1 28             LDA  (BASL),Y   ;cursor position
03BA:C9 20             CMP  £$20       ;to find the start
03BC:D0 F5             BNE  UPCUR      ;position ident.
03BE:4C 1A FC  CURSFND JMP  UP         ;exit via UP.

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

If you don't know anything about Pascal – or else find that this article goes over your head, read the first part of our Pascal Tutorial series which starts this month on Page 34.

And once you've learned Pascal (from scratch) come back to this article and work through it – because it is fun!

In Practical Pascal JONATHON LEWIS will examine interesting theories and things you can do with the language. In his first column he explains how you can perform PEEKs and POKEs. Next month, once you have mastered this technique, he'll show you how to apply it to animation.

Safe way to peek into Pascal

ALTHOUGH it is contrary to the spirit of Pascal's portability – and sometimes dangerous anyway – to do things like PEEKing and POKEing in Pascal, it is fairly easy to write a procedure to do just that.

My first choice of method was to write a set of simple machine code routines and call them as external procedures.

However, realising that many people would not be too keen on handling the assembler and subsequent use of Linker etc, I turned to Pascal itself.

There are a number of "byte-oriented built-ins" in Pascal, such as MOVELEFT, MOVERIGHT, which are frequently used to emulate PEEK and POKE behaviour.

A more elegant – and obvious method I think – is to use VARIANT RECORDS, and POINTERS. In other words, create a variable which can be used for more than one purpose. In this case, something that will be either an integer or the address of an absolute location in memory.

The Pascal declaration I have used looks like this:

```
ppd:record
        case b:boolean of
           true: (pointer:^byte);
          false: (address:integer)
      end;
```

This variable comprises two parts. The first is a Boolean part, which can have the values true or false; the second can actually be one of two different types of object.

When the Boolean part (ppd.b) is true the information the second part contains is treated as a pointer to the location of a thing called a byte, but when the Boolean part is false the information is recognised as an integer. (In fact, there is never any need to refer explicitly to ppd.b, as you can jump straight to ppd.pointer and ppd.address.)

To PEEK at a location then, you need only two lines of Pascal to define a suitable function:

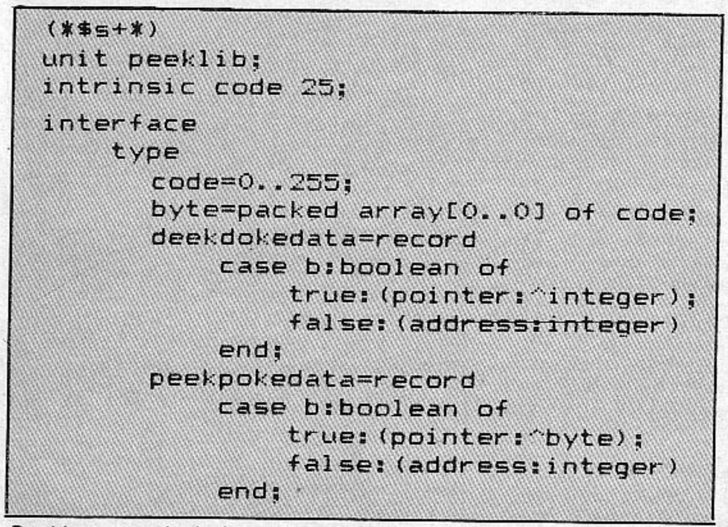
```
ppd.value:=addressyouwantto
   PEEKat;
peek:=ppd.pointer^(*the thing
   being pointed at*)
```

There is only one more problem to deal with. Pointers in Pascal tend to point at words (ie two consecutive bytes), so even if you define a byte as a number from 0 to 255, when Pascal points at it it will point at a number between –32768 and 32767!

To get round this you must make use of the technique known as PACKing, as Pascal can even point to the exact BIT when it is pointing at a PACKed item. The necessary declaration for a byte becomes:

```
byte=packed array[0..0]
        of 0..255
```

that is, an array of one entry, which is a number between 0 and 255.



```
(*$s+*)
unit peeklib;
intrinsic code 25;
interface
    type
       code=0..255;
       byte=packed array[0..0] of code;
       deekdokedata=record
            case b:boolean of
                 true:(pointer:^integer);
                 false:(address:integer)
            end;
       peekpokedata=record
            case b:boolean of
                 true:(pointer:^byte);
                 false:(address:integer)
            end;
```

Packing – or declaring a byte

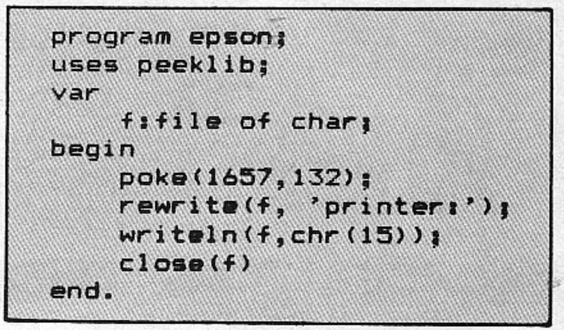
The code with this article is written in the format of a UNIT, so if you copy it exactly, compile it, and use LIBRARY.CODE to link it into your SYSTEM .LIBRARY (see Operating Manual p.188), you will be able to PEEK, POKE, DEEK and DOKE (the last two being the two-byte equivalents of PEEK, POKE) in any of your programs by putting in the 'USES PEEKLIB' in the usual place.

If you do not want to make a library unit you will need to copy the TYPE declaration into your program, and then whichever functions/procedures you want.

But remember to include the parameter lists (shown in the interface part although missed out in the implementation part).

Once you can POKE into Pascal, there are many useful things you can do. It is possible, for example, to tell the system to forget where it booted from, and to look somewhere else for the boot disc. (I use a Rocon 128k ramcard as my boot disc most of the time, which cuts compiling time to about 40 per cent.)

As a very simple example of "legitimate" uses of POKEing in Pascal, I present a little routine for setting an Epson printer to 132 condensed characters per line.

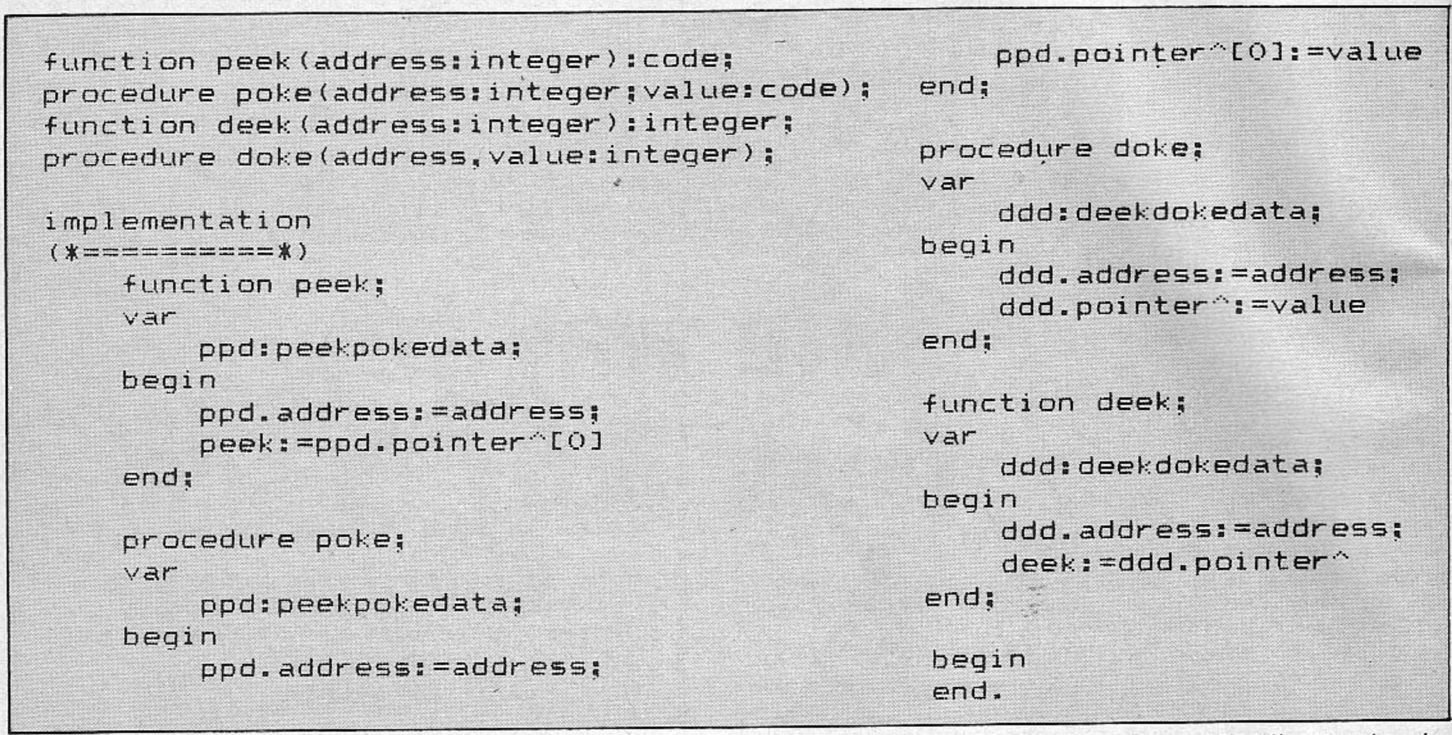


```
program epson;
uses peeklib;
var
     f:file of char;
begin
     poke(1657,132);
     rewrite(f, 'printer:');
     writeln(f,chr(15));
     close(f)
end.
```

Setting an Epson printer

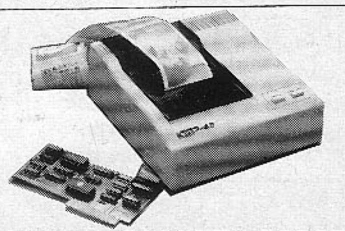
If you have time to spend PEEKing and POKEing around, with the operating manual at your side, you will be amazed at the clever (and silly) things you can do.

Be warned, however, that it is possible to wipe the directory of every single disc in the system with just one careless POKE. I suggest you experiment with just one (backed-up) disc whenever you want to try something new.



```
function peek(address:integer):code;
procedure poke(address:integer;value:code);
function deek(address:integer):integer;
procedure doke(address,value:integer);

implementation
(*==========*)
    function peek;
    var
        ppd:peekpokedata;
    begin
        ppd.address:=address;
        peek:=ppd.pointer^[0]
    end;

    procedure poke;
    var
        ppd:peekpokedata;
    begin
        ppd.address:=address;
        ppd.pointer^[0]:=value
    end;

    procedure doke;
    var
        ddd:deekdokedata;
    begin
        ddd.address:=address;
        ddd.pointer^:=value
    end;

    function deek;
    var
        ddd:deekdokedata;
    begin
        ddd.address:=address;
        deek:=ddd.pointer^
    end;

    begin
    end.
```

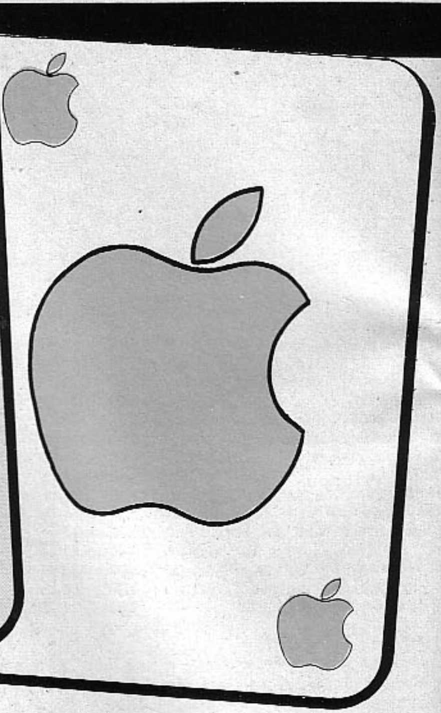
PEEKing at a location



APPLE GAME

This game listing will really try your PATIENCE

But don't worry. Card ace G.R. GILBERTSON has devised a finely structured, easy to follow program resulting in a game to keep you occupied for hours



THERE are many games of Patience or Solitaire. This particular version requires you to rearrange the shuffled cards from two packs into eight columns descending from king to ace in alternate colours. Black cards are shown in normal video and red cards as inverse.

Runs of cards may be built up in any column and transferred as a block to another column. Up to six spare cards may be held at any one time.

Several spare cards may be transferred in one move from the bottom of any column, but spare cards may only be replaced singly by specifying individually which spare card is to be moved to which column.

If you try to cheat, the program will prevent you from making an illegal move.

Surprisingly, the computerised version is more demanding than the same game played with real cards and can be remarkably addictive.

My favourite program tester, who is almost guaranteed to find in any program bugs which I had never even contemplated could exist, has been known to play game after game continuously for five hours or more at a stretch until long past her normal bedtime.

The program listing also demonstrates the modular approach which I use for all my programming.

The whole logic of any program is contained within the first 1,000 lines, with all the work being done in separate sub-routines at intervals of 1,000 lines.

This approach enables the outline of any program to be developed extremely rapidly at the keyboard without the need for charting.

Each step can be tested, as it is completed without necessarily waiting for completion of the whole and, with a versatile renumber program, amendments or additions incorporated as required.

