A Database Publication

apple user

Vol. 5 No. 5 May 1985 £1

Making Apples score on the sports field

Join TELECOM GOLD through MicroLink – see centre pages

HOW TO

protect your programs from COPYA
create your own pie charts
make the most of RWTS
binsearch in Forth and Basic
– and how Hungarians discovered the Apple

Your free ticket to appleUG inside

REVIEWS: Workbench • Macputer IIc • Copytext • Omnis 2 • Logo books

apple user
Volume 5
Number 5
May 1985

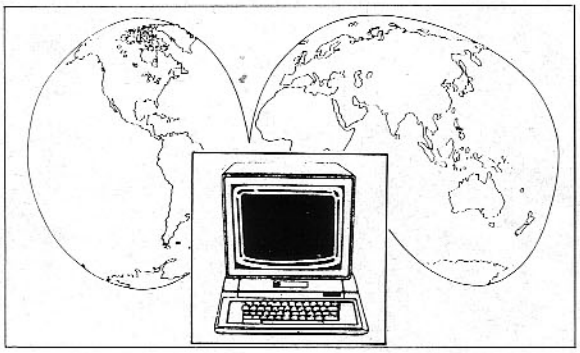
News

Apple '85

Graphics

Sports

Reviews



Introducing MicroLink

● Join MicroLink, the exciting new micro communications system tailored to your needs. Full details of the service and the facilities it offers, can be found in the centre pages.

Macintosh

Programming

Pascal

Protection

Timer

Searching

New products

Books

Application

Feedback

Classifieds

Fun & Games

Index

Order form

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

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

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

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

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.

Subscription rates for 12 issues, post free:
£12 UK
£13 Eire (IR £16)
£20 Europe
£15 Rest of world (surface)
£30 Rest of world (airmail)

European drive to weed out counterfeit Apples

A CAMPAIGN to eliminate counterfeit Apple computers is currently hotting up throughout the EEC.

Coordinated by an elite legal unit based in Paris, the crackdown is aiming to weed out copies of the Apple II range in the shortest possible time.

For the anti-counterfeit squad has received intelligence that it is shortly to be faced with another major problem – a flood of fake Macintoshes.

"That is certainly going to be our next battle", a team member told *Apple User* in an exclusive interview. "And we are preparing for it as of now".

Known as the "Turkey Shooters", the team has achieved significant results throughout Europe during the last 12 months.

"It would be fair to say that we appear to be winning", the spokesman said, "but with an enemy like this you cannot afford to be complacent, for they can hit you anywhere, any time".

In all the company team dealt with several hundred counterfeit cases throughout Europe during 1984.

"As all involved distributors a threat proved sufficient to get them to mend their ways", said the team member.

According to the latest information from the Paris-based organisation the counterfeiters appear to be on the run throughout the UK and Europe.

The current state of play is:

UK: Very little copy Apple activity following a number of warning actions six months ago.

Belgium: The counterfeiters are now keeping their heads down after a major clear out.

France: Le Crunch has come for the fake Apple merchants with the end result that nothing has been seen of them of late.

Holland: The campaign continues with always one or two cases on the boil.

Germany: This remains the only black spot on the European map. However conditions are starting to improve. Following a purge counterfeit sales have been dramatically reduced from the 800 a month being recorded six months ago.

Apart from the activities at grass roots market level, Apple is intensifying its lobbying of governments out in the Far East, the source of almost all "bad" Apples.

"We have already had some convictions in Taiwan", said the Apple spokesman, "but the courts tended to be fairly lenient with their sentences.

"However there is a move by countries like Taiwan to become more respectable. And part of this may soon be reflected in their attitudes to counterfeit manufacturers.

"If this happens it will be a major breakthrough for us, for that means the counterfeiters will be caught in a vice".

Wait for the 65C02

REPORTS that all Apple IIe computers produced since March contain the 65C02 central processor are wrong, a company spokesman told *Apple User.*

However it is understood the company feels it would be logical to upgrade the Apple IIe to benefit from the superior technology of the 65C02 once existing stocks of the 6502 chip are exhausted.

The spokesman would not comment about plans to implement the 65C02 for the IIe other than to say an official statement would be forthcoming.

The 65C02 – standard in the Apple IIc – uses CMOS technology which means it runs on less power than the 6502.

PRESTEL CHAT GOES LIVE

MICRONET has launched a major innovation in interactive viewdata – the first live programme on Prestel to be scheduled on a regular weekly basis.

Celebrity Chatline gives micro owners their first chance ever to interview well known personalities direct from their home computers over the Micronet system.

The service is a development of the highly successful Late Night Chatline which is second only to Micronet itself in the Prestel Top Ten of most popular areas accessed.

Celebrity Chatline is similar to Late Night Chatline's CB-style on-screen chat facility, except that Micronet editor David Babsky travels to the homes of selected celebrities.

As Micronet members electronically send questions on special message frames, the night's celebrity replies on-line straight away via his own home computer.

One of the first guests on Celebrity Chatline was Derek Meakin, managing editor of *Apple User* who commented: "It was gratifying for Database Publications to be chosen to help launch this exciting new development in interactive viewdata.

"This is yet another example of the pioneering spirit behind the Micronet operation and helps to explain why micro users are joining in ever-increasing numbers".

Celebrity Chatline is on Micronet 800 every Wednesday between 7 and 8pm.

Mac gets a Cobol compiler

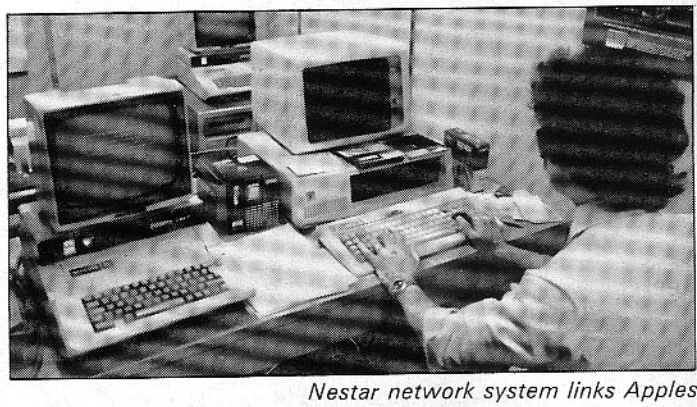
THE first Cobol compiler for the Macintosh has been announced by Micro Focus.

It was welcomed by Apple Computer's development tools product manager Dan Cochran as "a long awaited addition to the Macintosh family of languages which creates enormous potential for the Macintosh Office".

MacCobol allows the Macintosh window and menu capabilities to be built in to Cobol applications because it provides access to the 386 most commonly used Macintosh ROM routines.

Developers using MacCobol can directly access these ROM routines via the Cobol Call verb. Support for the remaining ROM routines will be provided in subsequent releases.

Apples help with £100m budget job



Nestar network system links Apples and IBM PCs at London Transport

APPLE'S are helping work out London Transport's signal and electrical engineer's department's budgets of over £100 million, a complex task involving many variables which until two years ago was done manually.

Computerisation onto London Transport's mainframes was tried, but many difficulties arose such as insufficient allocation of computer time, obtaining access to the machine and downtime over which there was no local control.

So the decision was taken to use networked personal computers – a network was necessary as working on budgets requires several people to have access to a central database of information.

The initial configuration used an early networking system from Nestar called Elf with a 10 mbyte hard disc and three Apples.

As this grew to eight Apples on the system, reaction time slowed and there was little room left on the hard disc.

Having proved the viability of networking, one of the latest systems from Nestar has just been installed. The Plan 4000 has a 137 mbyte disc and all the latest networking features, yet is compatible with the Elf system enabling all existing files and programs to be transferred onto the new system.

Nestar's networking system is compatible with IBM's recently announced PC Network. The Plan series uses Arcnet token passing technology and can have both IBM and Apple personal computers on the network.

So the eight Apples from the Elf and three IBM PCs are currently on the Plan 4000.

Scientific interface

A VERSATILE interface allowing Apple computers to be used in college science and technology control and measurement experiments has been promised by Eagle Scientific.

Production should begin this month on the interface which will cost in the region of £140.

Its analogue inputs will connect to instruments measuring temperature, voltage, pH and other data. Digital inputs are for timing, counting and similar functions.

Cut price modems

THE purchase of Prism Microproducts' communications stocks by Modem House has resulted in bargain-priced telecomputing opportunities for Apple owners.

A complete 300/300 baud colour supporting comms pack, including Modem 2000 and software on disc, is available from Modem House for £99.95 – £20 cheaper than the old Prism price.

Hebrew WP pack developed

GABI Miro, a 38-year-old Israeli software developer, has produced a Hebrew word processor for the Macintosh.

Miro, a physics graduate from Hebrew University of Jerusalem, is a consultant to software developers at Apple's European headquarters in Holland.

The new software is called MouseWrite and externally resembles MacWrite, an advantage because of the ease of use of the word processor.

It is a fully bi-directional and bilingual, left-to-right and right-to-left, program supporting various keyboards with English and Hebrew layouts.

MouseWrite has two Hebrew fonts and uses all standard MacWrite fonts, both proportional and fixed sizes. In the Hebrew mode it has right justification and in the English left justification.

Two of its main features are its ability to use both Hewbrew and English in the same sentence, and its ability to mix English and Hebrew paragraphs in the same document.

Bernard Marks and Associates of Tel Aviv has been appointed distributor for the software, which is expected to sell for $199.

Terminal on your wrist

THE world's smallest Apple compatible computer terminal will be available in the UK towards the end of June.

The Seiko RC-1000 wrist watch module connects to an Apple and once the information has been transferred to the watch it is ready for instant recall on a two line 24 character dot matrix LC display.

The RC-1000 has five operational modes giving the time, a user definable storage area, schedule alarm, weekly alarm and world times.

The memo function enables the user to create telephone directories, price lists, notes and other data.

Colour for the Macintosh

A BRITISH systems house is claiming a world first with its development of colour monitor output for the Macintosh.

The Pixel Artist colour painting system from Micro Core allows the graphics output from the Macintosh to be used for a colour display and interfaced with a video or camera system.

The system enables drawings, paintings and charts to be converted from standard Macintosh application packages into full colour display on a high resolution monitor and to be output, if required, through a video or camera system.

Using a camera system, transparencies and slides can be taken directly from the RGB colour output of the system. An option also exists for the input of video via a digitiser.

The system is based on a 512k Macintosh computer, a graphics controller and high resolution colour monitor.

Total system price is £9,750, including MacPaint, MacDraw and MacChart.

A word from the top...

APPLE computers have been endorsed as teaching aids from the highest level – President Reagan, no less.

The corporation's public relations men were said to have been "driven wild with delight" when the President – if somewhat unwittingly – provided the machines with his official blessing during a recent speech.

"When I was growing up, an apple was something you'd give to the teacher. Now it's something you learn on", Reagan is reported to have said recently.

"It comes to something when you have to be named after a fruit to get the backing of the President", moaned a man from IBM.

Deadly game

MASTER spy James Bond will be making his first appearance on a computer game for Apple users soon.

Software house Domark has secured the worldwide licence rights for the latest 007 adventure "A View to Kill" starring Roger Moore and Grace Jones and scheduled for release in June.

Contents of the game are being kept secret but Domark said it will be centred on four arcade games closely linked to major stunts in the film.

Business graphics on tap

SALESMEN, managers and trainers in industry are using the Apple IIe to originate high resolution business graphics for audio visual presentations.

And thanks to a new software package the resulting 35 mm slides have a resolution of 4,000 lines and a choice of 16 colours from a palette of 64.

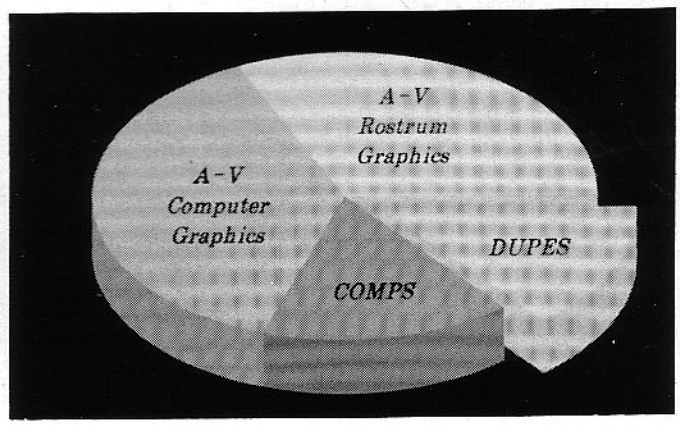
The software – Apple Presenter – was developed by Dicomed UK.

The user inputs his data and the resulting graph, pie chart or organisation chart is plotted and scaled on screen.

No artwork or typesetting is involved. All alterations, updates and colour changes are done electronically.

The graphics, in the form of electronic data, are sent via a modem or on floppy disc to a Dicomed service bureau. The mounted slides are returned in 24 hours.

The service bureau can also enhance the graphics beyond the capabilities of the Apple they were created on – with company logos, special graphic effects and different type styles.



FROM IBM DRIVES TO APPLE

FOLLOWING the launch of Crossdata, which allows transfer of software between different computers without cables or modems, System Constructors has released Apple Turnover, a product that transforms IBM, or clone, disc drives into Apple drives. The package allows users to format discs and transfer software to and from Apple DOS or Apple CP/M machines without wires or modems.

It incorporates a small board and software and runs on most true compatibles.

Once the board is in place the user loads the program, selects either Apple DOS or CP/M from the menu and converts his software by copying it from his Apple disc in the B drive to the MSDOS disc in his A drive – or vice versa.

The board does not affect day to day running of the PC.

Walk easy on an Apple

THE footsore folk of Shropshire are reaping the benefit of high tech developments at the county's district health offices.

An Apple-based system has slashed administration time in the busy chiropody department of Shropshire Health Authority, freeing trained staff to see more patients.

The Omnis management system, produced by Blyth Computers has also saved taxpayers more than £3,500 in expenses as well as the equivalent of a practitioner's salary over the last year.

Chiropody department head, Michael Allard-Williams, said: "We are amazed at the speed of access for data and the reporting facilities of the database.

"Monthly returns now take eight minutes to produce instead of a whole day and annual returns can be completed in 11 minutes instead of three days".

The organisation of staff and collation of their reports has, until now, involved the time equivalent of one senior chiropodist a week.

"This has always concerned me", said Mr Allard-Williams, "so having my own Apple IIe I tried out various database configurations using different languages.

"But I could not find anything that matched the speed and reporting facilities of the Omnis system.

"My actual savings in subsistence and travelling alone have produced a benefit of £3,587 and the indirect saving on salary of one practitioner".

Mr Allard-Williams paid tribute to his local Apple dealer, Cressage Computer Services in Shrewsbury, which had helped in the demonstration and loan of various databases.

High tech goes on show at Apple '85

ALL the latest technological advances in the Apple and Macintosh family will be on display during Apple '85 – the fourth national Apple User exhibition.

The show takes place at the Champagne Suite, Novotel, Hammersmith, London, Thursday May 9 to Saturday May 11.

The contribution from Apple Computers itself will be highlighted by laser technology as MacOffice and AppleTalk network 32 Macs with one laser printer.

Jazz, the integrated software package for the Macintosh from Lotus Development, will be previewed at the show before its May 27 UK launch.

It makes the most of the Macintosh mouse, icons, cut and paste, undo typing and pull-down menus and offers keyboard shorthand for mouse commands.

Jazz lets users look at several documents on the desk top at the same time and enables them to be created and modified to any size.

Its word processor offers all the professional elements of a stand-alone word processor, while the cut and paste facility makes copying and moving text easier than ever before.

Blyth Software marks its move to Saxmundham, Suffolk, with the launch of its Omnis 3 package which brings networking flexibility to the Macintosh for the first time.

The powerful database management program allows all kinds of applications programs to be designed.

It can be used for all data processing applications involving the maintenance of multiple connected files.

The package will be available immediately after its launch at Apple '85.

The wraps come off the hush-hush software project developed jointly by Symbiotic Computer Systems and SAMS – Systems Analysis and Micro Software.

This multi-user accounting package will run on Symbfile and Symbnet, giving a stand-alone or network option.

It is designed for the Apple IIe and IIc and the Macintosh and will be available this summer as a complete business accounting system, at a price yet to be set.

Elite will be demonstrating the latest version of its Format 80 Apple word processor, including Arabic, French and German disc-based versions, and also showing ProDOS hard disc support.

New for the Macintosh is Psion's Chess, winner of the world microcomputer chess championship and the first 3D graphics chess program in the world.

Apple '85 visitors will be able to play one of the 28 levels of the game from novice to grand master. It comes with 50 grand master games on disc.

Dark Star will be exhibiting and demonstrating its set of desk top utilities for the Apple, the Snapshot DeskSet which includes calculator, calendar/appointments diary and small notepad facility, and does for Apple II what Sidekick did for the IBM PC.

Lorne Computers is introducing Macputer Profile – sales invoicing, purchase, nominal stock control, VAT and management accounts on hard disc.

Ideal for the reasonably sized firm and easy to use, this program means the user can scroll through a lot easier than other accounts programs for the Apple because of the windowing facility.

Greengate Productions is offering the latest version of its digital sound sampler sequencer musical add on for the Apple.

New software for this product includes looping of sounds and Midi add on to the DS3 system.

From A.M. Technology come two new products, both named Vicom, for the Apple II and Macintosh.

One runs in colour on the IIc on Hayes protocol auto dial modems and can save 20 Prestel pages on disc with one button push.

The Macintosh version of this combined viewdata/Ascii package is said to be the only one in the world that does everything that MacTerminal does plus viewdata and is usable with all standard UK modems as well as Hayes protocol modems.

A plug-in card which gives the best-selling Nightingale modem from Pace an auto dial/auto answer capability will be launched at the show.

The additional enhancement to the modem for Apple users will go on sale in the summer.

The first live demonstration of the exciting new development in telecomputing Micro-Link will take place at the show.

With its access to the massive Telecom Gold database, it offers Apple users the most comprehensive international communications opportunities available.

Satellite-linked global communications, telex services, mainframe data and text management services, instantaneous electronic mail and a multitude of free software for downloading are just a few of the tremendous new experiences opened up for Apple users.

It's all as pie - or is

I HAVE now decided that "It's as easy as pie" is the most hateful phrase in the entire English language. I hadn't always felt that way of course – but then I hadn't tried to write pie drawing routines before either.

The ubiquitous pie charts are an effective method of displaying information in a concise and easy to understand format. A good graphics library should thus provide the facility to produce them with the minimum of effort to the user.

Unfortunately minimising the effort for the user maximises the effort required to write the routines – not to mention the mind bending trigonometry involved. (I knew looking out of classroom windows instead of paying attention would catch up on me in the end.)

Before setting out to write the routines I asked various people what features they would want when drawing pie charts. This produced the following set of criteria:

☐ The ability to have the pie any size and anywhere on the screen.
☐ The data can be in percentages or, if not, there should be automatic conversion.
☐ The slices should have variable shading options.
☐ The slices should have variable colours.
☐ The pie can be 'exploded' to prevent colours spilling over adjacent slices.
☐ Individual slices can be pulled out by a variable amount.
☐ The slices should have the option of being labelled.
☐ The orientation and shading of the first slice can be chosen if required.
☐ Default values for most of the options should produce a reasonable chart.

The routines provided here meet all of these criteria and they also provide additional control over the resolution used to draw the arcs and shading lines. These last features are particularly important if the routines are to be modified to drive a plotter.

Unlike most of the routines in the library they do not constantly check the HPLOT values against the screen window chosen. Instead a simple 'one off' test is employed to ensure the whole pie is in the screen window.

This sacrifice was made to maintain a reasonable drawing speed. However the individual tests could easily be inserted before each of the five lines containing HPLOT – and the global one removed – if desired.

Following the usual technique employed throughout the series, most of the work is relegated to subroutines which are called by a relatively simple main program. The routines should be typed in at the end of the previous library routines since they make extensive use of the earlier ones in the series.

We can now take a brief look at each of the routines in turn.

Pie slice routine

At the lowest level is the routine at 45200-45710 which draws a single slice anywhere on the screen. This routine would not normally be accessed directly since it requires a lot of computation to determine the information it needs.

Since all the information used by this routine refers to screen values it could however be extracted and used independently of the graphics library, apart from the labelling.

The instructions to the routine are placed in an array ZR( ) before calling it. The array locations have the following meanings:

1. *X coordinate of centre (screen value)*
2. *Y coordinate of centre*
3. *Radius of pie*
4. *Slice size (radians)*
5. *Angle to start drawing the slice (radians)*
6. *Offset from centre*
7. *Shading option 0-4*
   0 open
   1 perpendicular to radia line
   2 parallel to radial line
   3 cross hatch
   4 solid
8. *Resolution steps/circle*
9. *Shading step size*
10. *Labelling switch, 0=off 1=on*
    Label held in ZS$

The slice is drawn in whatever colour is presently selected. The 'solid' option is achieved by very close shading and is a little slow.

The labelling routine makes use of the machine code text routine in the library. If you have the Basic version of this then the GOSUB 43470 in line 45880 should be replaced by GOSUB 42800. The label itself must be in ZS$.

The single slice routine also uses two other small routines at 45800 and 45910 internally.

Pie chart controller

The routine at 46100-46510 is used to draw a complete pie. Its main function is to set up the values required for each slice and then to call the slice drawing routine.

