A Database Publication

appleuser

Vol. 4 No. 8. August 1984 £1

-the new name for Windfall

How to use your Apple to talk to the outside world

Hello there!

Getting to grips with the Pascal assembler

Easier text handling by machine code

Howzat!

Multiplan scores at calculating batting and bowling averages

REVIEWS

- Voice Input Module
- Codewriter IIe
- ShortCuts
- Ramdrive IIe
- Superfile
- Sage Integrated Accounting
- Scribe

apple user
Volume 4
Number 8
August 1984

Applications

● How an Apple has helped run a haulage company. 59

Books

● Hugh Dobbs casts a critical eye at a self-teaching guide to data-file programming. 62

CAD

● David Haughton has a look at Scribe, a 3-D CAD package for the Apple IIe. 54

CP/M

● Neville Ash reviews a single-disc accounting system for the Apple II. 32

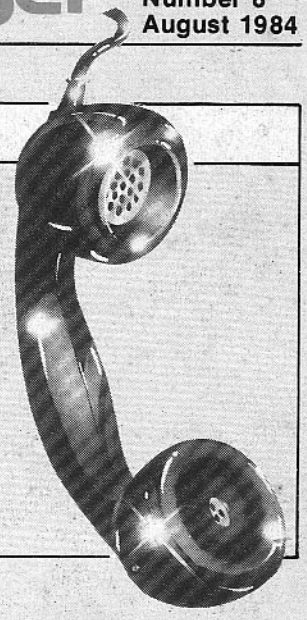
● Pitfalls of the Superfile database system are explored by Roger Heron. 57

Feedback

● Your letters requesting help on the Apple IIe and Epson printer, solving TK! Solver powers using Visicalc, requesting a teletext adaptor for the Apple II, more on Epson codes and how to enter a full string length of 255 characters from the keyboard. 66

Communications

● Peter Tootill describes the Bulletin Board system and Quentin Reidford continues his series with an explanation of some of the hardware requirements. Four pages aimed at getting Apple users communicating. 39



Games

● Thoughts on desirable features of games plus reviews of Plasmania (an arcade version of the Fantastic Voyage film), Bouncing Kamungas (a contender for Silly Game of the Year), and Pinball Construction Set (for DIY pinball addicts). 17

Graphics

● Part VI of ' Peter Gorry's regular "build your own graphics package" section. 26

Lisa

● Charles Fairfax and Geraldine Howe show how Lisa can streamline meetings and reduce paperwork. 29

New products

● Summary of all the latest products on the Apple scene. 69

News

● New head of operations for Apple UK, how to learn American football, Apples helping in medical research, how Taiwan is dealing with the copycats, and we take a creep around the atomic energy labs. 9

Pascal

● P.H.P. Harris shows how to make the most of the Pascal assembler. 44

Reviews

● Don Morrison tries out a popular program generator. 36

● A Ramdrive for the Apple IIe gets the Peter Gorry treatment. 51

● Max Parrott takes some short cuts with a utility package. 52

Speech

● Martin Keeson tests out the Voice Input Module. 24

Spreadsheet

● An unusual use of Multiplan helps John and Jonathan Nixon keep track of some sporting averages. 34

Apple User Managing Editor **Derek Meakin** *Production Editor* **Peter Glover** *Features Editor* **Cliff McKnight**
Technical Editor **Max Parrott** *Design* **Heather Sheldrick** *Editor-in-Chief Database Publications* **Peter Brameld**
Marketing Manager **Susan Lipman** *Advertisement Manager* **John Riding** *Sales* **Peter Nowell**

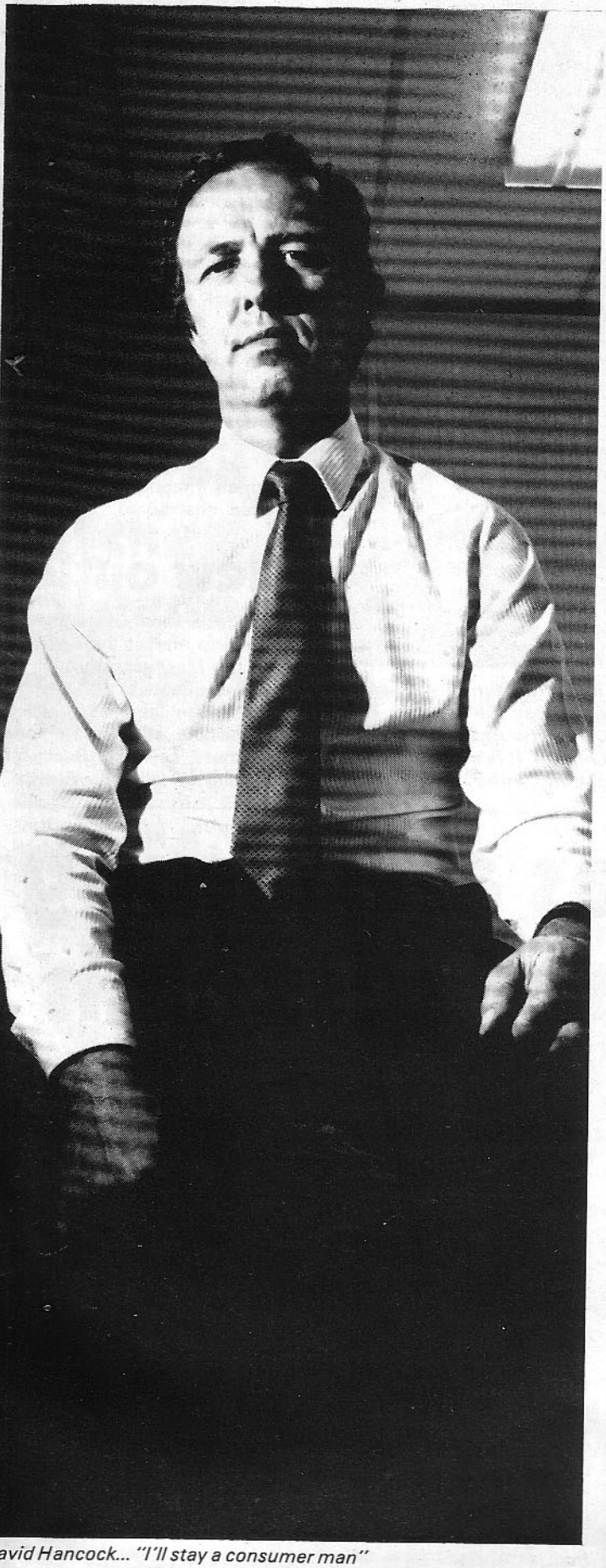
Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising) 061-480 0171 (Subscriptions) Telex: 667664 SHARET G. Prestel: 614568383.

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

Trade distribution in the UK and overseas: Contact Steve Fletcher, Circulation Manager of Database Publications at the above address or telephone him on 0293 26945.

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)
£18 Europe
£15 USA (surface)
£25 USA (airmail)
£15 Rest of world (surface)
£30 Rest of world (airmail)



David Hancock... "I'll stay a consumer man"

New Apple operations chief brings Cola-style sales flair

APPLE UK has appointed as its head of operations a high powered marketing specialist who readily admits his 13-year-old son knows more about computers than he does.

Not that David Hancock sees that as being any disadvantage to becoming general manager in the recent corporate shake-up. "I do not intend to become a technocrat", he told *Apple User*, "for I will stay a consumer man and in that way remain sympathetic to consumers".

David Hancock has been recruited from Gillette where he has at various times been in charge of the company's blade and razor products as well as the PaperMate pen subsidiary.

The move is in line with Apple president John Sculley's intention of injecting the company with all the marketing flair of the Pepsi Colas of this world.

Hancock has been brought in specifically because he is a man who fits this mould rather than someone with in-depth knowledge of the computer industry.

And no one could be more pleased with the move than the new general manager.

"All of a sudden I see a company that wants to do what I want to do – that is look after the consumer", says David Hancock.

For he has not been enamoured with the computer industry's approach to marketing to date.

"The trouble so far has been that the consumer has been left to think 'My God, I've not been allowed to join in'.

"For there is a terrible fear of computers – the fear to touch, the fear to be made to look like an idiot. That's why a whole generation gap has been created between father and son. There's a great deal of discontented parents out there – and I was one of them.

"Now with Apple all that's going out of the window. My job

Peter Cobb moving on

PETER Cobb, Apple's managing director, is reported to be staying on until the new team has time to establish itself. After that he is likely to move out to a new post in Apple's ever-expanding empire.

"I helped to set up Apple in Britain and that's what I enjoy most – the start up operation", he said recently.

"Once the job is done I like to move on – and besides you can only sit in the Apple seat for so long unless you want to become the most likely case for a coronary this side of the Thames".

AN Apple for the teacher. That's the tune increasingly being played in training centres across the country. Here is one of three Zynar Plan 4000 networks – which uses Apples – recently installed in London. It is at the Haringey Information Technology Education Centre.

From Page 9

is to break down the barriers between the consumers and the computers.

"From here on in people will no longer be made to feel stupid when they look at Apple computers. Our message will be that they don't need to know about ROMs and RAMs to take advantage of an Apple".

David Hancock believes that Apple will make dramatic inroads into the PC market with this consumer friendly strategy.

"Only the surface of the market has been scratched so far", he insists. "If you look at it on a scale of one to ten, the PC market is at about .3".

As part of the top level shuffle, Stuart Bagshaw is coming in as sales manager from Systems International where he was involved with managing an independent dealer network.

On the marketing side Apple has a void to fill since marketing director Keith Hall left earlier this year. This is to be filled by Bob Kissach, who has been taking care of Apple's marketing on a part time basis for the last three months. He comes to Apple UK from the company's European headquarters in Paris.

Soccer American style

AMERICAN football is now being taught by computer. A Texas company, Sterling Swift, has launched what is claimed as the first software package to teach the basics of a sport.

The first in a five-part series, "50 Defence vs. Run" costs $100 and runs on an Apple II.

The software helps players and those interested in American football to grasp the fundamentals of each team position, it is claimed.

The program combines tutorials, graphics and an automated chalkboard.

New outlet

CURRYS Micro-Systems has started to market the Apple IIc and the Macintosh through its Micro-C outlets.

"We see them as products that should sell well in local branches", says Jim Reed, the company's marketing director.

Apples are aiding medical research

A MAJOR medical research experiment to aid victims of spinal chord injuries using Apple computers has been launched.

Four subjects are currently undergoing tests arranged by the Medical Research Council at Northwick Park Hospital, Harrow, on machines controlled by two Apples in 15 minute sessions, five days a week.

One of the volunteers is PC Philip Olds, the Metropolitan Police traffic constable who was shot when he tackled two gunmen as they fled from a hold-up.

The experiment is being sponsored by the Daily Mail Fund for Victims of Spinal Chord Injuries, with the computers provided by Apple UK.

It is under the direction of Dr Jerrold Petrofsky, the leading expert in the field, from Wright State University, Dayton, Ohio.

PC Olds was a patient of Dr Petrofsky for most of last year and in December stood up on 15 occasions and took more than 50 steps.

Once the preliminary UK tests have been carried out the experiment will continue at St Vincent's Orthopaedic Hospital at Pinner, Middlesex.

The aim is to test Dr Petrofsky's exercise machines under research conditions.

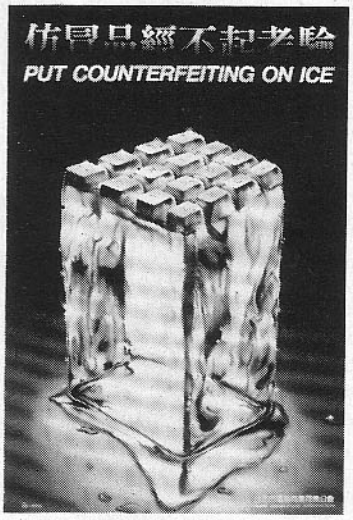
Subjects who have used the machines in America report feeling generally better, suffering fewer colds and in some cases there has been a partial return of sensation to paralysed legs.

Transfer package

IF you've ever wanted to send files from your Apple II or IIe to your less fortunate colleagues who use Sirius, Apricot or IBM PCs, now's your chance.

ACT Pulsar has announced a new file transfer package which will allow you to do just that.

The communication will be one-sided though, because the package only allows transfer from an Apple.



Exit the Taiwan copycat Apples

A DECLARATION of war against counterfeit products by the organisers of the Computex '84 show in Taiwan almost eliminated the usual crop of copy Apples being displayed.

Such was the impact of the message that only a few of the Apple lookalikes – for which Taiwan is famous – slipped through the security net to appear on stands.

In fact the show's organisers went out on a limb by placing a strict embargo on counterfeit products of any kind appearing during the seven day event in Taipei.

To hammer home the ban the show's sponsors ran a competition for the best essay or poster on the subject of why commercial counterfeiting is a dead end.

The winning poster – which won its designer $500 – is shown above.

A total of 145 firms sold an estimated $7,000,000 worth of micros, peripherals, software and accessories to more than 90,000 visitors to the show.

THE Taiwan government's anti-counterfeiting campaign is really beginning to bite.

Pirates have suffered the shock of seeing six local businessmen serving prison sentences for producing rip-off versions of Apple computers called "Orange" and "Golden Apple".

Aussie action, too

NEW copyright laws to provide protection for software are on their way in Australia – thanks to pressure from Apple.

A campaign has been waged Down Under since Apple brought an action against Computer Edge claiming copyright infringement.

After a Federal Court ruling declared software unprotected under Australian law, this was later overturned.

In the meantime the Australian computer industry has been lobbying for a change in existing laws. These have since been approved by Parliament and are now before the Attorney General.

Unfortunately for Apple, its current case is unlikely to be helped by the new laws. For a spokesman for the Attorney General has pointed out that the legislation will not be retrospective.

The Australian outback is proving fertile territory for Apple. The computer-conscious Department of Education in the Northern Territory has been installing Apple equipment in all its secondary schools.

On trail of the dreaded creep..

EIGHT Apples installed in the engineering metallurgy department at the United Kingdom Atomic Energy Authority's laboratories at Risley, Cheshire, are saving hundreds of hours in assessing the data produced.

One micro is used to analyse and report on creep data from 105 machines, enabling a four-man task to be done easily by one.

Creep is a metallurgical term used to describe the gradual extension or compression of metals when subjected to a constant load stress.

The other Apples are used to carry out and analyse new types of tests on the behaviour of metals in nuclear reactors.

Without the computers the analysis and presentation of data from each test would take hundreds of hours. The Apple takes 30 minutes.

The programs were developed at Risley about 12 months ago, and have been continuously extended and improved.

Risley is an important centre for the testing of reactor metals for strength and durability, producing data for the industry's design engineers.

Apples at work at Risley

Apple cards find a new market

MAKERS of Apple cards might have thought they knew the limits of their marketplace. But Xcalibur Computers have changed all that.

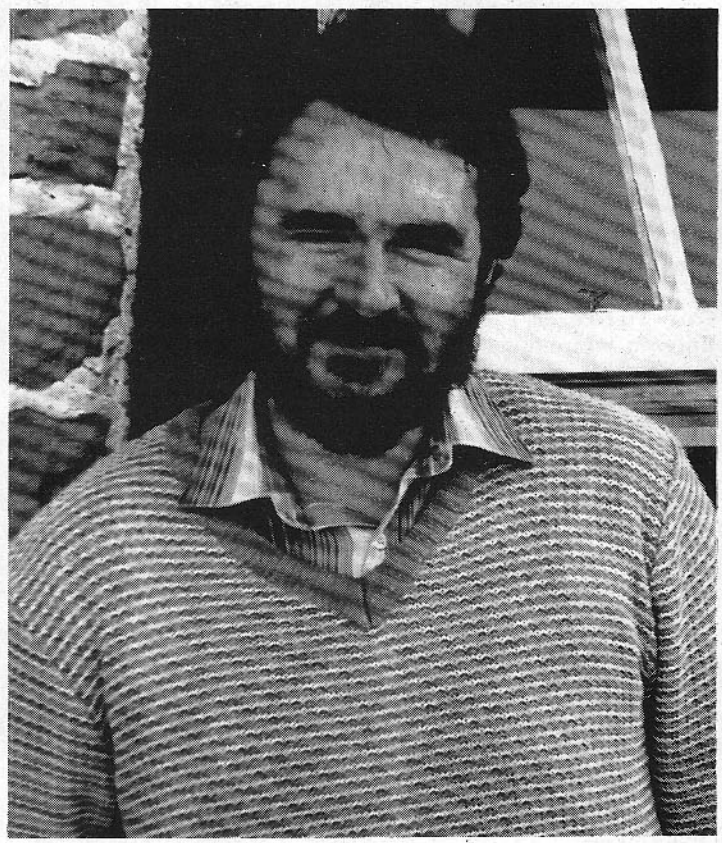
They've developed a backplane for the BBC Micro which allows standard Apple cards to be plugged in.

The input/output addresses of the Apple are mapped into defined addresses on the BBC, which can then drive the card through Basic commands or assembler.

The backplane has five slots and also has 64k of RAM on board.

Although masses of cards are available for the Apple, only experience will tell how successfully each will run through the backplane.

Who knows? It may give a whole new lease of life to the card manufacturers, at a time when they might have thought that the advent of machines like the Apple IIc – which doesn't have room for traditional expansion cards – meant a narrowing of their horizons.



Our Show winner

A NORTH Yorkshire licensee hopes to find more time to play golf now that he's been named as winner of an Apple IIe – the star prize for visitors to Apple '84.

I'll be able to use it for my bookkeeping which will allow me extra time on the links", said 35-years-old John Russell landlord of the Foresters Arms in Carlton-in-Coverdale.

However John is lucky to be walking let alone playing golf.

For some years ago he was seriously injured in a 50 foot fall from the balcony of the Queen's Hotel in Brighton, while he was manager there.

The balustrade collapsed and he plummeted down to land first of all on a parking meter before bouncing off onto the bonnet of his own car.

John Russell spent the next year in hospital paralysed with a broken back. In fact, he was unable to work for more than four years due to his terrible injuries.

Fortunately for the hotelier, the SAS came to his rescue. To be more accurate, it was a former physical training instructor in the undercover regiment who had taken up physiotherapy at Margate Hospital who got John Russell back on his feet again.

"He was a very hard man indeed", he recalls. "And if it hadn't been for him, I wouldn't be walking today.

"Anyhow, my luck seems to have changed since then. For I was able to use my compensation to buy this freehouse – and now I've won a computer as well . . ."

..and IBM men can read Apple discs

JEALOUSY has finally got them somewhere. Owners of the IBM PC and its lookalikes can now read files from Apple discs

Both Appledos 3.3 and Apple CP/M discs can be used in the IBM's disc drive, and files can be transferred to MS-DOS discs. In addition, blank discs can be Apple-formatted and MS-DOS data transferred to them. It's all done with a so-far unnamed file transfer board which Microware is about to introduce. Cost will be £250.

Says Microware: "With an installed base of nearly 2 million Apple computers and almost 1 million IBM PCs, a large world-wide market appears ready for effortless disc-to-disc transfer".

Franklin struggles to survive

A $2.5 million lawsuit by Apple is believed to have been the last straw for Franklin Computers, the troubled US personal computer maker, which has just gone into Chapter 11 bankruptcy.

That is a feature of US bankruptcy law which allows a company to carry on trading but keeps creditors off its back, thus giving it a chance to turn itself round.

In January the company was sued for copyright infringement by Apple, which alleged that Franklin had used Apple software in its Ace computers.

As a result Franklin had to pay damages and stop selling the offending models.

Meanwhile Franklin has laid off 160 people in its struggle to make enough profit to pay its debts and come back from the brink.

Manual for the 11c

WHAT may be the first popular Apple manual to include the IIc has been written by Graham Keeler, a physics lecturer at Salford University.

Keeler, who runs short residential courses at the university for Apple users, has condensed his expertise into a book called "Getting the most from your Apple".

He wrote it on his Apple II, using the Applewriter word processing package.

The book was then typeset straight from floppy disc by publishers Addison Wesley.

Scheduled for publication first in the United States, the manual should be in the UK shops before Christmas.

Time to raise the standard

CLIFF McKNIGHT puts forward a case for establishing quality guidelines for games

WHEN I was a lowly draughtsman, slaving over a hot pencil, everything had to be done to B.S.308. In fact, there is a British Standard to cover most things in industry, and I recently got to wondering what the British Standard Game would look like.

Apart from specifying the minimum number of aliens per level, the recommended points per power pill and the Klingon Constant – otherwise known as the acceleration due to warp factor 10 – what would a standard hope to achieve?

The area which would benefit from standardisation is that of "convenience controls". In many ways, the standard is a *de facto* one defined by existing successful games. Most of the following features can therefore be found in various games. However, I don't know of any game which incorporates all of them, and I know lots which ought to have more.

I'll deal first with arcade games. The single most important feature of any arcade game is a pause facility. There is only one thing worse than the phone ringing as you approach a new all-time high score, and that's approaching it with a full bladder and your legs crossed.

Most of the industry has already realised this, and indeed the Esc key has almost become the standard key for pausing a game.

Where practical, it would be nice to be offered a choice between keyboard and joystick control. In fact, for true arcade games this should be no problem. It only becomes a problem when the game gets complicated – for example in something like Aztec.

However many games which offer joystick control also require key presses at various times, if only to start a new game. Since the advantage of a joystick is that you can sit back with it, it's irritating to keep leaning forward to "press any key to continue".

Once I've opted for keyboard control I like to be able to choose which keys I'll use. Many games offer user-definable keys, and some like Snack Attack generate a different combination each time in order to spread the keyboard wear.

The possibility of keyboard damage means that I wouldn't want to standardise on particular keys. Even so, at one time the industry standard movement keys were A, Z, → and ←, and I dread to think how many aliens have been consigned to the hereafter with a timely thumb on the spacebar.

Many games offer a sound toggle. It's invaluable if you want to sneak a game, although twitching and cursing usually demonstrate to all but the very naive that you're not using Visicalc.

The standard I would like to see, though, is a half-way point too. In many games the sounds are a real part of the fun, but on many occasions it would be nice if the volume could be reduced.

A "Hall of Fame" is incorporated in most games and these can be fun in their own right. In our house we have contests to see who can devise the most creative pseudonym as well as trying to get the highest score. In this respect games which only allow three letters to be input are boring and discriminate against people like me who have only two initials.

The standard, then, should make provision for a reasonable length of name. I would also like to see more games offering the facility to wipe the high-scores clean. When I have hammered a game for review purposes it's nice to wipe the board and give the kids a chance to see their name in lights.