It also greatly simplifies the understanding of a program listing.

In my view good structure depends on the way in which a language is used and not upon the language itself.

This is equally true of natural languages, such as English or German as it is of computer languages, and yet is not a theme I have seen developed in any of the endless discussions on the relative merits of Basic when compared with any of the so-called structured computer languages.

The most effective language must surely be that in which one's meaning can be conveyed most fluently.

Brevity may be the soul of wit, but rarely of understanding.

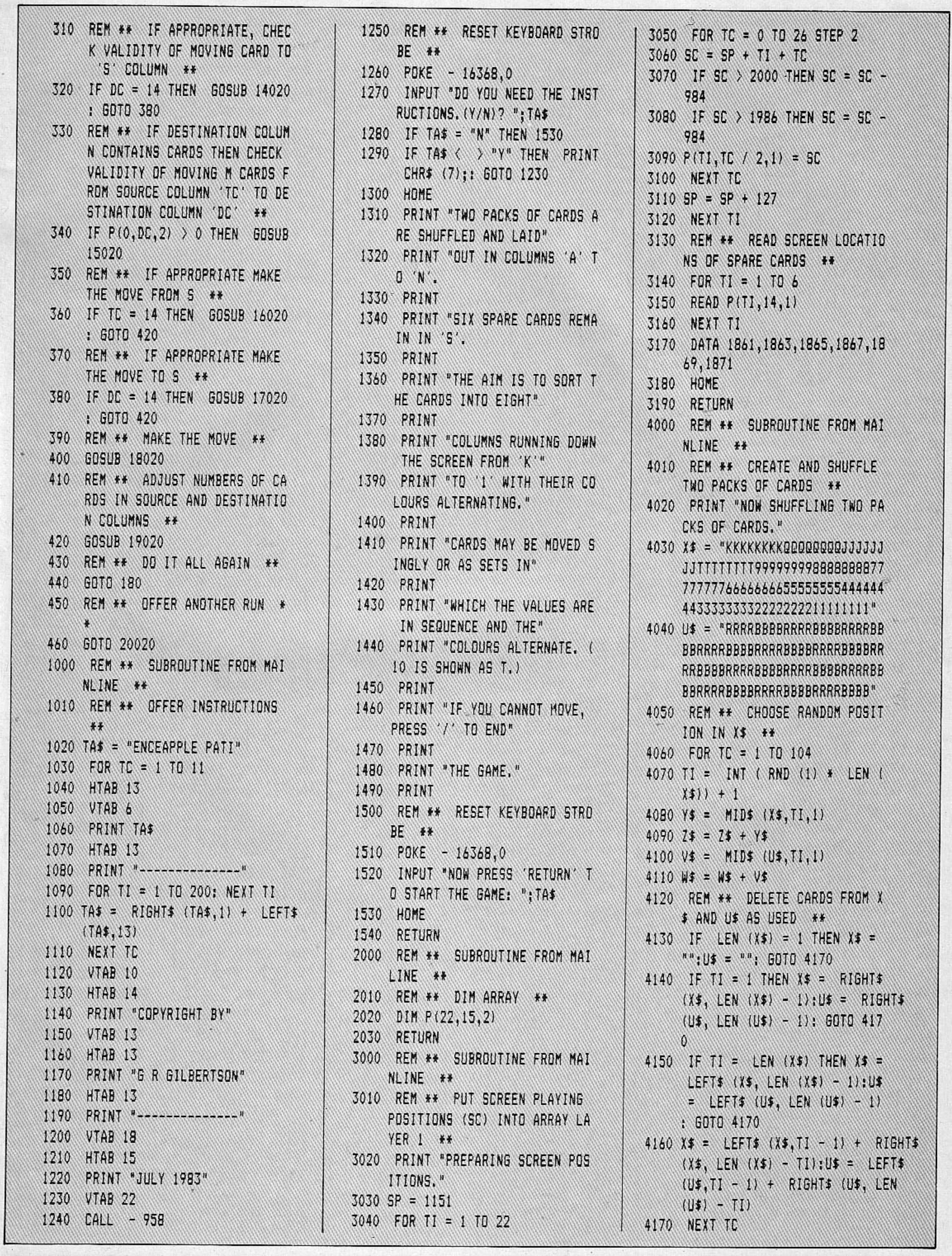
However, for those of you who disagree, the modular approach illustrated can be greatly condensed using a program such as Listmaster's Applespeed which can remove REMs, reduce variables to two significant characters, combine lines and renumber them in steps of one.

Typically, this would reduce the program length by half, speed its execution and, if that is what you want, make the resultant listing totally incomprehensible.

Having tried your patience, you can now try mine.

PATIENCE LISTING

```
10  TEXT : HOME : NORMAL : SPEED=
    255
20  REM **  PATIENCE  **
30  REM **  APPLESOFT AND APPLE D
    OS 3.3  **
40  REM **  COPYRIGHT G R GILBERS
    ON JULY 1983  **
50  REM **  OFFER INSTRUCTIONS  *
    *
60  GOSUB 1020
70  REM **  DIM ARRAYS  **
80  GOSUB 2020
90  REM **  PUT SCREEN POSITIONS
     (SC) INTO ARRAY LAYER 1  **
100  GOSUB 3020
110  REM **  CREATE AND SHUFFLE T
    WO PACKS OF CARDS  **
120  GOSUB 4020
130  REM **  PUT CARDS IN PLAYING
     ARRAY LEVELS  **
140  GOSUB 5020
150  REM **  DISPLAY PLAYING ARRA
    Y  **
160  GOSUB 6030
170  REM **  OBTAIN SOURCE COLUMN
     FROM WHICH CARD(S) MOVED  *
    *
180  GOSUB 7020
190  REM **  IF MOVE IS FROM SPAR
    E CARDS, FIND CARD TO BE MOV
    ED  **
200  IF TC = 14 THEN  GOSUB 8020:
     GOTO 240
210  REM **  OBTAIN NUMBER OF CAR
    DS TO BE MOVED  **
220  GOSUB 9020
230  REM **  OBTAIN DESTINATION C
    OLUMN  **
240  GOSUB 10020
250  REM **  REMOVE INPUT LINE  *
    *
260  GOSUB 11020
270  REM **  IF MOVE IS NOT TO S
    AND MORE THAN ONE CARD TO BE
     MOVED, THEN CHECK M CARDS T
    O BE MOVED ARE IN SEQUENCE
    **
280  IF DC < 14 AND M > 1 THEN  GOSUB
    12020
290  REM **  IF APPROPRIATE AND I
    F DESTINATION COLUMN CONTAIN
    S CARDS, CHECK VALIDITY OF M
    OVING SPARE CARD 'TI' TO DES
    TINATION COLUMN  **
300  IF TI > 0 THEN  IF P(0,DC,2)
     > 0 THEN  GOSUB 13020: GOTO
    360
```