The parameters for the controller routine determine what the pie chart will look like and are set up by the user in the main program. The parameters themselves are stored in an array ZP( ) and have the following meanings:

ZP(0). This determines how the data is to be treated. The data itself must be put in an array ZY( ) with the number of data points in ZN. This is exactly the same format as

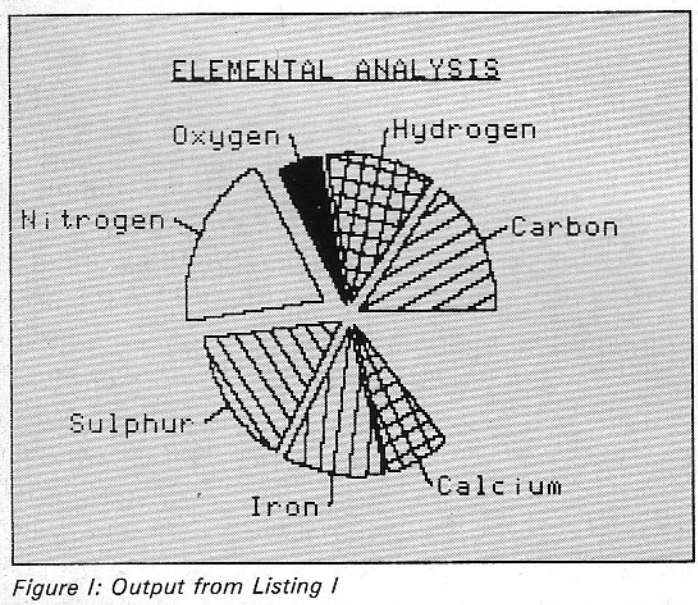


Figure I: Output from Listing I

easy as it?

PETER GORRY turns pie-maker for Part XIII of the Apple User Graphic Library

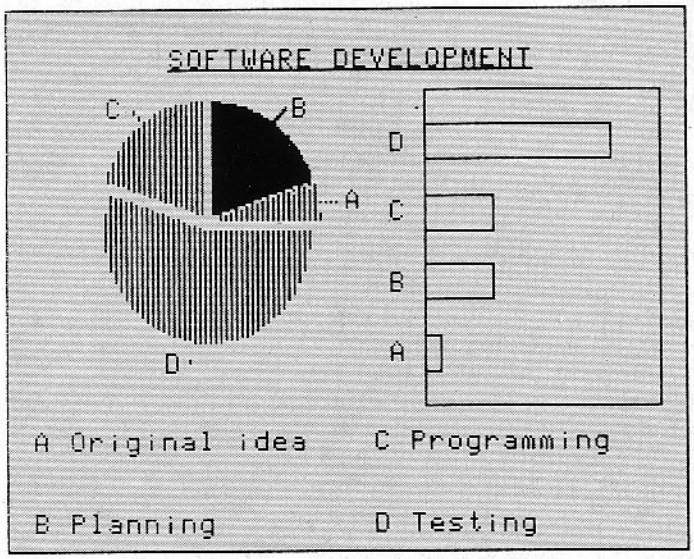


Figure II: Using the pie and histogram routines

used for data plotting and the histogram routines. If ZP(0)=0 then the data is treated as straight percentages – the program checks to see that the total is not greater than 100. Using percentages is useful if you don't have a complete pie and want to leave a gap. If ZP(0)=1 then the program will assume that the data constitutes a complete pie and will calculate the slice sizes accordingly.

ZP(1), ZP(2), ZP(3). These hold the X,Y value of the pie centre and the radius of the pie respectively. In line with all other routines in the library the values are in 'user units' defined by the mapping routines. Since there is no guarantee that the user units are defined in such a way as to ensure that a circle would look like a circle, only the X axis is used to calculate the radius. For instance, if you have defined the X axis to go from 0-25 then a radius value of 5 would produce a pie whose radius was one fifth of the X axis in length.

ZP(4). This is the orientation of the first slice, relative to the horizontal – measured in degrees. This allows one to start the pie at any place on the circle. The default value of zero starts the pie off with the first edge horizontal. The pie is then built up in an anticlockwise direction.

ZP(5). This contains the general offset – as a percentage of the radius. Thus a value of 10 here will offset all slices by 10 per cent from the pie centre. A small value often produces more pleasing pie charts than a value of zero, since there is no colour 'spillage' from neighbouring slices.

ZP(6). This location determines whether any slices are to be pulled out further than the rest. A value of 0 means that no individual offsets are required. A value of 1 means that the program must check each slice for special offsets. If the second option is chosen the offsets must be set up in an array ZO( ) before calling the pie controller. Thus if you want slices 2 and 5 to be pulled out 15 per cent and 20 per cent more than the others you must set ZO(2)=15 and ZO(5)=20.

ZP(7). This switches on automatic cycling of the colours, except for black which is omitted. 0=off, 1=on.

ZP(8). This switches on automatic cycling of the shading options. This can be used individually, or in conjunction with the colour cycling to create even more variation. 0=off, 1=on.

ZP(9). This determines the shading type to be used for the first slice. Values are: 0=open, 1=perpendicular, 2=parallel, 3=cross hatch, 4=solid. If ZP(8)=0 then this shading type is used for all slices. Thus it is possible to draw say, solid slices, with different colours simply by setting the appropriate values for ZP(7)-ZP(9).

ZP(10). This location switches on automatic labelling of the slices, 0=off, 1=on. If this option is selected you must store the label for each slice in an array ZP$( ) before calling the pie controller. The labelling is carried out using the library text routines and no changes are made to the settings – so if you choose inverse-underline mode the slices will be labelled in that way.

The main program

Lines 110-180 should be fairly standard by now and are used to set up the screen size, values, colour and title. Lines 190 and 200 determine the number of slices, the pie size and its position. Line 210 provides the size of each slice (%), and line 220 gives a name to each slice.

Once the data has been set up lines 230-300 determine how the pie should be drawn and line 310 draws it. A great variety of pies can be produced simply by changing the values used in the ZP array – it's as simple as that.

Graphics listing

```
100  REM
EXAMPLE PROGRAM

110  GOSUB 42400: REM  SHAPE TABL
     E LOADER
120 ZC = 3:ZP = 1:ZF = 1: GOSUB 4
     0000: REM  SET PAGE
130 ZM(1) = 0:ZM(2) = 10:ZM(3) =
     0:ZM(4) = 10: REM  USER VALU
     ES
140 ZM(5) = 0:ZM(6) = 260:ZM(7) =
     191:ZM(8) = 0: REM  SCREEN V
     ALUES
150  GOSUB 40200: REM  SET MAPPI
     NGS
160 ZG(1) = 0:ZG(2) = 0: GOSUB 40
     400: REM  SET BORDER
170  REM  SET UP PIE
180 ZS$ = "ELEMENTAL ANALYSIS":ZX
     = 2.5:ZY = 9.0:ZS(6) =  - 1
     : GOSUB 43470:ZS(6) = 0
190 ZN = 7: REM  SEVEN SLICES
200 ZP(1) = 5:ZP(2) = 4.5:ZP(3) =
     2.0: REM  RADIUS 2.0,  X=5,
     Y=4.5
210 ZY(1) = 15:ZY(2) = 13:ZY(3) =
     5:ZY(4) = 19:ZY(5) = 15:ZY(6
     ) = 12:ZY(7) = 7: REM  SLICE
     S IN %
220 ZP$(1) = "Carbon":ZP$(2) = "H
     ydrogen":ZP$(3) = "Oxygen":Z
     P$(4) = "Nitrogen":ZP$(5) =
     "Sulphur":ZP$(6) = "Iron":ZP
     $(7) = "Calcium"
230 ZP(10) = 1: REM  LABEL GRAPH
240 ZP(0) = 1: REM  PERCENTAGES
250 ZP(5) = 8: REM  8% GENERAL OF
     FSET
260 ZP(7) = 0: REM  CONSTANT COLO
     UR
```

```
270 ZP(8) = 1: REM  CYCLE SHADING

280 ZP(6) = 1: REM  SPECIAL OFFSE
    T
290 ZO(4) = 15: REM  EXTRA OFFSET
     FOR SLICE 4
300 ZP(9) = 2: REM  SHADING TYPE
310  GOSUB 46100: REM  PIE CONTRO
    LLER
320  END :

45200  REM

PIE SLICE ROUTINE

45210  REM  ALL INFO IN SCREEN VA
    LUES
45220  REM  AND HELD IN ARRAY ZR
45230  REM  ZR(1) X COORDINATE OF
     CENTRE
45240  REM     2  Y COORDINATE
45250  REM     3  PIE RADIUS
45260  REM     4  SEGMENT SIZE (R
    ADIANS)
45270  REM     5  START ANGLE (RA
    DIANS)
45280  REM     6  SHIFT FROM CENT
    RE
45290  REM     7  SHADING OPTION
    0-4
45300  REM     8  RESOLUTION STEP
    S/CIRCLE
45310  REM     9  SHADING STEP SI
    ZE
45320  REM     10 0=NO LABEL, 1=
    LABEL ZS$ HOLDS LABEL
45330 Z6 = 4 *  ATN (1): REM  PI
45340 Z7 = (ZR(4) * ZR(8)) / (2 *
    Z6):Z7 =  INT (Z7): IF Z7 =
    0 THEN Z7 = 1
45350 Z8 =  COS (ZR(4) / Z7):Z9 =
     SIN (ZR(4) / Z7)
45360 Z5 = ZR(4) / 2 + ZR(5): REM
     SLICE CENTRE LINE
45370 Z3 = ZR(1) + ZR(6) *  COS (
    Z5):Z4 = ZR(2) - ZR(6) *
    SIN(Z5): REM  SLICE CENTRE
45380 Z1 = ZR(3) *  COS (ZR(5)):Z
    2 =  - ZR(3) *  SIN (ZR(5))
45390  REM  DRAW SLICE
45400  HPLOT Z3,Z4 TO Z1 + Z3,Z2
    + Z4
45410  FOR ZK = 1 TO Z7
45420 Z0 = Z1 * Z8 + Z2 * Z9:Z2 =
    Z2 * Z8 - Z1 * Z9:Z1 = Z0
45430  HPLOT  TO Z1 + Z3,Z2 + Z4:
     NEXT
45440  HPLOT  TO Z3,Z4: REM  CLOS
    E SLICE
45450  REM  NOW SHADE SLICE
45460 Z3 = ZR(4) / 2:Z4 = ZR(3) *
    ZR(3):Z0 = 1.0E5
45470  IF Z3 <  > Z6 / 2 AND Z3 <
    > Z6 * 1.5 THEN Z0 =  TAN (
    Z3): REM  GRADIENT
45480 Z8 =  COS (Z3 + ZR(5)):Z9 =
     SIN (Z3 + ZR(5))
45490  ON ZR(7) GOTO 45510,45610,
    45510,45500: GOTO 45690
45500 ZR(0) = ZR(9):ZR(9) = 0.5:
    GOTO 45610: REM  SOLID
45510  REM  PERPENDICULAR STRIPES
45520  IF ZR(4) < Z6 THEN Z7 = ZR
    (9):ZZ = ZR(3) *  COS (Z3)
45530  IF ZR(4) >  = Z6 THEN Z7 =
    - R *  COS (Z6 - Z3):ZZ = -
    0.001
45540  FOR ZK = Z7 TO ZR(3) STEP
    ZR(9)
45550 Z1 = ZK:ZU = ZK: IF ZK > 0
    THEN 45580
45560 Z2 = Z0 * ZK:ZV =  SQR (Z4
    - ZK * ZK): GOSUB 45910
45570 Z2 =  - Z2:ZV =  - ZV:
    GOSUB 45910: GOTO 45600
45580 Z2 = Z0 * ZK: IF ZK > ZZ
    THEN Z2 =  SQR (Z4 - ZK * ZK)
45590 ZV =  - Z2: GOSUB 45910
45600  NEXT : IF ZR(7) = 1 THEN 4
    5690
45610  REM  PARALLEL STRIPES
45620 ZZ = ZR(3) *  SIN (Z3)
45630 Z7 = ZR(3): IF ZR(4) < Z6
    THEN Z7 = ZZ
45640  FOR ZK = 0 TO Z7 STEP ZR(9
    )
45650 Z2 = ZK:ZV = ZK:ZU =  SQR (
    Z4 - ZK * ZK):Z1 =  - ZU
45660  IF ZK < ZZ THEN Z1 = ZK /
    Z0
45670  GOSUB 45910:Z2 =  - Z2:ZV
    = - ZV: GOSUB 45910
45680  NEXT : IF ZR(7) = 4 THEN Z
    R(9) = ZR(0)
45690  IF ZR(10) = 0 THEN  RETURN

45700  GOSUB 45800
45710  RETURN :

45800  REM

LABEL SLICE

45810  REM  STRING IN ZS$, X,Y SE
    T UP IN RP ARRAY
45820 Z1 = ZR(3):Z2 = 0:ZU = ZR(3
    ) * 1.2:ZV = 0
45830  GOSUB 45910
45840  IF Z8 < 0 THEN XC = XC -
    LEN(ZS$) * 7
45850  IF Z8 >  = 0 THEN XC = XC
    +5
45860 ZX =  FN UXCN(XC + ZR(1)):Z
    Y =  FN UYCN(YC + ZR(2))
45870 Z2 = ZS(3):ZS(3) = 1: REM
    NO BLANKING
45880 Z1 = ZI: GOSUB 43470:ZI = Z
    1: REM  PLOT STRING
45890 ZS(3) = Z2: REM  RESTORE BL
    ANKING
45900  RETURN :

45910  REM

TRANSFORM AND PLOT

45920 Z1 = Z1 + ZR(6):ZU = ZU + Z
    R(6)
45930 XC = Z1 * Z8 + Z2 * Z9:YC =
    Z2 * Z8 - Z1 * Z9
45940  HPLOT XC + ZR(1),YC + ZR(2
    )
45950 XC = ZU * Z8 + ZV * Z9:YC =
    ZV * Z8 - ZU * Z9
45960  HPLOT  TO XC + ZR(1),YC +
    ZR(2)
45970 Z1 = Z1 - ZR(6):ZU = ZU - Z
    R(6)
45980  RETURN :

46100  REM

PIE CHART CONTROLLER

46110  REM  ALL INFO PASSED VIA Z
    P ARRAY
46120  REM  ZP(0)  0=CONVERT, 1=A
    S GIVEN
46130  REM     1   X COORDINATE O
    F PIE CENTRE (USER UNITS)
46140  REM     2   Y COORDINATE
46150  REM     3   RADIUS (USER U
    NITS - X AXIS)
46160  REM     4   START ORIENTAT
    ION (DEGREES)
46170  REM     5   GENERAL OFFSET
     (%)
46180  REM     6   0= NO SPECIAL
    OFFSETS, 1 = OFFSETS IN ZO()
     ARRAY
46190  REM     7   0=SINGLE COLOU
    R, 1=STEP COLOURS
46200  REM     8   0=SINGLE SHADI
    NG OPTION, 1=STEP SHADING
46210  REM     9   SHADING TYPE/S
    TART 0-4
46220  REM     10  0=NO LABLES, 1
    =LABELS
46230 ZQ = 0: FOR ZI = 1 TO ZN:ZQ
    = ZQ + ZY(ZI): NEXT
46240  IF ZQ > 100 AND ZP(0) = 1
    THEN PRINT "ERROR - GREATER
    THAN 100%": RETURN
46250 ZR(3) =  FN XCN(ZP(3)) -  F
    XCN(0): REM  RADIUS IN SCREE
    N COORDINATES
46260 ZR(3) =  INT (ZR(3))
46270 ZR(1) =  FN XCN(ZP(1)):ZR(2
    ) =  FN YCN(ZP(2))
46280 ZR(0) = ZR(3) * (100 + ZP(5
    ) + 20 * (ZP(10) = 1)) / 100
    : REM  RADIUS + OFFSET + LAB
    EL
46290  IF ((ZR(1) + ZR(0)) > ZM(6
    )) OR ((ZR(1) - ZR(0)) < ZM(
    5)) THEN  PRINT "NOT ON SCRE
    EN": RETURN
46300  IF ((ZR(2) + ZR(0)) > ZM(7
    )) OR ((ZR(2) - ZR(0)) < ZM(
    8)) THEN  PRINT "NOT ON SCRE
    EN": RETURN
46310  REM  DRAW PIE
46320  IF ZR(8) = 0 THEN ZR(8) =
    100: REM  STEPS PER CIRCLE
46330  IF ZR(9) = 0 THEN ZR(9) =
    6
46340 ZR(5) = ZP(4) * 0.0174532:
    REM START ORIENTATION -
    RADIANS
46350  IF ZP(0) = 1 THEN ZQ = 100
    : REM  %
46360  FOR ZI = 1 TO ZN:
46370 ZR(4) = ZY(ZI) * 6.2831853
    ZQ: REM  CONVERT TO RADIANS
46380 ZR(6) = (ZP(5) + (ZP(6) = 1
    ) * ZO(ZI)) * ZR(3) / 100:
    REM OFFSET
46390 ZR(7) = ZP(9): REM  SHADE O
    PTION
46400 ZR(10) = ZP(10): IF ZP(10) =
    0 THEN 46420
46410 ZS$ = ZP$(ZI): REM  SLICE L
    ABEL
46420  GOSUB 45200: REM  DRAW SLI
    CE
46430 ZR(5) = ZR(5) + ZR(4): REM
     NEXT SLICE ORIENTATION
46440  IF ZP(8) = 0 GOTO 46460
46450 ZP(9) = ZP(9) + 1: IF ZP(9)
     = 5 THEN ZP(9) = 0
46460  IF ZP(7) = 0 THEN  GOTO 46
    300
46470 ZP(7) =  INT ( PEEK (228) /
    32): REM  HCOLOR
46480 ZP(7) = (ZP(7) + 1 + (ZP(7)
     = 3) - 7 * (ZP(7) = 7)) * (
    ZP(7) > 0)
46490  HCOLOR= ZP(7):ZP(7) = 1
46500  NEXT
46510  RETURN :
```

WORKBENCH
son of
DOS TOOLKIT
By MALCOLM WHAPSHOTT

I WILL assume in this review you are familiar with the old DOS Toolkit and only mention the changes that have been made in order to turn it into the Workbench.

The package is supplied as two discs, one for Applesoft and one for assembly language, and a manual of over 300 pages containing tutorials on the programs and documentation for them. More about the manual later.

The Applesoft disc contains a slightly improved programmer's aid and a faster scrolling hi-res character generator, but the main change is the addition of the "Boston window", an editor for Applesoft programs.

Typing Ctrl-E to activate the program causes the screen to clear and the current options of the assembler to be displayed in inverse on the bottom line.

Default conditions are auto line numbering, insert mode and window open, and the start of the program to be displayed in its own format. That is, each line number consists of five digits including leading zeros, and a full stop is placed on the right hand edge of the screen to indicate the end of the line. This is because the program puts a "window" around any line you are currently editing.

The commands allow you to edit, insert or delete a line and there are also commands to search the program globally as well as replace characters.

The assembler disc contains not only a new version of the editor/assembler but also Apple's debugger, Bugbyter. Both of these programs seem similar to those described in the April 1984 issue of *Apple User* under the review of ProDOS except they run under DOS 3.3, of course.

The new assembler is co-resident with the editor on machines with more than 48k and is considerably faster than the old version in loading, saving and especially assembling programs.

Any program written with the old assembler will load and assemble on this version, the exception being very large blocks of source code, as the work space is about 2k smaller and so these will not load.

Some added features are the ability to have two blocks of source code in memory at the same time and to switch between them. Macros can be included, and a program that resides entirely in memory can be assembled straight to disc without the usual need to save it. It is also possible to INCLUDE files instead of having to CHaiN them.

A feature I like is the ability *not* to generate any output so I can quickly find out if there are missed-out labels, etc, although it is intended for only listing to the printer.

A feature I dislike is the BUFFER FULL message when I try to assemble a program and I have not typed NEW to clear it first. We are told that this is so you remember to save the source code first, but this happens just after I have typed SAVE!

An example of the sort of increase in speed you can expect by switching to the Workbench is that 8334 lines of source code – the assembler now counts the lines – which used to take some 23 minutes to be assembled now only takes five. These are approximate timings from the date and time stamps that my version of DOS puts on the files, but indicate the sort of speed increase you can expect.

Some of the editor is still written in Sweet16, making commands like FIND extremely slow compared to other editor/assemblers, although the manual says Sweet16 has been re-written for speed. It is also possible to include Sweet16 op-codes in your source file.

The assembler has only crashed once on me when it tried to assemble a corrupted line. It went into the monitor but pressing Reset put me back into the editor with a message saying at which line the error had occurred. The line was corrected and the program was assembled correctly.

Bugbyter is the debugger program supplied with the assembler. It is a 6.7k program that allows you to trace and modify a machine language program by setting break points and there is a built in mini-assembler.

I have not used Bugbyter as I find debugging programs takes a lot of time to set up properly and the time saving is not that great. Also I do not always have the memory to spare for it.

However if you are writing small machine code programs that are called from Basic this may be just the program you are looking for.

Unfortunately the manual is designed to fit in a three ring binder which Apple expect you to buy yourself. This may be easily found in America, but very hard to come by in this country, so they sell an official binder.

However it is not easy to get even one of these either. I had to phone Apple to get the name of a dealer! Even then, it got damaged in the post so I am still waiting for an undamaged binder which has to be flown in from the States to Apple, sent to the dealer (who shall remain anonymous for now) and then to me.

What neither the assembler or Bugbyter will do is recognise the new 65C02 instructions, although they can be made to execute them.

Product: *DOS Workbench Programmers ToolKit*
Price: *£60 excluding binder*

appletip

When you are writing programs in a more or less structured way you can start your subroutines at line 1000,2000,3000 etc.

It will happen that you have to renumber these subroutines once in a while. This is necessary if you want to insert lines or when you want to have a nice listing with all the lines numbers at intervals of 10, for example.

Now you can do this renumbering job by hand with the RENUMBER program. But why don't you use your computer for it?