Of course, once you've bashed away for a few games it's handy not to have to start from scratch each time. The choice of starting level should be offered, even if it means foregoing the chance to get in the Hall of Fame as it does in Lode Runner.

The only other feature I'd like to see more use made of in arcade games relates to the question of colour. Some look just as good in monochrome as they do in colour. However many look great in colour but messy in monochrome.

The best ones offer a choice between monochrome and colour displays and tailor the graphics accordingly.

In the same way that arcade games need a pause facility, adventure games need save and restore options. Practically all adventure games have these, but some are better than others.

For example, although most people have two disc drives, most games assume a single drive. Hence saving a game becomes a major exercise rather than a quick precaution when danger looms.

There are two solutions to this problem. One is to ask how many drives the player has. The other is to save the game to the actual master disc.

Although practically every game allows the restoring of a saved game, very few offer the option when you've just been killed off.

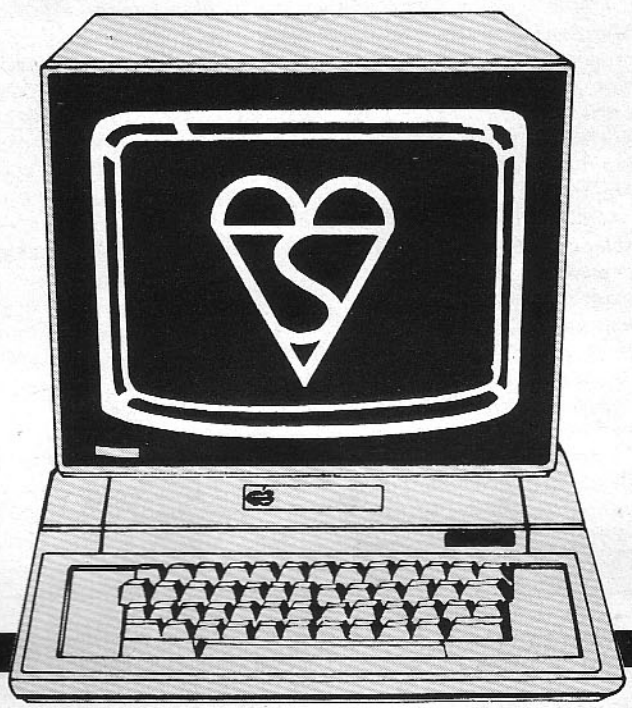
Rather, you are typically offered the chance to start again, often involving tedious introductory sections of the game.

Indeed many of the standard features I would like to see follow from the fact that you are likely to get killed off regularly and have to retrace your steps several times. For example, once I'm sick of the pictures, I like to be able to opt for a text-only mode which is usually quicker.

It is essential that common commands can be abbreviated – say Inventory should only require a single letter. However being told that the program only looks at the first four letters of each word doesn't seem to help me. I find it easier to type whole words rather than half words.

The ability to issue multiple commands can also be a time saver, particularly when you know where you want to get to.

Well, that's my list of requirements for the standard game. Since they concern the process rather than the content, I don't think their adoption would lead to all games becoming boringly similar.



Plasmania – a matter of life and death

IF you ever saw the film "The Fantastic Voyage", you'll know the plot of a rash of games which have recently appeared.

Maybe rash is the wrong word, but clot of games doesn't sound right – people might think I'm talking about myself again.

Perhaps I'd better explain for the benefit of the uninitiated.

You and your submarine have been shrunk to microscopic size and injected into a patient. Your task is to get through the various blood vessels in order to destroy the clot which threatens the patient's life.

En route are a variety of organisms, some of which are good for the patient and some of which aren't.

For example, enzymes are to be blasted since this causes healing properties to be released and improves the chance of success. However, allowing one to pass by has no adverse effect.

In contrast, antibodies and bacteria *MUST* be destroyed because of their harmful effects. Not everything has to be blasted, though. Blood cells and clotlets must be avoided.

Your submarine is perpetually moving forward, presumably going with the blood flow.

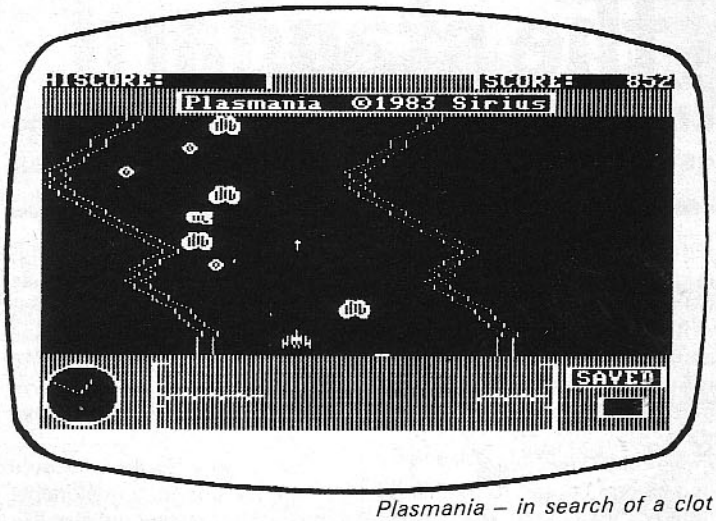
You can go faster if you want and then slow down again, but there is a minimum forward speed which drives you inexorably onward – typically forcing you to crash into something you'd much rather avoid.

The vein takes a rather tortuous path so you must move from side to side in order to avoid the vein walls. It's a bit like the arcade driving games where you have to stay on the road. Bumping into the vein walls releases antibodies which must be destroyed.

Failing to zap the nasties or accidentally zapping the goodies leads to a reduction in the patient's health. This is being constantly monitored and displayed as both an auditory and visual signal.

Additionally, the clot isn't exactly benign so you only have a certain time to reach it.

All these factors go to produce a game not unlike Starblaster and many others, although the requirement to avoid certain things rather than simply blast everything in sight makes for a more complicated game.



Plasmania – in search of a clot

Of course, the Sirius quality is there too, so the game is well implemented with good graphics and some use of sound.

Control is via joystick or keyboard, the game can be paused and the sounds can be toggled off . . . all things we expect from Sirius.

There are three levels available – easy, normal and difficult – and each can be played in normal or extended form. So far I've only managed to save patients on the easy level, and never more than two even then.

One really distinctive feature of Plasmania is that it talks to you. It's not exactly a great conversationalist – all it says is "Sirius presents Plasmania, ha, ha, ha".

However, it produces this sound even if you haven't got a voice card installed. If you've never heard a voice emanating from your Apple it comes as a bit of a shock.

The game was originally called "The Vein Game" and the Atari version is called "Fantastic Voyage", but presumably it was harder to coax either of these from the speaker.

Even so, it opens up some interesting possibilities. Any suggestions on what your Apple should say to you?

Title: Plasmania
Author: David Lubar
Publisher: Sirius Software
Requirements: Apple II/II+/IIe with 48k.

Silly, but fun as well

WHEN it comes to Apple games those funny folk at Penguin Software certainly know what they're talking about. So when they say that Bouncing Kamungas is "among the silliest of games" you'd better believe it.

Bouncing whats? Oh, come on, don't say you don't know

CLIFF'S COLUMN

WHEN the film of Michael Crichton's novel The Terminal Man was shown on Channel 4 recently, The Guardian's television guide reported that the central character had "a minicomputer implanted in his brain". Now ***that's*** *what I call big-headed!*

Strangely enough, although science fiction authors are often seen as 'visionaries' and are commonly thought to extrapolate current science into the hypothetical future, The Terminal Man was one of the few stories which have taken the microcomputer seriously.

In Crichton's novel, The Guardian notwithstanding, the computer was the size of a postage stamp. In contrast to this, most authors have depicted 'Incredibly Big Machines' as the shape of computers to come.

Even HAL, (one jump ahead of IBM – geddit?) in 2001 was a vast structure into which Bowman was able to climb and mess about with the "solid state logic units".

Fourteen years later, Arthur C Clarke wrote 2010 using Wordstar and sent a disc to the publisher.

Has the micro industry developed so fast that science fiction authors can't keep up, or have I been reading the wrong books?

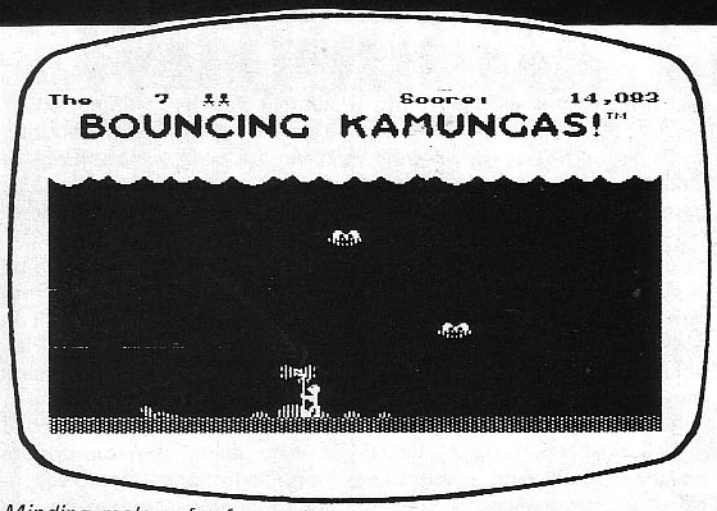
Did they burn themselves out on space technology, plasma drives and force fields at the expense of VLSI, micro-drives and icons . . . or was Douglas Adams really prophetic wi regard to mice?

The Hitch-Hiker's Guide the Galaxy was just on example of a computer gam being produced from a book or more accurately a rad show. Films, too, have produc games like Alien, Microbe an Dark Crystal.

Even War Games seems have spawned a game calle Global Thermonuclear Wa marketed by Starfire Games an including "HAL (tm) speec synthesis"! Did you spot th Apple III in the film Tron?

Not being a visionary myse I'd like to see what thes authors make of computi games when they finally catc up with the micro revolution.

With writers of Jer



Minding melons for fun and profit

what a Kamunga is! Neither did I, but I can now reveal that they are "cute little furballs that rain from the overcast Dakota sky".

Why Dakota? That just happens to be where your melon farm is.

That's right – a melon farm! You play the part of a farmer who plants melon seeds, but unfortunately the Kamungas take great delight in squashing them as they ripen.

They drop down from the sky and bounce all over the place, while you try to defend your produce with a pitchfork.

Being an overcast sky, there's lightning about, so you can't keep your pitchfork up too long. If the Kamungas squash your melons you can plant more, but if they land on your head you lose a life.

Also Dakotan weather snakes appear on the scene. They're not deadly, but if you frighten one all the melons you have harvested are spoiled.

If you manage to successfully harvest nine ripe melons the scene changes as you take them to market in your truck.

Ideally, you'd like to get there as fast as possible, but the Peronies which lie on the road cause the melons to bounce out of the truck.

If you drive over them slowly your melons stay aboard – but the longer you take the lower the price you get.

Once you make it to the market you return to the melon fields and start trying to raise another crop. This time the Kamungas come in increasing numbers, so it gets increasingly harder to protect the melons and let them ripen.

Control is via joystick or keyboard, and there are the usual convenience keys for controlling sounds, pausing the game and restarting from the beginning. The top five scores are saved to the disc.

Bouncing Kamungas certainly is a silly game, but it's also fun to play. Once you've played a game or two, it's not too difficult to get a reasonable score.

By the time you start on your third crop, though, it's quite difficult to keep any melons from the hordes of Kamungas raining from the sky – particularly if you don't want to join the Ben Franklin school for ex-farmers.

Title: Bouncing Kamungas
Author: Thomas Becklund
Publisher: Penguin Software
Requirements: None stated

Pournelle's calibre being involved in the micro scene – his column in Byte magazine makes fascinating reading, even if he rarely mentions Apples – I'm surprised I haven't encountered anything yet.

I don't believe that the space travellers of the future will really be limited to playing computers at chess, checkers and pantominoes.

It's a bit like suggesting that the Stainless Steel Rat uses Visicalc to ask "what if?" questions about the economy of the planet whose bank he is about to rob.

Perhaps science fiction authors are visionaries ... perhaps the micro revolution will have died out by 2001. Somehow I doubt it.

Make your own vidieo pinball with the PCS

TWO years ago an acclaimed pinball game, Raster Blaster, was released by one of America's leading Apple programmers, Bill Budge.

Now Bill has gone one stage further by releasing what he describes as a "software toy", the Pinball Construction Set, used, as its name implies to build pinball games for the Apple II.

The PCS comes as a protected single sided disc with four demonstration pinball tables in a short but informative manual.

Note that there is no in-built game present on booting up the PCS, which makes it easier to get started.

When using the PCS, an on-screen "hand" acts as a cursor, controlled by your joystick. The joystick button causes the hand to grasp or select whatever it is touching, in the same way as the mouse-controlled arrow on Apple's Lisa.

The first step in designing a pinball table is to draw a rough sketch of where everything is to go. Drawing the outline of the table itself is done by dragging a polygon onto the screen, colouring it in, then manipulating it to the shape you desire by grasping, or adding, knobs and pulling it to the required position.

This is possibly the weakest part of the PCS. It is awkward trying to grasp the knobs and pretty near impossible to move them to exactly the correct place. An easier method would be to draw the shape freehand and fill it in later.

Polygons are filled in instantly by selecting the paintbrush, touching the desired colour and then touching the polygon.

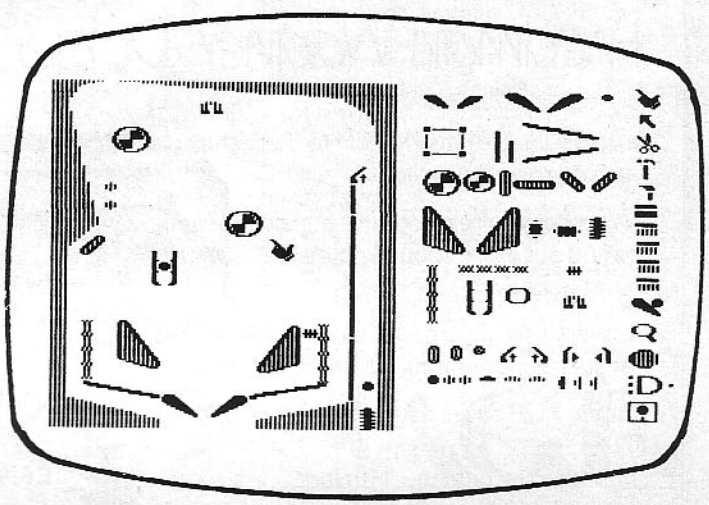
Knobs are added by selecting the hammer, positioning it halfway between two existing knobs, and pressing the joystick button. The knob is dragged by holding the button in and moving the joystick.

Knobs are removed by selecting the scissors and touching the joystick button when the tip of the scissors is near the unwanted knob.

Fine detail may be painted in by selecting the magnifying glass with the hand pointer. A small portion of the pinball table will then be shown magnified on the lower right hand portion of the screen.

Once again, colour is selected by touching the desired paintpot with the brush.

The paragraph on painting requires to be read at least twice



It's not rude to point in PCS

to ensure you understand how the program is going to react when you start dragging your paintbrush all over the place.

The magnifying glass option can also be used to add your own title to the upper right hand corner of the screen.

The PCS does not add anything to indicate that it was used in the game's creation.

Once you have drawn the basic outline, parts from the pinball library may be added by touching them with the hand and dragging them to the desired position on the table.

Up to 128 pieces may be on the board at once and they can be placed wherever you want, even inside other objects – although it seems that if you paint over them with the paintbrush PCS ignores them.

Parts available include two sizes of right and left flippers, a ball, two sizes of round bumpers, four rectangular bumpers, right and left slingshots, two knockers and a launcher.

All of the bumpers have equal kick strength – which is determined by the world settings – directed at right angles to the surface tangent at the point of collision with the ball.

Slingshots and knockers are similar to bumpers but only react when hit on certain points.

The launcher will kick the ball if the joystick button is pressed. Its kick strength is determined by the position of the joystick lever.

In addition, there are two drop target sets, which are really four pieces grouped together. When all four parts of a drop target set are hit by balls it turns on and returns to its initial state.

There is also a ball hopper which catches up to two balls and releases them when a third comes along.

A ball disintegrator can be included which will eat any balls coming its way, while the spinner will spin when a ball hits it and the magnet will hold onto the ball for a second or two.

To direct the balls in a certain direction there are lanes and gates, while rollover lights and targets add the finishing touch to the table.

After setting up the table you define the score, sounds and bonuses. Pieces come with predefined scores and sounds, but they may be changed at will.

Bonuses are defined by selecting an AND gate, depicted on the screen, and using the screwdriver to select the pieces which are conditions for the highlighted AND gate.

Up to three pieces may be selected as conditions for a single gate. The bonus and sound for that AND gate will then be processed only when all of the selected conditions for the gate are on.

Conditions may be removed by touching the offending conditional piece with the pliers.

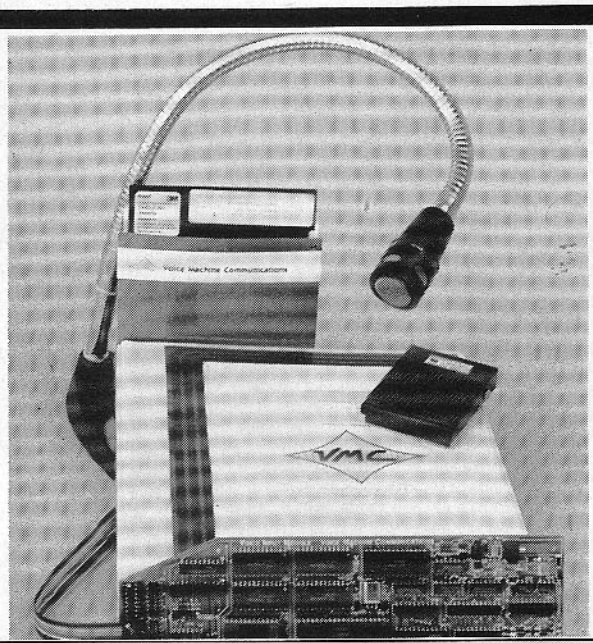
The Pinball Construction Set has four sliding switches which control "gravity", "time", "kick" and "elasticity". On booting these are set for the physics of the Earth, but they can be changed by selecting the World icon.

Your pinball game may be tested at any time in the proceedings by placing a ball on the table and selecting the Play icon. Play is quitted by pressing the Esc key.

Disc operations are carried out by using the appropriate icon. Operations catered for include loading and saving the game as data, changing the slot and drive and making the final game. This last option creates a machine code program which occupies 121 sectors and caters for up to four players.

Playing video pinball was certainly a pleasant change from doing anything else on an Apple. Apart from the few, relatively minor, niggles I have mentioned, PCS is an excellent package and well worth the money. **Philip Colmer**

A word in your Apple's ear...



Voice Input Module reviewed by MARTIN KEESON

THE Voice Input Module (VIM) allows Apple users to control their micro by speech.

Made by Voice Machine Communications, the module recognises isolated spoken words or short phrases and sends easily predefined data to the Apple using the utility software provided.

Any existing Apple applications software can use VIM without modification. Voice and keyboard data can be input concurrently.

The package consists of interface card, microphone (Shure model VR300), footswitch, adapter board for the Apple IIe, System Master disc, Standalone demo disc and user's manual.

A head-worn microphone and a programmer's manual are optional.

On the board is a 16 channel audio spectrum analyser, a 6803 microprocessor, 8k RAM (used to store voice patterns and vocabulary) and 4k ROM (recognition software). The microphone is unidirectional and can be switched off and on with the footswitch.

Installation in the Apple II is easy. The ribbon cable from the keyboard is disconnected from the Apple main board and connected to the VIM. Another ribbon cable connects the VIM and the main board.

Installation in the Apple IIe is more complex and an adapter board is needed. Three ICs have to be removed from the main board and plugged into this adapter board. This is then plugged into the empty sockets on the main board.

A disadvantage of this approach is that the two boards are very close. The ROM is so close to the adapter board that it could cause heating problems.

It seems strange that VMC did not develop a better adapter board, but this could be because it was designed for use in the American version of the Apple IIe which is slightly different from the European versions.

The System Master disc contains programs to create, train, test and recognise vocabularies (AVIM), to merge existing vocabularies (MERGE), for quick change of vocabulary (CHANGE VOCABULARY), and to test the board (VIM TEST).

It also has several ready-to-use vocabularies and two demonstration games – a maze and a blackjack program.

Building a vocabulary is done by defining the word or phrase to be recognised by the VIM and then defining the response of the VIM to the Apple.

For example:

Spoken word	*Input*
hello	**hello**
catalog	**CATALOG<0D>**
backspace	**<08>**
start	**RUN DEMO<0D>**

As you see, the input can contain anything which can normally be inputted by the keyboard. The hexadecimal digits between brackets indicate control characters.

A given vocabulary may be edited so that the VIM responses can be adapted without having to change the voice patterns.

When you are satisfied with a vocabulary you can save it on disc. You can now start training with a vocabulary and save your voice patterns on disc too.

The test feature allows testing of a given vocabulary and voice patterns which makes optimising easy.

The VUP is a user friendly menu-driven program written in Applesoft Basic, so adaptation to your own needs is quiet simple.

Unfortunately the program is another example of bad programming without structure and not many comments. It is quite a puzzle to find out how everything is done.

Existing vocabularies with their voice patterns can be merged by using the MERGE program, saving double work and training time.

To load a vocabulary to the VIM, the CHANGE VOCABULARY program is used which takes about 10 seconds from standard Apple disc drives. It is possible to load by spoken commands, giving you complete voice control of your Apple.

The vocabularies mentioned are for use with Wordstar, Apple Pie, Magic Window, Visicalc and List Handler. Just train these vocabularies and run one of these software packages by voice.

Another vocabulary contains a complete set of Applesoft statements which allow the user to program by speech.

There are few other vocabularies containing digits, DOS commands, spelling words and commands to control the maze and blackjack games.

On the demo disc there is an autorunning program to show how to work with the AVIM program. It is a good addition to the user's manual.

The standard software allows you play with the VIM, which is enough when using software packages such as Wordstar.

When you want to use VIM together with specially developed software there are more possibilities. But it is necessary to buy the programmer's manual which details the use of the VIM under machine language control.

Unfortunately this manual contains a lot of mistakes and even wrong codes, so take care.

Once under machine language control it is possible to mask parts of the vocabulary, increasing substantially the accuracy of recognition.

Another possibility is to get information about the word nearest to the recognised word.