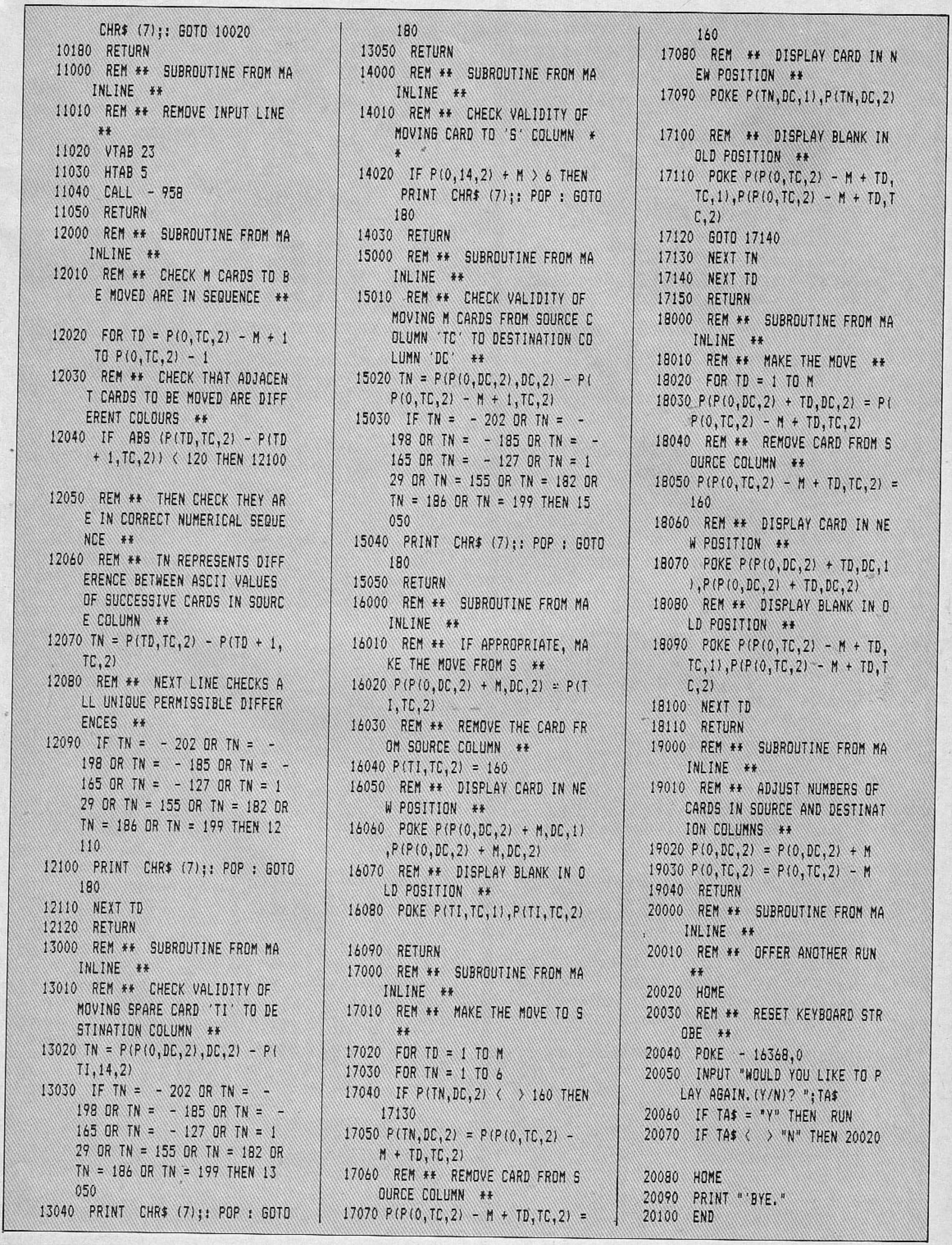


```
310  REM **  IF APPROPRIATE, CHEC
     K VALIDITY OF MOVING CARD TO
      'S' COLUMN  **
320  IF DC = 14 THEN  GOSUB 14020
     : GOTO 380
330  REM **  IF DESTINATION COLUM
     N CONTAINS CARDS THEN CHECK
     VALIDITY OF MOVING M CARDS F
     ROM SOURCE COLUMN 'TC' TO DE
     STINATION COLUMN 'DC'  **
340  IF P(0,DC,2) > 0 THEN  GOSUB
     15020
350  REM **  IF APPROPRIATE MAKE
     THE MOVE FROM S  **
360  IF TC = 14 THEN  GOSUB 16020
     : GOTO 420
370  REM **  IF APPROPRIATE MAKE
     THE MOVE TO S  **
380  IF DC = 14 THEN  GOSUB 17020
     : GOTO 420
390  REM **  MAKE THE MOVE  **
400  GOSUB 18020
410  REM **  ADJUST NUMBERS OF CA
     RDS IN SOURCE AND DESTINATIO
     N COLUMNS  **
420  GOSUB 19020
430  REM **  DO IT ALL AGAIN  **
440  GOTO 180
450  REM **  OFFER ANOTHER RUN  *
     *
460  GOTO 20020
1000  REM **  SUBROUTINE FROM MAI
     NLINE  **
1010  REM **  OFFER INSTRUCTIONS
      **
1020 TA$ = "ENCEAPPLE PATI"
1030  FOR TC = 1 TO 11
1040  HTAB 13
1050  VTAB 6
1060  PRINT TA$
1070  HTAB 13
1080  PRINT "--------------"
1090  FOR TI = 1 TO 200: NEXT TI
1100 TA$ =  RIGHT$ (TA$,1) +  LEFT$
     (TA$,13)
1110  NEXT TC
1120  VTAB 10
1130  HTAB 14
1140  PRINT "COPYRIGHT BY"
1150  VTAB 13
1160  HTAB 13
1170  PRINT "G R GILBERTSON"
1180  HTAB 13
1190  PRINT "--------------"
1200  VTAB 18
1210  HTAB 15
1220  PRINT "JULY 1983"
1230  VTAB 22
1240  CALL  - 958
```

```
1250  REM **  RESET KEYBOARD STRO
     BE  **
1260  POKE  - 16368,0
1270  INPUT "DO YOU NEED THE INST
     RUCTIONS.(Y/N)? ";TA$
1280  IF TA$ = "N" THEN 1530
1290  IF TA$ <  > "Y" THEN  PRINT
      CHR$ (7);: GOTO 1230
1300  HOME
1310  PRINT "TWO PACKS OF CARDS A
     RE SHUFFLED AND LAID"
1320  PRINT "OUT IN COLUMNS 'A' T
     O 'N'.
1330  PRINT
1340  PRINT "SIX SPARE CARDS REMA
     IN IN 'S'.
1350  PRINT
1360  PRINT "THE AIM IS TO SORT T
     HE CARDS INTO EIGHT"
1370  PRINT
1380  PRINT "COLUMNS RUNNING DOWN
      THE SCREEN FROM 'K'"
1390  PRINT "TO '1' WITH THEIR CO
     LOURS ALTERNATING."
1400  PRINT
1410  PRINT "CARDS MAY BE MOVED S
     INGLY OR AS SETS IN"
1420  PRINT
1430  PRINT "WHICH THE VALUES ARE
      IN SEQUENCE AND THE"
1440  PRINT "COLOURS ALTERNATE. (
     10 IS SHOWN AS T.)
1450  PRINT
1460  PRINT "IF YOU CANNOT MOVE,
     PRESS '/' TO END"
1470  PRINT
1480  PRINT "THE GAME."
1490  PRINT
1500  REM **  RESET KEYBOARD STRO
     BE  **
1510  POKE  - 16368,0
1520  INPUT "NOW PRESS 'RETURN' T
     O START THE GAME: ";TA$
1530  HOME
1540  RETURN
2000  REM **  SUBROUTINE FROM MAI
     LINE  **
2010  REM **  DIM ARRAY  **
2020  DIM P(22,15,2)
2030  RETURN
3000  REM **  SUBROUTINE FROM MAI
     NLINE  **
3010  REM **  PUT SCREEN PLAYING
     POSITIONS (SC) INTO ARRAY LA
     YER 1  **
3020  PRINT "PREPARING SCREEN POS
     ITIONS."
3030 SP = 1151
3040  FOR TI = 1 TO 22
```

```
3050  FOR TC = 0 TO 26 STEP 2
3060 SC = SP + TI + TC
3070  IF SC > 2000 THEN SC = SC -
     984
3080  IF SC > 1986 THEN SC = SC -
     984
3090 P(TI,TC / 2,1) = SC
3100  NEXT TC
3110 SP = SP + 127
3120  NEXT TI
3130  REM **  READ SCREEN LOCATIO
     NS OF SPARE CARDS  **
3140  FOR TI = 1 TO 6
3150  READ P(TI,14,1)
3160  NEXT TI
3170  DATA 1861,1863,1865,1867,18
     69,1871
3180  HOME
3190  RETURN
4000  REM **  SUBROUTINE FROM MAI
     NLINE  **
4010  REM **  CREATE AND SHUFFLE
     TWO PACKS OF CARDS  **
4020  PRINT "NOW SHUFFLING TWO PA
     CKS OF CARDS."
4030 X$ = "KKKKKKKKQQQQQQQQJJJJJJ
     JJTTTTTTTT9999999988888888877
     777777666666665555555544444
     443333333322222222111111111"
4040 U$ = "RRRRBBBBRRRRBBBBRRRRBB
     BBRRRRBBBBRRRRBBBBRRRRBBBBRR
     RRBBBBRRRRBBBBRRRRBBBBRRRRBB
     BBRRRRBBBBRRRRBBBBRRRRBBBB"
4050  REM **  CHOOSE RANDOM POSIT
     ION IN X$  **
4060  FOR TC = 1 TO 104
4070 TI =  INT ( RND (1) *  LEN (
     X$)) + 1
4080 Y$ =  MID$ (X$,TI,1)
4090 Z$ = Z$ + Y$
4100 V$ =  MID$ (U$,TI,1)
4110 W$ = W$ + V$
4120  REM **  DELETE CARDS FROM X
     $ AND U$ AS USED  **
4130  IF  LEN (X$) = 1 THEN X$ =
     "":U$ = "": GOTO 4170
4140  IF TI = 1 THEN X$ =  RIGHT$
     (X$, LEN (X$) - 1):U$ =  RIGHT$
     (U$, LEN (U$) - 1): GOTO 417
     0
4150  IF TI =  LEN (X$) THEN X$ =
      LEFT$ (X$, LEN (X$) - 1):U$
      =  LEFT$ (U$, LEN (U$) - 1)
     : GOTO 4170
4160 X$ =  LEFT$ (X$,TI - 1) +  RIGHT$
     (X$, LEN (X$) - TI):U$ =  LEFT$
     (U$,TI - 1) +  RIGHT$ (U$, LEN
     (U$) - TI)
4170  NEXT TC
```

```
4180  HOME
4190  RETURN
5000  REM **  SUBROUTINE FROM MAI
      NLINE  **
5010  REM **  PUT CARDS INTO ARRA
      Y LEVEL 2  **
5020  PRINT "LAYING OUT CARDS."
5030  REM **  PUT NUMBER OF CARDS
       IN EACH COLUMN INTO P(0,TC,
      2)  **
5040  FOR TC = 0 TO 14
5050  READ P(0,TC,2)
5060  NEXT TC
5070  DATA 10,10,9,9,8,8,7,7,6,6,
      5,5,4,4,6
5080  REM **  PLACE CARDS IN ARRA
      Y LEVEL 2  **
5090  FOR TC = 0 TO 14
5100  FOR TI = 1 TO P(0,TC,2)
5110 TD = TD + 1
5120 P(TI,TC,2) =  ASC ( MID$ (Z$
     ,TD,1))
5130  REM **  AMEND ASCII VALUE T
      O MAKE CARDS NORMAL OR INVER
      SE TO REPRESENT BLACK OR RED
        **
5140  IF  MID$ (W$,TD,1) = "B" THEN
      P(TI,TC,2) = P(TI,TC,2) + 12
      8
5150  IF  MID$ (W$,TD,1) = "R" THEN
       IF P(TI,TC,2) > 64 THEN P(T
      I,TC,2) = P(TI,TC,2) - 64
5160  NEXT TI
5170  NEXT TC
5180  HOME
5190  RETURN
6000  REM **  SUBROUTINE FROM MAI
      NLINE  **
6010  REM **  LABEL PLAYING POSIT
      IONS  **
6020  VTAB 1
6030  INVERSE
6040  PRINT "A B C D E F G H I J
      K L M N"
6050  NORMAL
6060  REM **  POKE LABELS FOR SPA
      RE CARD AREA  **
6070  POKE 1354,211
6080 TD = 193
6090  FOR TC = 0 TO 10
6100  POKE 1477 + TC,173
6110  POKE 1733 + TC,173
6120  IF TC / 2 =  INT (TC / 2) THEN
       POKE 1605 + TC,TD:TD = TD +
      1
6130  NEXT TC
6140  REM **  LAY DOWN CARDS  **
6150  FOR TC = 0 TO 14
6160  FOR TI = 1 TO P(0,TC,2)
6170  POKE P(TI,TC,1),P(TI,TC,2)
6180  NEXT TI
6190  NEXT TC
6200  RETURN
7000  REM **  SUBROUTINE FROM MAI
      NLINE  **
7010  REM **  OBTAIN SOURCE COLUM
      N FROM WHICH CARD(S) MOVED
      **
7020  VTAB 23
7030  HTAB 5
7040  CALL  - 958
7050  REM **  RESET KEYBOARD STRO
      BE  **
7060  POKE  - 16368,0
7070  INPUT "MOVE FROM WHICH COLU
      MN? ";TA$
7080  REM **  PROVIDE ESCAPE  **
7090  IF TA$ = "/" THEN 460
7100  REM **  PROVIDE FOR RETURN
      ONLY PRESSED  **
7110  IF  LEN (TA$) = 0 THEN  PRINT
       CHR$ (7);: GOTO 7020
7120  REM **  PROVIDE FOR ESCAPE
      IF NO MOVE POSSIBLE  **
7130  IF TA$ =  CHR$ (47) THEN 46
      0
7140 TC =  ASC (TA$) - 65
7150  REM **  IF MOVE IS FROM S A
      DJUST TC TO COLUMN 14  **
7160  IF TC = 18 THEN TC = 14
7170  REM **  CHECK VALIDITY OF R
      ESPONSE  **
7180  IF TC < 0 OR TC > 14 OR TC <
       >  INT (TC) THEN  PRINT  CHR$
      (7);: GOTO 7020
7190  RETURN
8000  REM **  SUBROUTINE FROM MAI
      NLINE  **
8010  REM **  CHOICE WAS TO MOVE
      FROM SPARE CARDS  **
8020  VTAB 23
8030  HTAB 5
8040  CALL  - 958
8050  REM **  RESET KEYBOARD STRO
      BE  **
8060  POKE  - 16368,0
8070  INPUT "WHICH CARD FROM S.(A
       B C D E F)? ";TA$
8080  REM **  CHECK VALIDITY OF C
      HOICE  **
8090 TI =  ASC (TA$) - 64
8100  IF TI <  >  INT (TI) OR TI <
      1 OR TI > 6 OR  LEN (TA$) =
      0 THEN  PRINT  CHR$ (7);: GOTO
      8020
8110  REM **  ONLY ONE CARD AT A
      TIME FROM S. THEREFORE M = 1
       **
8120 M = 1
8130  RETURN
9000  REM **  SUBROUTINE FROM MAI
      NLINE  **
9010  REM **  OBTAIN NUMBER OF CA
      RDS TO BE MOVED  **
9020  VTAB 23
9030  HTAB 5
9040  CALL  - 958
9050  REM **  RESET KEYBOARD STRO
      BE  **
9060  POKE  - 16368,0
9070  INPUT "MOVE HOW MANY CARDS?
       ";TA$
9080  REM **  PROVIDE ESCAPE  **
9090  IF TA$ = "/" THEN 460
9100 M =  VAL (TA$)
9110  REM **  M = NUMBER OF CARDS
       TO MOVE  **
9120  REM **  CHECK NUMBER OF CAR
      DS IS PRACTICAL  **
9130  IF  LEN (TA$) = 0 OR M < 1 OR
       M <  >  INT (M) OR M > 22 OR
       P(0,TC,2) < M THEN  PRINT  CHR$
       (7);: POP : GOTO 180
9140  REM **  MOVE IS NOT FROM SP
      ARE CARDS SO SET TI TO 0  **

9150 TI = 0
9160  RETURN
10000  REM **  SUBROUTINE FROM MA
       INLINE  **
10010  REM **  OBTAIN DESTINATION
        COLUMN  **
10020  VTAB 23
10030  HTAB 5
10040  CALL  - 958
10050  REM **  RESET KEYBOARD STR
       OBE  **
10060  POKE  - 16368,0
10070  INPUT "TO WHICH COLUMN? ";
       TA$
10080  REM **  PROVIDE ESCAPE  **

10090  IF TA$ = "/" THEN 460
10100  REM **  PROVIDE FOR RETURN
        ONLY PRESSED  **
10110  IF  LEN (TA$) = 0 THEN  PRINT
        CHR$ (7);: GOTO 10020
10120  REM **  DC  = DESTINATION
       COLUMN  **
10130 DC =  ASC (TA$) - 65
10140  REM **  IF MOVE IS TO S AD
       JUST DC TO COLUMN 14  **
10150  IF DC = 18 THEN DC = 14
10160  REM **  CHECK COLUMN EXIST
       S  **
10170  IF DC < 0 OR DC > 14 OR DC
        <  >  INT (DC) THEN  PRINT
```



```
     CHR$ (7);: GOTO 10020
 10180  RETURN
 11000  REM **  SUBROUTINE FROM MA
      INLINE  **
 11010  REM **  REMOVE INPUT LINE
      **
 11020  VTAB 23
 11030  HTAB 5
 11040  CALL  - 958
 11050  RETURN
 12000  REM **  SUBROUTINE FROM MA
      INLINE  **
 12010  REM **  CHECK M CARDS TO B
      E MOVED ARE IN SEQUENCE  **

 12020  FOR TD = P(0,TC,2) - M + 1
      TO P(0,TC,2) - 1
 12030  REM **  CHECK THAT ADJACEN
      T CARDS TO BE MOVED ARE DIFF
      ERENT COLOURS  **
 12040  IF  ABS (P(TD,TC,2) - P(TD
      + 1,TC,2)) < 120 THEN 12100

 12050  REM **  THEN CHECK THEY AR
      E IN CORRECT NUMERICAL SEQUE
      NCE  **
 12060  REM **  TN REPRESENTS DIFF
      ERENCE BETWEEN ASCII VALUES
      OF SUCCESSIVE CARDS IN SOURC
      E COLUMN  **
 12070 TN = P(TD,TC,2) - P(TD + 1,
      TC,2)
 12080  REM **  NEXT LINE CHECKS A
      LL UNIQUE PERMISSIBLE DIFFER
      ENCES  **
 12090  IF TN =  - 202 OR TN =  -
      198 OR TN =  - 185 OR TN =  -
      165 OR TN =  - 127 OR TN = 1
      29 OR TN = 155 OR TN = 182 OR
      TN = 186 OR TN = 199 THEN 12
      110
 12100  PRINT  CHR$ (7);: POP : GOTO
      180
 12110  NEXT TD
 12120  RETURN
 13000  REM **  SUBROUTINE FROM MA
      INLINE  **
 13010  REM **  CHECK VALIDITY OF
      MOVING SPARE CARD 'TI' TO DE
      STINATION COLUMN  **
 13020 TN = P(P(0,DC,2),DC,2) - P(
      TI,14,2)
 13030  IF TN =  - 202 OR TN =  -
      198 OR TN =  - 185 OR TN =  -
      165 OR TN =  - 127 OR TN = 1
      29 OR TN = 155 OR TN = 182 OR
      TN = 186 OR TN = 199 THEN 13
      050
 13040  PRINT  CHR$ (7);: POP : GOTO
      180
 13050  RETURN
 14000  REM **  SUBROUTINE FROM MA
      INLINE  **
 14010  REM **  CHECK VALIDITY OF
      MOVING CARD TO 'S' COLUMN  *
      *
 14020  IF P(0,14,2) + M > 6 THEN
      PRINT  CHR$ (7);: POP : GOTO
      180
 14030  RETURN
 15000  REM **  SUBROUTINE FROM MA
      INLINE  **
 15010  REM **  CHECK VALIDITY OF
      MOVING M CARDS FROM SOURCE C
      OLUMN 'TC' TO DESTINATION CO
      LUMN 'DC'  **
 15020 TN = P(P(0,DC,2),DC,2) - P(
      P(0,TC,2) - M + 1,TC,2)
 15030  IF TN =  - 202 OR TN =  -
      198 OR TN =  - 185 OR TN =  -
      165 OR TN =  - 127 OR TN = 1
      29 OR TN = 155 OR TN = 182 OR
      TN = 186 OR TN = 199 THEN 15
      050
 15040  PRINT  CHR$ (7);: POP : GOTO
      180
 15050  RETURN
 16000  REM **  SUBROUTINE FROM MA
      INLINE  **
 16010  REM **  IF APPROPRIATE, MA
      KE THE MOVE FROM S  **
 16020 P(P(0,DC,2) + M,DC,2) = P(T
      I,TC,2)
 16030  REM **  REMOVE THE CARD FR
      OM SOURCE COLUMN  **
 16040 P(TI,TC,2) = 160
 16050  REM **  DISPLAY CARD IN NE
      W POSITION  **
 16060  POKE P(P(0,DC,2) + M,DC,1)
      ,P(P(0,DC,2) + M,DC,2)
 16070  REM **  DISPLAY BLANK IN O
      LD POSITION  **
 16080  POKE P(TI,TC,1),P(TI,TC,2)

 16090  RETURN
 17000  REM **  SUBROUTINE FROM MA
      INLINE  **
 17010  REM **  MAKE THE MOVE TO S
      **
 17020  FOR TD = 1 TO M
 17030  FOR TN = 1 TO 6
 17040  IF P(TN,DC,2) <  > 160 THEN
      17130
 17050 P(TN,DC,2) = P(P(0,TC,2) -
      M + TD,TC,2)
 17060  REM **  REMOVE CARD FROM S
      OURCE COLUMN  **
 17070 P(P(0,TC,2) - M + TD,TC,2) =
      160
 17080  REM **  DISPLAY CARD IN N
      EW POSITION  **
 17090  POKE P(TN,DC,1),P(TN,DC,2)

 17100  REM **  DISPLAY BLANK IN
      OLD POSITION  **
 17110  POKE P(P(0,TC,2) - M + TD,
      TC,1),P(P(0,TC,2) - M + TD,T
      C,2)
 17120  GOTO 17140
 17130  NEXT TN
 17140  NEXT TD
 17150  RETURN
 18000  REM **  SUBROUTINE FROM MA
      INLINE  **
 18010  REM **  MAKE THE MOVE  **
 18020  FOR TD = 1 TO M
 18030 P(P(0,DC,2) + TD,DC,2) = P(
      P(0,TC,2) - M + TD,TC,2)
 18040  REM **  REMOVE CARD FROM S
      OURCE COLUMN  **
 18050 P(P(0,TC,2) - M + TD,TC,2) =
      160
 18060  REM **  DISPLAY CARD IN NE
      W POSITION  **
 18070  POKE P(P(0,DC,2) + TD,DC,1
      ),P(P(0,DC,2) + TD,DC,2)
 18080  REM **  DISPLAY BLANK IN O
      LD POSITION  **
 18090  POKE P(P(0,TC,2) - M + TD,
      TC,1),P(P(0,TC,2) - M + TD,T
      C,2)
 18100  NEXT TD
 18110  RETURN
 19000  REM **  SUBROUTINE FROM MA
      INLINE  **
 19010  REM **  ADJUST NUMBERS OF
      CARDS IN SOURCE AND DESTINAT
      ION COLUMNS  **
 19020 P(0,DC,2) = P(0,DC,2) + M
 19030 P(0,TC,2) = P(0,TC,2) - M
 19040  RETURN
 20000  REM **  SUBROUTINE FROM MA
      INLINE  **
 20010  REM **  OFFER ANOTHER RUN
      **
 20020  HOME
 20030  REM **  RESET KEYBOARD STR
      OBE  **
 20040  POKE  - 16368,0
 20050  INPUT "WOULD YOU LIKE TO P
      LAY AGAIN.(Y/N)? ";TA$
 20060  IF TA$ = "Y" THEN  RUN
 20070  IF TA$ <  > "N" THEN 20020

 20080  HOME
 20090  PRINT "'BYE."
 20100  END
```