To do the job make sure that RENUMBER is active then execute a textfile which has the following contents:

```
&F1000,S1000,E1999
&F2000,S2000,E2999
&F3000,S3000,E3999
```

and so on, as long as you have subroutine entries.

You could make this file with a word processor. Save the textfile on your working disc, for instance under the name CLEANUP.

Sometimes it will happen that the error message "No lines in range" hits the screen. This is not fatal, the RENUMBER program was only looking for lines which aren't there.

It is always nice if you know when a program has ended its task. You can accomplish that by making the last line in CLEANUP something like PRINT "OK make it a mess again".

Martin Keesen

IF you are one of the many parents who get roped in to help with school sports day, this suite of programs by PETER POPHAM is just what you need.

The programs are too long to list, but readers who would like copies should send a blank disc and return postage to *Apple User*.

Give your Apple a sporting chance

DURING the summer term of 1980, I was involved as one of the recorders in my school's annual sports day. It was after a particularly hectic part of the afternoon that I decided that there had to be a more efficient way to record the results and also keep the scores.

The more that I thought about it, the more I was convinced that my Apple computer could solve both problems. I therefore set about writing a suite of programs that would be easy to use – even for someone not familiar with computers. Sports Meeting84 is the result. My sports days are now far more enjoyable and relaxed.

The suite of programs is designed to maintain the results of events and scores for a sports meeting with four year groups and a maximum of 26 events. It also enables the printing out of individual year groups or the complete results at the end of the meeting.

System requirements: Apple II, II+ or IIe with 48k RAM, Applesoft in ROM or in language card, one disc drive, DOS 3.3 and, if a print out is required, a printer. The binary file CHAIN *must* be on the disc.

The first program, 7k long, is used to set up the name, date of the sports day, the number of teams, the team names and the number of points for the first place in each event. It also sets up the files necessary to store both the above information and the event results and scores as they occur during the meeting.

The second program is the main one. It is 15k long and is accessed from the first program. Menu driven, it contains all the routines necessary to enter, display and correct events as well as those to display and correct the cumulative scores.

The third program is 7k long and contains the print routines necessary to produce the hard copies. It can be accessed from the main program or loaded and run independently. The routines are designed for a Seikosha GP100 printer but the program is easy to adapt for other printers.

The program makes extensive use of the GET function and one key responses with no need to press Return. The main exception to this is when entering data and pressing Return is required to complete entry.

On running the program, SPORTS MEETING84, the user is asked "Is this the first use of the program for this meeting? (Y/N)". The answer to this question is very important, because it determines whether or not new files are set up.

The answer N is used for restarting after a shutdown for lunch or some other reason. Also use N in the unlikely event of a power cut losing the program.

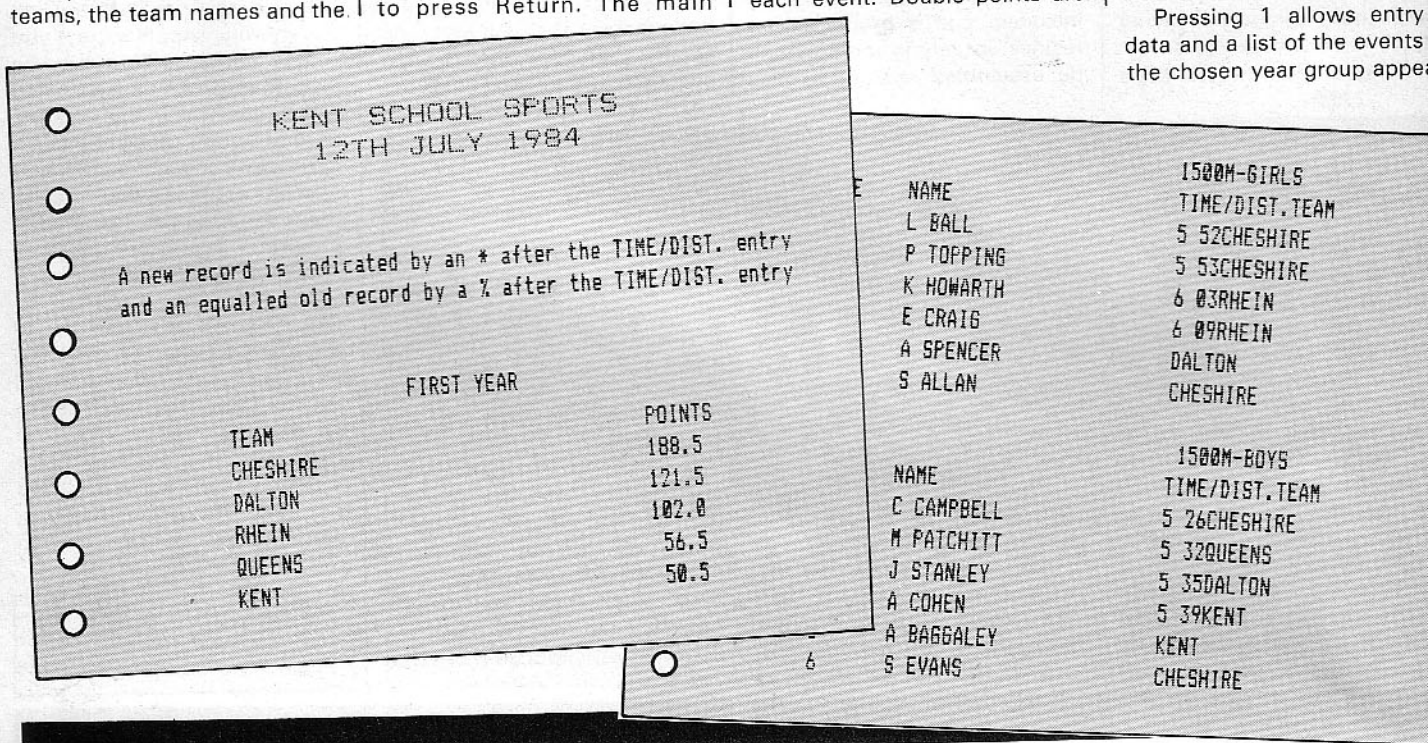
Entry of the title of the meeting and the date are the next steps. These are both completed by pressing Return.

The final stage of the initialisation is the entry of the number of teams, the team names and the number of points to be allotted to the first place in each event. Double points are automatically awarded to rela races. As these entries are s important, the user is asked confirm them and may chan any incorrect ones.

When all is correct, the da are stored in files called YEA and CODES and the year fil are set up. On completion, suggested method of indicati old and new records is give Pressing Return loads and ru the second program. CHAIN used so that all the necessa variables and data are saved use in the second program. It also used to reduce the size the second program.

The second program, call SPORTS.MAIN84, after loadi and running, asks the user enter 1, 2, 3 or 4 for the y group that is required. pressing one of those numb the full menu appears. Exce when entering data, pressi Esc returns the user to menu.

Pressing 1 allows entry data and a list of the events the chosen year group appe



Entering the event number and pressing Return allows the program to continue. On completion of entry the data as entered is displayed – the scores are automatically amended.

The entry of the team name controls the scoring and a check is made to ensure accuracy of spelling. If it is incorrect or the name is not one of those entered in the set up routines, the user is prompted to re-enter correctly.

The program caters for the entry of two competitors who are placed equal and the scores are adjusted automatically. The team name is again checked to ensure that the equals sign, used with the team name, has been entered correctly.

An event that has already been entered is noticed by the program and the user is informed – the results displayed – finally the program returns to the menu.

Menu options 2 to 5 operate in a similar way to option 1, the entry routine, and the user is prompted for entries when necessary with instructions given when they are needed.

Option 6, the print out routines for the final results, should only be used at the end of the meeting or at some other time convenient to the user.

Re-running SPORTS MEETING84 after a shut down requires the response N to the question "Is this the first use of the program for this meeting? (Y/N)". It also requires the code letter for the meeting to be entered when asked.

The program checks to see if there are files with that code letter and, if so, loads all the variables and data files. When this has been completed, the user is prompted to press Return after which the SPORTS.MAIN84 loads and runs. The entry of data may now continue as before.

All the files are sequential. Data are saved to the file using the append and write commands to reduce waiting time to a minimum.

When correcting data, the original file is renamed TEMP, and a new file is created, after which the corrected data with the other data are stored in it sequentially. Finally TEMP is deleted. This method is used for the other files that are updated.

Printing specifics are:

SPORTS.MAIN84

If the printout of the current team positions and scores is used and problems are experienced with the positioning, adjustments may have to be made with the TAB values in lines 5130, 5145, 5155, 5165, 5175.

The non-display of events excluded from particular year groups is controlled by lines 1105 to 1120.

SPORTS.PRINT84

Lines 320 to 350 control the printing of the title and date in enlarged text by using CHR$(14) to switch on and CHR$(15) to switch off the system.

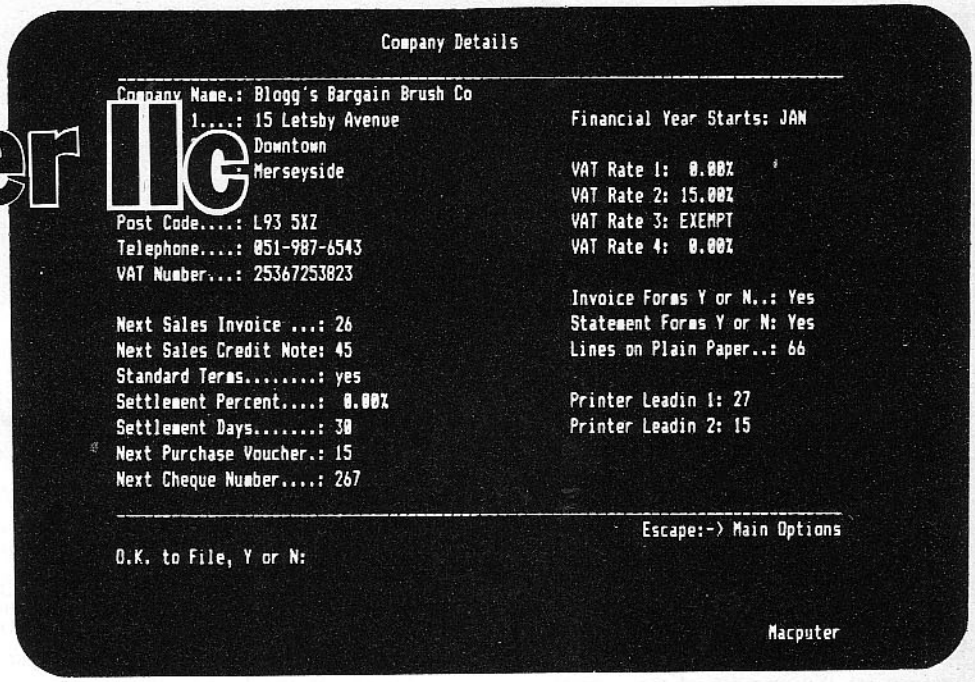
In lines 5265, 5330, 5430, 5530, 5630, the POKE 36,X function is used in conjunction with TAB to ensure the correct positioning of the text on the 80 column Seikosha printer.

If problems are experienced with the positioning of the TEAM and POINTS text, adjustment of the TAB values may be necessary in lines 5130, 5145, 5155, 5165, 5175, 5265, 5330, 5430, 5530, 5630.

The non-printing of events excluded from particular year groups is controlled by lines 5205 to 5230. The present program excludes event 23, JAVELIN-GIRLS, from the first and second year events and event 21, ROUNDERS BALL-GIRLS, from the third and fourth year events.

Macputer IIc

Easily followed accounting package for small businesses



By ROBBIE McLAREN

MACPUTER IIc is an accounting package for the Apple IIc from Lorne Computer Services. Like the IIc, Macputer IIc is aimed at small businesses. To run it you need an Apple IIc with an external disc drive.

The packaging and manual are attractive and are similar to the style used by Apple. The manual is well printed and I found it easy to follow. Again, the style of the manual is very Apple-like.

After initial chapters introducing the package the manual divides into chapters covering sales accounts, purchase accounts, cash accounts, etc. The manual makes good use of screen pictures, examples and plain words. Full marks for both the manual and packaging.

The integrated package is fairly simple to follow. It is controlled from a network of menus. The main screen asks you to choose between the main sections. You are then presented with further menus until you are at the desired point such as entering an invoice. The divisions between sections are logical and consistent, so it's quite easy to find your way around.

The first main option is the sales ledger. As well as keeping records of all your customers, including their current turnover and balance owed, you can record details of all your sales transactions. You can enter invoices, credit notes or direct credits into your account. Macputer uses forms on the screen to ask you for information. Filling in these forms is straightforward, you type in the details and use the arrow keys to move around the screen.

As well as recording sales invoices, Macputer can be used to prepare the invoices. On the screen you see a copy of the invoice which you can fill in. All the totals and VAT calculations are carried out automatically.

Once the invoice is complete you can print it out on your hard copy printer. When you receive payments, Macputer helps you allocate the money to unpaid invoices or credit notes. It will show you all the outstanding invoices or credit notes for the customer.

You can then say how much of the payment should be allocated to each invoice or credit note.

The sales ledger allows you to print a wide variety of sales reports, including a list of your debtors together with their telephone numbers!

One feature I found very helpful was connected with code numbers. If, like me, you can never remember codes you have created for customers, suppliers, nominal accounts and so on, Macputer has the answer.

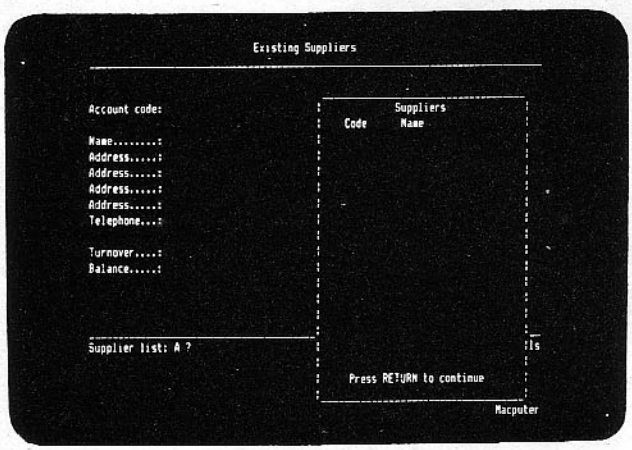
Whenever you are asked for a code you can call up a window listing all your codes. For instance, suppose you were filling in a purchase invoice and needed to give the supplier's code. You simply press a couple of keys and up pops a window showing all the suppliers' names together with their codes.

The purchase ledger is similar to the sales ledger. It keeps records of all suppliers and purchase transactions. Purchase invoices, credit notes and direct debits from your account can all be entered. Macputer handles settlement discount on both purchase and sales invoices.

There are facilities to help you decide which suppliers to pay. You can ask for a list of creditors due by any date. You can go on to display lists of outstanding invoices and credit notes for each creditor. It's then up to you whether you pay all you owe, or pay only specific invoices.

As with the sales ledger, a wide range of reports can be printed, including supplier ledgers.

The cash ledger records cash sales, purchases and cash deposits and withdrawals from your bank account. It's intended



for those parts of a business which operate on a cash rather than credit basis.

Throughout the package heavy emphasis is laid on managing accounts as well as recording them. In order to allow you to monitor and control your accounts it provides a facility to set up nominal accounts. You may know these as analysis codes. They are simply different categories of income and expenditure. For instance, postage and fuel bills might be recorded as part of a nominal account called expenses. Each nominal account can be given targets or budgets for each month of the year. As the year progresses you can compare your actual performance against these budgets.

Another useful management account facility is the ability to examine your cash flow position at any time. You can also prepare for the coming of the VATman by automatically completing your VAT return. Payments to or from Customs and Excise can also be recorded.

A number of other facilities cover the various housekeeping activities such as backing up data discs, procedures at the end of a year, and so on.

Obviously to run on a machine such as the IIc, even with the external disc drive, there must be limits on the amount of information you can store.

You can record details of up to 250 suppliers or customers. You can have up to 1,000 transactions (sales invoices, credit notes, and so on) outstanding at a time. Your income and expenditure can be analysed into 20 sales and 40 purchase accounts.

So Macputer IIc can cope with the demands of a small business, but larger companies or businesses with very complex accounts would be better with a system with more capacity for storing information.

Incidentally, Lorne have now launched Macputer Profile, a hard drive version which increases the maximum number of transactions to 32,000. However, this version is only available for a 128k Apple IIe, Apple III and costs £395.

Macputer is very easy to learn. The standard of the software and manual is high. At £195 it is good value for money and should be attractive to many small businesses.

Product: Macputer IIc
Price: £195
Requirements: Apple IIc with external drive, or IIe with 128k and two drives.
Distributor: Lorne Computers, Oban, Argyll, Scotland. Tel: 0631 65635.

MacReview

It's horses for courses

• Omnis 2 for pure power
• PFS duet spells economy

CHRIS BURRIDGE concludes his comparative review of two databases

FOLLOWING last month's look at PFS:File/Report, I'll now concentrate on Omnis 2.

The Omnis 2 manual is sensibly laid out. It begins with a short interactive tutorial of a typical client/order database which I found *does*, as intended, give an essential overview of the program's structure.

Following chapters logically describe step by step the building of a database and all facilities. Text is to the point and well illustrated with screen dumps, but on occasions the sheer multitude and complexity of the options is hard to digest.

More examples of typical applications would be helpful, although overall the manual is a model for such a powerful program.

Omnis 2 differs from PFS:File in that form designs are maintained in a Library File which must be available to work with the applicable data. This library concept is an inherent part of the program operation, but does facilitate a wide range of field attributes, for example entry of numerical, dated or character data only – default settings are also supported.

Once my Library File was established, setting up my personnel test database proved no more difficult than before, although report layouts take longer.

Labels and fields can be designed in free format in up to 80 columns wherever the mouse is pointed on the blank screen. Editing can use the usual cut and paste. This is certainly necessary to re-position labels, as infuriatingly I could find no way to insert blank spaces, as on PFS.

Labels can occupy up to 12 screen pages. The individual fields are then created by double-clicking in the appropriate position which displays the field description screen – see Figure I.

Field lengths must be specified, but the comprehensive mix of attributes and descriptions selectable illustrate the many idiot proofing possibilities, so that data must conform to your input wishes or not be accepted.

Entering actual data is similar to PFS:File, but with defaults making life easier. Option Boxes in the right margin allow more entries or find the previous or any record. A special timesaver is that any record can be re-presented for editing as a subsequent similar record.

Pleasingly, the Mac Notepad and Scrapbook can be shrunk and always ready for use while using the full Omnis 2 menus. This is not possible with PFS.

To retrieve records clicking the Find box quickly locates records by any of up to 10 indexed fields. This is fast, as indexed fields are automatically maintained in sorted order.

Exact matches are possible, but one practical frustration is that Omnis 2 distinguishes between upper and lower case (unlike PFS), although this is overcome by selecting the option "Upper case only".

A built-in multiple update/delete is an authoritative tool – you could, for example, increase all stock price records by 10 per cent.

Search facilities are really impressive and include calculated searches. Easy to set up requirements may be defined on up to a mind-boggling 50 lines using the screen shown in Figure II. Formats may b saved/modified and printed, ar the logic functions incorporate should satisfy the most dema ding user.

Setting up a format is simil to creating a library file, that is must be redesigned fro scratch by means of a tree ty menu structure.

This is much slower tha PFS:Report, which exploits t original page design but do not enjoy the same wealth facilities.

Layout of labels and fields designed on a blank screen in enormous total area of 24 characters across by 240 lin down. Up to 13 self-contain sections must be separated special symbols – Head Detail, Total, etc.

After customising fields t comprehensive Parameters R Report screen shown in Figu III is used to set constants the current report. Unfo tunately no automatic meth of aligning decimal poin calculating averages or ev supplying total "box-lines" supplied as with PFS, althou these difficulties can overcome.

For example, I puzzled some time how to produce average salary as I did with P The Omnis manual did not h but I eventually solved problem by dividing the total salary field by a temporary fi #R holding the number records read.

It is easy to set up mail labels, 1 to 3 or more across page – either different

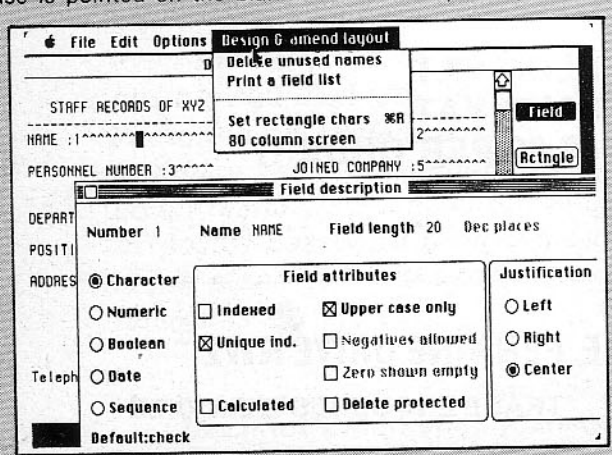


Figure I: Omnis 2 field description screen offers many field protections

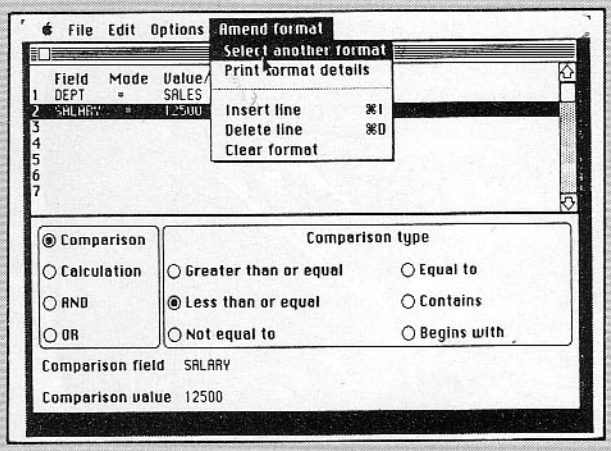


Figure II: You can define up to 50 searches with Omnis 2's search format screen

repeated. A sorting option, faster if disc space available, allows a choice of ascending or descending order in a maximum of nine assorted levels within a particular database.