Product: Voice Input Module.
Description: Converts spoken words to commands or data for your application program.
Price: About £900.
Distributor: Cascade Graphics Development, 185 Lower Richmond Road, Richmond, Surrey.

Full speed ahead for text handling by machine code

PETER GORRY expands his series on graphics routines

IN June's column I provided a shape table and Basic subroutine as part of the *Apple User* graphics library, to put a variety of text formats on the hi-res screen. Those of you who have typed them in will have found that Basic text handling is rather slow.

So here is a machine code program that takes over most of the tasks performed by the Basic routine. If you can face the prospect of typing in yet more hexadecimal code you'll find the resulting speed increase well worth the effort.

The success of such a program requires the ability to use DRAW, XDRAW, SCALE, ROT and HCOLOR from machine code. These are complicated machine code routines deep in the heart of Applesoft, but providing you know how to set them up they can be called from a machine code program quite easily.

So before getting down to the program itself I want to spend a little time explaining how this can be achieved in practice. If you're not a machine code programmer you can skip the next bit – you don't need to understand the code in order to use it!

Not all Applesoft routines can be called from machine code, since several of them contain parts that check for the correct syntax in a Basic program.

For instance, the normal entry points for DRAW and XDRAW are $F769 and $F76F, but these will fail if called from machine code. The task is to find which routines we can make use of and how they are set up.

Figure I shows the sequence of events needed to DRAW or XDRAW a shape on the hi-res screen. It assumes you have already loaded a shape table into memory and selected the hi-res screen.

The first step is to set $E8 and $E9 to point to the start of the table – this is identical to poking 232 and 233 from Basic.

Next we must set the SCALE, ROT and HCOLOR values we require. The first two are easy – just store the required number in $E7 and $F9 respectively.

Setting the HCOLOR value is more tricky, and there are two ways of doing it. We can make use of the Applesoft routine as follows:

```
LDX     load with 0-7 to set colour
JSR $F6EC
```

Alternatively you can store a value directly at $E4 – but it must be a "colour mask", not the 0-7 HCOLOR value. The mask values for the 0-7 colours are:

BLACK	**$00**
GREEN	**$2A**
VIOLET	**$55**
WHITE	**$7F**
BLACK	**$80**
ORANGE	**$AA**
BLUE	**$D5**
WHITE	**$FF**

The next task – to set which shape out of the table to use – is easily done using DRAWPNT2, located at $F730. The X register must contain the shape number before calling it.

The screen location values (X 0-279, Y 0-191) must be put into the X, Y, A registers as indicated before calling HPOSN at $F411. This routine calculates the screen memory address corresponding to the X, Y value given.

Finally we must load the ROT value into the A register and call DRAW1 ($F605) or XDRAW1 ($F661) to achieve our desired result. That's all there is to it.

Now to incorporate the routine into the library. Listing I is a hexadecimal dump of the code situated at $7000. It should be entered just as you did for the shape table:

```
]CALL-151
*7000:85 FF 86 EB 84 35 A9 00
*7008:F0 19 00 60 64 00 64 00
"
"
*71C8:E8 A5 FF A4 35 A6 EB 60
*BSAVE TEXT.BIN, A$7000,L$1D0
```

A Ctrl-C returns you to Basic. The code is relocatable, so it can be put anywhere in memory for use – just as the shape table can be.

This offers maximum flexibility, but results in the code being more complex and a little bit longer than for a fixed set of locations.

Listing II is the assembly version for those who want to study the code in detail.

For the machine code programmers among you I make use of the last 15 bytes of the input buffer ($2F0–2FF) as a temporary store – this is a useful piece of absolutely located memory which can cut down the indirect indexing required otherwise and won't cause any problem unless your code uses the INPUT routines.

The Basic routine, supplied in Listing III, will find where you have put the table and code automatically – but it is essential that you call them "CHAR TABLE" and "TEXT .BIN" respectively to ensure this works.

Following our usual procedure, the code is to be loaded below hi-res page 1 and you will have to tell the TABLE LOADER routine where to put it:

```
42440 ZT(10) = 3 :REM THREE TABLES
42470 ZT(3) = 3456: ZT$(3) = "TEXT.BIN"
```

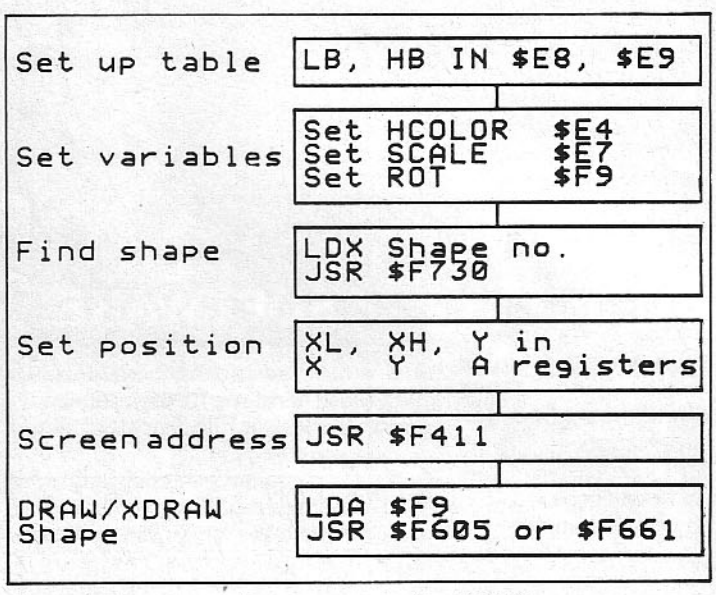


Figure I: Sequence of events needed to DRAW or XDRAW a shape on the hi-res screen

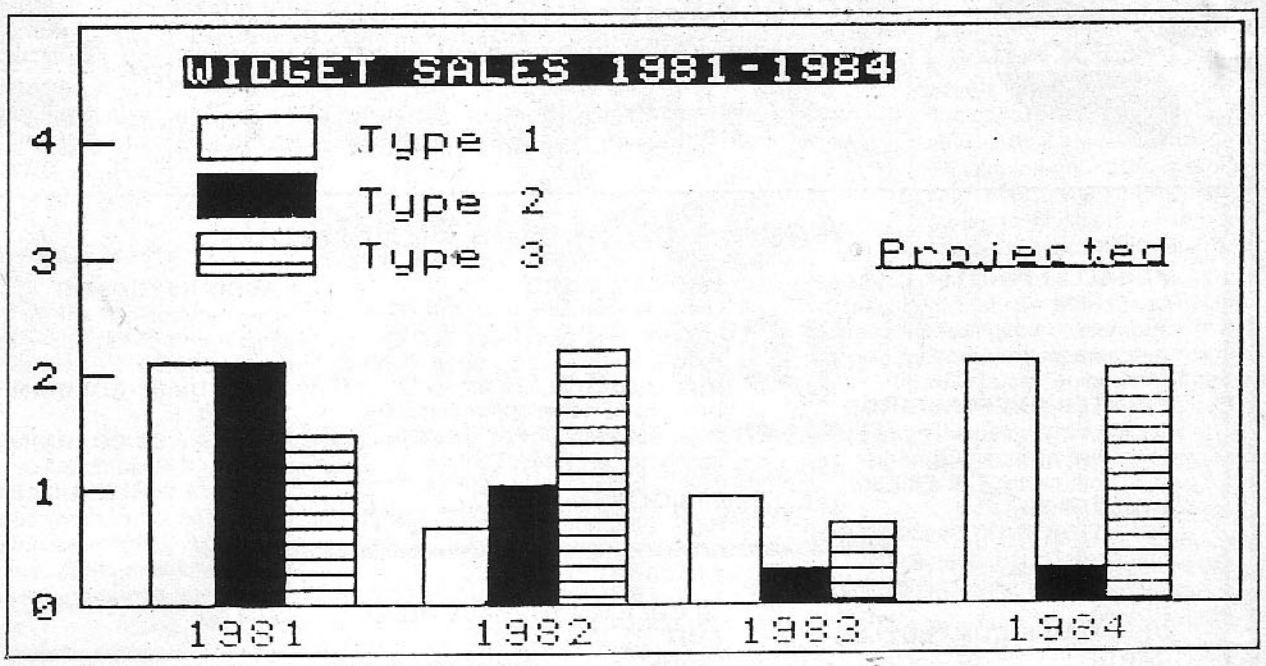


Figure II: Example of text combined with graphics

The job of the machine code is to handle the positioning and text format of a single letter. It is designed to take the place of the normal Character Output routine which handles single letter output.

Memory locations 55 and 56 contain the address of this routine – so all we have to do is put the address of our machine code program there instead.

In fact this can be a dangerous thing to play with, since DOS also changes these locations, and horrible results can occur. The attraction of doing it this way is that the PRINT command can be used to perform the string "unpacking" for you.

The Basic routine in Listing III performs the delicate task of altering the Character Output routine addresses in lines 43720-43750.

The rest of this routine is involved in initialising all the addresses first time round and telling the code where, and in what mode, you want the text to be.

The routine should be typed in after the routines from June and you can use the same example program as then – just change every GOSUB 42800 to GOSUB 43470.

The text type, orientation, windowing etc, is set via the ZS array, just as before. The string, too, is held in ZS$ and plotted at ZX, ZY (user co-ordinates) so there's nothing new to learn there. The only difference is that the routine can be entered in two places. One place is much faster than the other.

The machine code routine has to know what values you have set in the ZS array, so the Basic routine copies these values into a table in the machine code. This takes a little while. If you haven't altered the ZS array there's no need to do this step, and it can be bypassed.

GOSUB 43400 Plot string at ZX, ZY.

GOSUB 43470 Copy ZS array first then plot string.

There's no harm in using the second option every time – it's just a bit slower.

I realise just how dull it is typing in all this hexadecimal code (I had to write it!) so I promise to stay away from it for a while.

However once you start producing those graphs and pictures with impressive annotation it will all have seemed worthwhile.

THIS monthly series of articles is designed to produce a set of integrated graphical routines that form a flexible and easily expandable package. The complete package can be added to any program to produce hi-resolution displays with minimal effort.

The first routines in the Apple User graphics library were published in our February, 1984, issue.

```
7000- 85 FF 86 EB 84 35 A9 00  70E8- 7F 85 E4 A2 60 20 30 F7
7008- F0 19 00 60 64 00 64 00  70F0- AE F2 02 AC F3 02 AD F4
7010- 00 00 00 00 00 00 00 03  70F8- 02 20 11 F4 A5 F9 20 05
7018- 00 03 14 01 BD 03 00 03  7100- F6 A5 E4 49 7F 85 E4 A5
7020- 14 01 BD A9 0A 85 CE A9  7108- FF 29 7F 38 E9 1F C9 01
7028- 70 85 CF A5 E7 8D FD 02  7110- 90 49 C9 61 B0 45 AE F9
7030- A5 F9 8D FE 02 A5 E4 8D  7118- 02 F0 0A C9 1F 90 06 C9
7038- FF 02 A2 00 A0 0A B1 CE  7120- 3D B0 02 69 20 AA 20 30
7040- F0 04 A0 13 D0 02 A0 0D  7128- F7 AE F2 02 AC F3 02 AD
7048- B1 CE 9D F0 02 C8 E8 E0  7130- F4 02 20 11 F4 A5 F9 20
7050- 06 D0 F5 A0 05 B1 CE D0  7138- 05 F6 A9 00 F0 02 F0 1B
7058- 76 88 B1 CE CD F2 02 90  7140- AD FB 02 10 16 A2 40 20
7060- 6E CD F5 02 F0 02 B0 67  7148- 30 F7 AE F2 02 AC F3 02
7068- 88 B1 CE CD F1 02 90 5F  7150- AD F4 02 20 11 F4 A5 F9
7070- D0 09 88 B1 CE CD F0 02  7158- 20 05 F6 A9 00 8D F0 02
7078- 90 55 C8 B1 CE CD F4 02  7160- 8D F1 02 A0 0B B1 CE C9
7080- F0 04 B0 4B 90 0A 88 B1  7168- 01 D0 07 A9 08 8D F1 02
7088- CE CD F3 02 F0 02 B0 3F  7170- D0 0E A0 06 B1 CE 29 01
7090- A0 00 B1 CE 99 F0 02 C8  7178- AA A9 07 9D F0 02 B1 CE
7098- C0 0D D0 F6 A5 E8 48 A5  7180- A0 02 A2 00 29 02 F0 17
70A0- E9 48 AD F0 02 85 EB AD  7188- 38 B1 CE FD F0 02 91 CE
70A8- F1 02 85 E9 A9 01 85 E7  7190- C8 B1 CE E9 00 91 CE C8
70B0- AD FB 02 F0 06 30 04 A9  7198- E8 E0 02 D0 EB F0 15 18
70B8- 00 F0 07 AD F6 02 0A 0A  71A0- B1 CE 7D F0 02 91 CE C8
70C0- 0A 0A 85 F9 A2 7F AD FC  71A8- B1 CE 69 00 91 CE C8 E8
70C8- 02 F0 08 A2 00 F0 04 A9  71B0- E0 02 D0 EB AD FF 02 85
70D0- 00 F0 6B 86 E4 AD F7 02  71B8- E4 AD FE 02 85 F9 AD FD
70D8- F0 06 A5 E4 49 7F 85 E4  71C0- 02 85 E7 68 85 E9 68 85
70E0- AD F8 02 D0 22 A5 E4 49  71C8- E8 A5 FF A4 35 A6 EB 60
```

Listing I: Hexadecimal dump of code

More listings on page 63 〉

Lisa speeds meetings on their way

CHARLES FAIRFAX and GERALDINE HOWE show how bringing in a micro can help demolish mountains of paperwork

THE introduction of Lisa into the daily work of the Emergency Planning Department of the North Western Regional Health Authority has simplified the tasks of preparing information for meetings and processing the decisions taken.

The long established style of dealing with meetings has always involved staff in large amounts of tedious, time consuming work. This was necessary in the past – but not now, thanks to Lisa!

During the pre-Lisa period there was a growing acceptance of the help that could be given by the Apple. In particular, word processing and file sorting had become established, and graphics were beginning to emerge as an added bonus.

However until the advent of Lisa, all we seemed to have achieved was the putting together of a series of separate computer processes leading to the production of the old familiar piles of printed papers.

Essentially, this was modelled on the old manual system. The use of the computer improved the efficiency of the work it handled, but afterwards each individual piece of work had to be linked up with the next by hand.

Each system used would only operate on its own and had to be loaded, run and saved afresh each time a different system was needed. Word processing and graph plotting did not mix easily, while graphics seemed in another world.

The result was inevitable. Some procedures that would have been better done by computer were still done manually because of the effort of setting up the systems.

The arrival of Lisa with its greatly improved capability using new integrated software, changed all this. Meetings could now be managed from beginning to end as one entity.

Straight away, considerable benefits ensued.

To begin with, information was fed in from a variety of different sources, not all of computer origin. Rapidly, as more work accumulated in the Lisa files and Lisa played a greater role in the daily working of the department, techniques improved and the benefits escalated.

Within a mere six months, practically all the office work was being carried out in a Lisa orientated form. Paper that previously was piled high vanished from the office desk!

Today, all Emergency Planning meetings and many others as well, are prepared and subsequently processed on Lisa. In many instances, the meetings themselves are conducted without reference to paper at all.

To some people this might seem to be going overboard, but enthusiasm for technical advance has to be controlled by the dictates of good sense.

Employing computer techniques to this extent is limited by the availability of hardware and such factors as the size and venue of a particular meeting.

Total dependence on Lisa is most applicable to the small group of up to six or eight people in the office where the micro is installed.

Screen size is the limiting factor – it needs to be seen by all. The larger the monitor, the more people can be encompassed, perhaps of the order of 20 to 30.

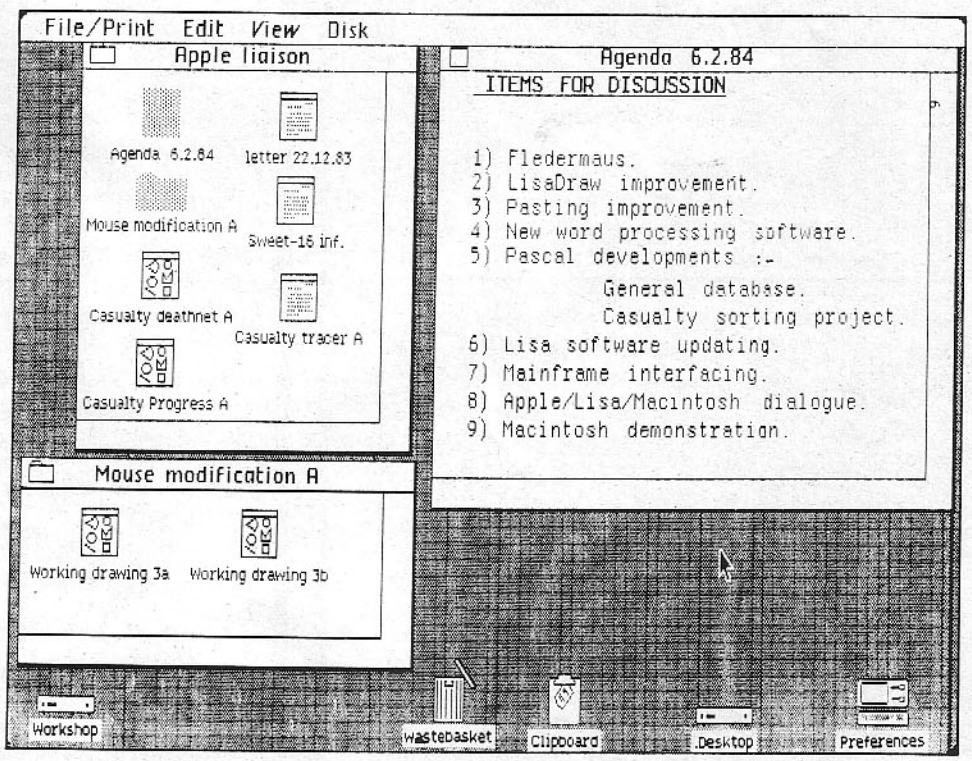
When meetings are conducted away from one's base there is always the problem of what papers to carry. Bags full of documents are heavy, yet the backup information left behind, since it will not fit the overnight bag, could well influence the outcome. It is so easy to pop a floppy in with the paperback one reads on the train!

The new hard microdisc now in use on the updated Lisa and Macintosh is even more suited, being really robust against physical damage, having its disc permanently encased in a hard plastic cassette. One such disc is usually more than enough to cope with a sizeable meeting.

As with most work on Lisa it is good practice at the outset to open the Desktop Profile, designate a working area and copy into it the necessary tool pads and the documents required for the meeting.

The preliminary letter of invitation sent to those attending, together with its enclosures is processed on LisaWrite. If at all possible it is a worthwhile saving to dispense with paper and simply use a copy disc in place of an envelope full of documents.

This will depend upon the number of people with whom one communicates, whether they too have Lisas, and perhaps whether they might have an unfortunate proclivity for retaining the as yet slightly



Dr Fairfax is a consultant in Community Medicine and Emergency Planning Officer to the North Western Regional Health Authority. Miss Howe is a personal secretary and computing assistant.

expensive disc.

No doubt as the technology becomes commonplace, exchange of information in this fashion will become a matter of everyday acceptance, not subject to such abuse. Public authorities could well pool discs on a quid pro quo arrangement.

Better interchangeability of software between different manufacturers and even between different versions from the same source is essential to make progress.

It is pleasing to see that at least one manufacturer is alive to this need. Lisa and Macintosh *will* be able to talk to each other.

A great saving can be made when a series of letters are sent out, each one the same in its essentials but containing its own slightly different details.

Often a master copy of the basic letter can be created from the essentials, enough duplicates made, and the individual differences pasted in.

The pasted information can either be cut or copied from existing documents using the selections from the Edit menu box. Cut makes a pasteable copy on the clipboard removing the original, while Copy does the same but leaves the original intact.

Paste simply inserts what is currently on the Clipboard into the place on the screen where the cursor mark rests.

Alternatively, entries may be created specifically for the purpose on a separate document. Where tables from LisaCalc or graphs from LisaGraph will be required, these may all be pasted into a parent document created from LisaDraw.

This also allows the use of graphics and a very much wider choice of presentation.

The final printed papers become more easily understood since they are set out in the form required, not being limited by the restrictions of archaic methods such as hand or typewriting.

The following example is taken from real life and illustrates the application of the methods discussed.

A long meeting was to be held between officers of the Authority and a team of Apple staff to discuss difficult technical matters relating to applications of new developments.

Two officers visited Apple at Hemel Hempstead, travelling from Preston and Manchester specifically for the purpose.

Had we followed old traditions we would have needed a team of sherpas to carry all the papers that might have been required. In fact all we needed was one Lisa disc.

The disc carried the workpads required for its processing plus the files containing the material to be discussed.

As the meeting progressed, these were updated so that all the information was ready for putting onto the Profile on return.

We did not have to rely on our own memories or need to compare notes later on to see if between the two of us we could recall at least a proportion of what had transpired.

At the meeting, the contents of the disc were displayed on one of the firm's Lisas.

The save and put away routine was used to prepare the disc before departure for Hemel Hempstead so that on arrival, by reversing the steps, the exact layout of the screen could be reproduced.

All that was needed was simply to open the disc from the menu box.

The screen was divided into two halves, the one on the right was kept to display the meeting agenda, and the one on the left used to carry the file material under discussion.

With only one exception, all the agreed updating was incorporated within the preformed files or on a duplicate.

The one exception was when we used a LisaDraw document, the computer equivalent of the ordinary paper scrap pad on which to express our ideas during discussion.

Sage puts an end to disc shuffling

NEVILLE ASH reviews a single disc accounting system for the Apple II.

IF you want to buy an accounting program for the Apple II there is a large choice.

You can spend under £100 for a program like Hilderbay Bookkeeper, or lots more on a complete accounting system such as the ones offered by Tabs, Peachtree and Jarman.

So with all these programs already available is there a need for another one?

Most are of the module variety and you add extra parts as your business grows.

In doing so you add extra program discs too, so disc shuffling becomes a part of the everyday use of the program.

Many people have accepted this situation and put it down to the 143k capacity of the Apple II disc drive, tiny by today's standards.

Sage is promoted as the single disc integrated accounting system. Everything you need is on one disc.

So there is no more disc shuffling between the different ledgers and no more confusion by using the wrong disc.

But there is still the problem of the Apple's small capacity. How many accounts can actually fit on the single 5¼in disc?

In practice the answer is 200 sales ledger accounts, 200 purchase ledger and 300 nominal – sufficient for many small businesses just deciding to computerise.