Cheap but cheerful alternative to costly word processing systems

JOHN EDAM describes how he has interfaced an electric typewriter with an Apple II

FOR several months I have been using Applewriter on an Apple II interfaced to an electric typewriter, mainly to help produce a number of small technical manuals.

I am not a whiz-kid at computing, using my Apple principally for iterative engineering calculations that previously I had to do with an electric calculator. I am only a self taught, two finger typist.

When I bought my system I was looking for a simple cost-effective tool that I could use without first having to take a degree in computer science or word processing. I have found my typewriter/printer, the Olivetti Praxis 35, linked to the Apple, quite capable of providing just this – and a much better proposition than a £3,000 standalone word processor.

The Praxis 35 is a very well-equipped electronic typewriter on its own. It comes in a 17 x 19 x 6in plastic carrying case and the complete unit weighs around 11 kg (24.2 lb).

The Apple II is also portable, and when I work away from home I can take the complete 'typewriter-computer-word processor' system away with me, packed on the back seat of the car.

The typewriter is a daisywheel machine and produces letter quality copy. The daisywheels can be changed to give different fonts and a simple switch allows the individual letters to be printed at either 10, 12 or 15 to the inch.

It can be used as a typewriter even when it is connected to the Apple and the word processing system is up and running. This has meant that I have been able to get the system overall to print out a section of the text and then stop. I have then been able to type in, say, an algebraic expression *(shown above)* before continuing with the printing.

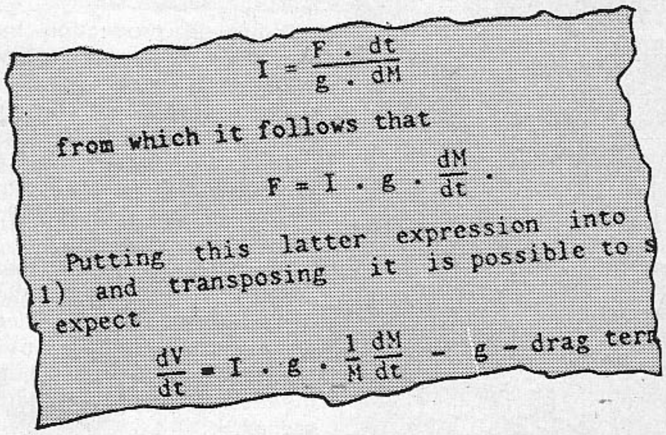
It is also a fairly simple matter to stop the Praxis so that the daisywheel can be changed, perhaps to provide the algebraic expression in a different font.

However it cannot automatically provide emboldening of special words like some of the more expensive machines and word processing packages. But it is a fairly simple matter to get the system to stop at the bottom of each page and, using the typewriter mode, go back and type over the letters which should be in bold.

The roller of the Praxis 35 is wide enough to take a sheet of A4 paper sideways, something quite impossible with many of the manual portables. This can be useful in many ways, such as typing the title on a diagram.

The typewriter takes both fabric ribbon and carbon tape in separate cartridges. The latter provides the darker print, but lasts for only 40,000 characters (only a few pages of A4 text). The fabric ribbon lasts 10 times longer, reputedly for 400,000 characters.

I type out my proof sheets using the fabric ribbon, changing to the carbon paper tape only for the final copy I send off for reproduction.



$$I = \frac{F \cdot dt}{g \cdot dM}$$

from which it follows that

$$F = I \cdot g \cdot \frac{dM}{dt} \cdot$$

Putting this latter expression into
1) and transposing it is possible to s
expect

$$\frac{dV}{dt} = I \cdot g \cdot \frac{1}{M}\frac{dM}{dt} - g - \text{drag term}$$

There are some minor problems with the system, however. For example there is no clear line underneath the print, which makes it very difficult to align a sheet of paper vertically when a page has to be put back for an alteration or correction. And it is annoying to have to remove the ribbon cassette before the daisywheel can be changed.

Other niggles concern the interfacing with the Apple.

An exclamation mark (!) on my Apple keyboard, comes out on the typewriter as an apostrophe (') – and I have to add the dots (.) underneath whenever I want to exclaim.

More seriously, the margin stops on the typewriter override the Applewriter commands. To overcome this, I have to switch on the typewriter and open out its margins to the full. If this is not done all the words on the right of a page are likely to be missing.

Despite these problems the complete Apple system has been a good buy – especially considering the low price tag (under £2,000), plus the fact that it provides not only a portable word processing system for letters, papers, manuals and even books, but also a separate portable electric typewriter, and a separate portable micro when needed.

appletips

The Apple IIe runs 98 per cent of the software available for the Apple II according to the official statistics. This doesn't help very much if you happen to own a piece of software in the odd two per cent, such as the APP-L-ISP from Datasoft.

One problem some programs have lies in calls to the monitor ROM routines as the IIe ROM is not the same as the II ROM. With this in mind, there is a way to rescue some of the non-running software.

Get a friend with a II to copy the FPBasic file from the old system master disc onto your boot disc, then alter the HELLO program on your disc so that the first three lines read as follows:

```
10  PRINT  CHR$ (4);"BLOAD
    FPBASIC, A$D000"
20  LCK = 12 * 4096 + B * 16:
    REM $C080
30  POKE LCK,0: POKE LCK,0
```

This loads the old Apple II ROM data into the top 16k of the IIe memory and locks that 16k so that the following programs will refer to it rather than the IIe ROMs.

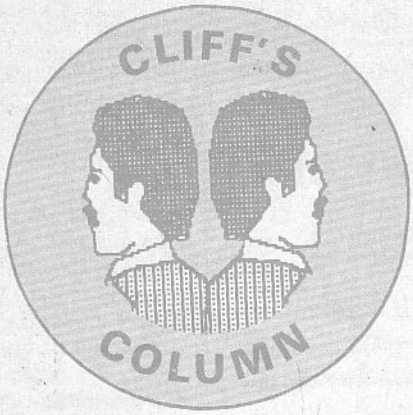
Jonathon Lewis

There are several ways to send data to a printer in Apple III Business Basic. They all require that a file be opened to a suitable device driver. Applesoft users will be quite comfortable with:

```
10 OPEN#1,".PRINTER"
20  PRINT "TO SCREEN"
30 OUTPUT#1
40  PRINT "TO PRINTER"
50 OUTPUT#0
60  PRINT "TO SCREEN, AGAIN"
70 CLOSE#1
```

Now, if you want to only print a line at a time to the printer:

```
10 OPEN#1,".PRINTER"
20  PRINT "TO SCREEN"
40  PRINT #1;"TO PRINTER"
60  PRINT "TO SCREEN, AGAIN"
70 CLOSE#1
```



Cliff McKnight muses on the state of play in the world of computer games

JANUS, the guardian deity of gates, is usually represented as having two heads because every door looks two ways. Consequently, January is traditionally a time of looking backwards on the old year and forward to the new year.

I've been compared to Janus in my time (Well, actually we called him two-faced. Ed.) *so I'll take the opportunity of doing a "retrospect and prospect".*

Last year we were Windfall. It was an interesting title, but it wasn't without its disadvantages. In America they don't seem to use the term so I was repeatedly having to explain the connection with apples when I was over there.

Even in Britain, when I attended press launches that weren't specifically Apple-related I would have to explain why a magazine called Windfall should be represented.

You wouldn't believe some of the guesses I've heard of the magazine's content!

During the year, we reviewed about 50 games, of which about half were arcade games. Adventure games were the next biggest group, with strategy games also represented. Some would be difficult to categorise.

For example, **Aztec** *combines arcade-type keyboard skills with a sort of adventure flavour. Games like* **Scrabble** *are in a class of their own.* **Prism** *was a storybook in the Masquerade mould, and* **Pot O' Gold Plus** *was a compendium of games.*

We try to achieve a balance in the games pages, but the main influence on the content is ultimately in the hands of the manufacturers.

We review those programs that we get sent for review, and in roughly the order we receive them. Having said that, I'd be interested to hear from anyone who thinks we neglect a particular category of game.

Last year's games included Sydney Development's first foray into the micro games market – **Evolution** *– and the coming year should see another exciting venture from them.*

They've teamed up with Sierra On-Line to bring the cartoon world of "BC" and "Wizard of Id" to the micro.