The mail merge facility not available with the PFS package is clearly the pièce de résistance. Standard letters can be formatted as above and printed using information selected from a database, utilising the usual search and sort choices.

You could therefore decide to mail every female employee under 35 who has worked in the sales department for say two years!

At the printing stage alternatives are presented to send the output to screen or disc as well as hard copy. One minor inconsistency noticed when screen printing – the menu options should be dimmed, as they are unavailable.

Overall I liked the superior flexibility of Omnis reporting, but feel that the setting up procedures do not reflect the user friendliness of Mac or the rest of the package.

The manual section on the layout structure symbols, for example, is over-concise, particularly on calculations, and challenging to follow.

Omnis 2 provides other attractive features, including full reorganisation/reindexation of data. These are vital routine procedures if some data has already been entered – and fields are subsequently added or modified at the layout stage. Unfortunately the manual does not give this important warning in the design report chapter.

On several disconcerting occasions data typed into newly created fields just blanked out again. The secret was eventually found near the end – a salutary lesson to read all of the manual!.

An interesting function is the capability to produce DIF text files – short for Data Interchange File and well known to Apple II users of programs like VisiCalc.

This allows the passing of raw data to and from Omnis 2 or 3 databases, and even to MacWrite for editing, which I found works well. Unlike PFS's built-in ability to divide and merge files, Omnis 2 relies on DIF to do a similar job but in a more cumbersome way.

Again, any records inserted must conform exactly to the receiving fields, but with the added complexity of special field values.

Other utilities manage and verify library/data files, display field names and can change current Mac date. Password protection is also offered with a separate screen, see Figure IV, permitting the setting of specific options to which other users have access. The manual ends with Appendices covering error messages, data security and index.

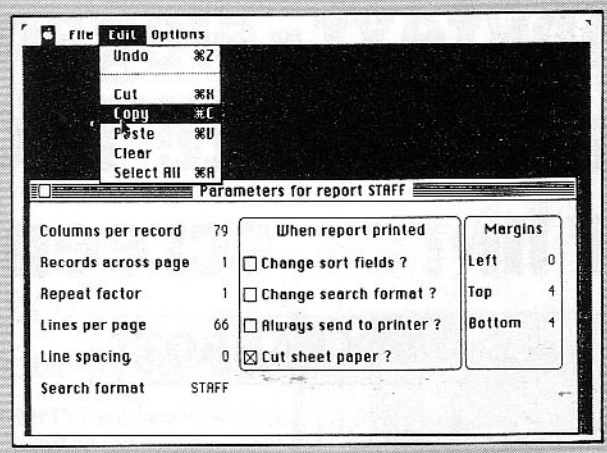


Figure III: Flexible Omnis 2 print options – here is my "staff" report selection

Conclusions

Taking an overall view based on my practical experiences working with my test personnel database on the two programs, there is no clear winner. It is a question of horses for courses.

Omnis 2 undoubtedly wins hands down on pure power, flexibility and scope of the facilities. If you require all the high level options, especially mail merge, high level searching, data input and password protection, there is no contest – Omnis 2 is for you. I found Omnis 2 a "tight" program – it is difficult to do anything other than the program intended.

The PFS duet however is some £135 cheaper, simpler in use and has sufficient facilities for many people.

In many ways using PFS is akin to using the excellent QuickFile on a IIe – it's not the best but it is good value.

The manual is certainly the more friendly of the two – witness the superbly clear error-message explanations.

Indeed PFS:File on its own may suit if you can tolerate the lack of on-screen sorting and limited printer formats.

Adding PFS:Report provides a fully competent database capable of coping with most small businesses and home needs.

Certainly producing totalled reports complete with box-lines is simpler than with Omnis. Against this is the inconvenience of two separate discs.

For extra sophistication demanded by larger businesses Omnis 2 will fit the bill. The reassurance for bosses where employees will be users is intrinsic in the program design – individual fields can be set up to guard against improper data entry, plus password protection to allow access only to specific options.

In such situations Omnis 2 rules OK!

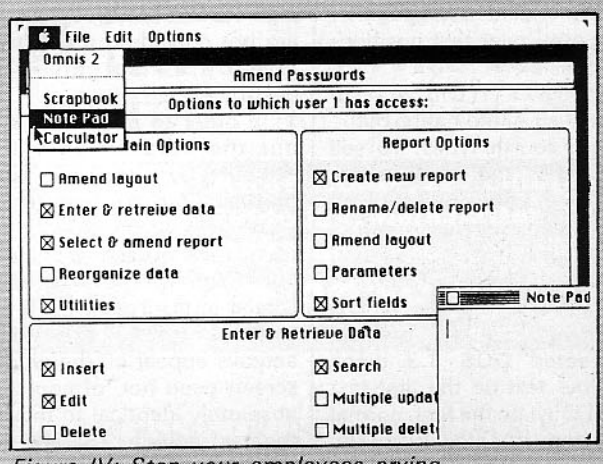
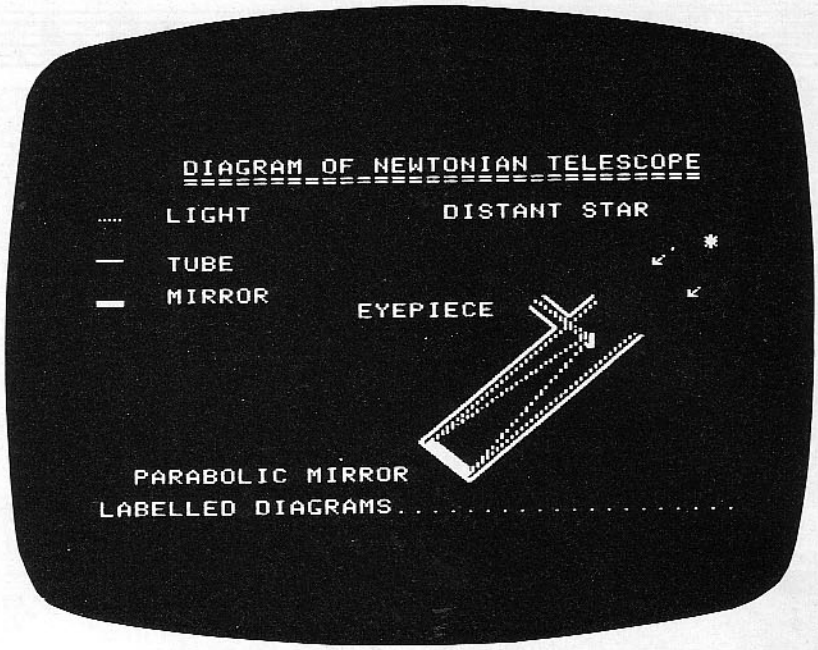


Figure IV: Stop your employees prying with Omnis 2 password protection



Copytext-elegant way to write text on hi-res screens

By MAX PARROTT

THERE are four ways of writing text on to the high resolution screens of the Apple II family. The first is to use the HPLOT TO command to form the letters. The second is to use shape tables to DRAW the letters where required.

The third is to use a high resolution character generator utility which essentially turns the high resolution screen into a normal text screen and the fourth is another kind of character generator which puts text at a given position on the screen when you invoke it.

The first of these methods is too lengthy, clumsy and laborious to be considered further. The second has a disadvantage in that the shape table has to be designed and implemented and it is slow to operate. However it does have the advantage that the characters can be placed anywhere on the screen.

The third is very easy to use – just position text on the normal text screen as you want and if the character generator is in operation the text is reflected on the high resolution screen, almost as if it were the normal text screen.

This is possibly the best known kind of character generator because many people have seen either Applevision – a program on the DOS 3.3 system master which uses such a generator – or they know the character generator distributed as part of the DOS Toolkit. This kind suffers from not being able to scroll the high resolution screen and not automatically clearing it.

The fourth is slightly harder to use since you have to move an invisible cursor (well that's one way of imagining the process) to the required start position on the high resolution screen and then invoke the character generator to write your text. The advantage is that finer control over text positioning is possible.

Copytext from Altim Software is a combination of the third and fourth types. To sell well against the competition, such a product has to work well and be cheap. Copytext answers both charges. It costs less than £10 and is very easy to use.

It is simple to use: BRUN COPYTEXT, (a file on the unprotected DOS 3.3 disc), write your text on the first text page (it must be the first, normal page) and transfer it to the current high resolution screen by printing Ctrl-F. This acts in exactly the same way as Ctrl-D does when invoking disc commands, thus it may be printed from the keyboard directly or by issuing the PRINT CHR$(6) command.

The command may be given under program control or in the direct mode. Any text on the text screen is then copied to the corresponding position on the high resolution screen. Spaces are not copied so that artwork already on screen is not destroyed.

In order to make text stand out the area which will be occupied by a character is first cleared before writing it. Characters which appear on the text screen as flashing or inverse will appear on the graphics screen in inverse.

The characters which actually appear on the graphics screen need not, of course, be absolutely identical to those on the text screen. There are 64 possible characters in Copytext which, in the version on disc, are identical to the upper case characters which can be displayed by a standard Apple II or II+. That is, the characters with Ascii codes $20 to $5F (32 to 95).

The disc also contains an editing program which will take the characters in the file Copytext, allow you to view and edit them, and then save them back to disc. It always saves the file as Copytext, so to achieve different fonts for different occasions you have to do some nifty work, changing either discs or file names.

However the editing program is easy to use. The standard keyboard characters are displayed on the right of the screen and the matrix (five wide by seven deep) of dots or pixels which you can change are displayed on the left.

The cursor is moved around by the familiar I,J,K,M keys and pixels are set by pressing 1 and reset by pressing 0.

Pressing Return quits th mode, first drawing the chara ter more clearly in the middle the screen. You then choose accept or discard it.

There are four more files the disc. The first of these, th Hello program, asks if you wa to run the editor program or th second file which demonstrat the capabilities of Copytext. Th third file is a backup of Copyte and the fourth is another versi of it, with a slightly differe font, used by the demonstrati program.

When BRUN COPYTEXT issued as a command it must the first statement in a progra because it loads at the top memory under DOS a protects itself by settin HIMEM: at $9351.

Commands such as MA FILES:4 (or greater) and FP w therefore destroy Copytext. position means that it cannot used with Integer Basic a because of this and the way interfaces with DOS you will unable to use it with ProDos

The other major drawback that you cannot easily put low case text on to the graphi screen. Of course it can be do at the expense of other chara ters, by editing.

Product: Copytext.
Price: £9.95.
Distributor: Altim Software 10 Madeley Street, Tunstal Stoke-on-Trent, Staffs ST 5AT.

Dodge DOS with RWTS

THIS article on the Apple's versatile RWTS routine by MICHAEL KING mentions three programs. The first, OOPS, restores a file which has just been deleted, the second tells you the length and loading address of a binary file and the third, PRETTY PRINT, provides a formatted listing of a Basic program held on disc.

Readers who would like all three programs should send a blank disc and return postage to *Apple User*.

THE RWTS (Read or Write Track and Sector) routine is documented in the DOS manual (pages 94-98 in my edition), but it starts off with a warning to the reader to skip it if you are not familiar with machine language.

This is unduly pessimistic as you need to know nothing about machine code to use the routine, and it is a very useful tool because it allows you to read and write data to/from the disc directly without going through DOS.

The RWTS routine has four parts:

- The calling routine itself, nine bytes long.
- The device characteristic table or DCT which tells the routine the physical characteristics of the disc drive you are using, that it is an Apple Disk II. It is four bytes long.
- Another table, the input/output control block or IOB which tells the routine about the actual disc you are using and what you want to do with it, read or write; which slot the disc is in, which bit of the disc you want, etc. (17 bytes).
- An area in the computer's memory – a buffer – where the data from the disc will be stored or from where it will be copied to disc by the routine (256 bytes).

These parts are entirely separate and can be put into any parts of the computer memory you like. However in order to ensure that they are not destroyed or corrupted by the program, they must be in protected parts of memory. The simplest way to do this is to put all the parts at the top of memory, then lower the ceiling of memory available to Basic to below them. The RWTS parts are POKEd into memory then protected by HIMEM:.

Let's now look at the four parts:

The Calling Routine

The nine bytes of this are:

first byte	– always 169
second byte	– HI-BYTE of the IOB address
third byte	– always 160
fourth byte	– LO-BYTE of the IOB address
5th to 9th	– always 32,217,3,96,0

The HI-BYTE and LO-BYTE of the IOB address tell the routine where to find the IOB in memory, and are worked out like this:

HI-BYTE = ADDRESS/256 (integer part only)

LO-BYTE = ADDRESS – (256 * HI-BYTE)

So if you have put the IOB in memory starting at 38027, then

HI-BYTE = 38027 / 256
= 148.54296
= 148 (integer part only)

LO-BYTE = 38027 – (256*148)
= 38027 – 37888
= 139

The Device Characteristic Table (DCT)

For Disk II the 4 bytes are always 0,1,239,216

The IOB Table

The format of the IOB table is shown in Figure I. It occupies 17 bytes.

The buffer

This is simply a copy in memory of that part of the disc.

To use RWTS you must first set aside some part of the Apple memory. The top of free memory is given by:

PEEK(115) + (256* PEEK(116))

so you can give yourself 400 bytes of reserved memory by:

```
OLDHI = PEEK(115) +
(256 * PEEK(116))
NHI = OLDHI – 400
HIMEM: NHI
```

Now you can load up the various parts of RWTS into memory from NHI upwards.

The three programs all start in essentially the same way, setting up the RWTS routines. The relative locations used for the various sections are shown in Figure II.

The programs then work out the address of the IOB, the DCT and buffer and then POKE these values into the tables. They then POKE the disc to be used in to NHI+12 and 0 (=INITIALISE) into NHI+22.

CALL NHI then calls RWTS to do the required function.

All the programs then read the disc directory, which they do by reading all the sectors of track 17 in succession. Track 17 is where the directory is held in Disc II systems, and it starts in sector 15 and ends in sector 0.

Each call to RWTS results in one sector-worth of data being transferred to the buffer. The program then looks through the buffer for what it wants. You can find details of the layout of the directory data on pages 129-131 of the DOS manual.

OOPS then uses that fact that the DELETE command doesn't actually remove data from the disc but simply marks the directory entry. It does this by putting the value 255 as the first byte of the directory block (and transferring what was there to byte 32).

An active file's directory entry, and the equivalent deleted entry, is shown in Figure III.

So provided you have not tried to save anything since the DELETE, you can restore the original file by putting back the moved value into the first byte.

Apple DOS may destroy the directory entry of a DELETEd file when it does a SAVE (or CLOSE after WRITE for TEXT files), so you can only be *sure* that OOPS will work if you have only just deleted the file. OOPS does actually list ALL the files on the disc. Those which are deleted but not destroyed appear in inverse video.

I find OOPS useful when either I have deleted the wrong version of a program under development, or when a program which DELETEs TEXT files before re-writing then suddenly crashes.

The other two programs show how to use RWTS to read the actual data of the files. To do so, RWTS must find the file on the disc, which means reading the track/sector list (TSL). The TSL is the list of where the file is and is kept separate from the directory entry on the disc. Figure IV gives an example of how Apple DOS organises its file management.

The data

Binary files start with four bytes which give the length of the file (bytes 3,4) and where in Apple memory it will be put by BLOAD (bytes 1,2). All the remaining bytes of data on disc are the same as will be put in memory by BLOAD. The program WHERE IS BINARY FILE reads these four bytes and prints their values on the screen.

TEXT files have no introductory bytes – they are simply the Ascii values of each byte of data. DOS recognises a value of 0 as the end of the file.

Basic is the most complicated. The first two bytes give the length of the file: subsequent bytes are the program, arranged as shown in Figure V for each line of Basic.

L1 and L2 are the "link" bytes which are used by the computer to keep track of where the lines are in memory (so it knows where the next line starts and where to return after a GOSUB). If they are both 0, it is the end of the program.

N1 and N2 are the program line number (LO-BYTE, HI-BYTE).

P1,P2....Pn are the Ascii and token values of the actual

Byte	Value and function
1	Always 1.
2	Defines the disc SLOT and has the value of 16 * SLOT NO (so for discs in slot 6, you must put 60 here).
3	Disc DRIVE (value 1 or 2).
4	Disc identifying number (or put 0 to ignore).
5	Track to be used (0 to 34).
6	Sector within that track (0 to 15).
7,8	Address of the DCT: byte 7 is LO-BYTE; byte 8 is HI-BYTE.
9,10	Address of the buffer: byte 9 is LO-BYTE; byte 10 is HI-BYTE.
11,12	Always 0.
13	Command code, telling RWTS what to do. 0: Initialise for RWTS. 1: Read the disc. 2: Write to the disc. 4: Format the disc, deleting everything on it.
14	Error code. The usual ones are: 16: Write protected disc. 32: Volume mismatch (the disc ident no (byte 4) did not match the one on the disc. 64: Other error. 128: Read error – probably nothing on the disc sector.
15	After RWTS the disc ident number actually found is put here.
16,17	These contain the slot and drive last used by RWTS (put any valid numbers (60,01) here).

Figure I: The IOB Table

The CALLING section starts at memory location	NHI
The IOB	NHI + 10
The DCT	NHI + 27
The BUFFER	NHI + 37
So the variables are at:	
Address of IOB: Hi-BYTE is at	NHI + 1
LO-BYTE is at	NHI + 3
Address of DCT: LO-BYTE is at	NHI + 16
HI-BYTE is at	NHI + 17
Address of buffer: LO-BYTE is at	NHI + 18
HI-BYTE is at	NHI + 19
Disc to be used is at	NHI + 12
READ or WRITE command is at	NHI + 22
Track to be used is at	NHI + 14
Sector to be used is at	NHI + 15

Figure II: Memory locations used

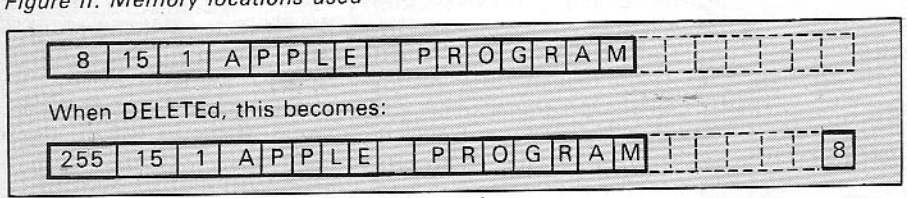


Figure III: Active and deleted directory entries

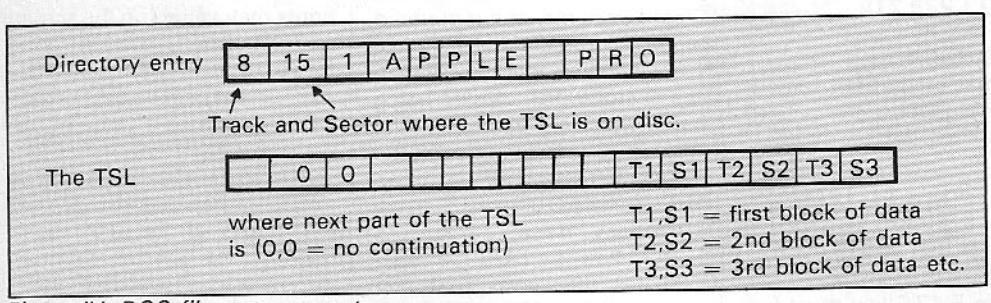


Figure IV: DOS file management

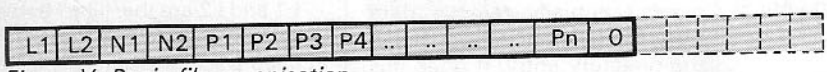


Figure V: Basic file organisation

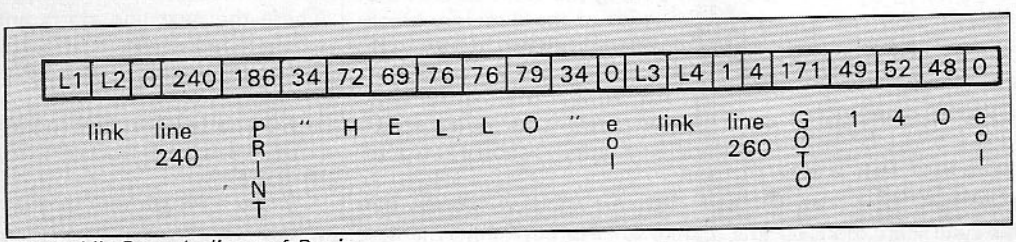


Figure VI: Sample lines of Basic

program. These are listed in the Applesoft manual.

0 marks the end of the line of BASIC code.

So the lines of BASIC

```
240 PRINT "HELLO"
260 GOTO 140
```

would appear on disc as shown in Figure VI.

The Pretty Print program uses these features. After RWTS has found the file in the directory it reads the TSL. It then uses the TSL data to read each of the blocks of program data from the disc.

Ignoring the first two bytes, the program steps through the data translating each of the non-link bytes into "English" by looking up its value in a table. It prints out these until it finds a link value of zero, but starts a new line whenever it finds a colon in the text or when the line is about to become too long.

It also counts the lines being output and inserts a few extra blank ones to allow you to skip over paper perforations.

I hope that these three programs show the potential of the RWTS routine and have given readers (whether 'expert' or not) enough information to allow them to adapt it to their own applications.