Even so, in these smaller businesses computer knowledge is likely to be non-existent, while the accounting staff will know their job using manual methods.

What is needed is a package following the correct accounting principles, which is also easy to use for the first time computer user.

The Sage package comprises the program, a disc with sample data, the manual and a cassette.

It takes you through the program step by step and is far superior to any manual.

Most people don't read software manuals because they are often very hard to understand.

This single disc program provides sales, purchase and nominal/general ledger facilities, aged analyses, monthly daybooks, monthly and year-to-date accounts and a balance sheet.

Sage runs under CP/M, so some form of Z80 card and CP/M is needed. I tried the program with the U-Microcomputer's card and also the Starcard supplied free with Wordstar, and it worked well with both.

You should also have an 80 column card in place. I tested both the Videx and the Elite Software 80 column cards and had no problem at all.

To determine the exact number of ledger accounts that can be stored on a single disc Sage provides a simple method of calculation.

The authors say that neither the combined number of sales and purchase accounts nor the nominal accounts should exceed 999.

After loading CP/M, just type Accounts and after a few seconds the message to insert a ledger disc into drive B appears. Press Return and the main menu appears with 17 choices spread into six categories.

These cover initialisation routines, data entry routines for sales invoices, credit notes and receipts, purchase invoices, credit notes and purchase payments, cash book receipts, payments and journal entries, creating ledgers, sales/purchase ledger reports, statement routines, accounts and management reports, and finally a category dealing with the information trail (audit trail), reconfiguration routine and exiting from the program.

Before using the program create back-up copies of the program disc and initialise a blank disc to hold the ledgers using the first option on the main menu.

In a practical test I went through starting by creating the ledgers – option 11.

As Sage loaded this section, the message "Loading programs please wait..." appeared. Here I was able to enter the purchase accounts and sales accounts.

Although I could have carried on and entered the nominal ledger details, the manual advised me to first make a copy of that disc as it contained valuable information.

Now the nominal ledger accounts are entered.

Each account name can be up to 25 characters in length and the system reserves four manual codes which can't be changed – debtors' control account, creditors' control account, VAT account and bank account.

This area is best dealt with in terms of the trading account, profit and loss account and balance sheet.

The longest part of using Sage involved creating ledgers and entering data. Once this had been done I was able to produce everything from the invoices right through to the labels needed on the envelopes just by following on-screen prompts.

To evaluate the package from an end user's and accountant's point of view, I invited two people to test it.

One accounts person found it so simple she didn't believe it – and this was the first time she had used a micro. And a chartered accountant grudgingly conceded Sage was quite good.

The only time I encountered problems was when the instructions were not followed.

Failing to initialise your ledgers, for example, could create trouble when information has to be entered. But this is a user fault rather than a software problem.

In another test I asked an accounts clerk to give Sage a full try too. Her only query was directed at her employer, who has previously been against computerising the accounts. Now things should change.

Another equally practical test of such a program is to see what type of printouts can be produced – again by following the instructions.

Sage is going to provide strong competition for existing Apple II accounting packages. Not only is it simple to use but it is also contained on a single disc and the makers have a hotline service too.

And this cassette form of instruction supplied doesn't appear to be offered by anyone else.

A Z80 card, CP/M and an 80 column card are required, but these are usually already installed on most business Apples. If not, the package is certainly worth the extra investment.

	Gross Profit	7,839.25	Gross Profit	7,839.25
Salaries	864.01		864.01	
Rent & Rates	250.00		250.00	
Electricity	146.35		146.35	
Travel & Entertaining	92.50		92.50	
Motor Expenses	178.10		178.10	
Postage & Carriage	27.10		27.10	
Bank Charges & Interest	14.70		14.70	
Sundry Expenses	89.20		89.20	
H/P Interest	23.90		23.90	
		1,685.86		1,685.86
Nett Profit		6,153.39	Nett Profit	6,153.39

Example of an easy to follow print out using the Sage accounting system

Product: *Sage*
Type: *CP/M-based accounting system*
Price: *£375*
Supplier: *Sage Systems*

Apple scores above average at cricket

JOHN and JONATHON NIXON demonstrate a sporting application for Apple

BY the time this article reaches you the cricket season will be drawing to its close. So now is the time to set about those averages.

My son Jonathan is scorer for Knaresborough first team, so it seemed logical to use the power of my Apple (with a little help from Multiplan) to give an up-to-date and continuous picture of the team's batting and bowling averages.

While it is always possible to add the score after each match into the totals, it is very messy, and there is no record of each match.

We therefore devised a spreadsheet which meets most of our requirements as well as illustrating a number of Multiplan's procedures.

Batting averages were the easiest to start with, although there is the problem of how many innings are not out. The formulae are shown in Figure I, with the finished spreadsheet in Figure II.

Individual scores are entered under each match with "not outs" being recorded with an additional 0.01. These scores are added up in column 8 by the usual SUM routine.

Innings are totalled by using the COUNT procedure, which only registers when an entry is made. And by subtracting the integer value of the batting total from the full total it is possible to obtain the number of not out innings by multiplying by 100.

The highest score is obtained by using the MAX procedure, which is shown in column 4. However this does not allow the usual notation for a not out high score. We have therefore introduced a three cell column, whose entry is controlled by an IF statement, again using the extra 0.01.

The batting average is then calculated by normal means, and finally the list can be sorted to give the correct batting average order.

The bowling averages presented a different problem in that each match generates four sets of figures for each bowler.

Because we have eventually to sort our list of averages, these four numbers have to be transposed from vertical to horizontal format. This can be achieved with a minimum of fuss by using the technique of naming individual cells, since it is not possible to copy vertical vectors into horizontal vectors.

The four totals are obtained in the normal way in column 3, and these cells are then named with two or three letters using the bowler's surname initial followed by O, M, R and W for overs, maidens, runs and wickets respectively.

We can then refer to these names in columns 2-5 against each bowler thereby allowing us to calculate a running average which can be finally sorted.

Spreadsheet formulae

Batting averages.	
Overall format	Column width 9, general alignment, integer.
Col. 2, Row 4 onwards	INT (RC(+6)) Column width 7
Col. 3, Row 4 onwards	COUNT (RC(+6):RC(+34))
Col. 4, Row 4 onwards	MAX (RC(+5):RC(+33))
Col. 5, Row 4 onwards	IF((RC(−1)−INT(RC(−1))>0), "* "," ") column width 3.
Col. 6, Row 4 onwards	100*(RC(+2)−INT(RC(+2)))
Col. 7, Row 4 onwards	RC(−5)/(RC(−4)−RC(−1)) Fixed point, two decimal places
Col. 8, Row 4 onwards	SUM (RC(+2):RC(+30)) Column width 8
Bowling averages.	
Overall format	Column width 10, General alignment, integer.
Col. 6, Row 7-17	RC(−2)/RC(−1) Fixed point, two decimal places
Col. 7, Row 22 onwards	SUM(RC(+2):RC(+32)) Column width 9
Col. 8 onwards	Column width 9

Figure I: Spreadsheet formulae

Knaresborough First Team Batting Averages 1984

Batsman	Total Runs	No. of Innings	Highest Score	Times Not Out	Average	Total Runs	Date Opponents	21.4.84 Alw'dley	22.4.84 Holmfirth	28.4.84 Guiseley	5.5.84 Ilkley	6.5.84 Elland
Robert Todd	387	11	95	2	43.00	387		21	7	84	1	13
David Nudds	150	8	46	4	37.50	150		14	45	1	46	
Brian Smart	376	13	67	2	34.18	376		25	15	51	1	25
Tony Quick	223	10	59	3	31.86	223		25	59	16	0	12
John Barton	103	8	27	2	17.17	103			7			27
Chris Darley	168	11	49	0	15.27	168		13	11	36	49	17
Nigel Leech	128	10	36 *	1	14.22	128		13	21	2	2	0
Mike Baxandall	40	6	16 *	3	13.33	40		16	1		1	
Robert Binns	106	9	32	1	13.25	106		1		22	32	13
John Knibbs	20	5	9 *	3	10.00	20		0			0	

Knaresborough First Team Bowling Averages as at 9.6.84

Bowler	No. of Overs	Maidens	No. of Runs	No. of Wickets	Average
Trevor Mead	44	21	57	7	8.14
Tony Quick	89	20	291	21	13.86
David Nudds	115.83	20	347	22	15.77
Mike Baxandall	137.5	38	371	22	16.86
Irving Waithe	60.5	6	263	15	17.53
John Knibbs	85	13	312	10	31.20

Bowler	Details	Totals	Date Opponents	21.4.84 Alw'dley	22.4.84 Holmfirth	28.4.84 Guiseley	5.5.84 Ilkley	6.5.84 Elland
Mike Baxandall	Overs	137.5		10	7	17	7	4
	Maidens	38		4	2	9	2	
	Runs	371		28	13	29	11	20
	Wickets	22		2	1	1		
Tony Quick	Overs	89		3	9	8	8	5
	Maidens	20				5	2	
	Runs	291		14	31	17	18	29
	Wickets	21		1	2	2	2	
David Nudds	Overs	115.83		8	7	8	13	6
	Maidens	20				1	3	
	Runs	347		28	23	27	40	18
	Wickets	22		1	3	1	2	
John Knibbs	Overs	85		7		5	6	9
	Maidens	13		3			1	2
	Runs	312		15		43	31	25
	Wickets	10		1			1	1
Trevor Mead	Overs	44		9			11	
	Maidens	21		6			6	
	Runs	57		12			10	
	Wickets	7		1			2	

Figure II: Finished spreadsheet

Dream come true for non-programmers?

DON MARRISON tries out a program generator with its promise of simplyfying the writing of Basic

CODEWRITER is a package which enables anyone without programming experience to write their own tailor made programs with the minimum of fuss and with a speed which has to be experienced to be believed.

Anyone who would like to become involved in creating custom built programs should certainly give Codewriter their consideration. They have no need to be apprehensive about using it.

Programs dealing with personnel records, invoices, statements, sales records, mailing lists, personalised letters or memos, financial data analysis, stock control and inventory recording can all be designed using Codewriter.

Its strength stems from the fact that the user is liable to know exactly what he requires and, if in the future modifications are necessary, these can be carried out without difficulty.

Codewriter comes with three discs, an introductory handbook and an easy to read manual. In addition to the two system discs, there is a demo disc showing how to generate data entry, report and menu programs.

The master disc is used to generate data entry programs and the second system disc is used to create report programs and menus to link together those created by the user.

Copy protection is provided by a dongle in the games I/O socket of the Apple IIe motherboard. With this in place working copies of both discs can be made.

The general manual covers several machines – the differing notes for individual micros are included on the master disc and are displayed on the monitor screen when the appropriate command is chosen.

This is not particularly convenient for a first time user but once he has the hang of it the notes are rarely needed.

The manual is not overpowering in any way and is easy to understand, progressing step by step through the various stages of screen design and program creation. Excluding the index, there are only 72 pages to read.

Apart from one minor inaccuracy and contradiction regarding the justification of monetary fields, the instructions are excellent and easy to follow – even for someone not used to this type of program.

Having completed the example in the manual, it is simple to start designing one's own program. What soon becomes apparent is the fact that they can be designed and coded very quickly indeed.

At first, the claims of Dynatech regarding speed seem exaggerated. The sample program on the demo disc containing a data entry program, three report programs and a menu linking all four together took only two hours to design and write.

However, with experience, this feat is by no means impossible. The expertise required is soon learnt and less and less reference has to be made to the manual as the screen instructions and prompts are clear and concise.

Once coded, the program can be run by simply typing RUN. A menu giving the options of entering data, updating a record, looking-up a record, searching records either selectively or by scanning them all, or deleting a record is provided without any involvement by the programmer.

Alternatively one can proceed to report creation by using the second system disc.

This is where the real power of Codewriter comes into play and where the programmer can, by using "accept if:" statements, create a program which can obtain the information he requires from the data already entered in the data entry program.

By studying the examples contained in the manual, the user can construct quite complex search formulae which will enable sophisticated reports to be produced by the end program.

At this stage, the only restrictions are imposed by the programmer's ingenuity. It is the most rewarding and satisfying stage of program generation and additional calculations can be performed if desired.

The end result can be sent to the screen or printed out as required – the latter choice given every time the report program is run. Pages can be numbered and the date of the report can also be shown.

When the report is being designed, the user has the option of having a static report – one which will look for the same information time after time – or a flexible one, which will enable the user to enter variable search criteria.

Apart from printed reports, this section can be used to provide mailing labels or personalised letters.

The report can be divided into several sections and can give summaries of specified data, including minimum and maximum figures as well as totals and averages.

A sort program is also loaded with the report and so the finished product can be sorted on a particular field before being printed.

Perhaps the simplest section is the final one, the creation of a menu program to link all the others together. This, literally takes only a few minutes, obviously depending on the number of programs involved.

Once done, a suite of programs can be run together and when booted-up, the disc containing them shows the main menu.

The only thing to spoil the finished package is that when a report program has been run, the user is given the options of running another report, exiting to Basic or returning to the Codewriter Disc 2 main menu.

The latter option, however, will not make sense to those who have not written the suite, as they may never have heard of Codewriter Disc 2.

If this option is followed, references to Disc 2 ignored and the program disc left in the drive, the user will be taken back to the main menu of the suite created by Codewriter, which is what one would have wanted anyway.

With a little bit of programming experience this could be rectified quite easily by listing the program and altering the wording of the prompts.

The finished programs are written in Basic and therefore are not the quickest running, which may be a drawback to some people. But as they can be compiled for faster execution those to whom speed is important can be catered for.

This review has been carried out through the eyes of a non-programmer but it is felt that Codewriter would be of great benefit also to the experienced.

Just by listing the created program, it can be seen that writing it from scratch in Basic would take a long, long time and any typing error would cause problems and delay.

Programs are written with more speed and accuracy than is possible by an individual and can be listed and modified quite easily. Codewriter, therefore, could be a great timesaver.

Product: Codewriter IIe.
Type: Program generator.
Price: £185.
Author/Distributor: Dynatech Microsoftware.

Get your message across on a bulletin board

PETER TOOTILL explains the advantages of using your Apple for interactive communication.

ONE of the most interesting of modern methods of communication is the bulletin board system – often referred to as a BBS.

So what is such a system? The idea started in the USA, when a computer club decided that pinning messages to other members and notices onto a cork board on the wall was a bit too much behind the times for a high technology club like theirs.

They decided to hook a computer to a telephone and allow people with modems and micros, or even ordinary terminals, to dial in and leave their messages on the computer instead.

As modems were even then priced within the reach of the hobbyists in the USA, the idea was very successful.

It was soon taken up by other clubs and also by individuals, with the result that today there are at least 1,000 public and private BBSs operating in North America – and probably considerably more.

Since those early days the software that controls such systems has developed enormously and you can run a BBS on most of the popular micros available in the USA.

British BBS software is virtually non-existent as yet, but some programs are being developed.

Bulletin boards will allow callers to do a wide range of things, but the main emphasis is still on the traditional message and mailbox facilities.

For example, on most BBSs messages can be private or public, general interest or collected into special categories. They will tell you if you have a message waiting when you call, and also let you search for messages on a particular subject.

Other features that you will find on a BBS include information and news files, help for inexperienced users, software to download, games and diversions, commercial sections and even advertising.

The reason I prefer using bulletin boards to Prestel or Micronet, which are the other systems readily available to the home micro user, is that they provide a very interactive type of environment, completely different in character to viewdata systems such as Prestel.

With a BBS you are always able to respond to the information on the system.

If you see something that interests you, you can leave a message about it, either to the originator of the item or to anyone else who calls the system.

In fact exchanging messages is really what BBS are all about.

There will always be a good bit of straightforward information on such systems, but it is usually there in a secondary role. For example there will probably be:

- Information about the system itself and how to use it.
- Information about particular subjects in the special interest sections.
- Files you can download into your own micro.
- Telephone numbers of other systems.

But unless the system has a particular theme of its own, you are unlikely to find much information of general interest on it.

If you are looking for train times, weather reports, financial information, hotel bookings and suchlike the large commercial systems such as Prestel are the place to go.

There is, of course, a certain amount of overlap between Prestel systems and BBS. You will find some news about the micro scene aimed at Apple users on Micronet and Viewfax 258 on the Prestel system, and also on some BBS. But on the

BBS

Standard RS 232 setting for use with on line systems:
7 bit word
(excluding parity bit)
Even parity
1 stop bit

Alternative setting that should work with most BBS:
8 bit word
No parity
1 stop bit

I recommend that you use the second setting, as most terminal software just ignores parity anyway, and it will normally work satisfactorily.

It also gives the added possibility of transferring binary data using standard CP/M protocols or XMODEM protocols which give error free file transfer – and which aren't just for CP/M systems.

whole the Prestel systems are very much more one way.

They do have response frames, or places where you can leave messages to the operators of the database concerned, but they don't normally have the public message areas that BBS have.

The other advantage of BBS is that they are open to anyone with no subscriptions or membership, and are free of charge (except for the cost of the phone call).

On the other hand ordinary Prestel will cost you £5 a quarter (+VAT), and Micro Prestel (which has the software and some other features from the Micronet system on it these days) is a further £8 a quarter.

However most people can get Prestel with a local call, which is something that cannot be said for BBS. A long distance call can soon chalk up a couple of quid.

If you are choosing a modem my advice would be – if you can possibly afford it – to look for a multi-mode type that will allow you to use both V.21 (300 bit/sec – most BBS) and V.23 (1200/75 bit/sec – Prestel) systems.

The same applies to software. Buy a terminal program that will enable you to use both systems, but remember that to use both you need the right modem AND the right software.

There's no such thing as software that will allow you to use a V.21 system via a V.23 modem, or vice versa.

Happy communicating!

Hooked on the comms idea? Now get down to making it work

Part Two of QUENTIN REIDFORD's series on microcommunications

MY last article – in the June edition of *Apple User* – was intended as an introduction to communications, and this time I would like to go into more detail about the various bits and pieces required.

The Apple is an 8 bit computer and data travels around inside it on an electronic 8 lane motorway. The combination of 1's and 0's which make up a character all travel at the same speed, reaching their destination together.

However we require the data to change from this parallel format to a single file or serial format before we can use it to communicate via the telephone lines.

The strength of the Apple is due in part to all those slots at the back of the case and it is into one of these – normally Slot 2 – that we will put the serial card which will handle this conversion of data from parallel to serial format. Peripheral card manufacturers have fun developing serial cards slightly different to each other, and this can cause some problems with software.

However they do follow a standard which rejoices under the name of the RS-232-C – catchy isn't it?

The serial card in Slot 2 acts like a funnel, taking the 8 bit wide bands of data and reducing them to a single stream.

'Peripheral card manufacturers have fun developing serial cards'

This leads to the obvious problem of finding out where one 8 bit character stops and another begins – enter protocol. This adds a start bit to the character and either one or two stop bits to the end of it.

There is another option available called parity which is a means of checking the validity of the transmitted character.

The parity bit can either be odd, even or none. Depending on choice, it will make sure that the number of bits in the character, including the stop-bits, which have a binary value of one, add up to either an odd or an even figure.

Typical formats in common use are 8 bit – no parity – 1 stop, and 7 bit – even parity – 1 stop.

If the distant computer is set to receive 8-N-1 then a calling computer set to either 8-N-1 OR 7-E-1 will be able to communicate freely. However if the calling computer was set to 7-E-2(stop), there will be corruption of the data received at the other end.

Incompatibility of the data-word format is a common cause of character corruption between devices, second only to poor telephone lines.

When you get your serial card you will find that it can be set to a variety of speeds, known as the baud rate – roughly the number of bits which can be sent in one second.

More precisely it is a mathematical calculation based on the time taken to send the shortest part of the signal.

If you have a 7 bit character preceded by one start bit and with one stop bit you should be able to send or receive 29.7 characters in a second.

For reading scrolling data as it comes into your computer 300 baud is a good choice. But file transfer at 300 baud is slow and a preferable speed would be 1200 baud.

Prestel on 1200/75 baud means that you type letters which are sent at 75 baud but you receive a screen of information at 1200 baud.

You only notice how slow 75 baud is when you are entering a message frame. For normal one-key selection of pages the speed is satisfactory.

However at 1200 baud a screen full of characters fills up very quickly and as it does not scroll you can read it at your own pace.

While on the topic of serial cards, you must get a card which will handle full-duplex not just half-duplex, and a serial cards are also meant to drive printers and plotters, make sure that the card has been set

NEXT MONTH sees the launch of a unique communications package, specially designed for Apple users, that incorporates all the latest developments in technology.

It includes a specially designed universal communications card, an all-standards modem, and a suite of communications software that meets the demanding specifications laid down by Apple User.

All these items will also be available separately, which means you can select whichever modules you need to enhance the equipment you may already have.

All the experts now agree that communications is going to be the fastest-growing field in micro-computing. This is YOUR chance to be right in the forefront of these exciting developments.

Full details of what we have to offer will be given in next month's issue of Apple User.

up to talk to a modem . . . more of which later.

In passing it is worth mentioning that Pace are about to introduce their own serial card on to the market which should be resonably priced and support all baud rates and interrupts.

This will make an ideal companion to their multi-function modem which will soon have auto-answer capability.

If the serial card is now firmly placed in Slot 2, you can either part with quite a lot of cash for the interface cable or join the Midnight Solderers' Club and make your own.

Either way you will soon find out that the only standard thing about the serial interface cable is the name – it's the RS-232-C again.

What this really means is that each wire coming from the serial card has a specific name and function, but there are no rules as to how many wires have to be connected or what type of connector is used to plug into the serial card or the modem.

The most common connector for RS-232 cables is a large 'D' shaped plug with 25 pins arranged in two rows. This will allow full use of all possible data and control signals passing between the modem and the serial card.

With the Apple I used to be pleased that every serial card I saw used this plug. Pity the BBC owner who has only a DIN plug, and reversible at that!

I was however dismayed to find that the new Apple IIc also has a DIN plug for modem connection. I hope that it is of the type that will allow enough control signals, as you could very well need them in the future.

In basic terms the only wires you need to connect a modem to your serial card are data-in, data-out and signal-ground.

However if you need auto-answer or auto-dial then you have to be able to control the modem from the Apple and vice-versa.

You should look carefully at the RS-232 socket on the modem you are contemplating buying, particularly one with auto-answer, as without these controls you could have problems.

There is usually enough information in the serial card manual to prevent me having to go through the pin configurations for connecting a modem to a serial card.