They plan "at least eight educational and entertainment programs within the next year and a half" and the advance press releases certainly looked interesting. Watch this space, as they say.

What else might the coming year hold? Well, Denise (my wife) and I are still working on our adventure game and Julian Brewer is still working on his arcade game.

With luck, both might be released on an unsuspecting public this year – but don't hold your breath!

And finally, Cyril . . . and finally, Esther . . . I did try to think of a new name for this column to go with the new magazine title. I wanted something as distinctive as His Master's Voice is in the hi-fi world, with all the imagery it conjures up.

However, the concensus of opinion was that His Captain's Hand was too obscure, so it will probably default to Cliff's Column.

We'll consider giving a prizette to anyone who can explain His Captain's Hand and suggest something better.

It's a hard life for a spider

Title: Apple Cider Spider
Author: Ivan Strand
Publisher: Sierra On-Line
Requirements: Apple II/IIe

ARACHNOPHOBES should skip to the next review, entomologists should read on, and budding etymologists should rush to the dictionary to find out what arachnophobes and entomologists are, because in Apple Cider Spider you take the role of the eponymous octoped.

Your night-time adventures have kept you out a little too late and you must make your way to your web at the top of the cider factory.

However, production has started for the day so you have to dodge all the apples, slicers, crushers, bottlers and cappers.

Also, hygiene isn't too good in the factory because, as well as your web in the rafters, the factory is also populated by frogs, birds and wasps.

Strangely enough, all of them seem to have amended their dietary habits to include spiders. I'm not sure what David Bellamy would make of it, but it makes an interesting game scenario.

The game has three screens and you need to work your way through each of them to get to your web. There are seven levels of difficulty numbered from 0 to 6, with difficulty being a function of speed and number of predators.

If you select level 0 (the "Teddy Bear" level – a phrase which On-Line seems to like), then you stay on level 0 throughout the game. However, if you select any of the other levels and make it to your web, the level is incremented.

There is a "hall of fame" of the top ten scores, with an option to wipe it clean.

I must admit this is a useful facility because after I've been hammering the game for review purposes, it's nice to let the children loose on it with the bonus of getting a high score.

There are various control keys to toggle the sounds off, change level during the game, restart the game and so forth. The movement keys can be selected, and the game also supports a Mockingboard.

There is a fair amount of skill involved in negotiating the moving belts and jumping onto the drop-lines.

But once you have worked out the route for a particular screen it then becomes a matter of practice. Even the predators move in predictable patterns.

Any new game from Sierra On-Line is worth investigating, and Apple Cider Spider is well written and well animated.

I think it is more a game for the children because they didn't seem to tire of it at all. I enjoyed it until I'd mastered it, and if it had more screens or levels and some randomness I'd be playing it still.

Cliff McKnight

Fantasy for a realistic price

Title: Theseus and the Minotaur
Authors: Bruce Nesmith and Keith Enge
Publisher: TSR
Requirements: Apple II

FOR quite a while now I have been looking out for successors to the two greatest creations I have ever come across. I am speaking about The Lord Of The Rings and Wizardry.

Any reader who has not heard of them until now deserves to have his Apple taken away! I am happy to say that I have recently found a trilogy better than The Lord Of The Rings, I have not, however, found any Apple game which surpasses Wizardry yet.

I am telling you this because TSR have just sent me a copy of Theseus and the Minotaur to review.

It is another Dungeons & Dragons maze type fantasy game: This time you take the place of Theseus in his quest to slay the minotaur.

As you may have guessed, the bad news is that it is nowhere near as good as Wizardry.

The good news, however, is that it only costs £12.95 – half the price of most game software!

The maze graphics in this one are standard, with the maze being shown by simple hi-res lines in 3D.

The time it takes to draw the view is very slow indeed. In fact, the program has very slow execution overall.

Every time the player enters a command involving forward motion, for example, there is a four-second delay before the new view is drawn. Also the maze is often completely redrawn when there hasn't been a single change in the situation.

Within the 3-level maze roam half-human monsters and venomous snakes, along with strange creatures who may well aid you in your quest.

All these creatures are shown by colourful pictures within the display of the maze. These pictures are the best point of the game, and there is a particularly good one of Princess Ariadne who must be retrieved from the maze.

Unlike other games with this type of monster graphics, the pictures actually appear smaller as they get farther away from you.

I was very impressed by the fact that the minotaur is shown from behind if he is walking away from you.

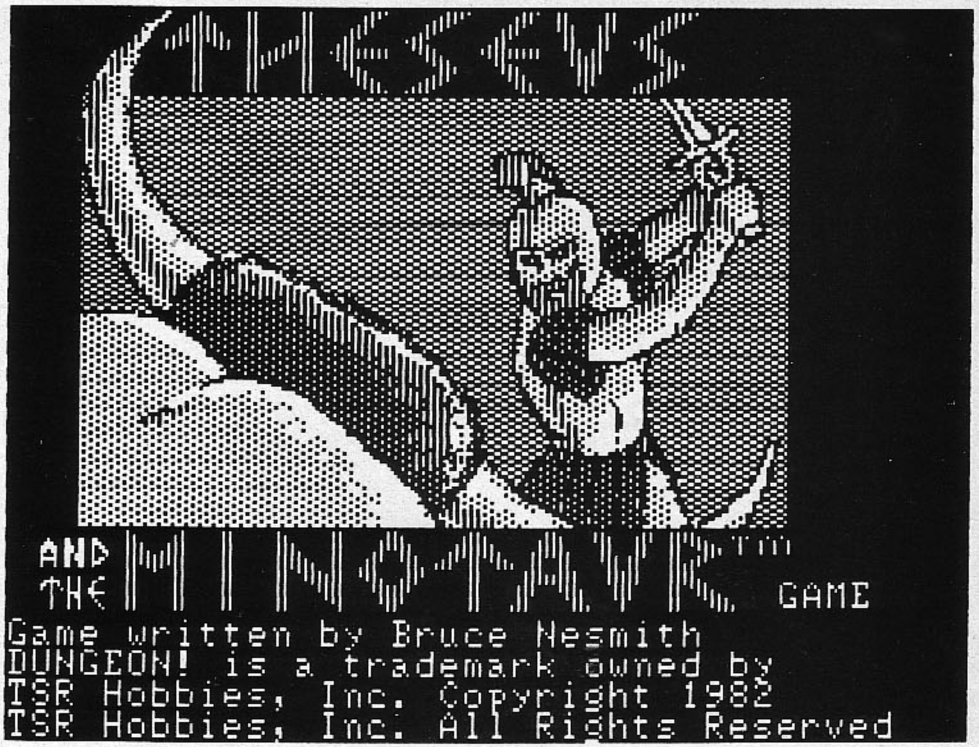
In most such games, different types of armour, weapons and potions are obtainable. In Theseus and the Minotaur unfortunately, the only ones are weapons.

Theseus starts with just his bare firsts, but is able to find better weapons by searching defeated monsters.

Extra options to maintain interest in the game include secret doors, selectable maze complexity, and "halls of mirrors". There is an excellent piece of graphics to be seen inside the "halls of mirrors" – it's a real work of art!

If the game is completed (by exiting the maze), the player is allotted points according to his performance in the maze, and to how difficult his selected options were. I found it possible to score 1,200 points in just three moves.

Somehow this points system seems to be a bit of an anti-climax after spending a long time getting to the finish. I'd much rather receive a half-kingdom reward.



```
Game written by Bruce Nesmith
DUNGEON! is a trademark owned by
TSR Hobbies, Inc. Copyright 1982
TSR Hobbies, Inc. All Rights Reserved
```

As I have said, this one is not the best of its kind, but it is fairly cheap.

If you find that your Christmas allowance doesn't quite have enough left for a normally priced game, it's probably worth getting Theseus and the Minotaur on its graphics merit alone.

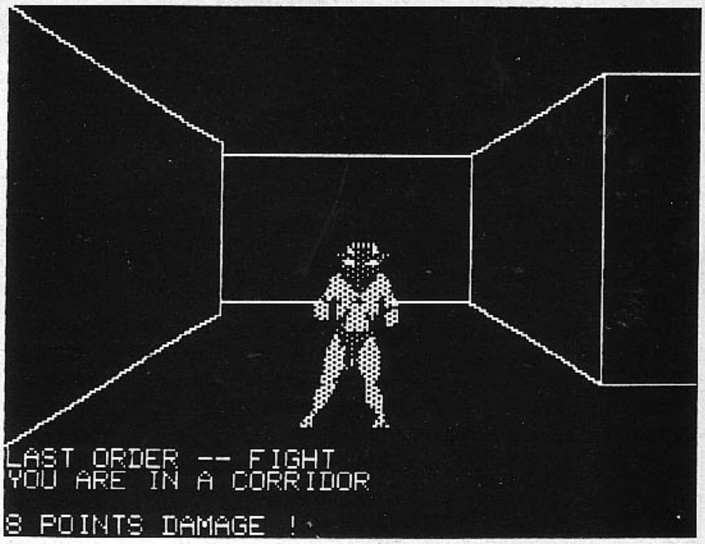
Julian Brewer

Bombs, bombs and yet more bombs

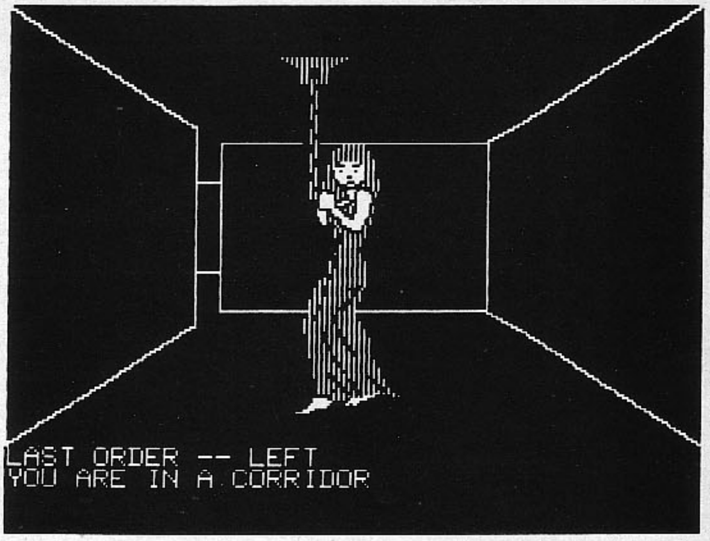
Title: Thunderbombs
Author: Thomas Becklund
Publisher: Penguin Software
Requirements: None stated

I THINK I'm beginning to feel like the film reviewers must have felt after seeing their thousandth Western. I mean, what can you say? "Bang, you're dead!"?

Thunderbombs from Penguin Software is my thousandth Western. In this game you move your cloud ship up and down the centre of the screen, shooting



```
LAST ORDER -- FIGHT
YOU ARE IN A CORRIDOR

8 POINTS DAMAGE !
```



```
LAST ORDER -- LEFT
YOU ARE IN A CORRIDOR
```

enemy ships on either side.

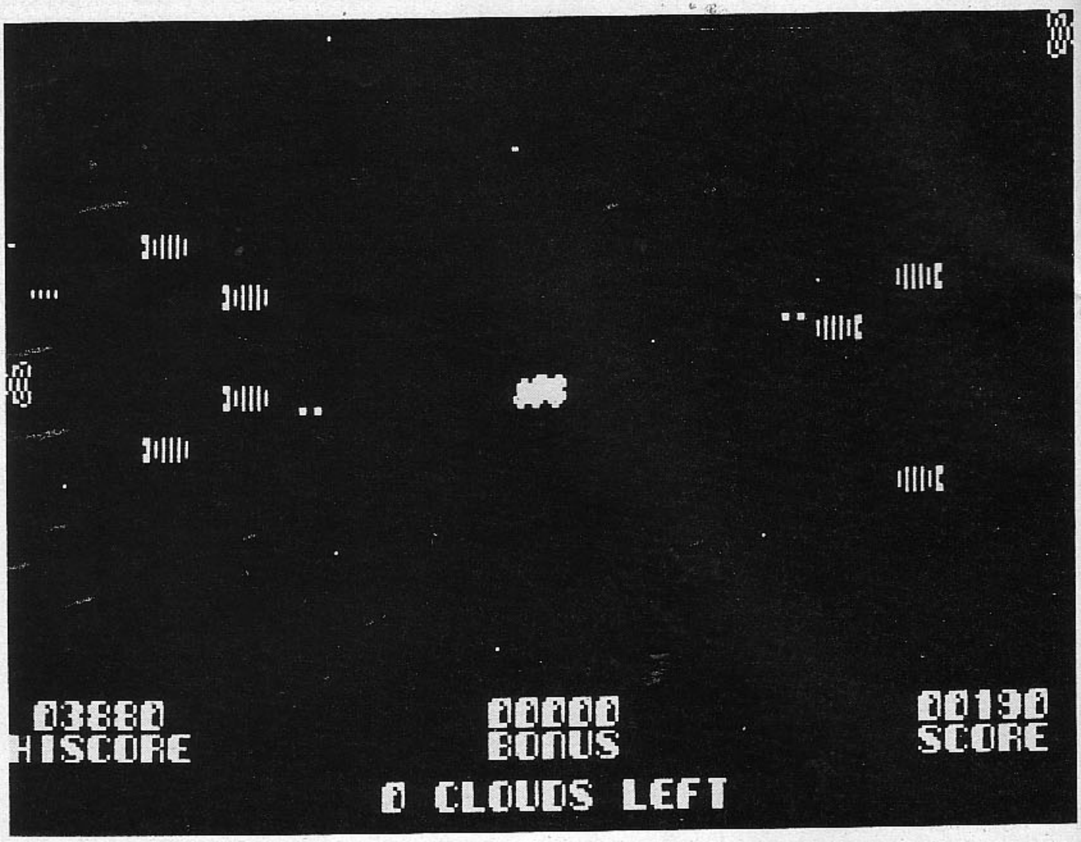
These are replaced unless you manage to hit the drone-clone replenishers that occasionally appear at the extreme sides of the screen.