Because RWTS allows you to read and write to a disc without using DOS, it can be used to read non-Applesoft-DOS discs on the Apple. In a later article, I shall show how RWTS can be used to read files created under CP/M and Apple Pascal.

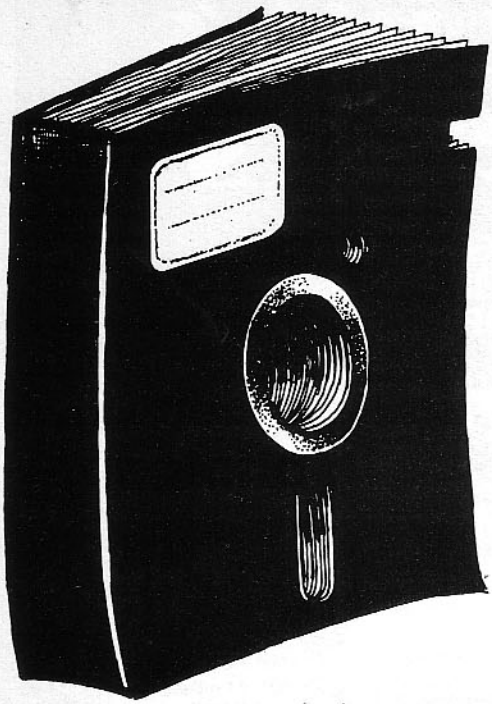
NOW that you've read Michael King's explanation of the RWTS, no doubt you're itching to unleash its power on your own

discs. While you're waiting for Michael's programs to arrive, why not type in D. POIRIER's Verify routine.

The program uses the RWTS to read every sector of a disc, displaying a dot if no errors are signalled and an asterisk if it is unable to read.

10	POKE the IOB table into memory at $300.
15	POKE Machine code subroutine to set up table address and call RWTS. Put in memory at $320.
20	Input slot.
30	Input drive.
35-38	Set up display.
40-50	POKE slot and drive into table.
65	POKE track and sector into table.
80-100	Display result.

```
1  HOME
2  PRINT "  DISK SECTOR
   VERIFY": PRINT "
   ==================="
10 BA = 768: POKE BA,1: POKE
   BA + 3,0: POKE BA + 6,17:
   POKE BA + 7,3: POKE BA +
   8,0: POKE BA + 9,32: POKE
   BA + 10,0: POKE BA + 11,0:
   POKE BA + 12,1: POKE BA +
   14,0: POKE BA + 17,0: POKE
   BA + 18,1: POKE BA +
   19,239: POKE BA + 20,216
15  POKE 800,169: POKE 801,3:
   POKE 802,160: POKE 803,0:
   POKE 804,32: POKE 805,217:
   POKE 806,3: POKE 807,96
20  VTAB 5: PRINT " SLOT :";:
   INVERSE : PRINT "6": NORMAL
   : GET A$:S =  VAL (A$): IF
   S = 0 THEN S = 6
21  VTAB 5: HTAB 8: PRINT S
30  VTAB 7: PRINT "DRIVE :";:
   INVERSE : PRINT "1": NORMAL
   : GET A$:D =  VAL (A$): IF
   D = 0 THEN D = 1
31  VTAB 7: HTAB 8: PRINT D
34  PRINT : PRINT "INSERT
   DISK. PRESS <SPACE> TO
   CONTINUE": GET A$
35  HOME
36  VTAB 3: PRINT "
   1111111111111111222": PRINT
   "
   0123456789ABCDEF0123456789AB
   CDEF012"
37  FOR L = 0 TO 9: VTAB L +
   5: HTAB 2: PRINT L: NEXT :
   PRINT " A": PRINT " B":
   PRINT " C": PRINT " D":
   PRINT " E": PRINT " F"
38  VTAB 2: HTAB 18: PRINT
   "TRACK": VTAB 10: PRINT
   "S": PRINT "E": PRINT "C":
   PRINT "T": PRINT "O": PRINT
   "R"
40  POKE BA + 1,16 * S: POKE
   BA + 2,D
50  POKE BA + 15,16 * S: POKE
   BA + 16,D
60  FOR T = 0 TO 34: FOR SE =
   0 TO 15
65  POKE BA + 4,T: POKE BA +
   5,SE
70  CALL 800
80 E =  PEEK (BA + 13): HTAB
   + 3: VTAB SE + 5:T$ = "."
90  IF E = 64 THEN T$ = "*"
100  PRINT T$: NEXT : NEXT
```



How to Krunch volumes with a Pascal directory

GORDON FINDLAY shares the secret

SOME time ago in writing a fairly large program intended to be used by computer novices, I needed to get a directory of files on a disc from within a Pascal program.

Once I had worked out how to do so, I was asked by several people to explain what I had done, and it may be that this information is of use to others.

The procedures presented in the listing allow the programmer to capture a directory into an array for later processing and to delete a file from the disc.

Armed with the directory it is relatively simple to recover deleted files, to K)runch a volume, to check that there is sufficient space on a disc before writing a file, M)ake a file, or perform virtually any of the Filer tasks from within a program, if necessary totally transparently to the user.

This is obviously most desirable – why should a user need to master the intricacies of the UCSD p-system just to find out what he called his data file last time?

First let me describe, in simplified form, the directory structure of a "blocked volume". The primary reference for this is an IAC (International Apple Core) Technical Note which gives the structure as a succession of Pascal declarations. This I have supplemented by capturing and analysing directories from discs with a known history of file creation and deletion.

A directory may be treated as an ARRAY[0..77] of entries. The zero-th element of the array is volume information while the other 77 entries each refer to a separate file. Thus there is a maximum of 77 files per disc.

Each directory entry is a PACKED RECORD, with a variant tag field. First let me give a brief commentary on the fields – referring to the record structure given in the program listing as TYPE DIRENTRY.

The first two entries locate the file on the disc: DFIRSTBLK is the first block occupied by the file, DLASTBLK is the block following the last used block. This will either be the first block of a free space or the first block of the next file in the directo
As files must occupy contiguo
blocks these two integers su
fice to locate all of a file.

The rest of the directory ent
depends on the type of t
file. The first case, labell
SECUREDIR, UNTYPED in t
declaration, refers in App
Pascal only to the first,
volume entry. DVID is t
volume name, DEOVBL
apparently is a time – it's hard
say with the Apple having
clock.

DNUMFILES is the numb
of active files in the directo
This is most important, and
will return to it later.

DLOADTIME is another tim
and DLASTBOOT is the mo
recent date set for this volum

appletip

With the IIc in mind

Here are three Apple tips which will only work with the IIc.

Sixteen colour double low-res graphics (80 x 40 instead of 40 x 40) can be achieved by typing **PR# 3, POKE 49246,0** and GR.

To test it, try typing **COLOR=15:HLIN 0,79 AT 0:VLIN 0,39 AT 79**, or type in this Applesoft program and RUN it:

```
10 PRINT CHR$(4);"PR# 3"
20 POKE 49246,0: GR
30 FOR I = 0 TO 79 STEP 16:FOR J=0
   TO 15: COLOR=J: VLIN 0,39
   AT J+I:NEXT:NEXT
```

Display Mousetext

To display the IIc's special graphics character set called MouseText by Apple, try this Applesoft program:

```
10 PRINT CHR$(4)"PR# 3":PRINT
CHR$(17):REM Set to 40 columns
20 PRINT CHR$(27):REM Activate
   Mouse Text
30 FOR I = 64 TO 95 : INVERSE:
   PRINT CHR$(I);: NORMAL:PRINT
   " ";:NEXT
```

The 40/80 switch beside the Reset key can be read with a PRINT PEEK (–16288). If the value printed is greater than 127 then the switch is on (set at 40 columns).

To test try this program and while it is running play with the switch and watch the value change. Press Ctrl-C or Ctrl-Reset to stop.

```
10 HOME
20 VTAB 1:HTAB 1: PRINT "PEEK
   (–16288) = ";PEEK (–16288);" "
30 GOTO 20
```

Wes Reimer

Files, which may be of many different types as you can see not all supported in the Apple, have a "status", which is used to indicate whether this file matches a "wildcard" name given by the filer.

DTID is the file name including its suffix, DLASTBYTE is the number of bytes used in the last block of the file – essentially used to locate the actual end of file. DACCESS is the date the file was last modified, and is the date displayed by an extended directory listing.

You will notice features of the declaration which are not supported by the Apple, including two "filler" fields, which merely waste space. These reflect the adoption of the UCSD system without specific hardware changes for compatibility with other machines.

Given the appropriate declarations, and a VARiable DD of type DIRECTORY, how are we to capture the directory from disc? The directory starts in block two of each disc, and the UNITREAD procedure allows us to read any block(s) we require.

The other tool at our disposal is the SIZEOF function. This returns as an integer value the number of bytes occupied by a variable given as its argument. We can use this to specify the number of bytes to be read from the disc – try to work it out from the declaration if you like, but it isn't easy!

Thus the procedure call UNITREAD(4, DD,SIZEOF(DD),2) reads the appropriate number of bytes, from unit 4 (change this if you need to), into the array DD, starting at block 2. With the directory entries now in the array, ordinary string and record manipulations may be used to extract the required information. The volume name, for example, is given as DD[0].DVID and the number of active files is DD[0].DNUMFILES.

The program listings are in the form of a unit DIRSTUFF, which I have made INTRINSIC so it may be linked into SYSTEM.LIBRARY, and used in any programs which may need it.

The public part, the interface, declares three types: FILENAME, FILEARRAY and VOLUME. The first procedure, GETNAMES(VOL, NUMBEROFFILES, NAMES) reads the file names only into the array NAMES from the volume VOL, and gives the number of files in NUMBEROFFILES (what else!).

The method is simple – capture the directory as I've described above, and fish out the file names one at a time.

DELFILE is a little more difficult. This deletes a file from a given volume – DELFILE (SYSTEM.SYNTAX, 4) would delete the file given from the volume in the first disc drive.

Deleting a file is a little unusual. What is done is not simply the removal of the directory entry involved. Rather the entry for the file to be removed is transferred to the end of the list, and the number of active files decreased by one.

This means that PROCEDURE DELFILE must first find the entry to be deleted, then save it (in the variable DUMMY in the listing), move the rest of the directory down one element in the array, put the DUMMY entry at the end, then decrement the number of active files. The directory is then returned to the disc with the UNITWRITE function.

Also in the unit listed is a Boolean function ISFILE(NAME,VOLUME) which is TRUE if file NAME exists on the volume VOL. This is done without reference to the directory at all, to show the use of IORESULT. The procedure attempts to RESET the file. An I/O error means the file wasn't there.

This procedure requires an untyped file to be declared. Unfortunately private file declarations don't work in a unit, so the declaration VAR f:file must be in the public part. This necessitates a DATA segment for the unit.

The listing here is an excerpt from the full unit I have inserted in SYSTEM.LIBRARY and the file declaration is needed elsewhere. You might like to rewrite the function making use of the directory directly, rather than through IORESULT.

This information should allow Pascal programmers to make full use of the disc directory, and make is possible for users to remain completely ignorant of the operating system.

A word of warning. If you are going to type these declarations in and use them, please keep backups, and use an expendable disc for debugging.

One little typing error could corrupt a directory and ruin a disc forever.

> **"This makes it possible for users to remain completely ignorant of the operating system"**

```
{$S+}
{Swapping must be on to compile a unit rather than a program.}

UNIT dirstuff; INTRINSIC CODE 27 DATA 26;

{The 26, 27 are segment numbers in SYSTEM.LIBRARY - Adjust them if you
 have already added routines to the library in these segments.}

INTERFACE {- The public information, which can be used by calling programs.}

  TYPE
    filename = STRING[15];
    filearray= PACKED ARRAY[0..77] OF filename;
    volume   = 4..12;

  VAR f:file;

  {See the text about the reason for this being a public rather than private
   declaration, and about the need for a DATA segment.}

  PROCEDURE get_names (vol:volume; VAR number_of_files:INTEGER;
                                   VAR names:filearray);

  FUNCTION is_file (name:filename; vol:volume):BOOLEAN;

  PROCEDURE del_file (g:filename; vol:volume);


IMPLEMENTATION {- the private details, which are hidden from calling programs.
                  Programs using the unit may not use TYPE or VAR declarations
                  which appear only in the implementation part.}

  TYPE
    daterec = PACKED RECORD
                month : 0..12;
                day   : 0..31;
                year  : 0..100;
              END;

    filekind = (untyped, xdsk, code, text, info, data, graf, foto, securedir);

    direntry = PACKED RECORD
                 dfirstblk : integer;
                 dlastblk  : integer;
```

Directory Handler program listing

```
    CASE dfkind : filekind OF

      securedir, untyped : (filler1 : 0..2048;
                            dvid : STRING[7];
                            devoblk:INTEGER;
                            dnumfiles:0..77;
                            dloadtime:INTEGER;
                            dlastboot:daterec );

      xdsk, code, text, info, data, graf, foto :
                           (filler  : 0..1024;
                            status  : BOOLEAN;
                            dtid    : STRING[15];
                            dlastbyte: 1..512;
                            daccess : daterec)
    END;  (direntry)

    directory = ARRAY[0..77] OF direntry;

  { Note: the procedures declared in the interface part above are now
    defined.  Their parameter lists (if any) were given in their declarations
    and so cannot be repeated, except as comments, as is done here for
    clarity.)

  FUNCTION is_file ((name:filename; vol:volume):BOOLEAN);

    (.. returns TRUE if file "name" exists on volume vol.  vol must be
        in [4,5,11,12])

    VAR g:STRING;
        i:INTEGER;

    BEGIN
      IF NOT (vol IN [4,5,11,12])
        OR (LENGTH(name)<1)
        THEN BEGIN
             is_file:=FALSE;
             EXIT(is_file)
        END;

        {Add prefix to filename: }

      CASE vol OF
        4:g:=CONCAT('#4:',name);
        5:g:=CONCAT('#5:',name);
       11:g:=CONCAT('#11:',name);
       12:g:=CONCAT('#12:',name);
      END;  {CASE}

      {$I-}
      RESET(f,g);
      i:=IORESULT;
      IF i=0 THEN CLOSE(f,lock);
      {$I+}
      is_file := i =0
    END;


  PROCEDURE del_file; { (g:filename; vol:volume)}

    {deletes file g from volume vol...
     vol must be 4,5,11,12.
     g is a file name without volume prefix }

    VAR
      i,j,number_of_files:INTEGER;
      dd :directory;
      dummy:direntry;
      found:BOOLEAN;

    BEGIN
      { First check if deletion is sensible }
      IF NOT (vol in [4,5,11,12])
        OR (LENGTH(g) < 1)
        OR NOT (is_file(g,vol))
          THEN EXIT(del_file);

      { Read directory }
      UNITREAD(vol,dd,SIZEOF(dd),2);
      number_of_files:=dd[0].dnumfiles;

      { Find appropriate entry }
      i:=0;
      found:=FALSE;
      WHILE NOT found DO
        BEGIN
          WITH dd[i] DO
          IF NOT (dfkind in [securedir,untyped])
            AND (dtid = g)
            THEN found := TRUE;
          i:=i+1;
          IF i =78 then EXIT(del_file)  (It shouldn't happen)
        END;

      i:=i-1;

      {Save entry found}

      dummy:=dd[i];

      {Move rest along}

      FOR j:=i TO number_of_files DO
        dd[j]:=dd[j+1];

      {Put saved entry back in list}

      dd[number_of_files + 1]:=dummy;

      {Update number of "active" files}
     dd[0].dnumfiles:=number_of_files-1;

      {Put directory back on disk}
      UNITWRITE(vol,dd,SIZEOF(dd),2);
    END;

  PROCEDURE get_names; {(vol:volume;
                        VAR number_of_files:INTEGER;
                        VAR names :filearray )}
    (..gets the file names (only) from the directory
      on volume vol (4,5,11, or 12))

      VAR i,k:INTEGER;
          dd:directory;

      BEGIN

        UNITREAD(vol,dd,SIZEOF(dd),2);
        number_of_files:=dd[0].dnumfiles;
        k:=0;
        FOR i:=1 TO number_of_files DO
        BEGIN
          WITH dd[i] DO
            BEGIN
              {reject inappropriate names}
              IF NOT (dfkind IN [securedir,untyped])
                AND NOT (LENGTH(dtid)=0)
                THEN BEGIN
                  k:=k+1;
                  names[k]:=dtid
                END {if}
            END {with}
        END {for}
      END;   {get_names}

    {The initialisation part - in this unit there isn't any initialisation,
     so the executable part of the unit is empty.}

    BEGIN
    END.
```

Listings to stop COPYA's capers

By CHEE WEI-LI

IN the October 1984 issue of *Apple User* there was a request in Feedback from Tim Stevenson asking for a way of preventing COPYA from copying a disc. I have here two short Applesoft programs in Listing I and Listing II to do just that.

Both programs assume that a standard DOS 3.3 (48k) is being used and patches are added to the DOS by the programs under that assumption. If the system is of a different memory size, changes to the program must be made accordingly. This program may not work with other types of DOS or on those that have been patched. It is suggested that the standard DOS 3.3 system master be booted on a 48k Apple II+ before running the programs.

The POKEs in Listing I from line 210 to 220 alter the address and data trailers on the protected disc. This is the simplest method, as it will not interfere with the booting process of the protected disc but is sufficient to deter COPYA and most of the standard DOS copy programs (unless the same patches are made to them).

However it will not be sufficient to deter bit-copiers and if that is desired more complex procedures have to be adopted which for the sake of simplicity I will not go into here.

The POKEs in line 230 alter DOS on the protected disc to allow use of the DOS commands only in the deferred mode of the Basic program and will force a reboot (PR#6) if any DOS commands are entered in the immediate mode.

This should be adequate to prevent listing of the Basic program, since at least part of it will be destroyed in the reboot. The line can be deleted if this is not desirable.

The POKEs from line 250 to 270 restore the DOS patches to their original values.

In order to keep the program in Listing I simple and to retain the standard DOS 3.3 commands, minimal patches have been added to the DOS. As such, running the program in Listing I will INIT the protected disc as with the standard command. A copy of the program in Listing I will be saved onto the protected disc under the specified boot file name. This file is then subsequently deleted by the program.

The protected disc cannot be booted until the boot file is placed onto the protected disc using the program in Listing II.

After running the program in Listing I, the system should be rebooted and the program in Listing II RUN.

This program will set up a small assembly language program at $0300 and by using the & key you will be able to toggle to and from the protected format and the standard format. You can then transfer files to and from the protected disc using the & key.

A typical sequence for transferring an Applesoft program from a standard disc to the protected disc will be:

CATALOG
See Catalog of standard disc
LOAD FILE
Load file from standard disc
&
Toggle to protected format
CATALOG
See Catalog of protected disc
SAVE FILE
Save file onto protected disc
&
Toggle to standard format

This process can be repeated as many times as necessary to transfer files to or from the protected disc.

B type files can be transferred in the same manner except those that use/load at $0300 as it conflicts with the & program. If this happens simply BLOAD the file at some other location.

One final point. An encoding program such as the one by Jonathan Lewis appearing in the April 1984 edition of *Apple User* or the one by Art Matheny appearing in Micro – for the Serious Computist (October 1984 edition) may be used to further protect your program.

```
10 D$ = CHR$ (4)
20 TEXT : HOME : PRINT
   "BOOT-FILE NAME TO USE":
   VTAB 3: INPUT "";N$
30 VTAB 5: PRINT "FILE TYPE
   (A/B): ";: GET T$: PRINT T$
40 IF T$ < > "A" AND T$ < >
   "B" THEN GOTO 30
50 ONERR GOTO 60
60 VTAB 7: INPUT "DISK VOLUME
   NO.: ";V%
70 V% = ABS (V%): IF V% < =
   0 OR V% > 254 GOTO 60
80 VTAB 7: HTAB 18: PRINT "
   ";V%: CALL -958: POKE
   34,7: POKE 216,0
90 HOME : VTAB 12: HTAB 2:
   PRINT "INSERT BLANK DISK IN
   SLOT 6, DRIVE 1"
100 VTAB 14: HTAB 7: FLASH :
   PRINT "ALL DATA WILL BE
   DESTROYED": NORMAL
110 VTAB 16: HTAB 7: PRINT
   "<ESC> TO ABORT,
   P-PROCEED";: GET A$
120 IF A$ < > CHR$ (27) AND
   A$ < > "P" THEN GOTO 110
130 IF A$ = CHR$ (27) THEN
   RUN
140 HOME : VTAB 12: HTAB 12:
   PRINT "FORMATTING DISK"
150 POKE 216,0: ONERR GOTO
   320
160 GOSUB 210: GOSUB 290
170 PRINT : PRINT D$;"INIT
   ";N$;",V";V%;",S6,D1"
180 PRINT D$;"DELETE ";N$
190 GOSUB 250
200 HOME : VTAB 12: HTAB 17:
   PRINT "DONE !"; CHR$ (7);:
   GET A$: RUN
210 POKE 47505,170: POKE
   48302,170: POKE 48312,170
220 POKE 47413,170: POKE
   47262,170: POKE 47272,170
230 POKE 42610,76: POKE
   42611,0: POKE 42612,198:
   POKE 42614,76: POKE
   42615,0: POKE 42616,198:
   POKE 40503,0
240 RETURN
250 POKE 47505,222: POKE
   48302,222: POKE 48312,235
260 POKE 47413,222: POKE
   47262,222: POKE 47272,235
270 POKE 42610,165: POKE
   42611,217: POKE 42612,48:
   POKE 42614,104: POKE
   42615,56: POKE 42616,96:
   POKE 40503,165
280 RETURN
290 IF T$ = "A" THEN POKE
   40514,6
300 IF T$ = "B" THEN POKE
   40514,52
310 RETURN
320 HOME : VTAB 12: HTAB 16:
   PRINT "ERROR!"; CHR$ (7);:
   GET A$: RUN
```

Listing I

```
10 FOR I = 768 TO 768 + 45:
   READ A: POKE I,A: NEXT
20 POKE 1013,76: POKE 1014,0:
   POKE 1015,3: TEXT : HOME :
   PRINT "& HAS BEEN SET UP"
30 DATA
   173,45,3,141,145,185,141,174
   ,188,141,184,188,141,53,185,
   141,158,184,141,168,184,201,
   222,240,6,169,222,141,45,3,9
   6,169,235,141,184,188,141,16
   8,184,169,170,141,45,3,96,17
   0
40 NEW : END
```

Listing II

TIME YOUR REACTIONS

SO you think you are quick? Well, have a go at this program and you will find out just how fast your reactions are.