If the serial card has been set up for a printer then you will have to cross two sets of wires. Pin 2 wire goes to Pin 3, Pin 3 wire to Pin 2, Pin 6 wire to Pin 20 and Pin 20 wire to Pin 6. Do this at only one end of the cable.

A cable made up in this way is called a null-modem and can be used to join two computers together using serial cards set for modem transfer and joined only by a cable – that is, without a modem being used.

The 25 pin 'D' plug is easy to use with all the pins clearly numbered. The DIN plug is a different matter with different numbers of pins in different positions and often with no numbers to help you.

Why there cannot be standardisation in the connector type I have no idea, but I suspect that it may not be for our benefit!

Now that you have a serial card and, hopefully, a suitable interface cable, we can look at modems. This alluring word comes from MOdulator/DEModulator, which describes the function of the device pretty well.

The serial data zips along the cable into the modem as a series of 1's and 0's but that information cannot be sent down the telephone wires without further changes.

The modem takes these signals and converts them to tones, one high and one low at specific frequencies to signify either 1 or 0. The rapid change in tone gives a musical sound over the phone line to the distant modem which then takes them and demodulates them back to a series of 1's and 0's, which is what the computer can understand.

Most modems also have the capability of generating two levels of tone for each baud rate – originate and answer.

Generally speaking if you are calling a remote computer which will answer your call then you should be in originate mode as the distant modem will be set to answer.

If you try to call with your modem also set to answer, then the carrier tones which overlay the data tones will not lock on and no connection will be made.

It is also important you understand that any modem will not work with any baud rate. For instance, a modem which can only deal with 300 baud will not be able work with 1200 baud, although it will probably manage 110 baud.

The latest series of modems

What is meant by 'duplex'?

FULL and half-duplex refers to the way the data is sent between computers.

In full-duplex the characters you type are passed along the phone lines as a modulating tone on one frequency and returned to you on a different frequency before being channelled to your screen.

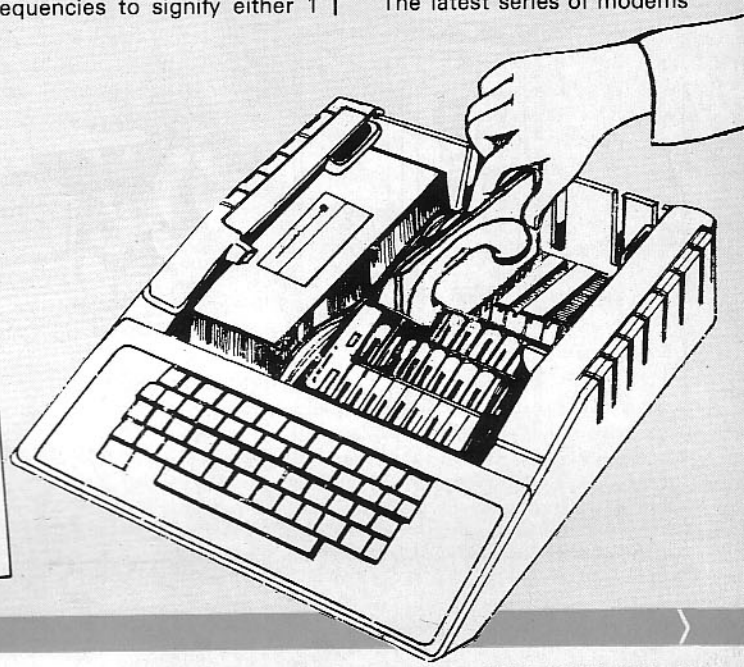
Therefore what you see on your screen is actually what arrived at the remote computer. This may not be what you typed, depending on line-noise or wrong formatting at either end. It also means that both computers can talk to each other simultaneously.

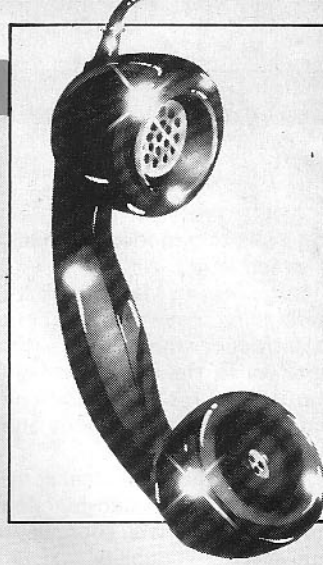
However some modems can only recognise one set of tones, and then as you can either receive or transmit data you have to use half-duplex.

This means that the characters you type are channelled directly to your screen and you have no idea what is actually getting through to the remote computer.

If the remote computer is expecting full-duplex and you are set for half-duplex you will get the character on your screen twice, once directly from your keyboard and again as the remote computer echoes it back to you.

The best analogy I can think of is to compare full-duplex with a normal telephone conversation where both people can talk and be heard simultaneously (with difficulty!) and half-duplex with a radio-telephone or CB radio where only one person can talk at a time and a signal sent to the other person when he/she can reply.





on the market which use the multi-standard chip can work with 300,1200 (half duplex) and 1200/75 baud. They will also be able to switch between American (Bell) and European (CCITT) frequencies.

CCITT, as used in the UK for data tone frequency, appears to be used world-wide, while the American (and Canadian) Bell frequencies appear to be unique to that continent.

But the number of dial up databases and the degree of modem sophistication makes Bell capability a desirable option. Both systems follow the same rules with data-word format, etc, and it is only the actual sound frequency which is different.

Apart from the considerations of baud rate, Bell/CCITT and full or half-duplex, the other principal category for modems depends on their method of connection to the telephone system.

Most of the new modems are 'direct-connect', which means they plug into a standard telephone wall socket and are usually fitted with a second socket to take the telephone handset which you will need for dialling out.

The second category is the acoustic coupler, which has two rubber cups that the telephone handset is pushed into after connection to a remote computer has been established.

The advantage of the direct-connect modem is a much cleaner line signal and the ability to operate with auto-dial and auto-answer, if available for the modem.

The acoustic coupler can operate on any standard telephone handset. With a self-contained battery, a modem such as the Sendata 700 series should, for a price, operate anywhere in the world regardless of telephone system or power requirements.

'Direct-connect modems give a much cleaner line signal'

This does not however mean that a CCITT modem of this type will connect to a Bell modem, even though you may be using it in the United States. You will still only be able to call another CCITT modem.

I am sure that some of you are now wondering about the American modems like the Hayes Micro-modem and the Novation Apple-cat. These would after-all remove the need for a separate serial card and the infernal interface cable. But would they work in the UK?

Well, I have been using a modem made in Taiwan called the Taitron. This is a plug-in-card type which, to software such as Ascii Express-Professional, looks like a Hayes Micro-modem.

However it is not a copy of the Hayes, and therefore works straight away with full auto-dialling and auto-answer facilities on our telephone system.

The Taitron seems to be completely compatible with all Hayes software, but it only works on 300 baud and Bell tones. However, the Texas Instruments chip on this card will apparently soon be available to give CCITT tones, which should make it very interesting.

The Hayes modem will work in the UK with a couple of small modifications which allow it to auto-dial and auto-answer on our telephone system.

However the Hayes MMII will only work on 300 baud and only on Bell tones. Additionally, it must be said that if British Telecom catch you using either of these modems – or in fact any unapproved modem – they will be extremely unhappy.

Considering that with the plug-in modems, it's Telecom's 50-100 volts coming into your 12 volt computer then you should be the injured par
These modems have
Telecom approval and I doub
they ever will, so be warned.

I am in no way advocati
the use of unapproved moder
on our telephone system, or
answering questions which a
bound to be in some of yo
minds.

The Novation Apple-cat
still an unknown quantity b
perhaps I can report on that ne
time.

Meanwhile the Min
Miracles auto-answer/auto-di
board has arrived and is provi
to be a good example of i
sufficient control lines betwe
modem and computer. It reli
entirely on software control.

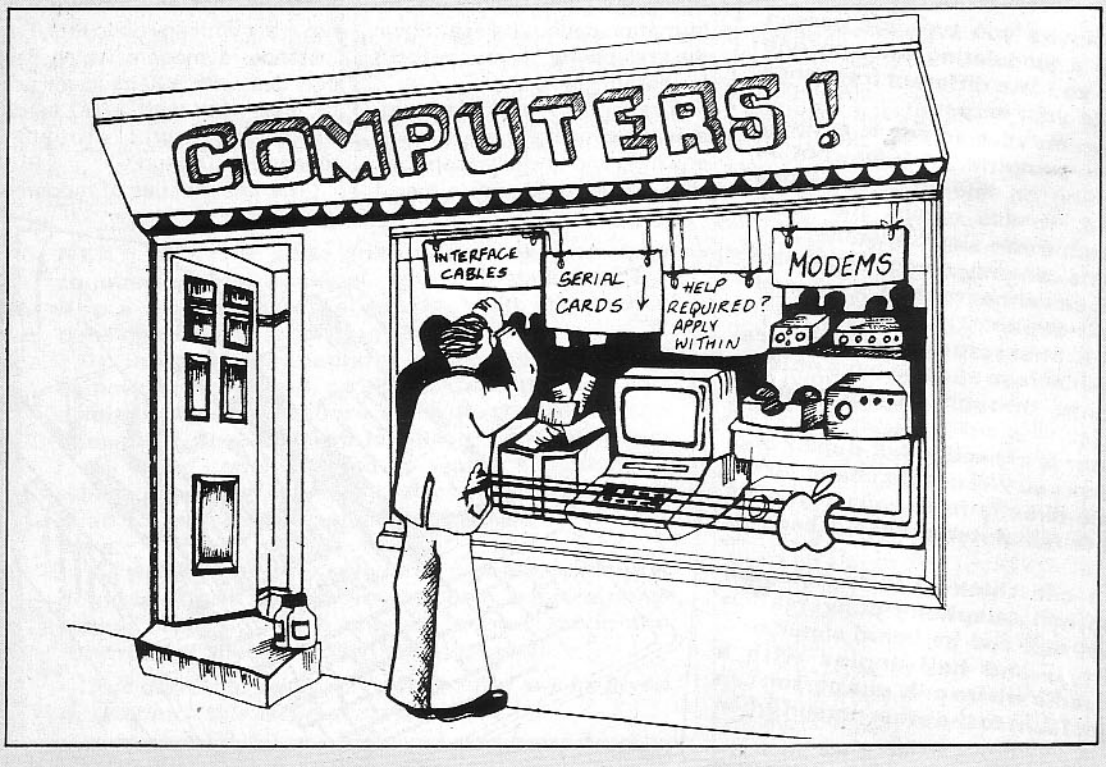
At the time of writing,
cannot prevent the auto-answ
unit from picking up the phon
line, say from a wrong numbe
and then refusing to drop
although the confused caller ha
long since gone!

I have seen the auto-diall
work with a BBC Micro, but st
cannot get a convincing resu
through the Apple.

Considering that you have
know the internals of yo
particular serial card intimatel
success in my case could b
some way off!

All in all, the auto-answe
dial board is not a ragin
success with the Appl
especially if you want to use
seriously. I am sure that Min
Miracles are looking
modifications with a degree
urgency.

● *Next time I will discuss som
of the things you can do with
this equipment; the softwa
that is available and some of th
information that can be gleane
from an enormous number
sources world-wide. Sta
saving now!*



Make the most of the Pascal assembler

P.H.P. HARRIS shows you how

THE Pascal language system is a comprehensive package providing both a text editor and an assembler.

Apple users may consider that access to these utilities, as well as to the Pascal language, may be a sufficient reason to justify the somewhat high outlay.

The text editor is indeed excellent, and requires only a complementary formatting and printout program (with which this article was prepared) to function as a wordprocessor.

The assembler, however, is not so simple to use.

The Pascal Reference Manual stresses that the chapter on its use is intended for readers who are already fluent in at least one assembly language, and does not provide a systematic tutorial.

Moreover, the assembler is of the one-pass type, with certain restrictions on the processing of labels.

Nevertheless, it is possible to use the Pascal assembler, not only for its intended purpose of generating small machine code programs to run in the Pascal environment, but also for generating machine code programs which run on their own or in a Basic environment.

The author has, for example, produced a utility program which can be used to read or edit any portion of a random or sequential textfile, machine code, or Basic program using & commands from the keyboard.

In addition, the assembler may be used to save published programs in a form which makes it easy to modify them later.

In order to achieve this, it is necessary to understand not only where the Pascal system places pieces of code in a program, but also what the directives .ABSOLUTE and .ORG do. Consider the assembly language program in Listing I. Assembled with neither .ABSOLUTE nor .ORG, it presents the output code as starting from address $0000.

This is not very useful to us if we want to use the code from a Basic program, because Page 0 of memory is used for all sorts of other purposes, and we would trample on various Basic functions if we left machine code programs in Page 0.

However, the code works. If we type it in from the Basic monitor and run it, the message COPYRIGHT 1983 appears in the middle of the screen.

This is not the case if we load the same code somewhere else, at $300 say, because it contains a reference to an address (LDA TABLE,Y) within itself.

The assembler equates the label TABLE with an address ($000E) appropriate for an origin of $0000.

When introduced into a Pascal program however, the code is not kept at $0000 but is slotted into the Pascal code stream by the Linker.

To do this, the Pascal assembler retains information to allow the code to be relocated by changing the address appropriately. The code will then work inside the Pascal program.

As far as a correct listing is concerned, the situation is improved by using the .ORG directive.

If we want our code assembled to start at $300, for example, we could put .ORG 300 before the start of the text to be assembled. The assembler will then produce a code stream containing $300 0s followed by our program.

This takes time, wastes disc space, and may generate an error message if the .ORG value is too high, but at least the code is correct when we type it in at this address.

However, only the listing is useful: the code is still not left at the right spot in the memory.

The use of the .ABSOLUTE directive before the .ORG prevents the assembler generating all those 0s and so saves time and space.

It brings the added benefit that the labels which have been declared at the start – using the .EQU directive – can be treated as absolute values which may be subjected to arithmetic processes, such as addition and subtraction.

This does not, unfortunately, apply to labels defined during assembly (such as TABLE in Listing I).

This prevents us, for example, from labelling the beginning and the end of a program START and FINISH and defining a label LENGTH .EQU FINISH-START, to provide a value which could be used in the program itself.

The restriction is due to the nature of a one-pass assembler. The use of the .ABSOLUTE directive, by cancelling the information necessary for relocating the labels, will in addition prevent the code from working from inside a Pascal program.

We now have a correct listing, starting at our chosen origin, but are still faced with the job of typing the code in.

It would be much simpler to find the code in the memory, for example by interrupting the Pascal program and searching for it in high memory just below $B000 (see Page 255 of the Apple Pascal Operating System Reference Manual).

A more sensible approach would be to write an assembly program which transfers the code to a safe place where it does not get overwritten when the Pascal disc is removed and a DOS disc booted.

The TRANSFER section of the code in Listing III does just this. It shuts off the language card, giving access to the ROM memory, and performs some initialisation routines.

It then finds out where it is situated in memory by performing a JSR to a ROM subroutine, and extracting the return address from the stack.

This address is then stored in $0,$1 after adding $47 to give us the starting address of the actual code we want transferred.

PAGE blocks of code, each 256 bytes long, are then

```
0000| A0 0D                    COPYRITE LDY #0D
0002| B9 0E00                  $1       LDA TABLE,Y
0005| 09 80                             ORA #80
0007| 99 B205                           STA 5B2,Y      ;CENTRE OF SCREEN
000A| 88                               DEY
000B| 10F5                             BPL $1
000D| 60                               RTS
000E|
000E| 43 4F 50 59 52 49 47  TABLE     .ASCII "COPYRIGHT 1983"
0015| 48 54 20 31 39 38 33
```

Listing I

transferred to the memory area starting at $2000, whereupon the program JMPs to the monitor, displaying the first 20 lines of the transferred code.

The method of extracting the return address from the stack is somewhat unorthodox. The normal method of finding a return address, using the sequence:

```
        JSR FINDADR
        .
        .
        .
        .
FINDADR  PLA
        STA TEMP
        PLA
        STA TEMPH
        .
        .
        .
        RTS
```

fails, as noted earlier, when the .ABSOLUTE directive is used and the code is run from a Pascal program.

We can now make use of the assembler as follows:

The necessary Pascal files are found on discs Apple1 and Apple2:

```
SYSTEM.APPLE
SYSTEM.PASCAL
SYSTEM.MISCINFO
SYSTEM.EDITOR
SYSTEM.FILER
SYSTEM.LINKER
SYSTEM.ASSEMBLER
6500.OPCODES
6500.ERRORS
```

The author, using a one-drive system (ugh!), uses disc Apple3 as the boot disc (for SYSTEM.APPLE), followed by a disc holding the rest of the above files.

A Pascal "carrier" program is written, as in Listing II. This program, PASDOSAS, is used to execute a single external procedure PUTCODE, which will be our machine code program, assembled and linked in with the linker. PASDOSAS should be compiled, and the compiled code left on the assembly disc.

An assembly program is written, as in Listing III. This must be given the name .PROC PUTCODE, so that it may later be recognised by the linker as the unit for linking into PASDOSAS. (The name may be changed, provided that the same name is used in the carrier program.)

The routine TRANSFER must come first, followed by the new assembly program (BORDER in the listing). TRANSFER regards BORDER as a table of code, to be transferred at execution time to the region of memory starting at $2000.

The program is then assembled, and the code linked into PASDOSAS. Reference to the appropriate chapters of the Pascal manuals should illuminate the procedure.

The last Linker question, as to the name of the output codefile, should be answered by a name indicating the nature of the assembly program, for example in this case BORDER.

This codefile should now be eXecuted. The Apple will beep and show you 20 lines of the transferred code and the monitor prompt.

It is now time to establish the DOS environment. Insert a DOS disc, and boot using 6CTRL-P RETURN. The disc MUST be a slave, NOT a master, otherwise the code will be overwritten by DOS during the boot.

The code may now be moved to its correct origin with the monitor M command.

The routine BORDER, when run, puts a border of asterisks round the text screen, and shows the message COPYRIGHT 1983 in the bottom right hand corner.

As an exercise in assembly language programming, it demonstrates several elementary techniques.

This includes the use of the X and Y registers as both incrementing and decrementing loop counters, and references to data tables and to subroutines not only within the program but also built into the Apple ROMs.

As a further exercise, readers may like to eliminate the data tables by replacing the calls to SETADDR with calls to VTAB ($FC22) after storing the contents of the Y register in CV ($25).

The following notes may prove useful when copying programs written on other assemblers:

The mnemonic ASL should be written ASL A, when the contents of the accumulator are to be shifted. (Beware when copying listings from the monitor disassembler.)

Hex numbers starting with A-F must be preceded by a "0" (for example 0FC22), otherwise the assembler thinks they are labels.

The syntax for indirect addressing is non-standard (see Page 105 of the Apple Pascal Reference Manual).

Symbols equivalent to the "<" and ">" used by the Apple Toolkit assembler to denote the high and low bytes of an address label are not available. A known address can be split by the following declarations, if the .ABSOLUTE directive is in use:

```
;Address < 8000
;          (hex)
ADDRESS  .EQU 3A2
ADDHI    .EQU ADDRESS/100
ADDLO    .EQU ADDRESS%100
;address >= 8000
;          (hex)
NXTAD    .EQU 9560
NXTHI    .EQU NXTAD-8000/100+80
NXTLO    .EQU NXTAD&0FF
```

During assembly, the labels ADDHI and ADDLO are given the values 3 and A2 respectively.