The enemy ships are gradually moving toward you – a sort of ambidextrous Space Invaders with nowhere to hide.

If you shoot the replenishers and then clear the screen, you move to the next level. You get a bonus cloud ship for every 2,000 points you amass.

Thunderbombs is a reasonable game, well written, and hard enough to be demanding . . . but we've seen it all before. I'm a bit surprised that Penguin thought there was room for such a game in an already-crowded market, even if their games are cheaper than most.

Cliff McKnight



Lack of creativity could sound the death knell for arcade games

"Re the computer game market. The designer/programmers are a hard lot to deal with. They seem totally uninterested in what is marketable and wholly consumed with showing off their technical prowess.

"The arcade-type game is fast becoming a vanishing species. Most of the smaller companies who flooded the market last year are out of business and the big guys are all taking a bath in the stock market since they released estimated losses.

"The arcade-style market is completely saturated here. Retailers are dumping once-upon-a-time $30 items for four and five dollars. The only gaming software which seems to be holding up are the ones which are actually games and not just hand/eye coordination tests, but even their prices have come down about 20 to 25 per cent."

THOSE encouraging words came from my US games publisher, a man not particularly noted for his pessimistic outlook.

Since the market for arcade-style games hasn't saturated this side of the Atlantic, it's no surprise to find that Buzzard Bait, the latest offering from Sirius, is another hand/eye coordination test.

The idea is that buzzards have taken to snatching humans as fodder for their chicks. And somehow you, with your hovercraft and missiles, have to put a stop to it.

Your problem is complicated by penguins (yeah, penguins) which will write off your craft by crashing into it if you don't manage to jump over them.

As an interlude between bouts of buzzard-bashing, you are invited to pick up ship parts while bouncing off an erratic flock of flying (yeah, flying) penguins and avoiding mines.

On top of all this, the buzzards will drop rotten eggs on you when they get really annoyed.

There are some nice touches to the program. It starts off with the buzzards mating, two by two, above their nests.

Parents need fear no overt eroticism. The process is symbolised by the appearance of stylised hearts which, perhaps inevitably, turn into eggs.

The eggs hatch and huge buzzard chick heads appear demanding food. After that, unfortunately, it's downhill all the way.

Technically the game is excellent. Sound effects are quite magnificent, graphics good and animation smooth.

You can keep track of your current score top left of the screen and there is a high-score feature which shows you the target you need to reach to become current reigning champ. Joystick, paddle and keyboard control options are all included.

Perhaps in an attempt to encourage addiction, the manufacturers have made the game automatically restarting. That is to say, you aren't given the option to play again when one game finishes – it simply happens.

Title: Buzzard Bait
Author: Mike Ryeburn
Publisher: Sirius Software
Requirements: Apple II or IIe

The program did, in fact, prove totally addictive to my favourite teenage arcade game maniac, who considered it as good as any he'd played.

As his scores and experience mounted, he discovered it was possible to shoot down high flying penguins and also that baby buzzards eventually started to use bombs.

But if you're over 21 (well, over 25 then), I doubt you will be playing this one for more than an hour or so before the whole thing starts to pall. As my publisher suggests, technical ingenuity isn't everything. Buzzard Bait does not lack colour or programming expertise – just creativity.

Cliff McKnight

Ikons come cheaper with Homeword

IF you are fascinated and amazed by Apple's Lisa, but feel left out in the cold by its price, you might try consoling yourself with a new word processing package that costs only £39.95.

Homeword has two similarities with Lisa. It runs on a member of the Apple family. And it employs icons (symbols representing objects or functions performed by the program).

Authors Sierra On-Line say that HomeWord is a writing tool designed specially for the home, offering "the power of a sophisticated word processor without the complications".

The package has three "screens" – a typing area (where you type your text), the icon diagrams (used to select command functions) and a picture of the page you are producing.

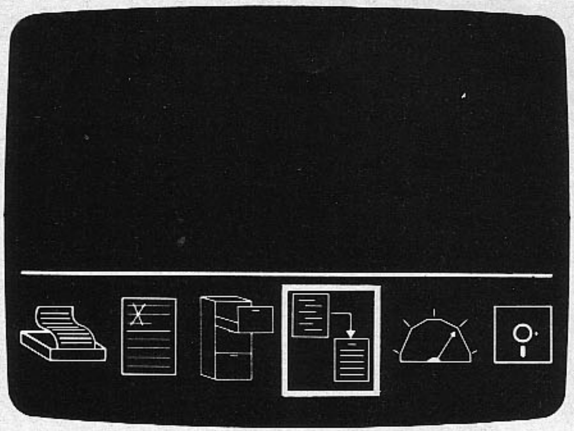
This picture, although very small, lets you see how your printed text will be laid out.

HomeWord is supplied with a tutorial tape cassette and a manual that is aimed at the complete beginner.

Features include edit functions such as copy, erase, find, find and replace, and move text; layout functions like boldface, underline, margin setting, headings/footings and line spacing; and file functions (insert document into text, load/save document, include file and help screens).

The program has a shift-lock facility (without keyboard modification) for earlier Apple IIs.

● *Contact your Apple dealer.*



Homeword menu options, represented by ikons.

Incredibly, Jack is here

IT has taken nearly a year for The Incredible Jack software package to reach the UK market from the US – although in that time the program has been upgraded to run on all Apple IIs including the IIe.

It had already been out for a couple of months when we first described it in the March 1983 issue of Windfall (page 73).

Authors Business Solutions Inc claim that "It does most of what you bought your Apple for".

The Incredible Jack offers "calc analysis, form letter generation, mailing list, filing, and word processing capabilities on one disc", claim the authors.

Jack is menu driven, its functions are integrated, and it operates in four main modes: Disc, File, Select and Print.

Its display is 20 x 80 characters. It uses vertical scrolling and can handle documents up to 60 pages long.

The program is available in the UK from most Apple dealers for around £129.50.

● *Business Solutions, 60 East Mann Street, Kings Park, New York 11754, USA.*

Graphing pack

THE first in a range of mathematical suites from McSoft Computing, "Cartesian" deals with graphing functions and their derivatives (differential curves) for an appreciation of maxima, minima and points of inflection.

It provides definite integrals, complex equation solutions, and the Newton Raphson method of precise root extraction.

The material is presented through high resolution graphics. The program also incorporates spreadsheet capabilities to allow plotting to extremely high or low x axis values. Magnification of scales is also possible.

The documentation included with the package explains the underlying principles and can be used with orthodox course material.

Cartesian for the Apple IIe costs £27.75.

● *McSoft Computing, Findrum, Convoy, Co. Donegal, Ireland. Tel: 074-22286.*

Blackboard's 700 options

IT is called The Blackboard, it is British-built and it is described by designers Leicester Computer Centre as a super intelligent printer card for the Apple II and IIe.

Designed to take full advantage of the new text and graphics facilities of the Epson and Apple DMP printers The Blackboard's 45 commands can produce over 700 different ways of printing just one hi-res screen.

Independent scaling of the X and Y axis is possible permitting the Epson FX series to print 1:1 graphs, previously only available in the 82 series.

The Blackboard prints single and double sized graphs and allows the X axis to be increased by a factor of five and Y by eight.

The automatic substitution of £s for $s is built in and line spacing can be set to single, double or none – and a text screen dump works from any language.

Basic, Pascal and CP/M are fully supported. The Blackboard costs £99.

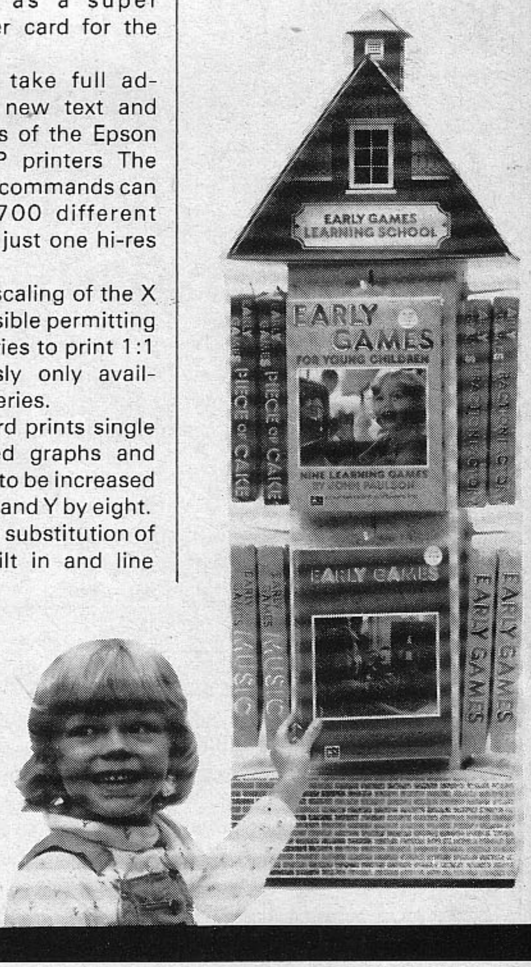
● *Leicester Computer Centre, 9 Jarrom Street, Leicester. Tel. 0533 556268.*

Games for tiny tots

UK distributor Pete and Pam Computers are now handling a series of programs for young children by Counterpoint Software of Minneapolis, USA. There are five programs in the Early Games series.

"Early Games for Young Children" is a set of nine activities for children aged two and a half to six. They can work without adult supervision to learn the keyboard, count colourful blocks, add and subtract stacks of blocks, work with the alphabet and their names, and draw colourful pictures.

In "Machmaker", colours, shapes and sizes combine with music to help children develop



discrimination skills.

Children of ages five to 12 learn to create tunes, play favourites, and become familiar with note names in "Early Games Music".

Visit a magic bakery where children learn to add, subtract, multiply and divide. The bakers explain how to do it and they make it fun. It turns maths problems into "A Piece of Cake".

"Fraction Factory" helps children to see and describe fractions, find equal values, multiply whole numbers by fractions, and add and subtract fractions.

Early Games feature multiple activities, easy-to-use prcture menus and colourful graphics.

Each program in the series costs £19.95.

● *Pete and Pam Computers, New Hall Road, Rawtenstall, Rossendale, BB4 6JG. Tel: 0706 212321.*

Accounting systems for Apple III

FIVE new accounting software systems for the Apple III, covering all the main business functions of sales, purchasing, invoicing, stock control and payroll have been launched in the UK by Jarman Systems.

The Jarman "Plus" series is designed to take advantage of a fully configured Apple III system with 256k memory and hard disc storage. It provides extensive management report facilities, time-saving routines and customer convenience/security aids, designed to speed business operations and maximise profit, claims Jarman.

The new Jarman systems comprise the J10 Plus (Sales Accounting), J20 Plus (Purchase Accounting), J30 Plus (Nominal Ledger), J50 Plus (Payroll) and J60 Plus (Invoicing) series.

The J10 Plus Sales Ledger system allows 999 analysis codes and sales analysis reports by profit centre, which can be extended to individual salesman level.

The package, states Jarman, can generate month by month sales for the whole year to date within the scope of one report – facilitating sales trends investigation.

Extended customer records available through the J10 Plus include: sales and payment by individual customer, discounts receivable, customer group categorisation, plus individual customer information, which can be defined by the system user to generate particular descriptions.

Much of the information in these records can be automatically transferred into the invoicing system. Customer records can be selected without account code keying. Non-VAT paying customers can also be indicated.

The J20-24 Plus Purchase Accounting system includes up to 1,000 departmental codes, an unlimited number of budget codes and a detailed suppliers record, allowing analysis of purchases by supply group.

Users can choose their preferred cheque layout and printing method.

The J30-34 Plus Nominal Ledger and Cash Book system allows sales and purchases to be transferred into it several times during the month. This permits spot checks on the company's position before the end of the month.

It provides "suspense accounts" for non-routine balance adjustments and the automatic entry of payroll figures to the nominal ledger. Relevant balances and the net profit position can be screened at any time.

The J54 Plus Payroll system can handle hourly rates payments and those calculated in fractions of one penny.

The J60 Plus Invoicing suite handles product record data and provides the user with recommended stock levels.