Press Space and after a random interval an O appears at the bottom of the screen. You must press Space again – as quickly as possible – and the time you took to react will be given.

If you do not press Space after the O a reading will be given after 10 seconds, which acts as a check on your computer operation time.

Reaction time is influenced by a number of things, but I have found that an average time for 12 to 14 year olds is about 0.2 seconds.

At past retiring age I am lucky to get 0.25 seconds!

You may also like the Exit routine. On exit the cursor is brought up to cover the R of Run.

If your greetings program is 'HELLO' then all it wants is Right Arrow and Rept to cover Hello and Return and the disc catalog is displayed.

The time is generated by the FOR . . . NEXT loop, lines 160-170. This contains the keyboard press Peek which causes the program to jump out of the loop and read the time elapsed, calculated as T.

The time given is fairly accurate. I have checked it with a stopwatch, and for the purpose in hand it is accurate enough.

Frank Mallett

```
10  HOME : CLEAR : POKE  -
    16368,0
15  PRINT : PRINT : PRINT
    "THIS PROGRAM WILL TEST
    YOUR REACTIONS !"
16  PRINT : PRINT
    "**************************
    ************"
20  PRINT : PRINT : PRINT "TO
    START THE TEST PRESS
    'SPACE'."
25  PRINT : PRINT "WHEN 'O'
    APPEARS AT THE BOTTOM OF
    THE "
30  PRINT : PRINT "SCREEN
    PRESS 'SPACE' AS QUICKLY
    AS"
35  PRINT : PRINT "POSSIBLE.
    THE TIME YOU TOOK TO REACT"
40  PRINT : PRINT "WILL BE
    SHOWN, IF YOU PRESS BEFORE
    'O'"
45  PRINT : PRINT "APPEARS YOU
    WILL BE TOLD !!"
50  PRINT : PRINT : PRINT "
    PRESS 'SPACE' TO START"
100  GET A$: IF A$ = "" THEN
    100
105 Y =  INT ( RND (1) * 1000)
    + 1
110  FOR X = 1 TO 1500
120  IF X = Y THEN 150
125  IF  PEEK ( - 16384) > 127
    THEN 240
130  NEXT X
150  VTAB 23: HTAB 20: PRINT
    "O"
160  FOR A = 1 TO 788
165  IF  PEEK ( - 16384) > 127
    THEN 200
170  NEXT A
200  HOME : POKE  - 16368,0
210 T = ((A / 788) * 10) -
    .0126903553
230  VTAB 10: PRINT "
    REACTION TIME = ";T;"
    SEC;"
235  GOTO 250
240  HOME : VTAB 10: HTAB 3:
    PRINT "YOU PRESSED BEFORE
    'O' APPEARED !!"
245  POKE  - 16368,0
250  VTAB 14: PRINT "PRESS
    'SPACE' TO RESTART, 'X' TO
    EXIT"
260  GET A$
270  IF A$ = " " THEN 10
280  IF A$ = "X" THEN 300
290  GOTO 260
300  CALL  - 936
310  VTAB 10: HTAB 15: PRINT
    "GOODBYE !"
320  VTAB 14: HTAB 2: PRINT
    "RUN HELLO"
330  POKE 37,12
340  END
360  REM  THIS PROGRAM,
    WRITTEN
370  REM  BY FRANK.H.MALLETT
    OF
380  REM  SHOREHAM.W SUSSEX.
```

appletip

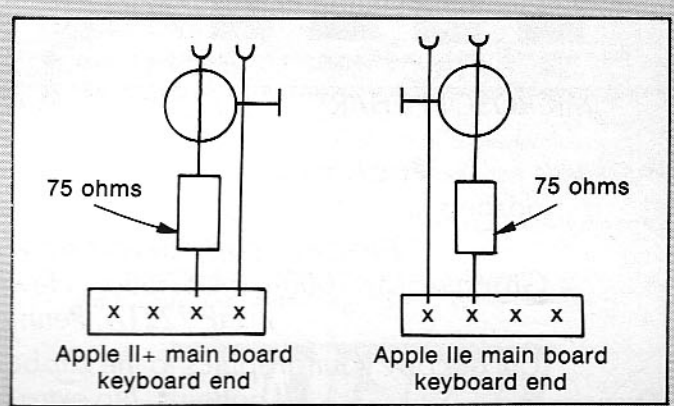
It is quite simple to construct an extra video output on your Apple. Just connect a video plug by a 75 ohms resistor to connector J14 in the Apple II+ and connector J13 in the Apple IIe as indicated in the figure.

You can use a Molex type of plug to make the connection. With the Apple IIe you can mount the plug at the back by drilling a hole in one of the plastic shutters. With the Apple II+ you just have to drill in the case.

The extra output gives you a second standard video signal (1v tt) at an impedance of 75 ohms. You can use this to connect your TV set to the Apple together with your normal monitor.

This saves changing the wiring, which is a great advantage when children are playing with your expensive Apple!

Martin Keesen



Apple II+ main board keyboard end

Apple IIe main board keyboard end

Try putting years on your clock

NOT all clock/calendar cards give the year part of a date – the Glanmire Microwatch is an example and there may be others. Programs using such cards to produce correspondence, accounts, reports and the like, where a full date is essential, must thus supplement the hardware-derived date, typically by adding a string variable.

Where a number of such programs are held on a disc it may become more convenient to hold the current year in a short text file to be accessed by any program requiring it.

Amendment of that single file at the beginning of the year then has the effect of updating all programs, without the need to identify and separately amend each. In practice if the year is accessed at the start of a program the extra disc operation is, both figuratively and literally, "lost in the noise" of the RUN (program) command.

From there it is a simple further step to make the program population itself maintain the year file and update it when the time comes, and I have used such an approach for the last 18 months as follows:

Create the Year file to contain two arithmetic variable, TY and LY, with TY set initially to this year and LY to last year. Instead of a simple disc read, incorporate into each appropriate program the standard sub-routine listed (for brevity the disc and calendar routines are described but not listed.)

In operation line 62020 returns truth values X and Y, according to the truth (1) or falsity (0) of the statement following each. A desk check of the logic will show that on all but two runs in the year, the state of the year file will satisfy the truth checks and the sub-routine will add little to a simple disc read.

Whichever program happens to be run first in each half-year however, will amend the year file in some way – lines 62030/62050 or 62040/62050 – before continuing with its normal task and subsequent users of the file will again find it "true". Line 62030 primes the year clock in the first run of any program after the end of June and line 62040 does the actual year change at the first run in the New Year.

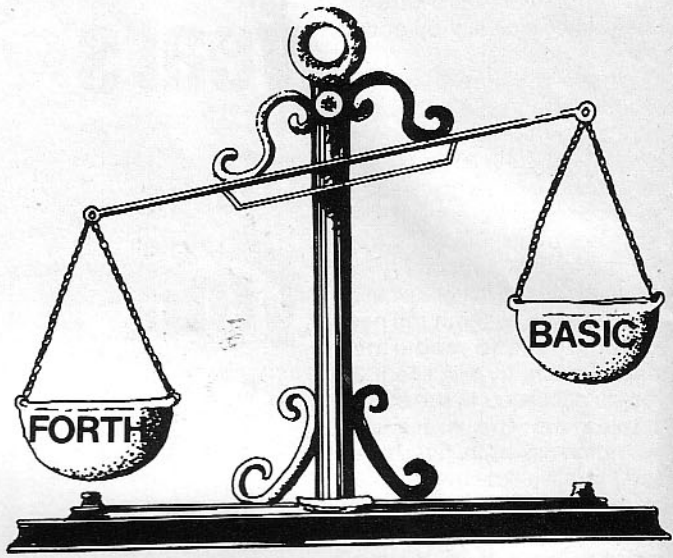
On balance the obvious temptation, to put the routine in the Hello program only and rely on a simple disc read in all other programs, should be resisted. Murphy's Law being what it is, you will not have booted from that disc at the first run of the year!

Richard Brown

```
62000 GOSUB 62080: REM Read
   Clock, returning incomplete
   datestring (DT$) and month
   no. (M)
62010 GOSUB 62100: REM Read
   TY,LY from year file
62020 X = M > 6: Y = TY = LY:
   IF(X AND Y) OR (NOT X AND
   NOT Y) THEN 62060
62030 IF X AND NOT Y THEN LY =
   TY: GOTO 62050
62040 TY = TY + 1
62050 GOSUB 62120: REM Write
   TY,LY to year file
62060 DT$ = DT$ + " " +
   STR$(TY)
62070 RETURN
```

BINSEARCH

COLIN DAVIES describes an efficient search algorithm and reports that if time is of the essence Forth beats Basic hands down



TO old hands, an article describing binsearch may seem old hat. However it is an extremely efficient algorithm and there must be many beginners who don't know the method.

This article includes implementations in Basic and Forth and so demonstrates the differences between the two languages.

Binsearch is used to find whether or not a given value is contained in a list of elements – an array – which has been sorted into ascending order.

If the value is present then the method will set a value K equal to the subscript of the array element which contains the value being sought.

If the value is absent then K is set to zero. The array element zero is not used, so there is no confusion.

The method is to examine the element at the middle of the array. If it is greater than the value sought then the top half of the array may be ignored and the process continued until the value has been found or the entire array discarded.

As half of the remaining elements are discarded after each comparison it can be seen that the maximum number of comparisons needed is $\log_2 n$ where n is the total number of elements.

This means that the greater the number of elements the more efficient is the method.

The demonstration finds an element containing the target value in an array of 10,000 elements. The number of the element is then displayed along with the appropriate part of the array (to prove you're not being kidded).

To sort an array of this size before the demonstration would take more than three minutes in Forth and more than half an hour in Basic.

To get over this problem point P in the array is chosen a random. The array is then fille with 1s up to this point, then a the point a 1 or 2 is inserted a random and the remainder o the array is filled with 3s.

The value searched for is se to 2, which may or may not b present. The bell is sounded a the start and end of the searc itself.

The time taken to find th target value in Basic was 1. seconds and in Forth it was fo all practical purposes instantan eous. The time taken to load th array was 54.5 seconds in Basi and 4.6 seconds in Forth.

```
100  REM  BINSEARCH DEMO
110  REM
120  HOME
130  LET N = 10000: DIM A%(N):
   LET X = 2
140  LET P =  INT ( RND (1) *
   9999) + 1:T =  INT ( RND
   (1) * 2) + 2
150  FOR K = 1 TO P - 1:A%(K)
   = 1: NEXT :A%(P) = T
160  FOR K = P + 1 TO N:A%(K)
   = 3: NEXT
170  PRINT  CHR$ (7): GOSUB
   340: PRINT  CHR$ (7); CHR$
   (7)
180  IF K = 0 THEN  PRINT "
   TARGET VALUE NOT PRESENT ":
   GOTO 200
190  PRINT "TARGET VALUE IS
   ELEMENT NO. ";K
200  PRINT
210  IF P < 3 THEN P = 3: IF P
   > 9998 THEN P = 9998
220  PRINT "ELEMENT ";
230  FOR L = P - 2 TO P + 2:L$
   =  STR$ (L)
240  IF  LEN (L$) < 4 THEN L$
   = " " + L$: GOTO 240
250  PRINT L$;" ";: NEXT :
   PRINT
260  PRINT "VALUE   ";
270  FOR L = P - 2 TO P + 2:L$
   =  STR$ (A%(L))
280  IF  LEN (L$) < 4 THEN L$
   = " " + L$: GOTO 280
290  PRINT L$;" ";: NEXT :
   PRINT
300  END
310  REM
320  REM  BINSEARCH
330  REM
340  LET LO = 1:HI = N:K = 0
350  IF LO > HI THEN 400
360  LET MID = (LO + HI) /
   2:MID =  INT (MID)
370  IF X < A%(MID) THEN HI =
   MID - 1: GOTO 350
380  IF X > A%(MID) THEN LO =
   MID + 1: GOTO 350
390  LET K = MID
400  RETURN
```

```
SCREEN #30
 0 ( BINSEARCH )
 1
 2 10000 CONSTANT NMAX   1 NMAX ARRAY A   VARIABLE X
 3
 4 : INITIALISE   X ! 1 SWAP ;          ( N,X---LO,HI )
 5
 6 : ANY-LEFT     2DUP <= ;             ( LO,HI---LO,HI,f )
 7
 8 : FIND-MID     2DUP + 2/ ;           ( LO,HI---LO,HI,MID )
 9
10 : MID-HIGH     DUP A @ X @ SWAP 2DUP < ;
11                  ( LO,HI,MID,---LO,HI,MID,X,A-MID,f)
12
13 : MID-LOW      > ;    ( LO,HI,MID,X,A-MID---LO,HI,MID,f )
14
15 -->


SCREEN #31
 0 ( BINSEARCH )
 1
 2 : SELECT-BOTTOM     2DROP 1- SWAP DROP ;
 3                                ( LO,HI,MID,X,A-MID---LO,HI )
 4
 5 : SELECT-TOP        1+ ROT DROP SWAP ;  ( LO,HI,MID---LO,HI )
 6
 7 : FOUND             -ROT 2DROP ;        ( LO,HI,MID---K )
 8
 9 : NOT-FOUND         2DROP 0 ;           ( LO,HI---K )
10
11 -->
12
13
14
15


SCREEN #32
 0 ( BINSEARCH )
 1
 2 : BINSEARCH      INITIALISE                  ( N,X---K )
 3                    BEGIN ANY-LEFT WHILE
 4                          FIND-MID
 5                          MID-HIGH IF
 6                              SELECT-BOTTOM
 7                          ELSE
 8                              MID-LOW IF
 9                                  SELECT-TOP
10                              ELSE
11                                  FOUND EXIT
12                              THEN
13                          THEN
14                    REPEAT
15                          NOT-FOUND ;  -->


SCREEN #33
 0 ( BINSEARCH-DEMO )
 1
 2 10000 CONSTANT N    VARIABLE P
 3
 4 VARIABLE RND   HERE RND !
 5
 6 : RANDOM       RND @ 31421 * 6927 + DUP RND ! ;     ( ---R )
 7
 8 : CHOOSE       RANDOM U* SWAP DROP ;     ( U1---U2 )
 9
10 : SET-P        N CHOOSE 1+ P ! ;         ( --- )
11
12 : LOAD-ARRAY   N 1+ P @ 1+ P @ P @ 1     ( --- )
13                DO 1 I A ! LOOP
14                2 CHOOSE 1+ SWAP A !
15                DO 3 I A ! LOOP ;  -->


SCREEN #34
 0 ( BINSEARCH-DEMO )
 1
 2 : RESULT       DUP 0= IF                    ( K--- )
 3                        DROP ." TARGET VALUE NOT PRESENT "
 4                      ELSE
 5                        ." TARGET VALUE IS ELEMENT NO. " .
 6                      THEN ;
 7
 8 : FIX-RANGE    P @ DUP                      ( --- )
 9                  3 < IF
10                        DROP 3 P !
11                      ELSE
12                        N 2- DUP -ROT
13                        > IF
14                          P !
15                        ELSE DROP THEN THEN ;  -->


SCREEN #35
 0 ( BINSEARCH-DEMO )
 1
 2 : DISPLAY          ." ELEMENT "                ( --- )
 3                    P @ DUP 3 + SWAP 2-
 4                    DO I 4 .R ."  " LOOP
 5                    CR ." VALUE   "
 6                    P @ DUP 3 + SWAP 2-
 7                    DO I A @ 4 .R ."  " LOOP ;
 8
 9 : BINSEARCH-DEMO   PAGE SET-P LOAD-ARRAY       ( --- )
10                    BELL N 2 BINSEARCH BELL BELL
11                    RESULT CR CR
12                    FIX-RANGE
13                    DISPLAY CR CR ;
14
15
```

Planner for Mac

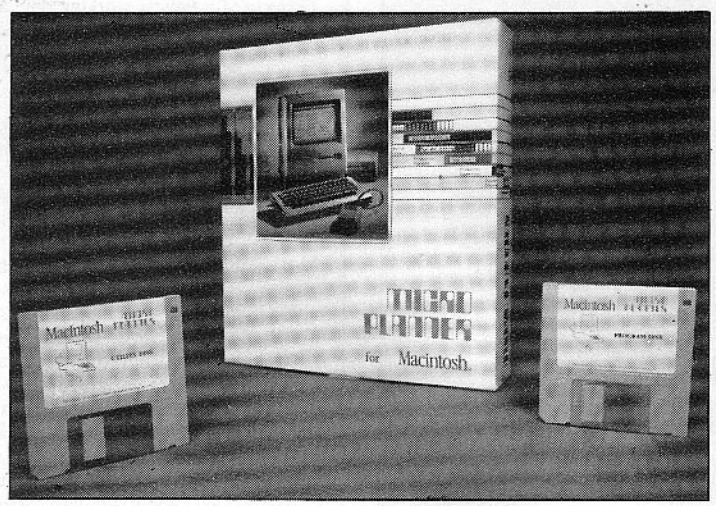
THE highly acclaimed Apple II program Micro Planner has been made available for the Macintosh by Micro Planning Software.

Micro Planner can be used for control of any project which requires the co-ordination of interdependent operations regardless of size, says the manufacturer.

The package is being included in Apple's "Test Drive a Macintosh" campaign aimed at the corporate office market.

The Macintosh version has been distilled from other versions currently in use at more than 800 installations worldwide for projects as varied as the construction of the space shuttle launch facilities, development of new drugs, planning of trade shows, magazine layout schedules, and planning and resourcing of software development projects. Price: £295.

● *Micro Planning Software, 34 High Street, Westbury-on-Trym, Bristol. Tel: 0272 509417.*



Micro Planner for Macintosh

Heavenly view

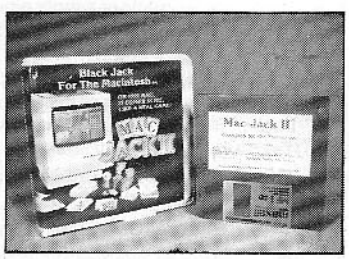
ASTRONOMY version 1.0 has been released for the Macintosh by E & M Software. Written in Microsoft Basic 2.0 the program produces two plots in separate windows.

The first window, Solar System, plots a Pole Star's view of the solar system for a given time and date.

The second window, Sky View, plots an image of the heavens as seen by an Earth-bound observer for a given date, time, longitude and latitude.

The Sky View plot locates the Sun, Moon, planets, comets and many major constellations. The windows can be thrown away, recalled, repositioned or resized. Price is $25.

● *E & M Software, 95 Richardson Road, No. Chelmsford, Massachusetts 01863, USA. Tel: 0101 617 251 7451.*



NOW you can take on your Macintosh at poker and blackjack with programs from DataPak Software.

Mac-Jack is a blackjack style game that makes use of the machine's mouse and animated graphics. As in real blackjack, the object is to have a higher card total than the dealer – in this case the Macintosh – without exceeding 21.

Mac-Poker is an electronic game of five card stud played against the computer. The player can peek at his cards, discard, draw and bet just like the real game. Price of each game is £44.95.

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

Dual processor

A DUAL processor board with Apple II compatible slots has been launched by U-Microcomputers.

The U-DP incorporates two 6502 microprocessors with shared access to up to 128k RAM on board – 572k with add-on boards.

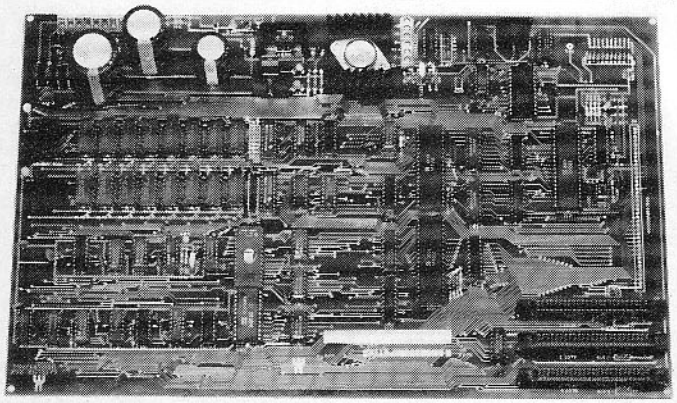
While one processor controls a serial port and parallel port the other has control of three Apple II compatible slots.

A wide range of boards is available to plug into these slots. The U-DP also has an on board power supply.

The choice of the dual 6502 processor and Apple II compatible slots allows an Apple II to be used as the development machine to produce machine code software to run in the U-DP.

The U-DP will not run standard Apple II software as such because it has two processors and can access a lot more memory. Price of the 128k model including on-board psu is £375.

● *U-Microcomputers, Winstanley Industrial Estate, Long Lane, Warrington Cheshire WA2 8PR. Tel: 0925 54117*



U-DP dual processor board

Automated pages

PUBLISHED by Manhattan Graphics, Ready Set Go automates the page design and pasteup process, turning the Macintosh environment into a professional graphic arts publishing system.