The listings are edited and simplified versions of the actual output files of the assembler, which generates further information, including symbol tables, not shown here.

```
PROGRAM PASDOSAS;

PROCEDURE PUTCODE;
EXTERNAL;

BEGIN
  PUTCODE
END.
```

Listing II

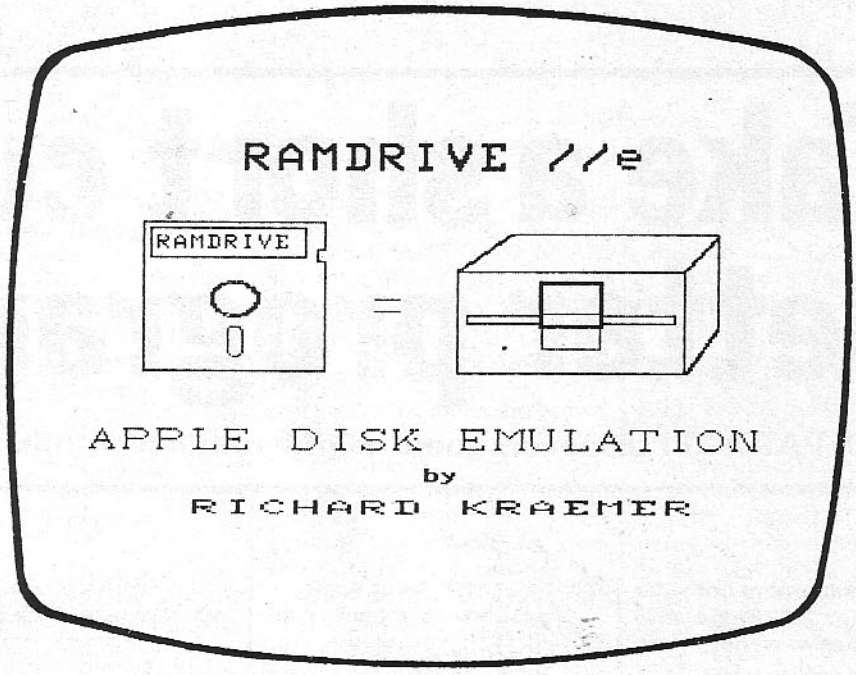
```
0000|                   ;Labels for BORDER
0000|                   ;
0000| 00AA              SYMBOL      .EQU 0AA        ;"*" as border
0000| FC58              HOME        .EQU 0FC58
0000| 0000              TEMPADL     .EQU 0
0000| 0001              TEMPADH     .EQU TEMPADL+1
0000| 0020              LEFTEDGE    .EQU 20         ;
0000| 0021              WIDTH       .EQU 21         ; Text window
0000| 0022              TOPEDGE     .EQU 22         ;  locations
0000| 0023              BOTTEDGE    .EQU 23         ;
0000|
0000|
0000|                   ;Labels for TRANSFER
0000|                   ;
0000| 0300              ORIGIN      .EQU 300
0000| 02AA              START       .EQU ORIGIN-56
0000| 0008              PAGE        .EQU 8          ;to transfer 2 kBytes
0000|                                               ;  of code
0000|                   0000|

0000|                             .ABSOLUTE
0000|                             .PROC PUTCODE
0000|                             .ORG START
02AA|
02AA|
02AA|
02AA| AD 82C0           TRANSFER LDA 0C082        ;Shut off language
02AD|                                             ;  card
02AD|
02AD|                       ;Monitor initialisation
02AD|                       ;----------------------
02AD|
02AD| D8                        CLD
02AE| 20 84FE                   JSR 0FE84         ;SETNORM
02B1| 20 2FFB                   JSR 0FB2F         ;INIT (Set text
02B4|                                             ; window etc.)
02B4| 20 93FE                   JSR 0FE93         ;SETVID
02B7| 20 89FE                   JSR 0FE89         ;SETKBD
02BA|
02BA|                       ;----------------------
02BA|
02BA|
02BA| BA                        TSX               ;Get return address
02BB| BD 0001                   LDA 100,X         ; from stack
```

Listing III

```
02BE| 85 01                    STA 1
02C0| CA                       DEX
02C1| BD 0001                  LDA 100,X
02C4| 18                       CLC
02C5| 69 47                    ADC #47
02C7| 9002                     BCC $1
02C9| E6 01                    INC 1
02CB| 85 00              $1    STA 0
02CD| A9 00                    LDA #0
02CF| 85 02                    STA 2
02D1| 85 3A                    STA 3A
02D3| A9 20                    LDA #20
02D5| 85 03                    STA 3
02D7| 85 3B                    STA 3B
02D9| A2 08                    LDX #PAGE
02DB| A0 00              $3    LDY #0
02DD| B1 00              $2    LDA @0,Y
02DF| 91 02                    STA @2,Y
02E1| C8                       INY
02E2| D0F9                     BNE $2
02E4| E6 01                    INC 1
02E6| E6 03                    INC 3
02E8| CA                       DEX
02E9| 10F0                     BPL $3
02EB| 20 61FE                  JSR 0FE61       ;List transferred
                                                ; code
02EE|
02EE| A9 69                    LDA #69         ;Redefine
02F0| 8D F203                  STA 3F2
02F3| A9 FF                    LDA #0FF        ;    <reset>
02F5| 8D F303                  STA 3F3
02F8| 49 A5                    EOR #0A5
02FA| 8D F403                  STA 3F4         ;      vector
02FD| 4C 66FF                  JMP 0FF66       ;End in monitor
0300|
0300|
0300| A2 00              BORDER LDX #0
0302| A0 00                    LDY #0
0304| 20 4B03            $1    JSR SETADDR
0307| E8                       INX
0308| E0 18                    CPX #18
030A| D0F8                     BNE $1
030C| C8                 $2    INY
030D| 91 00                    STA @TEMPADL,Y
030F| C0 27                    CPY #27
0311| D0F9                     BNE $2
0313| 20 4B03            $3    JSR SETADDR
0316| CA                       DEX
```

```
0317| 10FA                     BPL $3
0319| 91 00              $4    STA @TEMPADL,Y
031B| 88                       DEY
031C| 10FB                     BPL $4
031E|
031E| A9 01             STWINDOW LDA #1
0320| 85 22                    STA TOPEDGE
0322| 85 20                    STA LEFTEDGE
0324| A9 26                    LDA #26
0326| 85 21                    STA WIDTH
0328| A9 17                    LDA #17
032A| 85 23                    STA BOTTEDGE
032C| 20 58FC                  JSR HOME
032F|
032F| A0 0D             COPYRITE LDY #0D
0331| B9 3D03            $1    LDA TABLE,Y
0334| 09 80                    ORA #80
0336| 99 6907                  STA 769,Y       ;Bottom right
                                                ;  corner of screen
0339|
0339| 88                       DEY
033A| 10F5                     BPL $1
033C| 60                       RTS
033D|
033D| 43 4F 50 59 52 49 47  TABLE    .ASCII "COPYRIGHT 1983"
0344| 48 54 20 31 39 38 33
034B|
034B| BD 5A03            SETADDR  LDA TEXTADL,X
034E| 85 00                    STA TEMPADL
0350| BD 7203                  LDA TEXTADH,X
0353| 85 01                    STA TEMPADH
0355| A9 AA                    LDA #SYMBOL
0357| 91 00                    STA @TEMPADL,Y
0359| 60                       RTS
035A|
035A| 00 80 00 80 00 80 00  TEXTADL  .BYTE 0,80,0,80,0,80,0,80
0361| 80
0362| 28 A8 28 A8 28 A8 28           .BYTE 28,0A8,28,0A8,28,0A8,28,0A8
0369| A8
036A| 50 D0 50 D0 50 D0 50           .BYTE 50,0D0,50,0D0,50,0D0,50,0D0
0371| D0
0372| 04 04 05 05 06 06 07  TEXTADH  .BYTE 4,4,5,5,6,6,7,7
0379| 07
037A| 04 04 05 05 06 06 07           .BYTE 4,4,5,5,6,6,7,7
0381| 07
0382| 04 04 05 05 06 06 07           .BYTE 4,4,5,5,6,6,7,7
0389| 07
038A|                          .END
```



Ramdrive opens the door to extended memory

RAMDRIVE is a nice little extra facility if you have an Apple IIe with an extended 64k or 128k 80 column card. It enables you to use this memory as a simulated disc drive.

The advantage of such RAM drives is that they are very fast – the disadvantage is that switching off the computer loses the contents.

I had not been tempted to buy a dedicated 128k RAM card for this purpose but, since the extra memory on the extended 80 column card is there and often unused, Ramdrive IIe is an attractive proposition.

You get a half capacity disc with the 64k card and a full one with 128k.

The disc arrives without a manual but on booting, all is revealed. As well as copious instructions for the screen there is an option to print a nine page document to a printer.

After using Ramdrive for a while, it becomes apparent just what a good job Precision Software have done. All the potential problems have been forseen and allowed for.

Installing Ramdrive for DOS 3.3 is simplicity itself: BRUN RAMDRIVE, and instructions for creating a full turnkey system including automatic transfer of files to the Ramdrive are included.

The software also provides audio and visual indicators to show when it is in operation. These mimic the red light on a normal disc drive, and can be modified at the bootup stage.

You can tell Ramdrive that you still want to use 80 column display and even double hi-res graphic mode and this will be allowed for on allocating the free disc space.

There is thus no conflict when using the card both for 80 column facilities and as a Ramdrive.

Side two of the disc contains files for installing the facility into the Apple Pascal 1.1 system.

These four files have to be transferred to APPLE1.

This is to provide automatic installation of the Ramdrive on booting up Pascal. There is also the option to allow automatic transfer of the System.Filer and System.Editor to the Ramdrive.

With these files in, there is none of that tedious disc accessing every time the F(iler and E(ditor are invoked.

The Ramdrive can also simplify life considerably if you are using Pascal with only one disc drive.

The system is simplicity itself to use and I had no problem whatsoever with either DOS 3.3 or Pascal.

However, for the technically minded, the disc also includes a description of the memory usage and bank switching used with both operating systems.

If the above was all you got for your money, you would be well satisfied – but there is more.

The disc contains a Ramcopy program that operates like COPYA but uses the extra memory to store disc contents. The number of repeated disc accesses is then reduced to one (64k) or none (128k), which speeds the backing up of discs quite considerably.

Finally, the disc contains a copy of the Speedos utility – a public domain program written by Lee DeRaud – which speeds up LOADing and SAVEing programs and binary files dramatically, even on a normal disc drive.

In conjunction with the Ramdrive, the net increase can be a factor of 40. It is possible to LOAD a hi-res picture from the Ramdrive in a third of a second.

I used the Ramdrive a little with Pascal, but mainly with a variety of Basic programs and I soon forgot that it wasn't a *real* drive.

I also deliberately set about trying to lose the files with RESETs and reboots (Ctrl-Open Apple-Reset).

On re-installing Ramdrive, the files were still there and ready to use. Short of switching off the Apple IIe, they seemed very resistant to being destroyed.

I'm not saying that they can't be – but it does indicate that Precision Software have made an effort to ensure that valuable files are not lost easily.

The Ramdrive also supports CHAIN and this is another area where the high access speed is a real boon.

All in all it's been some time since I saw a package that offered as much at such a modest cost.

Product: *Ramdrive IIe.*
Description: *Software utility for disc drive emulation.*
Requirements: *Apple IIe with extended memory 80 column card. N.B. Versions for the Apple IIc or operating under ProDOS are not available.*
Price: *£29.95.*
Distributor: *Coastal Computing, 16 Malt Kiln Road, Newbiggin, Ulverston, Cumbria LA12 0RJ. Tel: 0229 88408.*

By PETER GORRY

Take a short cut to better programming

MAX PARROTT discovers some helpful routines in this utility software package

JUST for a moment consider these lines of Basic and guess what's wrong with them:

```
10 INPUT X%
20 PRINT SQR (X%)
```

Actually there's nothing wrong with them until the program user forgets what the program is doing, or maybe doesn't even know what to enter at the keyboard.

Clearly a prompt is needed to enter a number. Easy enough – change line 10 to:

```
10  INPUT "Enter an Integer
    (>0): ";X%
```

But suppose the user forgets, or is awkward and enters –12.03, or enters an O in place of zero or enters a number greater than 32768. To prevent these potential errors is quite a headache for the programmer. A great many extra lines of code are required and the whole input routine slows down.

Furthermore, suppose the output is desired to three decimal places, correctly rounded and neatly justified with all trailing zeros printed . . . Not easy if all possible Applesoft real numbers could be printed.

Now a quick and easy way to cope with these programming problems – and a great many more – has been provided by Kelly Puckett and Penguin Software. The software is named ShortCuts and it joins others such as &Array and the Routine Machine in using the ubiquitous Applesoft & command to control your Apple.

ShortCuts is mainly concerned with easing the programmer's burden in controlling I/O, but it also provides some sorting capabilities.

The system is powerful, but not necessarily very easy to use initially. I counted 43 new commands, most of which interact with each other. But it is easy to implement.

While creating and editing your program ShortCuts can sit under DOS or under most other utilities which usually sit in that memory space, and is protected by HIMEM: which is changed automatically.

The complete, running system then occupies 4602 bytes, but two shorter versions – one 3935 bytes long which handles just the I/O routines, and the other just 682 bytes long for sorting only – are available.

Actually ShortCuts can be sat at most locations and when your program is completed it is easily appended to the finished article so that both become one, with ShortCuts protected by LOMEN:.

What will ShortCuts actually do for you? I cannot fully describe possibilities, but perhaps can illustrate some of them. To start let's describe some of the many commands by going back to our two lines of Basic, and change them to:

```
10  &INPUT "Enter an
    Integer (>0): ";X% =
    > 0 AND X% < 32768
15 X1 = SQR (X%)
20  &PRINT , PDL (3),X1
```

and if ShortCuts is operative we will have gained our objectives.

Furthermore, with a few more simple commands we could have set input and output field lengths, have made the input field appear in inverse so that the user would have no doubt as to the required number of digits, have allowed the user to enter not a number but a string which will be evaluated, have right or left justified the output . . . and so on.

If the user now enters a negative number or one greater than 32768 or a decimal and presses Return, an error message – which can be set by the programmer – will be put on the screen and the program will not proceed until the space bar is used to acknowledge the error, when the number may be re-input.

Equally, by using the statement &ABS, only positive numbers could be input, an attempt to type '-' would elicit an immediate error message requiring the space bar's use to carry on.

String handling is also supported by ShortCuts. The range of allowable input characters can be set and a format mask can be set so that strings such as telephone numbers can be entered and at the appropriate points delimiting characters will appear.

For example, a postcode number which we want to be input in a form compatible with SK7 5NY can be controlled with lines such as:

```
10 PC$ = "LLD DLL"
20  &INPUT , FOR (PC$),X$
```

The Ls in line 10 will allow only characters A to Z and the Ds will allow only the characters 0 to 9. On input the space will be printed automatically in the right space and the target variable X$ will contain "SK75NY", that is without the space. To print the postcode correctly just use the statement &PRINT,FOR(PC$),X$.

I think that if only for these kind of utilities ShortCuts is worth its cost – but there is more.

A table of control characters can be set up – for example, the command &CONT"EGPT" does this – which can then control the program flow. The command &ON..GOTO.. uses the position of the control character in the table to branch to a set line number (Esc always has the value 0).

Thus in my example pressing Esc could be used to go to the main menu, Cntrl-E to exit the program, Cntrl-G to display a graphics screen, Cntrl-T to redisplay the text and Cntrl-P to switch on the printer. In a similar way a choice of subroutines rather than GOTOs may be made by the user.

Each of the I/O controlling & commands so far described can be set globally. They are operative until changed, or can be set locally – that is, they act with new values only in that line of Basic and then the global values are again valid.

Whether they are global or local depends merely on their position in the line. Furthermore, commands can be strung together in one line of Basic.

Commands will also interact

with each other and it is this which can cause the first-time user some problems. For example, with two decimal places allowed by the command &PDL(2) and an input field allowed of four places only numbers between −9.99 and 9.99 can be input. Attempting to enter two digits without a decimal point will elicit an immediate error message.

If the commands are near to each other in the program there is no problem in understanding the formatting. However if they are far apart there can be difficulties in debugging.

Unfortunately, there is no way of listing the values of the formatting commands in operation at any one time. Luckily I found that I quickly became used to the likely source of such problems and have had no real problems in debugging.

The error messages are printed on the bottom line of your video display. If you want a line of text to be displayed here ShortCuts will remember your text and redisplay it after the error message is acknowledged if you wish.

All commands are usable on both 40 and 80 column displays. When first invoked ShortCuts determines the display in use and makes the appropriate decisions about allowable I/O.

I have used the routines with both Videx and Vision-80 80 column cards and there were no major problems. The display device can be altered within the Basic program if required and ShortCuts informed of the change by a simple command.

The Vision-80 didn't like the &HOME command which has been provided for those cards which do not support HOME (which the Vision-80 does). The Videx did use the &HOME correctly, but it was rather slow and annoying in appearance.

The Videx was fitted with inverse character capability but ShortCuts' &INVERSE command would not work on it, nor did the error messages appear highlighted as on 40 column display. Text did print in inverse on the Vision-80.

Because the Videx doesn't recognise Basic commands such as HOME, ShortCuts' loader program is very untidy which tempted me to run it under normal 40 column display and then switch to 80 columns. This is okay if one remembers to inform ShortCuts of the changes. It is disastrous if one doesn't.

Both cards, and presumably all 80 column cards, do not support the &SAVE command. This will save a video screen to another area of memory and then allow it to be recovered with the &RECALL command, but it only moves 1k of memory.

This 1k is squeezed between the end of the program and LOMEM: – it's a pity that the language card RAM could not be used because memory is likely to be tight down in normal RAM space.

The I/O formatting commands are compatible with disc and printer I/O but generally a preliminary command has to be issued to suppress certain of ShortCuts' video-orientated output.

Other utilities are provided by the command &&. If an ampersand utility is in memory when ShortCuts is initialised this && command will cause an immediate jump to it. If the old ampersand vector did not point to page 3 and similarly the USR vector did not point to page 3 three more routines are loaded there when ShortCuts is initialised.

These are invoked by simple CALLs and allow the Applesoft ONERR stack fix, more or less total disablement of the Reset key and re-enablement of the Reset key.

I mentioned that ShortCuts is capable of data sorting. The command &LIST 1 TO N, A% will sort the first N elements of the integer array A% into ascending order. The command &LIST N TO 1, A% would sort into descending order. Similarly, real arrays and string arrays can be sorted.

The sorting is quite rapid. Sorting 1,000 randomly selected real numbers took 10 seconds, 1,000 single character strings took seven seconds.

While on the subject of time, I measured the time to print on the screen 1,000 real array elements, each rounded and displayed to three decimal places and right justified in the output field. This took 1 minute 36 seconds, which is quite impressive when compared to the time taken (34 seconds) just to output the same numbers in the usual higgledy-piggledy way that Basic has.

"I counted 43 new commands, most of which interact with each other"

Actually when sorting, a further integer-only array can be co-sorted. This is useful for keeping track, say, of disc records or any sort of indexing. Thus sorting on multiple keys is also possible, but some ingenuity on the part of the programmer may be required.

All of the useful routines described cannot, of course, be obtained without some expense. So what will it cost you?

First memory – between 682 and 4602 bytes as described earlier.

Second, calculations carried out within the &PRINT statement, in other words lines such as &PRINT SQR(X), will be flagged as SYNTAX ERROR.

Third, only one variable can be printed in a statement. Multiple variables separated by commas and semi-colons are not allowed, although strings – either literal or expressions – can precede the variable.

Fourth, variables are actually altered where necessary. For example, the lines

```
10 X = 1.43786
20 &PRINT , PDL (2),X
```

actually changes the value of X from 1.43786 to 1.44.

Fifth, ShortCuts may be partly, or even wholly, incompatible with your favourite Basic editing program (you do use one don't you?). The manual says that GPLE (Synergistic Software) appears wholly compatible with ShortCuts, PLE (also Synergistic Software) suffers only from the fact that it won't pass Esc on to ShortCuts and so it is difficult to test program branching when Esc is required, ES-CAPE (S-C Software) on the other hand disconnects ShortCuts which has to be reconnected by a couple of POKE's.

I tested ACE (Southwestern Data Systems), finding the two to be mutually incompatible, and also The Developer (Leicester Computer Centre), which appeared to behave perfectly (using the && command).

Sixth, the interaction of commands can mean that one line misbehaving could be affected by one very far away and hence difficult to track down. As mentioned above, the values presently in effect cannot be listed.

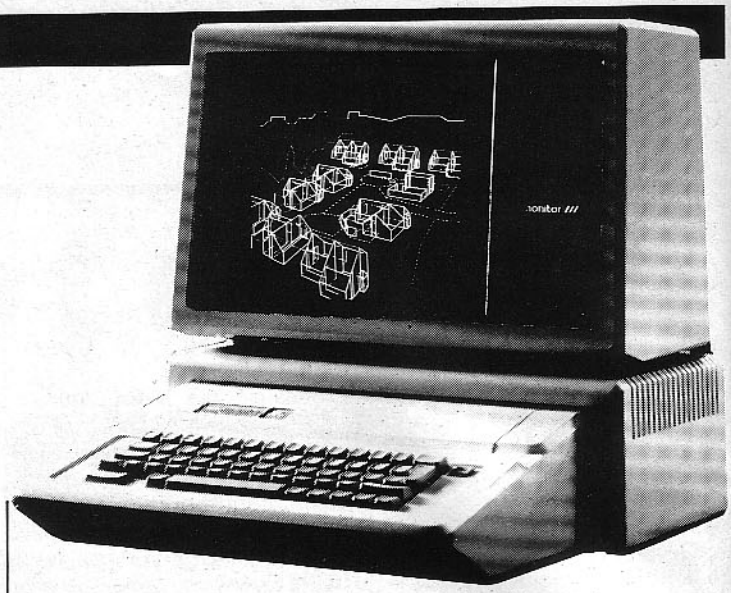
However a short perusal of the slim but adequate manual and of the sample programs provided on the non-copy protected disc soon make one feel at home with these extra commands. I have not as yet found any bugs in the system and certainly using ShortCuts saves much development time.

I do have one moan however. Why did it not appear a couple of years ago – it would have saved me so much time!

Title: *ShortCuts*
Author: *Kelly Puckett*
Publisher: *Penguin Software*

With architects in mind...

Design program for the Apple IIe reviewed by DAVID HAUGHTON



SCRIBE is a three dimensional modelling and drawing system developed by architect Cedric Green for use on the Apple IIe by the architectural profession in private practice, education and research.

The system is built around a three dimensional graphics program which is capable of generating full perspectives and all standard orthographic and three dimensional projections on a model containing up to 720 planes.

It can also produce plans and elevations which can be worked up into production drawings.

The screen simulates a drawing board and has facilities for overlaying a second page. The user specifies the scale – usually based on a modular grid – and the drawing is entered by means of hand controls for vertical and horizontal cursor movement.

Exact dimensions can be entered via the keyboard and curves are drawn using a polygon construction of short straights.

Drawings are generally in plan with heights of planes specified numerically. Alternatively, drawings in section or elevation can be used to insert items such as doors and windows.

Pitched roofs can be drawn in plan and tilted in space to the required slope. Each storey level can be separated as an overlay, including basements and foundations.

Once drawn the model can be viewed from any direction as a perspective, isometric or axonometric projection with the facility to remove hidden lines.

Every line or plane can be given a code number which may refer to its specification in terms of cost, thermal characteristics, mass or any other property on which a calculation may be performed.

A database allows the calculation modules to compute lengths, areas and volumes, and to calculate the overall characteristics such as cost, mass and thermal efficiencies.

Links with other programs for the Apple system are gradually being created. Enscribe is the first to be available and links with the Energy/1 package by Richard Twinch.

Energy/1 calculates 'U' values, predicts condensation risk and provides the basis for decisions on levels of thermal insulation required to satisfy building regulations.

Enscribe provides the link with Energy/1 through the Specalc and Spielcalc modules of Scribe.

It will load files created in these modules and allow the user to edit them sequentially or randomly from files used in the database of materials and their attributes derived from Energy/1.

The package is currently being improved by adding more program modules. There are 14 modules at present.

Scribe has been developed by Cedric Green over the last three years and it is claimed that there are currently more than 80 users.

Scribe's strengths are in the production of three dimensional drawings, while true production drawings await development of automatic dimensioning and shading facilities.

However Scribe does incorporate most of the facilities associated with the larger mini and mainframe based systems, yet at a very much lower cost.

Product: Scribe
Type: 3-D modelling and drawing system.
Price: From £900.
Distributor: CIC.

appletip

THERE may be times when you have a real variable in machine code and wish to locate it.

Apple uses five bytes to hold numbers. The first byte is most significant and centres, in hex, at 80 which is 0.5. Every unit increase above 80 doubles the number: 84 is 8 and 8F is 2 to the power 14, and so on.

Every unit decrease below 80 halves the number. For instance, 7F is .25.

The next byte holds 128ths of the first. If the first is X then X + X/2 gives a second byte of 40 (in hex) which is 64/128. The other three bytes hold descending orders of significance in a similar way.

The available range of inputs is from -1.7E38, which gives FF FF C9 9E 41, to 1.7E38, which gives FF 7F C9 9E 41.

Note that the second byte only runs up to 7F for positive numbers, and that 80 to FF is used for negative numbers.

The following short program, called Hexnos, enables you to enter a variable and then shows what you have to look for in machine code.