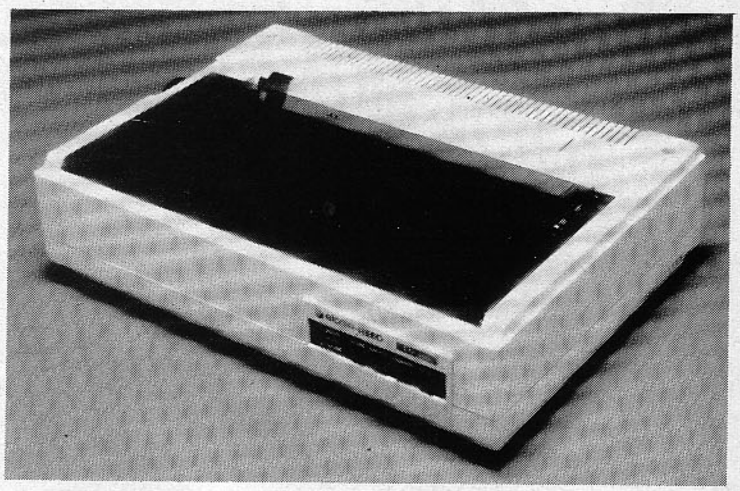
Features common to all packages are formatting to use 80-column screens, a choice of keyboard entry levels to suit the operator, password availability throughout for maximum security, and a high level of options in printer settings and preferred report layouts.

Options can be centralised for each package, allowing the option, once selected, to be applied automatically to all other packages used without further keying.

Report features include numbering of key reports, for example bank payments, so that reports cannot be lost and are easier to audit, and the use of "footings" as well as headings on reports to assist control.

The packages cost between £290 and £440.

● *Jarman Systems, 6A Dolphin Square, Tring, Herts HP23 5BW. Tel: (0442 82) 6841.*



A NEW series of daisywheel printers from Silver-Reed is being distributed in the UK by Star Computer Peripherals.

With all three models in the range – the EXP 500, EXP 550 and EXP 770 a user can control the carriage movement in units of 1/120in, and the forward/reverse paper feed in units of 1/48in.

Other features include subscript, superscript and bold typing plus a paper injector which automatically feeds the paper until the printhead is positioned an inch from the top.

Print speed for the EXP 500 is 12 cps, while the EXP 550 and 770 models achieve 17 cps and 31 cps respectively.

The machines cost from £299 for the EXP 500 (pictured).

● *Star Computer Peripherals, Unit 6, Imperial Way, Worton Grange Industrial Estate, Reading, Berks RG2 OHU. Tel: 0734-752273.*

Disc drive analyser

DISC drives are finely tuned instruments, and even in normal use can fall out of alignment, affecting the integrity and accuracy of data.

The Datalife disc drive analyser, introduced recently to test the performance of Apple II and III drives, has been improved to include informative graphics.

It is supplied with easy, step-by-step instructions and a special, preprogrammed testing disc.

The disc is simply placed in the drive to be tested, and a test is selected from the menu on the screen. The tests can be done individually or all of them may be run consecutively.

This is the only product that automatically diagnoses the four most critical areas of Apple flexible disc drive performance, claims manufacturer Verbatim.

It checks radial alignment, with the drive's read/write head centred directly over the appropriate flexible disc track. It also checks disc speed – drives are calibrated to revolve a flexible disc at an industry standard (RPM) rate. If the speed is incorrect, read/write errors and data loss may result.

The analyser also checks write/read. It tests the ability of a drive to accurately record and then play back a random series of numbers.

● *Verbatim Ltd, Wallington, Surrey. Tel: 01-773 1115.*

Invoicing system

A SYSTEM to raise and print sales invoices and credit notes, post the total value (goods and

VAT) to the appropriate sales ledger account, downdate quantity in stock and update the sales to date field in the stock files, is available from Integrated Micro Computer Accounting (Imca).

It is called Sales Invoicing and Stock. It works on floppy or hard disc. Stock numbers are alphanumeric and the stock file holds a product description, four selling prices, cost price, quantity in stock, minimum stock level, group code, gross profit earned to date, sales and receipts to date.

Imca says invoices can be printed on plain or two types of pre-printed stationery and are built-up on the screen. Non standard products can be keyed in, as can comments. Invoices are printed as each one is finished and the Stock and Sales Ledger files updated at the same time.

Discounts can be built in to the customer record in the sales ledger and applied to the whole of an invoice, keyed in as part of the invoice header, or added as an individual line.

From the stock file, reports on stock value, movement, re-ordering and profitability can be produced. The system will also produce price lists. All the reports can be for the full stock file, any selected product group, or any selected alph prefix, and can be sorted into a logical sequence.

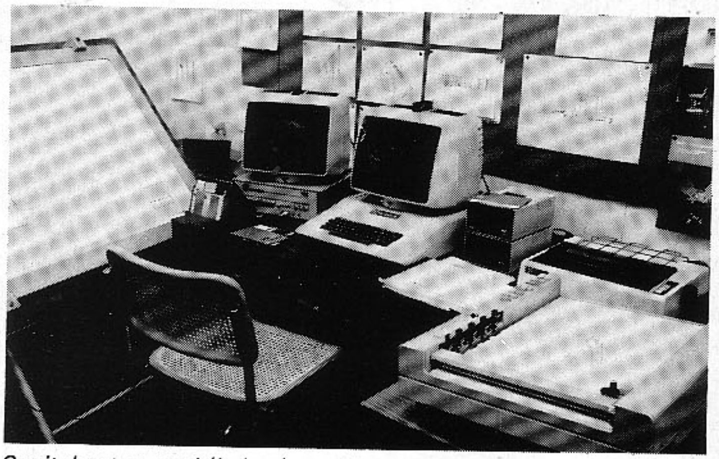
● *Southern Computer Software, 7 Park Hill Road, Torquay, Devon. Tel: 0803 212957.*

Spot the disc

THE latest disc storage box from BFI Electronics – the SEE 10 – presents 10 stored floppy discs in view when opened up. This allows the user to select and remove one disc without bending or dislodging others.

When closed it can be tucked away onto storage shelves or cabinets. When opened, the lid hinges down to provide a support for the storage compartment. The front panel swivels forward to present the 10 discs in "fan" formation. The box costs £2.74.

● *BFI Electronics, 516 Walton Road, West Molesey, Surrey KT8 OQF. Tel: 01-941 4066.*



Capitol setup, and (below) sample outputs

Taking the 3D view

A computer-aided 3D perspective visualisation system for architects and designers is called Capitol.

It allows designers to experiment on screen and to test ideas from any viewpoint, at any scale and in different line-styles or colours, depending on the hardware configuration chosen.

According to authors Graphicsaid, ideas can be copied, moved, rotated, scaled, twisted, wedged or partly erased with up to 50 objects in current memory at any time.

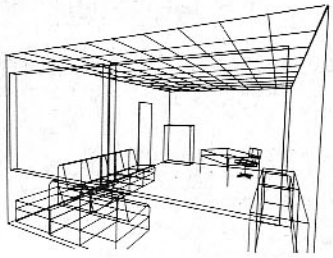
"Using English word commands via the keyboard, designers can interact with their ideas during the development of a scheme and store them on floppy disc if required", said a spokesman.

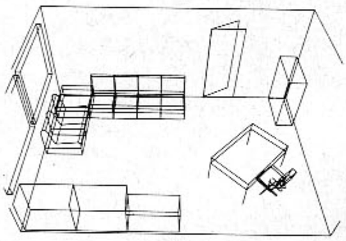
Two versions of Capitol are available for the Apple II – Rev 6.2 (£500) for the standard 48k machine with language card and two disc drives, and Rev 7.2 (£700), which gives enhanced resolution on the screen using the Digisolve VGP card.

Digitiser input and plotter output are also catered for.

The package comes with tutorial data files and lessons and an illustrated manual.

● *Graphicsaid, 17 Friarswood Close, Yarm, Cleveland TS15 9JG. Tel: 0642-781791.*





Mind your language with Speller

BANK Street Speller, a spelling checker designed for use with Bank Street Writer, the word processing package, has been introduced by Broderbund Software. It automatically proof reads your writing, finding spelling errors and highlighting them, in context.

When an error is detected the program suggests a correct spelling, if desired. If you are unsure of a word you can look up spellings, even using "wildcard" characters, in Bank Street Speller's electronic dictionary.

The program then automatically inserts the correct spelling directly into your text files, replacing misspelled words throughout, or instance by instance.

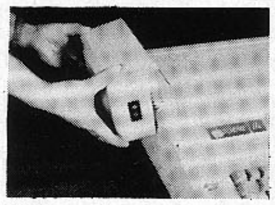
Other features include a facility to display or print out alphabetical word lists and provide a count of the total number of words used in a document; to establish the number of unique words you have used, and with two disc drives, to create your own customised dictionary.

Bank Street Speller looks and works like the Bank Street Writer, using the same screen layout, control keys, easy-to-follow menus and prompts. And there are no complicated set-up procedures, claims Broderbund.

The program costs $69.95

● *Broderbund Software, 17 Paul Drive, San Rafael, California 94903. Tel. (0101) 415-479 1170.*

Cool idea



A NEW cooling fan for the Apple has been launched by Pete and Pam Computers.

The unit measures about 7.5 × 4 × 3in. The fan itself is encased in plastic colour-matched to the Apple's own casing. It is easily installed to the left hand side of the Apple by hooking it onto the ventilation slots.

The fan costs £39.

● *Pete and Pam Computers, New Hall Hey Road, Rawtenstall, Rossendale. Tel: 0706 212321.*

Accelerator

A MUM utilities disc with selected routines, to take the delays out of Applesoft programming, is available from Heyden Datasystems.

The principal aids on the Macro Utilities Master disc are a set of macros, each called up by only one or two keystrokes.

With these you can renumber, shorten or merge programs, search for variables or strings, delete and insert, measure free memory and even define your own macros.

The disc costs £18.

● *Heyden Datasystems, Spectrum House, Hillview Gardens, London NW4 2JQ. Tel: 01-203 5171.*

What do you do when a computer won't compute?

I HAVE an Apple II computer with integer Basic on board and an Applesoft ROM card which I use for all my programs.

It had been working perfectly for the past four years until I had it serviced, when autostart ROM was fitted in.

Now my Apple works perfectly for hours until it decides to call it a day, and although the display is perfect, no key and no power on earth can make it go.

Can you imagine my disappointment when, in the middle of the program which I may be writing, I have to turn it off.

When trying to restart after a while the Apple displays a white screen with some coloured squares and again all keys are disabled. This happens even if the Apple is completely cold.

I have spent over £100 with several so-called experts who could find nothing wrong.

My Apple was working then unfortunately. Sometimes I succeed eventually to restart it, although I do not know how.

In my desperation I decided to write to you for advice, for which I would be very grateful. – **J.B. Suzin, London.**

● We sympathise with you – there is nothing worse than a computer which won't compute.

The short answer to your query is that we don't know any magic answers. However, what happens if you remove the autostart ROM and return to your old system set-up? If your system still plays up it would seem there is a conflict somewhere.

It may be worth taking out the Applesoft card and cleaning its gold-plated connector fingers or teeth.

This should be done very gently (away from the Apple) using a very soft, clean pencil rubber.

More important, try pressing every chip on the Apple mother board firmly back into its socket. Some chips seem to ride up with time (and maybe heat) and you may correct a fault by pushing them home.

Max Parrott

Faster way to the Y co-ordinate

THE routine used by Dr Parrott in the November issue for calculating the base address in memory for the y co-ordinate on an HGR screen is not the best one.

The routine I used in my HGR screen program published in the May 1983 issue of Windfall is about 11 bytes shorter and faster.

"Vert" is the y co-ordinate, "OFFSET" is the page reference ($20 for HGR, $40 for HGR2) and the result is given in "COLOUR" (hi) and "PIXEL" (lo).

Those names do not mean anything normally, but were used as temporary storage in hi-res screen. – **John Blaiklock, Norwich.**

Taking the Ascii Express

I AM seeking some advice in respect of modems and communications software that is approved by British Telecom.

I am in touch with DaCom Ltd, who make the Buzzbox (a 300 baud self powered modem), and the DSL 2123, an all singing and dancing 300, 1200/75, 1200/1200 machine.

The point is that I already have a copy of South Western Data Systems Ascii Express program. Will this work in the UK?

I also have a Hayes Micromodem II and the same question applies to this piece of equipment. I realise this is only a 300/300 board so would not be Prestel/Micronet compatible.

Can you or one of your colleagues supply a list of UK comms software? Your Windfall software book doesn't list much that isn't American.

Also, thanks for Max Parrott's graphics text in the November Windfall – very nice – but what assembler is he using?

I thought I would try to learn Big Mac and enter his source code – it didn't like the first few lines with the EPZ.

Also it was out of memory – Mac sets HIMEM at $8000 by default, so I assume I should assemble at a lower address and then change the OBJ location from the monitor. Am I on the right track? – **P Trinder, Sunningdale, Berks.**

● Ascii Express is a very powerful program and should certainly work here. Another very useful, possibly easier to drive program, is VisiTerm. We do not know of any relevant UK communications software.

I also regret we don't know much about the Hayes modem, not having one ourselves.

Neither do we have the assembler Big Mac (the published listing was Lisa-generated) but we are under the impression that it assembles the object code at HIMEM ($8000) upwards until it meets the downward growing symbol table and it stores the source code at $901 upwards until it meets HIMEM.

This seems to give plenty of space unless you used the OBJ$1000 pseudo opcode which would change HIMEM.

If you did use it, leave that line out and things should be okay. If there is still no room, assemble the code and character data separately.