Manhattan says the system overcomes the limitations of current Macintosh programs such as MacWrite, MacPaint and MacDraw encountered when trying to use them for sophisticated page makeup.

The package is for projects requiring a considerable artistic design component such as newsletters, reports, forms, flyers, brochures, presentations and all page proofing and layout design work.

Pages may be stored on disc as full documents or as "shells". The latter facility allows the re-use of page designs or the building of an inventory of layouts for future projects.

Pages can be printed on the Imagewriter or on the Laser Printer. The package will retail for about £125.

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

Tolerant discs

THE new range of 5¼in FlexyDisks from Basf is said to have been designed to withstand rigours of micro applications for which floppy discs were never intended.

Brand-named Science, the range is claimed to be suited to circumstances in which absolute data security is imperative, such as the fields of

New line in Basf 5¼in discs

medicine and scientific research.

The recording surface has been treated by an improved coating process which guarantees as many as 70 million head passes per track. Under normal operating conditions, the makers say, the disc should last for about 20 years.

It has also been designed to tolerate usage and storage in a wide range of temperatures and incorporates a special antistatic material in its jacket.

Basf says the disc can operate at temperatures up to 60 deg C, and be stored in temperatures as low as 4 deg C or as high as 70 deg C. Prices range from £5 for the 1D single-sided single density to £6.40 for the 2HD double-sided high density disc.

● *Basf Computer Division, 4 Fitzroy Square, London W1P 6ER. Tel: 01-388 4200.*

Back-up system

THE PC Megastore file-orientated, addressable back-up system has been launched in Europe by Ampex.

With its capability of direct access to files on tape it can locate data anywhere on the tape within 90 seconds, the company says.

The off-line streaming capability means a user can continue to work at the computer while a back-up or restore operation takes place.

By insertion of a new host adapter card into a computer the PC Megastore will work with an alternative computer making non obsolesence a key advantage.

The Apple version includes an interface card and a boot program. Prices for the four systems available start at £1,900.

● *Ampex, Acre Road, Reading. Tel: 0734 875200.*

Printer connection

A HARDWARE Expansion device that allows all Macintosh applications to use a wide range of IBM compatible printers has been announced by Microsoft.

The MacEnhancer provides four additional ports, allowing five different peripherals to be connected. These can be letter quality printers, dot matrix printers, modems, and other devices using RS232 and RS422 serial connections, as well as parallel connections.

As well as including all the software necessary to drive a wide range of printers, the system enables the Macintosh to emulate VT52 and VT100 terminals.

This permits Macintosh-to-Macintosh connection as well as connection to mainframe and minicomputers, and to other micros, regardless of which operating system they use.

MacEnhancer is compatible with 128k and 512k Macintosh models and costs £250.

● *Microsoft, Piper House, Hatch Lane, Windsor, Berks. Tel: 07535 59951.*

A library of LOGO

Seven Logo books reviewed by CLIFF McKNIGHT

WHEN I met Seymour Papert in 1970 I wish I'd known how famous he was to become – I might have paid more attention to him.

Lots of people obviously did pay attention to him and his MIT colleagues because Logo, the language they wrote, is now one of the most famous of computer languages. There are implementations of Logo on practically every major micro on the market.

It's not surprising, then, that books on Logo abound. Recently seven such books were sent for review so I'll try to give a brief overview of all of them.

Of the seven, five are written specifically for Apple implementations of the language. The odd ones out are **Forward 100** by Ray Hammond and **Logo for Micros** by Martin Lesser.

Forward 100 includes an overview of most implementations and advises you to "choose the version of Logo most suitable for your needs and only then choose the computer which will run it" (p.173).

However most people come to languages having already bought their computer so the advice is not likely to be well-heeded, good though it may be. Having said that, schools with several varieties of computer would find the book a useful source if they were contemplaing the purchase of Logo.

There are some interesting case histories of "Logo in action" presented, and if you're thinking of buying a version of the language for your Apple there's a discussion of the similarities and differences between the 'official' Apple version and the Krell/MIT version.

None of the Macintosh Logos are mentioned, but I would expect a future edition of the book to incorporate them since it seems fairly comprehensive.

Logo for Micros is a serious introduction to the language largely through its list processing capabilities rather than the more usual turtle approach. Like Hammond's book, it's not written as a hands on guide although it does aim to be a programming book.

While it is not written to be Apple specific, the fact that the author uses an Apple and takes the Apple LCSI Logo to be the reference version means that it is ideally suited to the Apple user. It points out that the Krell/Terrapin varieties lack the error handling capabilities of LCSI versions, and even mentions the Macintosh Logo.

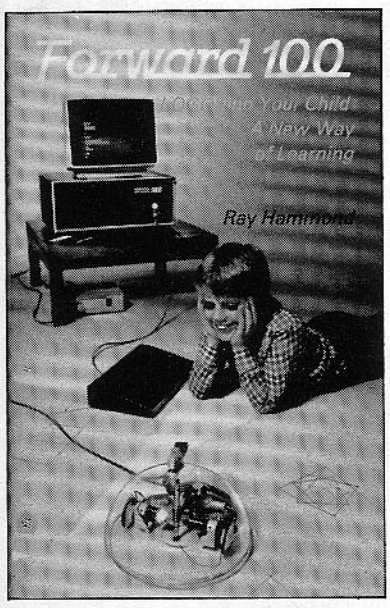
The CyberLogo Primer is by Gary Bitter and Nancy Watson. They obviously specialise in Logo, having published the Apple Logo Primer in 1983.

If you're not familiar with the Logo market, you'd be forgiven for not realising that CyberLo is a version for the App However once the instructi start telling you to "insert Systems Master 3.3 diske that came with your compute it's a fair guess that it's aimed at BBC or IBM owners.

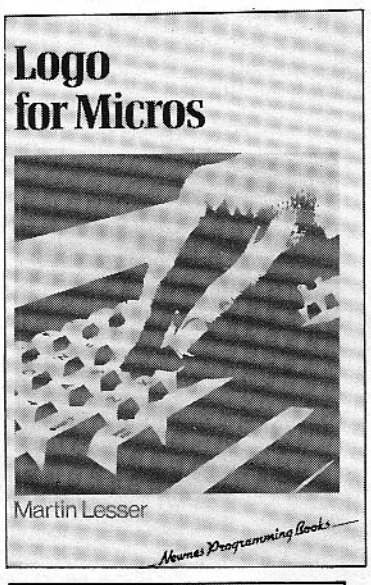
In fact the book alm manages without mention the Apple computer. The c reference is in a confus section which gives an overvi of the language. The confus is not in the description bu the counting. The text promi a list of nine versions, the ta is headed "Seven major v sions of Logo" and actua contains eight versions!

Most of the book is step-by-step guide to Cyb Logo but there is a crash cou for the experienced or impatient fast learner. The bo would seem to be most use then if you're consideri buying CyberLogo (it is covered in Forward 100).

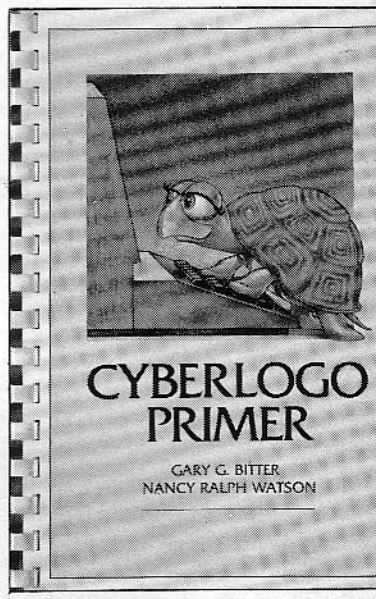
What makes CyberLogo ferent is that it fits in 48k can therefore be run by Ap II+ owners without the use language card. It concentra on turtle graphics and there no text manipulation c abilities. However, it has ot advantages, like a refere



Title: Forward 100
Author: Ray Hammond
Publisher: Viking
Price: £12.95



Title: Logo for Micros
Author: Martin Lesser
Publisher: Newnes Technical
Price: £7.95



Title: CyberLogo Primer
Author: Gary G. Bitter and Nancy Ralph Watson
Publisher: Reston-Prentice/ Hall
Price: £18.30

section in software which can be accessed at any time during use.

The book gives answers to practice activities and there are suggestions for further work. It also gives some rather heavy-handed printing instructions which involve jumping out via Ctrl-Reset and dumping the graphics page. It claims you need a Grappler card to achieve this, but of course many other printer cards will support a graphics dump.

Turtlesteps by Pamela Sharp and **Apple Logo** by Harold Bailey, Kathleen Brautigam and Trudy Doran are so similar in style of presentation that it's not surprising they come from the same publisher.

However they each have a different focus. The former aims to be an introduction to Apple Logo and Terrapin Logo while the latter is aimed at exploring turtle graphics.

Such is the similarity between these two versions of Logo that Turtlesteps manages to deal with them both simultaneously. The only major exception concerns procedures and a separate chapter 7 is devoted to each version. There is also an additional chapter describing some of the extra editing commands available in Apple Logo.

Apple Logo (the book) specifically ignores the arithmetic operations and list-processing in order to concentrate on turtle graphics. There are lots of examples, plus Turtle Testers and Turtle Teasers (with answers provided).

Logo For Apple Computers by Roger Haigh and Loren Radford aims to be a self-teaching guide. It claims to cover all versions of Logo available for the Apple II, but doesn't include CyberLogo.

As the rather formal title might suggest, it's much more 'dry' than the other books and has a more serious tone. Like Logo for Micros, I found it a refreshing change from the jolly, jog-along approach of many Logo books. They're more the sort of book I would expect to find in a university or college library rather than on a primary school teacher's shelf.

In complete contrast, **Step By Step Through Logo Turtle Graphics** is obviously written to be used by children while sat at the computer, as the cover illustration shows.

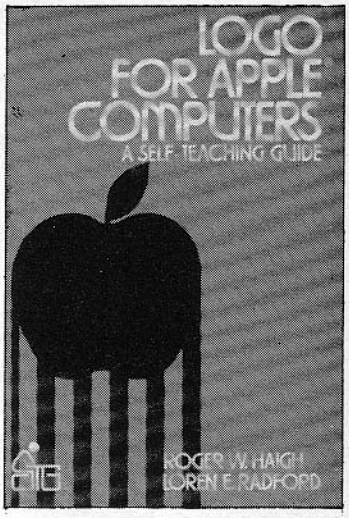
It's a large-format book with plenty of white space on the page and the child is guided right from the act of turning the Apple on. I say 'child' but the book gives the appropriate range as "for ages 8 to adult".

The lower limit depends very much on the child. My six-year-old had no problems with it other than finding the instructions on using the Apple a bit tedious. However, she has been using Apples for three years now.

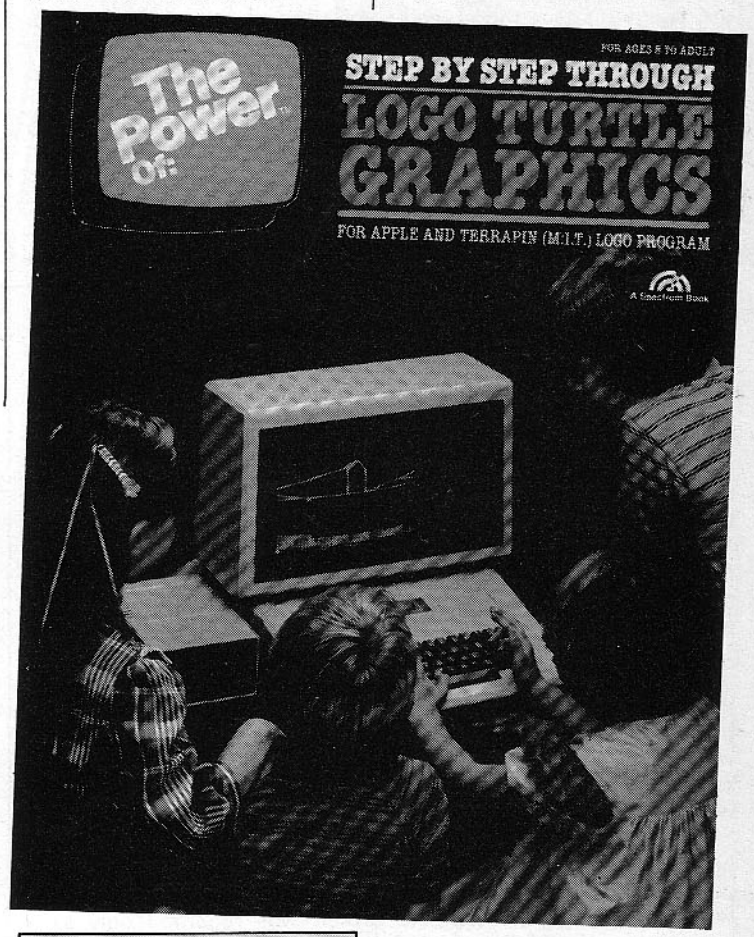
I was initially sceptical about the book being useful for adults but eventually decided that someone who knew nothing about programming might take to it. They would be able to work through it in much less time than a child and might just feel a little patronised.

By the time I'd finished looking at these seven books, I could appreciate why Ray Hammond called his book Forward 100. I got to the point where I was playing turtle as I moved around the house.

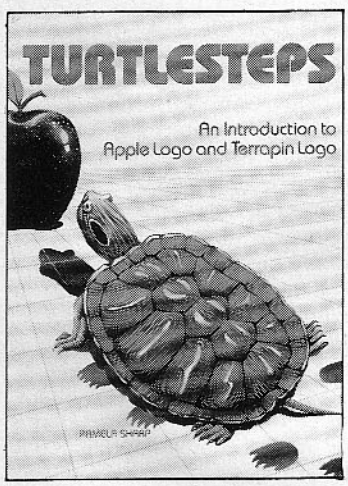
The interest teachers have shown in Logo was one of the more encouraging trends noted in a recent SSRC report on micros in education. If you add these seven books to the pile already available, there's something for everyone, from the toddler to the dodderer.



Title: Logo For Apple Computers
Author: Roger Haigh and Loren Radford
Publisher: John Wiley
Price: £14.95



Title: Step By Step Through Logo Turtle Graphics
Author: Ann Rose
Publisher: Spectrum-Prentice/Hall
Price: £7.50



Title: Turtlesteps
Author: Pamela Sharp
Publisher: Brady-Prentice/Hall
Price: £13.55

Title: Apple Logo
Author: Harold Bailey, Kathleen Brautigam and Trudy Doran
Publisher: Brady-Prentice/Hall
Price: £14.50

No dealers – but Apples can still thrive in Hungary

LASZLO KORANYI relates how they're beating home-grown products on both range and price

ALTHOUGH Hungary is not really far from the UK – for example Budapest is nearer to London than the South of Spain or Stockholm – for many people it seems to be a distant place.

That is the reason why I think that there might be a few readers who thought that the only kind of apple which might have any connection with Hungary would have been the tasty fruit, of which Hungary produces a significant quantity.

It is true that compared to the UK or West Germany, the number of Apple micros is very low. Even so, through the couple of hundred machines and through their many more users there are a few thousand people in Hungary who have had personal contact with an Apple.

The first machines arrived in the country about three years ago, and since then the number has risen to the current level, necessitating the formation of an Apple User Group in 1982.

The group is a member of the Hungarian John von Neumann Computer Society. (For those who might not know, since the 1940s almost all computers have followed principles laid down by John von Neumann, the Hungarian-born mathematician.)

The group's task is to keep Hungarian users in touch with each other and, as in any such group, the exchange of experiences and programs – and the most important – games.

The sessions are always held near to an Apple. We gather monthly at the home of one of our members.

It must seem rather strange that there is, as yet, no Apple dealership in Hungary. The nearest dealer is in Vienna, from where companies and institutes can order the necessary hardware and software.

Quite a few Apples were brought into the country as private purchases. These were bought by Hungarian scientists or computer professionals who worked abroad and had the opportunity to buy the equipment in that country.

Although both these methods are more complicated and slower than simply going to the dealer around the corner, this by no means prevents the spread of Apple computing in Hungary.

Why are Apples bought in Hungary? The country has a rapidly developing computer industry specialising in minis. In the past couple of years, Hungarian manufacturers have also appeared on the domestic market with personal computers, but Apples are often preferred because of the wide variety of peripherals and software and because of their relatively low prices.

The first personal acquaintance with Apples were made by professionals while abroad. Now we can get knowledge of the most recent hardware and software developments either through the foreign contacts of our user group (mainly German) or through magazines like *Apple User*.

In the beginning the Apples were used mainly by professionals who progressed from other types of computers to the micros. A smaller proportion of the users worked in other fields of science and started to use Apples only as tools.

In many cases they fell in love with computing – which until then was regarded only as a necessary evil – through the user-friendliness of the Apple.

At present the circle of users is widened with those who are users in the original sense of the word – using the machine only to carry out a specific task, either in science or in business.

It is not surprising then that in Hungary the Apple is used primarily for scientific purposes and it is interesting that more specifically it is extensively used in the life-sciences – in the fields of medical, biological and psychological research.

Applications include modelling of biological systems, calculation and graphical representation of atomic coordinates and parameter estimation with nonlinear regression.

A statistical package for evaluating biological and psychological experiments has been developed. An Apple extended with a video camera and a digitiser is being used for evaluating conventional (non computer thomographic) X-ray pictures.

A few Apples are used in business, and probably different from the situation in Britain the users are generally big companies rather than small ones.

This fact determines that the sort of applications are those in which, on the basis of relatively low data storage, meaningful analyses can be made for business decisions.

For these applications either software written in-house or such well known packages as VisiCalc or PFS are used. Business applications have a useful side effect because users claim that it was the first time they discovered that computing can be not only useful but human and friendly as well.

The third significant application field is education. Apples are used in the medical faculty of Budapest University for special introductory computing courses. They were used this summer for courses in programming held for children between 10 and 14 years old.

On the basis of the very good experience gained so far, we can assume that further Apples will continue to arrive in Hungary and that the field of applications will be widened.

Although Commodore 64s and ZX Spectrums are better known here than Apples – the main reason for this is the price – as one can guess – among professionals the Apple is very popular and well known.

The only problem which arises as a result of the lack of dealers is the rare occasion when maintenance is needed. We try to solve these problems ourselves, but it would be a great help if we could read a series in *Apple User* dealing with maintenance and repair. Probably this would be useful not only to Hungarian readers.

Also we would be very happy to contact a British user group or users if they think this might be interesting for them as well.

I hope that this short round-up of the Apple scene in Hungary brought this not so distant country nearer to the readers. But please don't forget about the other kind of Hungarian apples!

Recovering deleted Pascal disc files

HAVING read Stephen Lowe's letter in Vol. 5 No. 3 of Apple User on recovering deleted Pascal text files on disc, I would like to mention that recovery is fairly easy, although it requires access to a disc editor such as Bag of Tricks which can access Pascal discs and take into account the different sector skewing used from DOS 3.3.

The CIA Files is not appropriate as it only works on DOS 3.2 and 3.3 skewing although I must admit it is a package I couldn't live without.

Recovering deleted Pascal text files is fairly straightforward so long as you haven't used the Filer's Krunch utility to move files together on disc.

If you have done this then I'm afraid your file has probably gone unless it was the last one on disc when you deleted it, in which case it should still be intact.

The first thing to do is to get an extended list of the directory which will show the length and starting block of unused portions of the disc.

If you know the size of the lost file then you can usually make a fairly good assumption as to which portion of unused blocks is associated with it. If not then I'm afraid you'll just have to start searching from the beginning of each unused area until you find the file required.

Once you have noted these details down on paper remove your Pascal disc and boot your disc editor. The one I use is Zap on Bag of Tricks. Change sector skewing to the Pascal format, then read in the sector corresponding to the starting block of the unused area.

*To convert blocks to track and sector is very simple. The track is BLOCK DIV 8 and the sector is (BLOCK MOD 8) *2.*

Then read in sectors in turn until you recognise the start of your text file. This will always begin on an even numbered sector, 0,2,A etc, and in normal circumstances this will be the fifth sector read in as the first two blocks of a text file contain the editor environment information.

Continue reading through the sectors, check that the text is sensible and in order, until you reach the end of the text, the block containing the END. statement.

You must now calculate the size in blocks of the text file you wish to recover. The starting sector should be four before you met readable text, which incidentally should begin with the program heading, and the last sector is the one containing the END. statement.

If the last sector number is even – 2,4,8,E etc – add one as Pascal files are always saved as blocks which end on an odd numbered sector and start on an even number.

You should now have the starting track sector and ending track sector and can calculate the starting block and length of file in blocks as follows:

*Start block = (STARTTRACK * 8) + (STARTSECT DIV 2).*

*File length = ((ENDTRACK – STARTTRACK – 1) * 8) + ((17 + ENDSECT – STARTSECT) DIV 2).*

The next thing to do is to reboot Pascal, enter the Filer and put in the disc with the lost file. If the starting block of the unused area on disc is less than your lost files starting block then use the MAKE command to make a dummy file to fill in the gap up to your lost file's starting block.

The dummy file should be created with a length of (Lost file start block – unused area start block) for example, DUMMY.TEXT[5] will create a dummy file five blocks long.

Then simply use the MAKE command again to create a file of the file length as calculated above LOSTFILE.TEXT[24].

Make sure you use the .TEXT suffix in your file name otherwise the MAKE command will make the file as one of type DATA and the editor will not recognise it as a text file.

I have had cause to do this on several occasions as I had a faulty Apple IIc disc drive which tended to corrupt directories when writing to the disc.

As to Mr Lowe's other problem, of converting a code file back to text, this is impossible as the final p-code file contains no information on data structures such as type definitions, variable names or structure. This is all taken into account by the compiler, which then discards them.