K. Archer & J.O. Gill

```
10  REM    HEXNOS BY ARCHER
    & GILL
20  HOME : HTAB 5: VTAB 8
30  PRINT "HEXNOS:STORE OF
    VARIABLES"
40  PRINT " ENABLES READOUT
    OF APPLE'S CODE"
50  PRINT : PRINT "
    3023409 GETS 96 38 88 C4
    00"
60  LOMEM:4094
70  PRINT "        ENTER 0 TO
    END"
80  PRINT
90  INPUT "ENTER NO.: ";X
100  IF X = 0 THEN  END
110  S = 24576
120  IF D = 28 THEN 160
130  FOR D = 1 TO 27: READ J
140  POKE S,J:S = S + 1:
    NEXT
150  VTAB 14
160  HTAB 26
170  CALL 24576
180  CV =  PEEK (37): POKE
    37,CV
190  GOTO 90
210  DATA
    169,0,141,6,96,173,5,16,32

220  DATA
    218,253,238,6,96,162,1,32,
    74
230  DATA
    249,173,6,96,201,5,208,235
    ,96
220  DATA
    218,253,238,6,96,162,1,32,
    74
230  DATA
    249,173,6,96,201,5,208,235
    ,96
```

Keep track of your records

ROGER HERON checks out the Superfile database management system and finds a few pitfalls

MICROS are playing an increasingly significant role in the way records are maintained and used in small businesses and offices throughout the country.

The need is for cheap and efficient ways to manage information such as customer details, membership lists, financial transactions, stock inventories, mailing lists, bibliographies, newspaper rounds and patient histories.

In response, the software industry has spawned an extensive range of database management systems, many of which are available for the Apple II.

The problem for the purchaser is to find the right system to suit his particular requirements.

Will a simple menu driven card index program be enough, or would it be better to look for something more sophisticated and flexible?

The trade off is usually one of power against simplicity and ease of use.

On the surface Superfile appears to offer the best of all possible worlds. It seems to be almost infinitely flexible, but can also be put to immediate use as a straightforward card index system.

Among the particular advantages claimed by Southdata for Superfile are that it is:

Efficient: Disc storage requirements are cut by 50 per cent or more, arising from the use of variable length records.

Flexible: Record structures can easily be changed without destroying existing data.

Adaptable: The system can be attached to any standard programming language.

Easy to operate: Menu-driven screen form and report programs are supplied to enable the system to be put to effective use immediately.

User friendly: The underlying concepts are easy to grasp and use.

These are very seductive qualities. However in the course of testing Superfile for the purposes of this review, it quickly became apparent that it is crippled by a number of problems which place serious limitations on its usefulness.

Conventional systems have a fixed record structure and, when a database is set up, the user has to specify the number of character spaces which must be made available in every field of every record.

Disc space will therefore be wasted if a record field is either not used or is not completely filled by the information stored in it.

Superfile claims to avoid this problem by adopting a quite different approach. It stores the name of each field used within the record itself, and uses sentinel characters to mark the ends of fields and records.

However despite Southdata's assertions, disc storage requirements will not always be substantially reduced. On the contrary, there will be some cases where a significant increase may arise.

The reasons for this are two-fold:

- Space is consumed by the field names and sentinel characters which are stored with the data in each record.
- Because some Ascii values are reserved for use as sentinel characters it is not possible to use the more compact format for storing numerical data which is used, albeit at the expense of a degree of accuracy, by other systems.

A Superfile-created database consists of main and overflow files.

The main file will include records which have been deleted, together with the previous version of records which have been amended since the last "tidying" operation.

The overflow file consists of records which have been added, along with the new version of records which have been amended since the database was last tidied.

One or more index files are associated with the main file but not with the overflow file.

The standard index file contains an index to every field in every record. If its size slows the record search process down unacceptably it is possible to create smaller index files, based on a more limited number of fields, to suit the particular application in question.

Apart from the fact that records in the overflow file are not indexed, and can therefore only be found by a slow sequential search, this all sounds fine. However there are some serious difficulties in practice:

☐ If there is insufficient additional disc space to allow the tidying operation to be carried out the system will carry on regardless and crash. The database will then be left in an unusable state. This is completely unacceptable in any serious business application.

☐ The tidying operation deletes every single index file held on that disc. This includes not only those which relate to the database in use but also the indexes to any other databases which happen to be held on the same disc.

☐ A "search and amend" operation can be badly hampered by the way the system deals with record amendment. The same record will be encountered twice, and unless the user is very careful indeed the system can be thrown into an infinite loop.

☐ The indexing system decides whether a field contains text or a number on the basis only of whether the first text word with more than two characters contains more numbers than letters.

Moreover data will not be indexed at all unless there is a word in the field which is more than two characters long. This can cause considerable inconvenience to the user.

Superfile's database management functions can effectively be added to one's own favourite high-level computer language. Southdata automatically provides an interface to Microsoft's Mbasic-80 with the system, and offers to supply interfaces to any other standard language free on request.

In theory this is a tremendou-

appletip

Tricky problem solved

Users of the CIA programs may have experienced an annoying problem when disassembling a sector to printer with Tricky Dick. Unless the last instruction ends at $FF – that is, the next opcode starts at $100 – Tricky Dick will throw out approximately four pages of BRKs. This problem can be fixed by patching Tricky Dick as follows:

```
CALL -155
BLOAD TRICKY DICK
ICBA:C0 2F
BSAVE TRICKY DICK,A$0803,L$3800
```

Greg Elkin

sly powerful feature. However there are a number of problems with the interface to Mbasic-80 which considerably detract from its usefulness in practice.

For example, some of the functions which are listed in the manual are not included in the skeleton Basic program provided and do not work even if they are added by the user.

To add to the confusion some of the specimen programs provided in the manual have fundamental errors in them. One instance is a program for converting sort find-pointers to 8byte numbers to enable calls to be made to the Superfile system after records have been sorted.

The names of the variables have been incorrectly printed and a GOSUB call is made to the wrong line number.

Southdata need to do a great deal more work on the system if they really want to give credence to their claim that "the unique ability of Superfile to interface to most languages means that people with only a few months' programming experience can soon be writing professional quality software".

At present Superfile is certainly not a friendly programming tool for the novice, and is likely to give most experienced programmers a severe headache.

To save the user all the trouble of writing his own programs in order to be able to use Superfile in simple applications, Southdata provides two menu-driven packages, Superforms and Supertab.

Superforms enables the user to design a range of screen forms to access a database for the purposes of data input, amendment, deletion and display.

Supertab allows tabular report formats to be specified, so that user-designed reports can be sent to the screen or the printer.

The associated Sort program allows data in the reports to be sorted on up to 36 fields at one time.

To create a screen form the user paints the screen with the wording required and marks spaces in which information is to be entered or displayed when the form is used.

Superforms then prompts for the field names to be associated with the marked spaces.

For each field a useful range of attributes can be specified which will automatically validate data for accuracy as it is subsequently entered.

Fields can also be set so that the result of a calculation is displayed. Once created and saved, a form can later be edited if necessary.

To find a record or records, whether for screen display or for inclusion in a printed report, a model is entered by the operator on a screen form. Selection can be made on any or all of the fields in the records.

Wildcards for part of any field can be used, and a limited form of the "sounds-like" function is available.

Ranges of numbers can be specified for numerical fields, as can a list of possible matches for all types of field.

Such straightforward menu-driven programs will inevitably have limitations, but Superforms and Supertab also fall well short of acceptable standards of user-friendliness for a product of this price. Here are just a few examples:

- There are no menu options to allow the operator to move directly from module to module.
- It is not possible to change the database in use from within either Superforms or Supertab.
- The screen display following selection of the "Load form" menu option fails to inform the operator how to obtain a directory listing of screen forms held on other disc drives.
- The list of options at the foot of each screen form is incomplete and sometimes misleading. There is no prompting at all to assist the operator in entering models to find records.
- Data entry is made unnecessarily awkward by the fact that it is not possible to move the cursor backwards through the fields on a screen form.
- The operator is allowed, without receiving any warning, to overwrite calculated fields with data entered from the keyboard.
- There are certain characters — <, >,? =, *@ and % – which should not be placed in a record because they are used to perform functions in the record search process. Superforms does nothing at all to prevent the operator from entering them.
- When a record is amended the system marks it as erased and adds a new record at the end of the database. If the erased record is later "unerased" there is nothing to warn the operator that an updated version of the same record exists elsewhere in the database.
- Control of the printer is inflexible. Supertab insists on initiating a form-feed before each report, and there is no facility for sending control characters to the printer in mid-report.
- There are considerable limitations on the form that a report can take. For example the data to be printed from each record must fit on one report line.
- Insufficient attention has been given to the particular needs of Apple II Plus users. The common "shift-mod" method of enabling lowercase keyboard input is not supported.

Nor is any assistance given to the novice Apple user in relation to the need to configure CP/M to allow keyboard input of the ':' character, which is necessary to allow "stay-puts" to be specified on a screen form when records are added.

The fundamental concept behind Superfile is excellent. Make a powerful set of database management commands available. Allow those commands to be used either by incorporation in programs written in a high level language, or by means of a menu-driven index-card package supplied with the system. Make that system fast, flexible, economic on storage space, and user-friendly.

Unfortunately the implementation of this concept by Southdata is a major disappointment. Although the system they have come up with is not without some merit, its limitations will considerably outweigh its advantages in most prospective applications.

Superfile's user-interface could certainly be cleaned up, probably without great difficulty, and it would then become a much better product.

The manual, which is inadequate and riddled with errors, could definitely be improved.

However, there are a number of drawbacks to the basic design which would make it difficult to raise the whole system to the standard required to give it wide appeal.

My advice: Check it out very carefully before you buy.

Product: *Superfile*
Price: *£375*
Description: *CP/M-based database*
Distributor: *Ranmor Computing, 14 Nelson St, Southend-on-Sea, Essex SS1 1AL.*

appletip

MANY routines are available to obtain the memory location from the horizontal and vertical co-ordinates of the hi-res graphics screen. However this information can be obtained from within the Applesoft interpreter without using specially written machine code routines.

When using hi-res graphics Applesoft makes use of the following locations and routines to obtain the memory location and the bit position of the point just plotted (or proposed to be plotted):

Absolute memory location:

lsb = ($26 + $E5)

Absolute memory location:

msb = ($27 ORA $E6)

Bit mask:

= $30

In order to examine a point without plotting, use the HPOSN routine at $F411, place horizontal coordinate in Y, X registers and vertical coordinate in Accumulator, do a JSR $F411 then the byte of interest can be examined with LDA ($26), Y.

If this is EORd with $1C (colour mark) and ANDd with $30 (bit mark) the appropriate bits are isolated.

R.A. Royall
South Cleveland Hospital

Apple II laps up the transport business facts

EDDIE and Joan Farrall have been in haulage for 28 years, ever since they bought their first wagon – a second-hand Bedford 0 type 5-6 tonner back in 1956.

In those frenetic days it was all odd jobbing, short haul journeys, carrying farm produce and building materials – anything to make a buck or two.

Since that time the business kept growing steadily. Today there are 24 on the payroll, including their two sons, Mike and Mark, and 18 Mercedes 1625 32 tonners to keep busy.

Up till a year ago the office work was handled manually by Joan Farrall and transport manager Ron Parkinson. However with the rapid growth of the business there simply weren't enough hours in the day to hold back the flood of office work.

Decision time had arrived. An extra pair of hands in the office would mean building extra accommodation. On the other hand a computer could sit in the corner of the existing offices without too much hassle. The computer won hands down.

The only problem was that nobody in the business really knew too much about computers. Luckily they were aware of the pitfalls awaiting them if they didn't research the market thoroughly before buying.

The mind-boggling selection of hardware, software and peripherals presents such a formidable range of choice that most uninformed potential computer buyers don't. Jargon is also daunting to the uninitiated.

A well meaning friend suggested an IBM mini might help out. A demonstration was arranged which unhappily only confirmed Ron Parkinson's suspicion that they didn't really know enough to go out and spend £10,000 or more on computer, software and peripherals.

They talked to several computer users and read a great deal more on the subject. Gradually it all became clearer.

Joan Farrall and Ron next attended a Kalamazoo demo which helped them gain confidence. They liked the system, but felt it was a little expensive.

The demonstration was convincing and jargon-free. It also served to emphasise the diversity of contribution that the computer could make to the business.

Another friend who worked for Northern Computers in Frodsham proposed an Apple. A further demonstration was set up and this led to a deal.

The Apple II, three disc drives and an Anadex dot matrix printer were delivered with Jarman software covering PAYE, sales accounts, purchase accounts and nominal ledger.

The complete package cost Farrall Transport £5,000 – about a third of the cost of some of the alternative equipment they were offered.

It wasn't all plain sailing however. Several times during the first three weeks Joan Farrall felt like throwing the whole shooting match out of the window.

However, she persevered working with the computer from time to time while carrying on with the manual systems. Gradually the computer took more and more of the workload.

Both Joan Farrall and Ron Parkinson have learned to use the Apple. They consider it essential that each should be capable of handling all the computer operations in the other's absence.

They have also introduced a fail-safe system for the storage of discs – two duplicate sets being stored in different locations in case of fire or loss.

The PAYE software was the first system to be implemented. The discs also hold all personnel records in addition to the PAYE capability.

Typically the manual PAYE exercise used to take two or three days in between phone calls and other necessities of office routine. With the Apple the whole job is completed in one hour.

According to Ron Parkinson, queries for claims, letters and queries on loads just used to stack up while the manual accounting systems were in operation. This doesn't happen any more.

The Apple has changed the whole ball game. If you ask Farrall Transport today what they think of the computer you'll receive the classic reply: "However did we manage without it?"

Because of the Apple, Joan has a lot more free time to pursue other activities like horse riding, but is still able to keep in touch with the business.

Following the PAYE software, the next two programs to be introduced were the sales and purchase accounts. Invoicing is on high quality pre-printed continuous stationery giving an average throughput of 100 invoices in half a day.

Joan Farrall and trusty Apple II . . . "However did we manage without it?"

All the invoicing is prepared on a Monday to cover the previous week's business. The sales journal printout also provides a running report on weekly totals.

Previously when invoicing had been completed it was necessary to transfer the details to a sales ledger. With the computer the whole job is a one-shot operation, invoice information being available in ledger form from one input of data.

Next job for the computer is the nominal ledger but Farralls have been somewhat thwarted from introducing this system by the work of the company's auditors whose presence has held things up temporarily.

Joan intends to put Farrall's current financial year figures into the nominal ledger as soon as possible. To use her own words: "This will mean that experience and intuition can be replaced by hard copy from the computer".

"There's no way we would ever return to a manual system"

Does she now think that she is a computer expert?

"Not a bit – but I know enough about this machine to appreciate the major contribution it's making to our workload and I'll make sure that it continues to do so.

"For what it cost us the computer has been a minor miracle. We now manage to do everything in two days per week that previously took us five or more.

"The machine is so reliable. It's always sitting here when I arrive and here when I go", she said.

Are there any further uses for which the Apple could be used to streamline the business?

"We could computerise vehicle costs, wages, fuel and tyres. Some haulage contractors might find this desirable but we really haven't time, and our experience in the business does count for a lot.

"We know when a vehicle is not being operated profitably and then take the appropriate measures.

"We recently looked at a system to record tachograph information. It might be useful but would need to be completely automatic rather than requiring us to input all the information into the computer manually".

What about breakdown with the computer or peripherals. Is this a problem?

"We had a minor problem with the printer some months back but Northern Computers soon sorted that one out and we now have a maintenance contract with them.

"If any part malfunctions they will be over to repair it or replace the offending item. Touch wood, we haven't needed to call them".

Would they recommend other haulage contractors to buy a computer?

"We certainly would. There's no way that we would ever return to a manual system"

apple classifieds

APPLE II EUROPLUS 64k with two Apple II disk drives – each with control card. "Zenith" Monitor. Clip-on Fan. "Videx" videoterm 80-column card & switch. "Snapshot 2" program & card. Excellent condition. Offers around £650 considered. Tel: 0604 408798.

APPLE II EUROPLUS 48k disk drive with controller, Silentype printer with interface, Pal colour card, paddles, joystick and IIe joystick adaptor as new £750. Tel: 0203 325213 evenings.

APPLE II ACCESSORIES disc drive/controller £175, Grappler+ £40, 16k Microsoft £35, 16k RAM £25, Colour card/modulator £45, Versawriter £70, Visicalc £40, Applewriter £30. Tel: 0203 382861.

TABS Business Software, Sales, Purchase, Nominal ledgers, Stock control, Job costing, Payroll, Firmware card and manual. New cost over £2,000. Best offer over £350. Tel: Oxted 2891.

CLOCK CARD £65, S. Serial £65, Wildcard £50, Screen writer £60, General manager database £75, Visicalc £50, Visiplot trend £80. 47 Ormiston Grove, London W12.

WANTED Apple II+ with drives and monitor etc. If you also have a printer with graphics state your price. Tel: Alan 0734 868147 (work) or 0235 23663 (home).

SILENTYPE printer plus interface, Nine rolls of paper. Tel: 051-678 4279.

WORDSTAR and Micromodeller with manuals for Apple II+ accept any sensible offer. Tel: 0371 850254 evenings.

VISICALC for Apple II or IIe, immaculate £60. Also Applewriter II £25. Tel: 0582 63666.

IIe selling everything. To pay for Mac.. Software, Hardware. Tel: 0932 56956 for list.

IIe selling everything. Must go. Still under guarantee, Software, Hardware. Tel: 0932 56956 for list.

QUANTITY of brand new Apple IIe cases at £30 each. Tel: 0604 842175 after 6pm.

TEC STARWRITER daisywheel printer. 40 cps plus cable and interface. Fully compatible with Wordstar and Applewriter II. £465. Tel: 0732 845644.

apple classifieds

- Classified ads can only be accepted from private readers, not companies.
- The cost is 20p per word, with a minimum of 10 words prepaid.
- Your ad will be printed in the next available issue of Apple User.
- Your accompanying cheque should be made payable to Apple User.
- Ads can only be accepted on this form (or a photocopy of the form).
- There is no maximum to the number of words you include in your ad.
- Ads too long for the form should continue on a separate sheet of paper.
- Ensure your phone number or address is included in the ad.

					10 words £2.00
					15 words £3.00
					20 words £4.00
					25 words £5.00
					30 words £6.00

Cheque enclosed for £_____

Name_____ Address_____

POST TO: Apple Classifieds, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Guide's errors reflect bad planning

Apple Basic: Data File Programming. A self-teaching guide by LeRoy Finkel and Jerald R. Brown (£8.95).

THIS is a strangely old-fashioned book. The authors, for reasons which they clearly explain, have gone against the current trend towards long variable names and confined themselves to one or two character ones.

Bearing in mind that not all Basics allow longer names and even Applesoft only notices the first two characters, this makes a lot of sense.

You end up with an economical, self-consistent program which you have some hope of being able to modify to run on another machine.

Naturally they are not against structured programming. In fact they give a great deal of good advice on structure, especially on the careful and systematic use of REM statements to divide the programs into logical sections, and to list and explain all variables used.

This is all explained in the first chapter, and the same approach is used throughout the book.

The authors include a major section on data entry – how to get the information in, check it is the right type, and adjust its length to suit the record structure you are going to use.

You are shown how to store and recall data using sequential files, how to add new data to the end of a file, how to change records and how to merge two files and so on.

The rest of the book deals with random access files, all necessarily specific to Apple DOS 3.2 or 3.3.

The book is called a self-teaching guide and follows a pattern where a short passage of explanation is followed by one or more questions, with spaces for the answers to be written in. The suggested answers follow immediately.

At the end of each chapter is a self-test, and again the answers follow immediately.

The same approach applies even to the programming exercises. This is fine for the college student who owns the book but I am a bit doubtful of its desirability if more than one person is to use it.

The price is slightly high for an expendable workbook.

As a book it is not readable. I tried, and got about half-way through, and one of my colleagues also gave up.

I think it would be much more effective if one had the time to sit down and work through everything continuously with the computer at hand. It might suit the business user learning programming, or a school with a lot of Apples.

Unfortunately there are several errors. A few are trivial misprints, but many reflect insufficient experience with the Apple or bad program planning.

For examples of the former, the authors suggest testing for a null string input (someone just pressing 'Return') by:

```
IF ASC(A$)=0 THEN ....
```

but there is a bug in Applesoft which produces an error if you apply ASC to a null string. To test if the first character in a string is a number other than 0, they suggest:

```
IF VAL(A$)<>0 THEN ....
```

which ignores the possibility of A$ starting with 0.2 or even 007.

Page 90 suggests that a semicolon can be used instead of a comma for separating numeric data items in a file, which is not true. However the authors fortunately do not recommend this method.

There is a lot of confusion on Page 91 about the Applesoft string INPUT statement and storage of the resulting string on file if the string typed in contains commas.

In fact the sequence:

```
LET B$="PUBLIC, JOHN Q."
PRINT D$; "WRITE FILENAME"
PRINT B$
```

(where D$ is a Ctrl-D) results in the whole of B$ being written to the file and does not produce an error message. The error occurs when that record is read back from file using an INPUT statement.

If, however, the top line is rewritten as:

```
INPUT B$
```

it is in fact possible to type in:

```
"PUBLIC, JOHN Q."
```

when the program is run, using the double quotes, and this does not result in an error message.

However it is inadvisable to have keyboard input and writing to a file in the same section of a program, and clearly the authors have encountered these errors in their experimentation and misunderstood the causes.

They do not mention the problem of string inputs containing colons, where anything after the colon is simply ignored without an error message.

The handling of error conditions in many of the example programs is potentially disastrous. Typically, the program tests to see if the error was "END OF DATA". If not, it reports an "unusual error", and then frequently goes on as if no error had occurred.

Thus the actions appropriate to reaching the end of a file will be carried out if you have a syntax error in that part of the program, or even if you press Ctrl-C.

For example, the program fragment on Page 98 will loop forever printing "UNUSUAL FILE ERROR. PROGRAM TERMINATED." if you leave the disc drive door open.

This is trivial by comparison with what could happen on Page 151, which deals with amending a sequential file by changing the contents of one record.

The method used is to read from the existing file, say CUSTOMERS, until the entry is found, writing each record to a temporary file TEMP.

The desired record is then amended and written to TEMP, followed by all the remaining records in CUSTOMERS. The files are CLOSEd, and then CUSTOMERS is DELETEd and TEMP is RENAMEd CUSTOMERS.

Suppose there is an error during the last stage of writing to TEMP – perhaps someone opens the drive door by accident. The program then attempts to CLOSE both files, and becomes locked in a loop emitting "UNUSUAL ERROR" messages and trying to CLOSE ... until the operator notices and shuts the drive door.