Max Parrott

Lurid entry to hi-res screen

HAVING recently mastered the basics of machine language on the Apple II I cannot find anywhere the methods for using the high resolution screen. Please assist in this problem. – **R.T. Gibson, Lancs.**

● By far the best thing is to buy a copy of *"Apple Graphics and Arcade Game Design"* by Jeffrey Stanton, published by The Book Company. It will cost you about £12.

Despite the lurid cover, it is very well worth reading.

Max Parrott

ROMs take a dim view of being upended

I RECENTLY brought an Apple computer from New York to the UK. Before the journey, I removed all the ROM chips from the main board, since I was told they might be spoiled by X-ray scanning equipment when I went through Customs.

Unfortunately, when I replaced the ROM chips I put two of them in back to front. Although I have since put them back in the correct positions my Apple still doesn't work properly.

What damage could have been done to my Apple? Do you think if I renew the two ROM chips the computer would work normally again?

I would be grateful for any suggestion, and thanks for your excellent magazine. – **Patrick Massey, Birmingham.**

● It is very easy to make a mistake when replacing large chips – and leave, for example, one of the connector pins bent up underneath the body of the chip instead of correctly inserted in the socket.

Try taking out the ROMs and replacing them, slightly splaying the pins outwards before insertion.

If they still won't work you'll have to try more ROMs just to find out what is wrong. It seems to me that being a member of a user group would help with this type of problem.

Finally, are you sure it is the ROMs that are at fault and not the power supply?

If you bought an American Apple it won't run on a UK power supply without an adapter. Check with your dealer for a solution.

More, please, on HRCG

I WONDER if you could suggest a source of instruction for the use of the High Resolution Character Generator, other than the slim manual provided.

We bought the Toolkit with a

view to including graphic displays in programs I have written, which will require mixing text and graphics. To do this, I need to use the HRCG under program control.

The manual does not seem to give clear instructions, to me anyway, as to how to go about this – specifically, which programs to take off the Toolkit disc and where to put them on my program disc.

Also, the manual advises that to get out of the HRCG one must press RESET and type FP. How is RESET evoked in programmed mode? **T. Shreeve, Norwich.**

● To run your HRCG you need on your own disc, RLOAD, a character set and HRCG itself.

Study LOADHRCG (or RIBBIT) to see how it is used. Alternatively, the text printing routines published in the November 1983 issue of *Windfall* may, in fact, prove more useful.

The HRCG can be turned on and off under software control. The following two subroutines will switch back and forth.

The variable "ADRS" is assumed to be assigned the same as in the subroutine starting at line 2000 of MAXWELL on the Tool Kit disc.

```
3000 REM     TURN OFF HRCG
3010 PRINT  CHR$ (4);"PR#0"
3020 PRINT  CHR$ (4);"IN#0"
3030 HOME
3040 TEXT
3050 RETURN
4000 REM     TURN ON HRCG
4010 CALL ADRS + 3
4020 RETURN
```

Beware, the HRCG overwrites the text screen when outputting to the hi-res screen. It will destroy any information there. You will have to rewrite the text screen after using HRCG.

Max Parrott

More tasks for ampersand routine

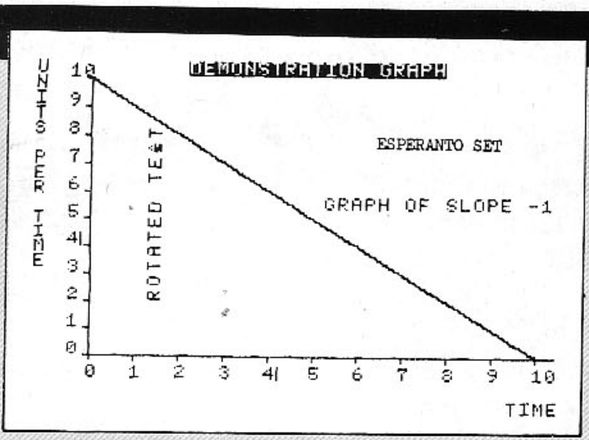
HOW pleased I was to find Mr M. Parrott's Ampersand routine to print alphanumerics on the hi-res screen in the November 1983 issue of Windfall.

After typing it in and running his demo programs I thought maybe the character sets available on the Tool Kit disc might also run using this routine.

They did actually run except that on a few of the character sets, Byte, Cyrillic, Esperanto, Gothic, Greek, Pudgy, Roman and upside down, the shift ½ bit right indicators were visible, especially on the figure 4 (see example above).

These could easily be removed using the Animatrix program and saving the character set on the required disc and loading it at $9400 straight after M. Parrott's routine.

Any non-used characters in the set could be removed by saving only the first 496 or so bytes of the set, and users could also define the character set they want by using Animatrix and combining the two. – **M. Evans, Bristol.**



A different font from the DOS Toolkit

Blackboard answer to a WP problem

I WAS very interested in recent articles by Mike Glover about fully utilising the facilities for dot matrix printing on Epson printers from word processors and hope this letter will be a warning to anybody about to consider purchase of the Applewriter IIe for the IIe machine.

Using an Apple II Plus, I was able to do quite sophisticated dot matrix printing from Applewriter 1.1 via a type 2 interface to an Epson MX82 using a program called Omnifont.

Any of the fonts from the DOS Toolkit can be used, including any graphic or double height characters you define using the Animatrix program in the Toolkit.

However, Applewriter 1.1 will not work properly on the IIe, apparently because of re-definition of certain screen characters.

Since an 80 column display is available on the IIe the obvious thing to do is upgrade to the more sophisticated, and rather more expensive Applewriter IIe.

It was therefore disappointing to discover that Applewriter IIe was unable to support this type of graphic printing itself and no suitable driver programs exist for it.

One of the principal reasons appears to be the fact that the Applewriter IIe cannot accept insertion of a Control-@ necessary for the 8132 interface (although it is possible to insert other control characters). It is also copy protected and so is practically impossible to modify.

Perhaps some enterprising enthusiast can come up with an alternative printer driver for Applewriter IIe, or maybe one of your readers has found a way of inserting Control-@ in what is otherwise a good program. – **Shaun Hope, Milton Malsor, Northants.**

● Shaun Hope is quite correct in his assertion that Applewriter IIe, unlike its predecessor Applewriter II, won't accept Ctrl @ (Ascii 0) in its text stream.

This makes the printing of bit image data impossible unless a special driver is in the printer card. For other commands substituting O (Ascii 48) will often work.

One small correction – it is the printer not the printer card that needs to see the Ascii 0.

A printer card the Blackboard, which can overcome this problem, is available from Leicester Computer Centre. – **Mike Glover.**

The darts that crashed

I TYPED in the Darts program from the March 1983 Windfall, saved to disc, typed RUN – and it crashed.

It came up with "illegal quantity error in 110. I seemed to have cured that by splitting the line up (it's a bit slower though).

Now it comes up with – "out of data" in 210, and over half the data is missing when you list lines 1210-1230.

As my head aches with puzzling can you suggest anything. – **Nick Trotman, Atherstone, Warwickshire.**

● First things first – line 110. Your error message and your cure both suggest that there was a typing error within 110 (or 100, or 90) such that some non-valid numbers are being plotted (or rather they aren't).

The most likely is the variable R0 (zero) being typed in as RO in one or more cases. I suspect that on retyping the line the error was cured.

The second problem, that of losing lines 1210-1230, is almost certainly due to the HGR command in line 990 wiping out the end of the program.

The program as written fits quite happily in the usual place ($800-$2000 if you are technically minded) so either you were running another program previously which reset the beginning of the program printer OR your version of darts is longer than our original for some reason.

If your programming experience on the Apple is limited you possibly typed in the REM statements (lines 19, 89, 119, 129, 139, 149, 189, 199, 219, 239, 389, 399, 509, 519, 569, 589, 729, 839, 859, 909, 1109, 1199 and 1209) with a lot of spaces in each so as to make them stand out as in our listing).

Is so then delete these lines and try it again.

Incidentally the REM statements were made to look like they do by putting line feeds into them using a commercial editing program.

Max Parrott

Handicapple.. computers in rehabilitation

SEVERELY spastic people are using Apples for communication at the rehabilitation centre "Het Roessingh" in Enschede, Holland.

Here MARTIN KEESEN explains how programs and hardware are developed to meet the special needs of handicapped people.

ONE way to establish communication with non-speaking people is through symbols.

At our centre we use the Bliss symbol language for this.

Bliss comprises a set of well-defined symbols standing for verbs, nouns, actions etc.

Each symbol consists of a number of basic forms arranged in a specific way. Figures I and II give an idea how this is done.

The definition of the symbols is supervised by the Bliss Communication Institute, Toronto, Canada. To date there are over 1,400 symbols and the language is still growing.

Bliss is normally used by pointing to a symbol drawn on a piece of paper.

This action is very difficult for severe spastics – and to make it possible for them to use Bliss we have implemented the language on an Apple II.

The Apple II is well suited for symbol drawing because of its shape-table utility, which allows complex symbols to be drawn from basic forms.

We developed a symbol editor program to construct Bliss symbols from a set of basic forms stored in a shape table.

To build a vocabulary the basic forms are positioned one by one to construct a Bliss symbol. The x and y coordinates are stored in a file on disc.

The next task involves putting a set of symbols on a "page", using a screen editor. A page is a screen filled with a matrix of symbols.

This editor makes it possible to personalise the symbols used. For example, the meaning of the symbol "sister" can be changed to the actual name of a user's sister.

The page data is stored on the Bliss user's disc, plus another program which controls the use of the Bliss pages.

This user program is adapted to meet the requirements (and the potential) of the user. Severe spastics can control the program with a single switch.

The matrix of symbols is scanned in a row-column fashion, that is first the rows are shown one by one and when the desired symbol is in the indicated row the switch is applied.

Next, only the symbols of that row are shown on screen in succession and the relevant choice made by flicking the switch. Now the symbol is displayed with its meaning.

After a short delay, the page is shown again and the selection process repeated.

If a user makes a wrong choice, he can reject his selection by flicking the switch twice in quick succession. This returns the program to the stage before the wrong choice was made.

To select another screenful of Bliss symbols, a user must choose an item out of the first row "menu" options.

The pages can be filled with symbols concerning specific subjects, grouping them, for example, by family, school and daily living.

A wide variety of controls is available – ranging from foot switches to "suck and puff" or even electromyographic switches which can be operated by contracting a muscle.

The communication becomes much faster if a user can control more than one switch.

For example, if a person can use three switches the row and column selection is achieved with two of them, and the third is used to confirm a particular choice.

This may still seem slow to the physically able. However, for the severely spastic, Bliss and the Apple open up a world of communication with the outside world.

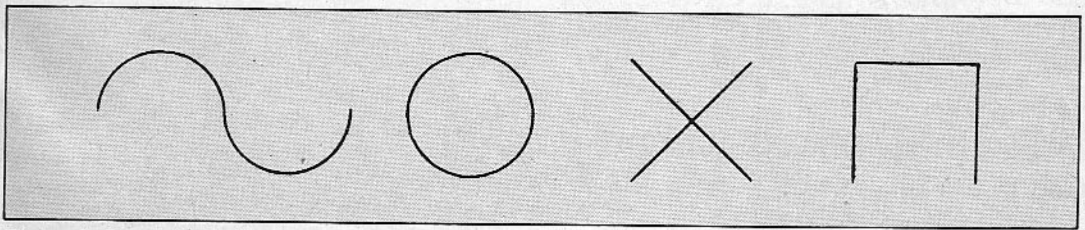


Figure I: Some basic forms

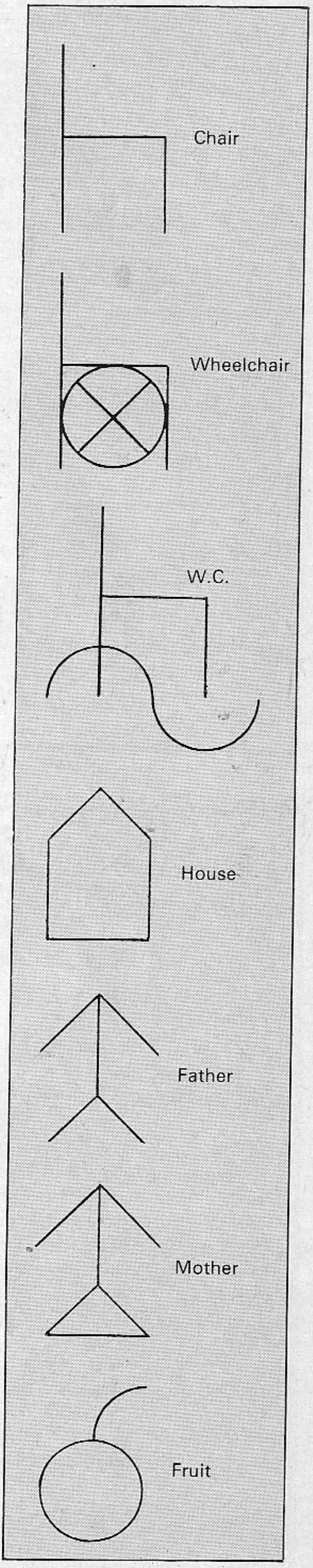


Figure II: Some Bliss symbols