Also several functions and procedures are handled directly at compile time such as CHR, ORD, PAGE, SUCC and PRED.

A code file may contain segments from several text files, library units and even external 6502 segments which come from assembler text files.

Hence any program which could convert a code file into a text file or even group of text files would be a very welcome Pascal utility, although I feel it would perhaps annoy those software houses selling such things as the PFS packages or Wizardry which are distributed as "untouchable" code files.

I hope this will be of use to your readers as I know the feeling of having just deleted a few hours' editing with no knowledge of how to recover the file. – **Allan Ogg, Glasgow.**

● **Peter Gorry** adds: It's a mystery to me why CIA doesn't have a command to switch between DOS 3.3, CP/M and Pascal sectoring, since the difference between them is minor.

All three disc systems have an identical format at the sector level – the only difference is the order in which the sectors

Getting back to Sterling

AS an enthusiastic Appleworks user on an Apple IIe with the usual bits and pieces, I was interested to see Peter Bradley's Appletip in the March issue of Apple User about replacing the $ sign with a £ sign.

I would like to make this change, but have had no luck with asking Apple via my dealer.

I've tried to do what Peter suggested, but found nothing at all at the disc position he mentions – track 14, sector 0F, byte AA.

I then tried the other address – track 13, sector 02, byte A5 – and found no $ sign there either.

There was a $ sign a little further on and I changed that just to see what would happen, but it simply stopped the spreadsheet from working.

Was there a printing error, have I got it wrong, or could there be several different versions of Appleworks? Mine is version 1.0. – **Mike Worth, Hawkhurst, Kent.**

● In view of your comments I rechecked my copy of Appleworks and its predecessor Three E.Z. Pieces (issued for use on the Apple III) and they are both all right.

I think you'll find that the reason for your lack of success is the order of the files on the program disc.

On mine, the files catalog as SEG.PR, SEG.ER, and SEG.MN – the last being the program.

If this file is in position one (as I suspect yours is), the modification is to change byte $AA from $24 to $23 on track 13 sector 02.

This is the only modification required. The other one given in the Appletip is the modification for Three E.Z. Pieces.

My Appleworks is version 1.0.1 and if yours is the same I'm sure that you'll be able to make the modification without any problem.

Peter Bradley

appear on the disc.

Thus "logical" sector 1 is the 13th sector on a DOS 3.3 disc, the second on a Pascal disc and the third on a CP/M disc.

All that is required to change between the different types is to alter the sector table in the RWTS routine. The table below gives the required values for each system.

DOS 3.3	PASCAL	CP/M
00	00	00
0D	02	03
0B	04	06
09	06	09
07	08	0C
05	0A	0F
03	0C	02
01	0E	05
0E	01	08
0C	03	0B
0A	05	0E
08	07	01
06	09	04
04	0B	07
02	0D	0A
0F	0F	0D

CIA uses two RWTS routines, one for reading and one for writing. Both must be altered to change disc types.

One table is held at the normal DOS location $BFB8-BFC7 (reading) and the other is at $3FB8-3FC7.

To change CIA you must quit Tricky Dick and enter the monitor (CALL-151). Next you must enter the appropriate table at $BFB8 and $3FB8. A Ctrl-Y will return you to CIA.

Unfortunately if you want to swap CIA modules you will have to restore the DOS 3.3 values first. If you want to use Pascal a lot it might be worth transferring the whole of CIA onto a DOS disc which has been initialised with the Pascal table.

Sorting figures

J.P. LEWIS'S clean-up utility in the March 1985 issue of Apple User caught my eye, not for its garbage-reducting action but because I feel it ought to be able to speed up the sorting of tables. Thanks for giving me the idea.

As it happens though, I am sorting only figures at the moment. Of course I could turn them into strings, sort, and then turn them back into figures, but that seems cumbersome and probably negates any speed advantage there might be.

Could you suggest a similar routine for figures only, please?

Also almost every published utility starts at $300. I already have another one there – where else could I put this one to try it? – **G.C. Balmain, Southampton.**

● The biggest problem with a number swapping routine is that there are many possible errors that need to be trapped, for example illegally swapping a real with an integer.

However if you are prepared to sort out your own error trapping, the key subroutine to handling the task is the machine code listed below.

This checks that the next character in the line is a comma, then calls the Applesoft's PTRGET routine, which tries to find the address of the variable following the comma.

When (if) it returns, the high byte of the address will be in the Y register, the low byte in the A register.

Around this routine – which you will have to call twice – you need to build the program to do the swapping. To swap an integer, real or arrayed real, you need to move five bytes. To swap an arrayed integer you need to move two bytes.

In answer to the second part of the question, there are four main solutions that I can suggest, all of which have been put forward in earlier editions of Windfall/Apple User.

● Put your code at the top of memory, and precede it with a call to move HIMEM down.

● Put it below your Applesoft program, and precede it with a routine to move the start of program pointer up.

● Put it in the middle of your Applesoft program, and protect it by setting the "next line" pointers to wrap past it (this one is more tricky).

● Keep a "software bank" in the top 16k RAM, and have a permanent $300 routine that copies in the one you want to run, then deletes it after use.

The first method is the most straightforward.

```
0300- 20 B7 00   JSR   $00B7
0303- C9 2C      CMP   £$2C
0305- D0 07      BNE   $030E
0307- 20 B1 00   JSR   $00B1
030A- 20 E3 DF   JSR   $DFE3
030D- 60         RTS
030E- 4C C9 DE   JMP   $DEC9
```

J.P. Lewis

Bracket bother

I HAVE come across a problem on which your advice will be appreciated.

When trying an 80 column card on a Taiwanese Apple II compatible, with 64k, Z-80 and Basic in ROM pressing Ctrl-K always gives the left square bracket ([), both in DOS 3.3 and in CP/M.

Using the CONFIGIO program one can see that CP/M is configured to this.

Now when using that program one deletes this, it happens only on screen, but when one gets back to MBasic or to CP/M, Ctrl-K again gives the left square bracket.

One can change practically all keyboard keys to his desire except Ctrl-K.

Could you please explain why this is happening and suggest any remedy?

An obvious handicap is that one can not use programs like Wordstar which make extensive use of Ctrl-K. – **S. Vouyoucalos, Athens, Greece.**

● If the 80 column board gives [under DOS 3.3 then the firmware on board is generating it, presumably for Pascal and general word processing.

Check whether [is produced by Ctrl-K with and without whatever shift lock mechanism you have.

Max Parrott

Mmagical poke?

HERE'S a strange poke for all you Apple IIe owners. If you have an 80 column card remove it and POKE 49165,0.

80 columns without an 80 column card!

However all is not as it seems and there is a major drawback with this apparently magical 50 quid-saving poke. It won't take you long to find out what it is. – **Frank Toal, Glasgow.**

Rrepeat interval?

I WONDER if there is any way of reducing the time interval before an Apple IIe key starts to repeat?

When "re-typing" it is often faster to multi-strike the key in question. YYYYYYYYours in repetition. – **Tim Shreeve, Norwich, Norfolk.**

● It would have to be a hardware modification, but I would consider seriously before doing it.

Watching many people use an Apple IIe I have come to the conclusion that the time interval could usefully be *lengthened.*

If any readers have tried we'll be glad to hear from them.

Max Parrott

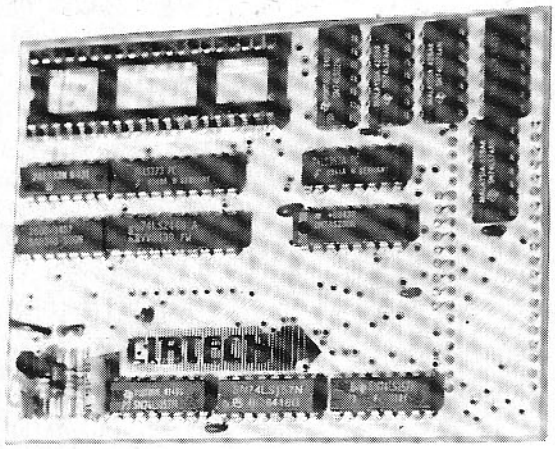
SPEEDING ALONG ON THE ICE

IN the April 1985 issue of Apple User there was a report by Max Parrott "Life in the fast lane?" which covered the SpeeDemon card for the Apple II+ and IIe.

In the last paragraph Max commented that he had been unable to get this card to work with Ice equipment.

We specialise in business network systems based on Apple IIe's and Ice equipment, mainly running under Pascal.

We have several installations where we are using the SpeeDemon to great effect. However I would add that considerable work went into the setting up of the card and the positioning of the other cards in the Apple before a workable solution was developed. – **Anton F. Szklarek, director, Decision Support Systems.**



THE APPLE //c CP/M PLUS SYSTEM!

The //c CP/M PLUS System is the finishing touch for the Apple //c computer. It consists of an adaptor (which fits totally inside the //c) and Digital Research's CP/M PLUS operating system fully configured for the Apple //c. A full bank switched implementation of CP/M PLUS allows use of the whole 128K of RAM in the Apple //c. The adaptor does not affect the normal opertion of the //c and is only activated when a CP/M disk is booted.

CP/M PLUS is an advanced operating system for 8-bit micro's. It features bank switched RAM, increased file sizes and larger disks. It is fully compatible with CP/M 2.2 and provides many features in addition to CP/M 2.2 such as BDOS error trapping, BDOS disk free space functions, BDOS program chaining, RSX modules, LRU (least recently used) sector buffering, date display or directory stamping and a HELP system. The //c implementation also includes several unique features such as the use of the mouse directly on the text screen, an 'invisible' print spooler and a selectable keystroke buffer, as well as supporting the normal printer, disk drive and modem ports. The modem port can also be used to drive a second printer on the system. The port configurations (baud rate, data bits, parity, etc.) can be changed using CP/M PLUS's DEVICE command.

The system will run all standard CP/M programs such as WORDSTAR, dBASE and MULTIPLAN on the //c with no patches or alterations necessary. The adaptor is also fully compatible with Softcard CP/M version 2.23 without any modification.

Also available separately is a CP/M PLUS PROGRAMMERS PACK containing a full set of programming utilities including MAC and RMAC (macro assemblers), SID (symbolic debugger), LINK, LIB, SAVE, HEXCOM, ED, DUMP and XREF programs with a full Digital Research reference manual.

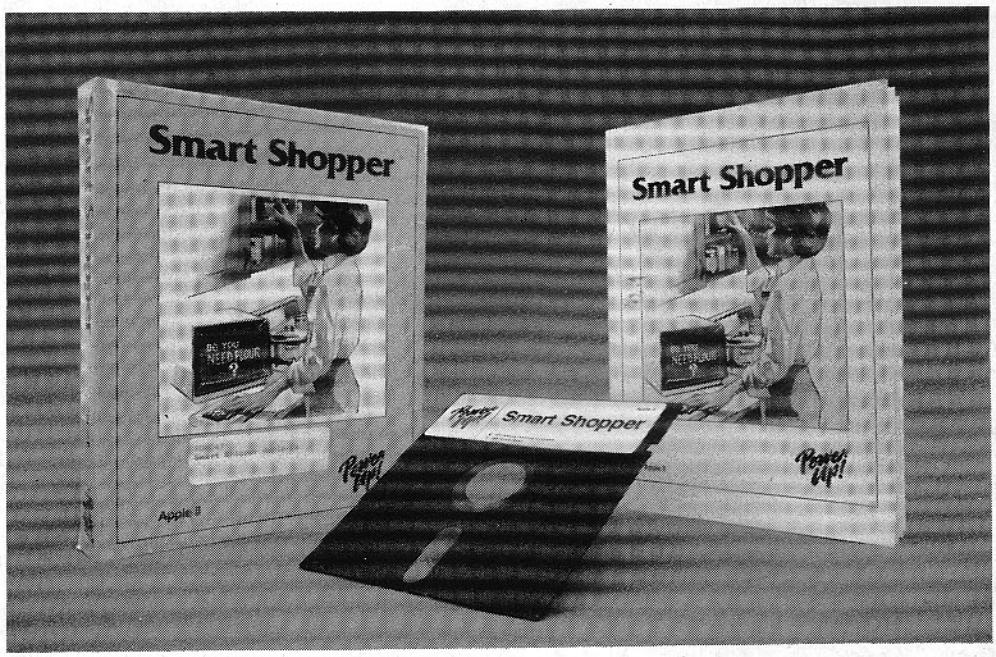
The system comes complete with the 'Standard' set of CP/M utilities plus disk formatting and copy programs.

The price an unbelievable £195-00.

*PAT. PENDING

CIRTECH, Currie Road Industrial Estate, Galashiels, Selkirkshire, Scotland, TD1 2BP Telephone (0896) 57790

The smart shopper picks up her pen



The smart gardener picks the right State

IF your house is like mine there are hundreds of bits of paper lying around. Among the notes from school, the double glazing leaflets and the kids' drawings, there are scraps containing old shopping lists, recipes, directions on where to find a wool shop and the list of seeds I meant to buy three months ago.

Of course the pundits tell us that the home of the future will be paper-free with the micro firmly established as the repository of all knowledge. I've been testing some software which aims to point us in that direction.

I must admit I didn't like **Smart Shopper** from the moment I saw the box. The cover illustration shows a woman looking in the cupboard while the micro screen contains the message "Do you need flour?" written in very large letters.

In fact the actual program never asks you a question explicitly and it uses standard sized type, but it's not improved for all that.

Basically, if you discount the sample shopping list which comes on the disc, the program allows you to type in an enormous list of all your possible purchases. When you want to go shopping, you can tick the ones you want and have a list of your selections printed.

You can prepare up to five different master lists – for example, one for the supermarket, one for the freezer centre, one for the off licence and so forth. However, despite what the sample list seems to suggest, there is no way to subdivide a master list.

This means that you can't skip a section – even if you know you don't want any vegetables, you've still got to step through them, from artichoke to zucchini.

At around £26, I don't think Smart Shopper is something which the smart shopper would buy. It seems like gratuitous use of a micro when a pencil and paper would be easier and quicker. The only possible people who would benefit from it are chronic amnesics or dedicated microphiles.

Title: Smart Shopper
Price: £25.95
Requirements: 48k Apple II family

I VIEW shopping as a chore, rather than a way of life. Gardening, on the other hand, is something I used to enjoy before we bought a clay-pit disguised as suburbia. I came to **Plantin' Pal,** then, with much enthusiasm.

The package aims to help you plan your vegetable garden efficiently – just the thing for wet days, I thought.

```
        Symbols for Vegetables
          (Press P for Print)
Asparagus              Lettuce(head)
Beans(pole)            Lettuce(leaf)
Beans(bush)            Mustard
Beans(lima)            Okra
Beets                  Onion
Broccoli               Parsnips
Brussel Sprouts        Peanuts
Cabbage                Peas
Cantaloupe             Peppers
Carrots                Potato
Cauliflower            Pumpkin
Celery                 Radish
Chard(swiss)           Rhubarb
Chinese cabbage        Spinach
Corn(sweet)            Squash(winter)
Cucumber               Squash(summer)
Eggplant               Sweet Potato
Kale                   Tomato
Kohlrabi               Turnip
Leeks                  Watermelon
      Press SPACE BAR to continue
```

Plantin' Pal's symbol set

The first thing you do is specify the size of your plot and designate which areas are unplantable. You then choose the vegetables you want to grow and whether you want them purely for fresh use or for storage too.

You are asked for the number of people that you're growing for and the program provides a length of run for the vegetable on that basis. On my first try, this resulted in the complete plot being dedicated to the humble runner bean. However, the program offers the option to reduce the length of run.

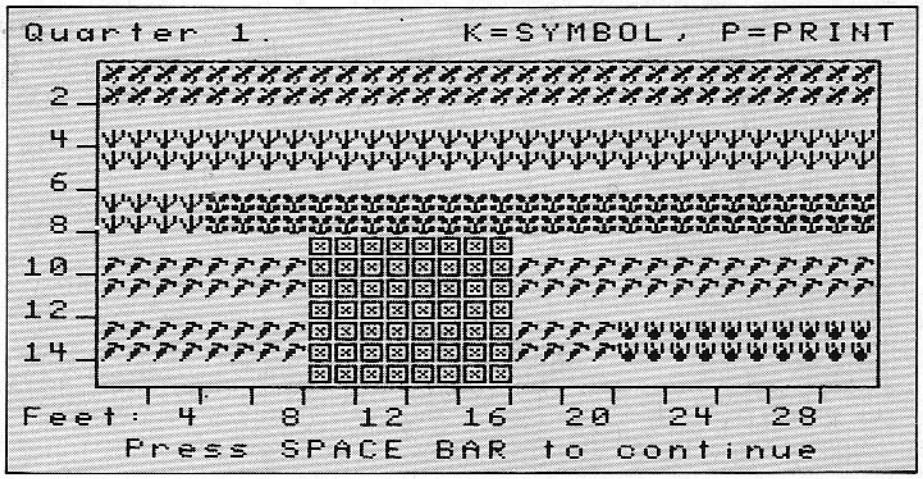
The disc also contains a database on the vegetables so you can see the growing tips for the vegetables you intend to plant. These give watering and

fertiliser requirements and a list of the diseases and pests to which each vegetable is prone.

The database only contains half of the information you need – the rest is contained in a 92-page manual. This seems a bit pointless, because you would need to write down the information displayed on the screen anyway.

The garden is planned for you in 20in "wide rows". However, despite the usual American penchant for being bigger, 'wide' in this case is considerably narrower than the 4ft recommended by the Gardeners' World team.

The program will also give you a planting schedule for your region . . . as long as you live in the USA. For non-Americans, the suggestion is that you enter the State most comparable with your own area. However, without resorting to a mass of physical geography texts, I can't say which state Cheshire most resembles.



Plantin' Pal lays out your garden in neat rows

All in all, then, Plantin' Pal would seem to be another one of those packages which I would trade for a decent book on the subject.

Title: Plantin' Pal
Price: £33.95
Requirements: 48k Apple II family

Cookery – menu driven, naturally

WITH these two experiences behind me, I came to the **Micro Cookbook** with little hope. I really enjoy lavishly illustrated cookery books and have previously reviewed an awful cookery program for the BBC Micro.

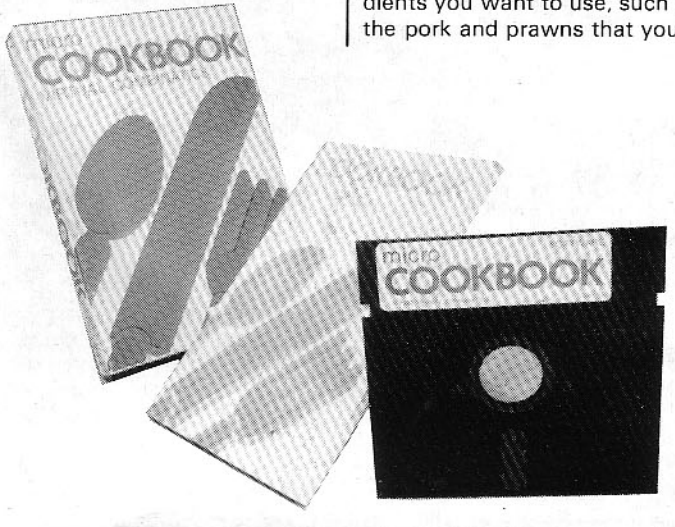
The first things I encountered on opening the box were two leaflets detailing the extent of malnutrition and hunger in the world today. This provides a wonderful perspective to those of us fortunate enough to be sat, well-fed, in front of an expensive micro playing with cookery software.

Virtual Combinatics go one step further than simply enclosing the leaflets. They will donate 50 cents to the Interfaith Hunger Appeal for every registration card returned. It's not quite on the Band-Aid scale, but the principle is a good one.

I was surprised to find that the software was quite good too, despite my previous misgivings.

The package allows you to do just about everything you'd want to do, with the possible exception of browsing through pictures. As you'd expect from a cookery program, it's *menu* driven and quite easy to use.

You can specify the ingredients you want to use, such as the pork and prawns that you'd like to use before they go 'off'. Or you can specify a type of cookery, like Mexican or Jewish. Or you can specify the course, like entrée or dessert. You can even specify combinations of these.



It was here I hit a snag because I asked to see the Jewish poultry dishes that used chicken. The program couldn't find any, but when I looked through the Jewish category alone I came across Honeyed Chicken. Was this the economical Polish version, I thought?

The answer lay – well, it would, wouldn't it? – in the fact that Honeyed Chicken was classed as an entrée and not as poultry. However it was very simple to add the extra classification to the recipe.

There are plenty of other useful functions too, like the program's ability to adjust the ingredient quantities for a given number of people, print a recipe and even compile a shopping list.

The package also contains a 'Terminology, Measurements and Reference' section containing much useful information on cooking terms, ingredient preparation, calorie and nutrition guidelines, measurement equivalence tables and so forth.

The package comes on a double-sided "flippy" with the main program on one side and the basic recipes on the other. It can be supplemented with several "additional chapters" – at extra cost, of course – on such topics as microwave cooking, special diets, desserts and wok cookery.

Of the three packages I've mentioned, it's the Micro Cookbook that I would bother to use again. If I had time, it would be nice to enter all my own recipes and get rid of all the bits of paper. Of course, to be really complete I'd have to enter all the recipes from all our cookery books too.

All three packages can be found in the Power Up catalogue published jointly by P & P Micro Distributors and the Software Publishing Corporation, and I'm grateful to P & P for lending the review copies.

Even if the home of the future *will* be paper-free, I don't think I'll get rid of my books just yet.

Denise McKnight

Title: Micro Cookbook
Price: £33.95 (+ £10.95 each for additional chapters)
Requirements: Apple IIe and 80 column card