At this point the CLOSE can be done successfully, and the program proceeds to DELETE your CUSTOMERS file, although not all of it has been copied to TEMP.

I find all this rather depressing. I don't feel that error handling should be taken lightly.

You should never use ONERR GOTO unless you are prepared to write failsafe (sometimes called failsoft) routines for all possible error situations, and at all costs the original file should be preserved when anything goes wrong.

In view of the likely incidence of syntax errors, etc, in real application programs, the error handling should be omitted until the program is fully debugged.

There are even program fragments here where the only executable instruction is ONERR GOTO 900 (for instance), and line 900 does not exist. This particular one is on Page 168, and line 900 enters the story on Page 172.

The overall problem is one of bad design. The authors' sequential files do not have at the start a record which holds the number of records on the file. Neither do they use a "rogue value" after the last record to mark the end of the file.

They are therefore forced to trap the inevitable END OF DATA error when the program tries to read past the end.

It would be so easy to teach good file handling methods rather than perpetuating sloppiness.

You will learn a lot from this book. But for goodness sake, combine it with some reputable book on file design and input validation.

Hugh Dobbs

Listing II: Assembler

```
0800                    1           PAG
0800                    2  ; TEXT HANDLER
7000                    3           ORG $7000
7000                    4           OBJ $800
7000                    5  ;
7000                    6  ; LABELS
7000                    7  ;
0036                    8  CSWL     EPZ $36          ;OUTPUT HOOKS
0037                    9  CSWH     EPZ $37
001A                   10  SHAPEL   EPZ $1A          ;SHAPE TABLE POINTER
00F9                   11  ROTZ     EPZ $F9          ;TEMP ROTATION POINTER
001B                   12  SHAPEH   EPZ $1B          ;LB,HB FORM
00CE                   13  BASEL    EPZ $CE          ;ZERO PAGE LOCATION FOR BASE ADDRESS
00EB                   14  SHAPEPNT EPZ $EB
00E7                   15  SCALEZ   EPZ $E7          ;SCALE VALUE
00E4                   16  HCOLORZ  EPZ $E4
001C                   17  HCOLOR1  EPZ $1C
02F0                   18  TBUFF    EQU $2F0         ;USE INPUT BUFFER FOR TEMPORARY STORAGE
F730                   19  DRWPNT2  EQU $F730        ;ENTRY POINT INTO DRWPNT
F411                   20  HPOSN    EQU $F411
F605                   21  DRAW1    EQU $F605
7000 85 FF             22  START    STA $FF          ;STORE REGISTERS - SAME
7002 86 EB             23           STX $EB          ;PLACE AS TOOLKIT
7004 84 35             24           STY $35
7006 A9 00             25           LDA #$00         ;FORCE BRANCH
7008 F0 19             26           BEQ BEGIN        ;PAST DATA
700A                   27  ;
700A                   28  ; DATA BLOCK
700A                   29  ;
700A 00 60             30  TABLE    ADR $6000        ;SHAPE TABLE ADRESS
700C 64 00             31  COORDS   HEX 6400         ;X COORDS LB,HB
700E 64 00             32           HEX 6400         ;Y COORDS LB,HB
7010 00                33  ORIENT   HEX 00           ;DEFAULT HORIZONTAL
7011 00                34  INVERS   HEX 00           ;DEFAULT NORMAL
7012 00                35  BLANK    HEX 00           ;DEFAULT BLANKING
7013 00                36  UPLOW    HEX 00           ;DEFAULT UPPER
7014 00                37  FULLW    HEX 00           ;DEFAULT FULL SCREEN
7015 00                38  VERTM    HEX 00           ;DEFAULT SET BY ORIENT
7016 00                39  BORW     HEX 00           ;DEFAULT WHITE
7017 03 00             40  XMIN     HEX 0300         ;FULL SCREEN CORNER VALUES
7019 03                41  YMIN     HEX 03
701A 14 01             42  XMAX     HEX 1401         ;LB, HB FORM
701C BD                43  YMAX     HEX BD
701D 03 00             44  XMINW    HEX 0300         ;WINDOW VALUES DEFAULT
701F 03                45  YMINW    HEX 03           ;IS FULL SCREEN
7020 14 01             46  XMAXW    HEX 1401
7022 BD                47  YMAXW    HEX BD
7023                   48  ;
7023                   49  ;PROGRAM START
7023                   50  ;
7023 A9 0A             51  BEGIN    LDA #TABLE       ;DUMMY ADDRESS - MUST
7025 85 CE             52           STA BASEL        ;BE SET FROM BASIC
7027 A9 70             53           LDA /TABLE       ;BEFORE CALLING THIS ROUTINE
7029 85 CF             54           STA BASEL+1      ;ZERO PAGE BASE SET
702B                   55  ;
702B                   56  ;CHECK IF X,Y COORDS CAN BE PLOTTED
702B                   57  ;
702B A5 E7             58           LDA SCALEZ       ;SAVE SCALE,ROT AND HCOLOR
702D 8D FD 02          59           STA TBUFF+$D
7030 A5 F9             60           LDA ROTZ
7032 8D FE 02          61           STA TBUFF+$E
7035 A5 E4             62           LDA HCOLORZ
7037 8D FF 02          63           STA TBUFF+$F
703A A2 00             64           LDX #$00
703C A0 0A             65           LDY #$0A         ;OFFSET IN TABLE FOR
703E B1 CE             66           LDA (BASEL),Y    ;SCREEN/WINDOW FLAG
7040 F0 04             67           BEQ FULLSC
7042 A0 13             68           LDY #$13         ;POINT TO WINDOW VALUES
7044 D0 02             69           BNE WINDOW       ;FORCE BRANCH
7046 A0 0D             70  FULLSC   LDY #$0D         ;POINT TO SCREEN VALUES
7048 B1 CE             71  WINDOW   LDA (BASEL),Y    ;TRANSFER TO TEMP BUFFER
704A 9D F0 02          72           STA TBUFF,X      ;FOR TESTING
704D C8                73           INY
704E E8                74           INX
704F E0 06             75           CPX #$06         ;DONE YET?
7051 D0 F5             76           BNE WINDOW
7053                   77  ;
7053                   78  ; NOW THE TESTS
7053                   79  ;
7053 A0 05             80           LDY #$05         ;CHECK Y COORD
7055 B1 CE             81           LDA (BASEL),Y    ;Y HI-BYTE
7057 D0 76             82           BNE DONE         ;MUST BE ZERO
7059 88                83           DEY
705A B1 CE             84           LDA (BASEL),Y    ;Y LO-BYTE
705C CD F2 02          85           CMP TBUFF+2
705F 90 6E             86           BLT DONE         ;Y<YMIN
7061 CD F5 02          87           CMP TBUFF+5
7064 F0 02             88           BEQ OK1
7066 B0 67             89           BCS DONE         ;Y>YMAX
7068 88                90  OK1      DEY              ;NOW X COORD
7069 B1 CE             91           LDA (BASEL),Y    ;HI-BYTE FIRST
706B CD F1 02          92           CMP TBUFF+1
706E 90 5F             93           BLT DONE         ;X<XMIN (HI-BYTE)
7070 D0 09             94           BNE MAXT         ;NOW TEST XMAX
7072 88                95           DEY              ;TEST XMIN LO-BYTE
7073 B1 CE             96           LDA (BASEL),Y
7075 CD F0 02          97           CMP TBUFF
7078 90 55             98           BLT DONE         ;X<XMIN (LO-BYTE)
707A C8                99           INY
707B B1 CE            100  MAXT     LDA (BASEL),Y    ;TEST XMAX NOW
707D CD F4 02         101           CMP TBUFF+4
7080 F0 04            102           BEQ OK2
7082 B0 4B            103           BCS DONE         ;X>XMAX
7084 90 0A            104           BCC XYOK
7086 88               105  OK2      DEY              ;LO-BYTE TEST NOW
7087 B1 CE            106           LDA (BASEL),Y
7089 CD F3 02         107           CMP TBUFF+3
708C F0 02            108           BEQ XYOK
708E B0 3F            109           BCS DONE         ;X>XMAX (LO-BYTE)
7090                  110  ;
7090                  111  ;COORDINATE IS OK TO PLOT
7090                  112  ;
7090 A0 00            113  XYOK     LDY #$00         ;MAKE TEMPORARY COPY OF TABLE
7092 B1 CE            114  TLOOP    LDA (BASEL),Y
7094 99 F0 02         115           STA TBUFF,Y
7097 C8               116           INY
7098 C0 0D            117           CPY #$0D
709A D0 F6            118           BNE TLOOP
709C A5 EB            119           LDA SHAPEPNT     ;SAVE EB, EC
709E 48               120           PHA
709F A5 EC            121           LDA SHAPEPNT+1
70A1 48               122           PHA
70A2 AD F0 02         123           LDA TBUFF        ;SET UP EB, EC
70A5 85 EB            124           STA SHAPEPNT
70A7 AD F1 02         125           LDA TBUFF+1
70AA 85 EC            126           STA SHAPEPNT+1
70AC A9 01            127           LDA #$01         ;SET SCALE
70AE 85 E7            128           STA SCALEZ
70B0 AD FB 02         129           LDA TBUFF+$0B    ;CHECK VERTICAL MODE
70B3 F0 06            130           BEQ SETROT       ;USE ORIENTATION VALUE
70B5 30 04            131           BMI SETROT
70B7 A9 00            132           LDA #$00         ;FORCE ZERO
70B9 F0 07            133           BEQ SETROTT
70BB AD F6 02         134  SETROT   LDA TBUFF+6     ;ORIENTATION
70BE 0A               135           ASL              ;MULT ORIENT BY 16
70BF 0A               136           ASL
70C0 0A               137           ASL
70C1 0A               138           ASL
70C2 85 F9            139  SETROTT  STA ROTZ
70C4 A2 7F            140           LDX #$7F         ;COLOUR MASK FOR WHITE
70C6 AD FC 02         141           LDA TBUFF+$0C    ;BLACK OR WHITE?
70C9 F0 0B            142           BEQ SETIT
70CB A2 00            143           LDX #$00         ;COLOUR MASK FOR BLACK
70CD F0 04            144           BEQ SETIT
70CF                  145  ;
70CF A9 00            146  DONE     LDA #$00         ;FORCED BRANCH
70D1 F0 6B            147           BEQ DONE1
70D3 86 E4            148  SETIT    STX HCOLORZ
70D5 AD F7 02         149           LDA TBUFF+7      ;CHECK INVERSE FLAG
70D8 F0 06            150           BEQ BLANKT
70DA A5 E4            151           LDA HCOLORZ      ;SET INVERSE MASK
70DC 49 7F            152           EOR #$7F
70DE 85 E4            153           STA HCOLORZ
70E0 AD F8 02         154  BLANKT   LDA TBUFF+8      ;CHECK IF BLANKING SET
70E3 D0 22            155           BNE SETSHAPE
70E5 A5 E4            156           LDA HCOLORZ      ;SWAP COLOUR FOR BLANKING
70E7 49 7F            157           EOR #$7F
70E9 85 E4            158           STA HCOLORZ
70EB A2 60            159           LDX #$60         ;SET BLANK
70ED 20 30 F7         160           JSR DRWPNT2
70F0 AE F2 02         161           LDX TBUFF+2      ;SET UP COORDS
70F3 AC F3 02         162           LDY TBUFF+3
70F6 AD F4 02         163           LDA TBUFF+4
70F9 20 11 F4         164           JSR HPOSN        ;SET UP TO DRAW
70FC A5 F9            165           LDA ROTZ
70FE 20 05 F6         166           JSR DRAW1
7101 A5 E4            167           LDA HCOLORZ      ;SWAP BACK
7103 49 7F            168           EOR #$7F
```

```
7105 85 E4      169           STA HCOLORZ
7107            170 ;
7107            171 ; GET THE SHAPE NUMBER NOW
7107            172 ;
7107 A5 FF      173 SETSHAPE LDA $FF          ;GET SHAPE
7109 29 7F      174           AND #$7F        ;HI-BIT OFF
710B 38         175           SEC
710C E9 1F      176           SBC #$1F        ;NO CONTROL CODES
710E C9 01      177           CMP #$01        ;IS IT A VALID SHAPE?
7110 90 49      178           BLT DONE2
7112 C9 61      179           CMP #$61
7114 B0 45      180           BCS DONE2
7116 AE F9 02   181           LDX TBUFF+9     ;CHECK LOWERCASE FLAG
7119 F0 0A      182           BEQ PLOTLET     ;NOT SET
711B C9 1F      183           CMP #$1F        ;IS IT A LETTER?
711D 90 06      184           BLT PLOTLET
711F C9 3D      185           CMP #$3D
7121 B0 02      186           BCS PLOTLET
7123 69 20      187           ADC #$20        ;MAKE LOWERCASE
7125 AA         188 PLOTLET   TAX             ;SET UP TO FIND SHAPE
7126 20 30 F7   189           JSR DRWPNT2     ;SET SHAPEL,H TO POINT TO SHAPE
7129            190 ;NOW SET UP POSITION
7129 AE F2 02   191           LDX TBUFF+2     ;X REG = XL
712C AC F3 02   192           LDY TBUFF+3     ;Y REG = XH
712F AD F4 02   193           LDA TBUFF+4     ;A REG = Y
7132 20 11 F4   194           JSR HPOSN       ;SET UP POSITION
7135 A5 F9      195           LDA ROTZ        ;SET ROTATION
7137 20 05 F6   196           JSR DRAW1       ;DRAW SHAPE
713A A9 00      197           LDA #$00        ;FORCE BRANCH
713C F0 02      198           BEQ UNDL
713E F0 1B      199 DONE1     BEQ DONE2       ;LONG BRANCH
7140 AD FB 02   200 UNDL      LDA TBUFF+$B    ;CHECK UNDERLINE FLAG
7143 10 16      201           BPL DONE2       ;NO UNDERLINE
7145 A2 40      202           LDX #$40        ;UNDERLINE SYMBOL
7147 20 30 F7   203           JSR DRWPNT2
714A AE F2 02   204           LDX TBUFF+2
714D AC F3 02   205           LDY TBUFF+3
7150 AD F4 02   206           LDA TBUFF+4
7153 20 11 F4   207           JSR HPOSN
7156 A5 F9      208           LDA ROTZ
7158 20 05 F6   209           JSR DRAW1
715B A9 00      210 DONE2     LDA #$00        ;ZERO DX, DY LOCATIONS
715D 8D F0 02   211           STA TBUFF
7160 8D F1 02   212           STA TBUFF+1
7163 A0 0B      213           LDY #$0B        ;INCREMENT X AND Y COORDINATES
7165 B1 CE      214           LDA (BASEL),Y   ;ACCORDING TO ORIENTATION
7167 C9 01      215           CMP #$01        ;AND VERTICAL MODE
7169 D0 07      216           BNE DXDYOR
716B A9 08      217           LDA #$08
716D 8D F1 02   218           STA TBUFF+1
7170 D0 0E      219           BNE ADDB
7172 A0 06      220 DXDYOR    LDY #$06        ;ORIENTATION
7174 B1 CE      221           LDA (BASEL),Y
7176 29 01      222           AND #$01
```

```
7178 AA         223           TAX
7179 A9 07      224           LDA #$07        ;OFFSET
717B 9D F0 02   225           STA TBUFF,X
717E B1 CE      226           LDA (BASEL),Y
7180 A0 02      227 ADDB      LDY #$02        ;PREPARE FOR ADD/SUB
7182 A2 00      228           LDX #$00
7184 29 02      229           AND #$02
7186 F0 17      230           BEQ ADD
7188 38         231 SUBXY     SEC
7189 B1 CE      232           LDA (BASEL),Y
718B FD F0 02   233           SBC TBUFF,X
718E 91 CE      234           STA (BASEL),Y
7190 C8         235           INY
7191 B1 CE      236           LDA (BASEL),Y
7193 E9 00      237           SBC #$00
7195 91 CE      238           STA (BASEL),Y
7197 C8         239           INY
7198 E8         240           INX
7199 E0 02      241           CPX #$02
719B D0 EB      242           BNE SUBXY
719D F0 15      243           BEQ FINISHED
719F 18         244 ADD       CLC
71A0 B1 CE      245           LDA (BASEL),Y
71A2 7D F0 02   246           ADC TBUFF,X
71A5 91 CE      247           STA (BASEL),Y
71A7 C8         248           INY
71A8 B1 CE      249           LDA (BASEL),Y
71AA 69 00      250           ADC #$00
71AC 91 CE      251           STA (BASEL),Y
71AE C8         252           INY
71AF E8         253           INX
71B0 E0 02      254           CPX #$02
71B2 D0 EB      255           BNE ADD
71B4            256 ;
71B4            257 ; FINISHED
71B4            258 ;
71B4 AD FF 02   259 FINISHED LDA TBUFF+$F     ;RESTORE SCALE,ROT,HCOLOR
71B7 85 E4      260           STA HCOLORZ
71B9 AD FE 02   261           LDA TBUFF+$E
71BC 85 F9      262           STA ROTZ
71BE AD FD 02   263           LDA TBUFF+$D
71C1 85 E7      264           STA SCALEZ
71C3 68         265           PLA             ;RESTORE E8, E9
71C4 85 E9      266           STA SHAPEPNT+1
71C6 68         267           PLA
71C7 85 E8      268           STA SHAPEPNT
71C9 A5 FF      269           LDA $FF         ;RESTORE REGISTERS
71CB A4 35      270           LDY $35
71CD A6 EB      271           LDX $EB
71CF 60         272           RTS
71D0            273           END

***** END OF ASSEMBLY
```

Listing III: Basic routines

```
 43400  REM
MACHINE CODE HI-RES S
    TRINGS
43410  REM  USES ZS ARRAY AS IN 4
    2800
43420  REM  USES ZB ARRAY INTERNA
    LLY
43430  REM  TWO ENTRY POINTS DEPE
    NDING ON WHETHER
43440  REM  ZS ARRAY VALUES ARE A
    LTERED
43450  IF ZB(0) <  > 0 THEN 43670
    : REM  PLOT STRING
43460  REM  SECOND ENTRY POINT-SE
    TS ZS VALUES
43470  IF ZB(0) <  > 0 THEN 43590
    : REM  COPY ZS VALUES INTO T
    ABLE
43480  REM  INITIALIZATION
43490  FOR ZI = 1 TO ZT(10)
43500  IF ZT$(ZI) = "CHAR TABLE" THEN
    ZB(1) = ZT(ZI)
43510  IF ZT$(ZI) = "TEXT.BIN" THEN
    ZB(0) = ZT(ZI): REM  CODE AD
    DRESS
43520  NEXT : IF ZB(0) = 0 OR ZB(
    1) = 0 THEN  RETURN :
43530  REM  SET SHAPE TABLE ADDRE
    SS IN CODE
43540  POKE ZB(0) + 11, INT (ZB(1
    ) / 256): POKE ZB(0) + 10,ZB
    (1) - ( INT (ZB(1) / 256) *
    256)
43550 ZB(2) =  INT (ZB(0) / 256):
    ZB(1) = ZB(0) - ZB(2) * 256:
     REM  LB,HB OF CODE
43560 ZB(3) = ZB(0) + 10: REM  CO
    NTROL TABLE IN CODE
43570  POKE ZB(0) + 40, INT (ZB(3
    ) / 256): POKE ZB(0) + 36,ZB
    (3) - ( INT (ZB(3) / 256) *
    256)
43580 ZB(4) = ZB(3) + 2:ZB(5) = Z
    B(3) + 4:ZB(6) = ZB(3) + 5:Z
    B(7) = ZB(3) + 19
43590  REM  NOW COPY ZS INFO INTO
    TABLE
43600  FOR ZI = 1 TO 7:ZA = ZS(ZI
    ): IF ZA < 0 THEN ZA = ZA +
    256: REM  FOR -1 VALUE
43610  POKE ZB(6) + ZI,ZA: NEXT
43620  IF ZS(5) = 0 THEN 43670: REM
    FULL SCREEN
43630 ZA = ZM(5) + 3:ZB =  INT (Z
    A / 256): POKE ZB(7),ZA - ZB
    * 256: POKE ZB(7) + 1,ZB: REM
    LOW X
43640  POKE ZB(7) + 2,ZM(8) + 3: REM
    YMIN
43650 ZA = ZM(6) - 3:ZB =  INT (Z
    A / 256): POKE ZB(7) + 3,ZA -
    ZB * 256: POKE ZB(7) + 4,ZB
43660  POKE ZB(7) + 5,ZM(7) - 3
43670 XP =  FN XCN(ZX):YP =  FN Y
    CN(ZY): REM  SCREEN COORDS
43680  IF XP < 0 THEN XP = XP + 6
    5536
43690  IF YP < 0 THEN YP = YP + 6
    5536
43700 ZA =  INT (XP / 256): POKE
    ZB(4),XP - ZA * 256: POKE ZB
    (4) + 1,ZA: REM  X
43710 ZA =  INT (YP / 256): POKE
    ZB(5),YP - ZA * 256: POKE ZB
    (5) + 1,ZA: REM  Y
43720 ZB(9) =  PEEK (54):ZB(10) =
     PEEK (55): REM  SAVE CSWL,C
    SWH
43730  POKE 54,ZB(1): POKE 55,ZB(
    2): REM  SET TO CODE
43740  PRINT ZS$;: REM  PRINT STR
    ING
43750  POKE 54,ZB(9): POKE 55,ZB(
    10): REM  RESTORE CSWL,CSWH
43760  RETURN :
```

No problems with American software

I WONDER if you could supply a list of Apple manuals or books that will help me to understand my Apple IIe.

I have twin disc drives, an 80 column card and a RX80FT Epson printer plus Forum 80.

I can put a program from your magazine onto disc, correct my mistakes and run it, but have no idea about the Apple IIe itself – basics such as that Ctrl S stop and starts listings. I used up a lot of paper on my Epson before I found out that little piece of information.

I would also like to do italic writing and suchlike on my Epson but don't know how or even if it will without added software. If software is needed could you suggest some please.

I'm a captain in the Merchant Navy and have plenty of time to read but not a lot of hands-on time, I'm hoping my wife will lose her fear of the Apple and also try and get to use it.

Thanking you for your very interesting magazine, some of which sometimes leaves me quite baffled but in time I hope to fully understand. – **M.F. Cross, Chislehurst, Kent.**

PS: As I sometimes go to the USA, if I purchase software such as games and the like will they run on my English Apple?

● Most American software will run OK, in fact most of the software we see in Britain is American!

The printer is controlled by Esc sequences and control codes. With a control code a non-printing character is sent to the machine.

An Esc sequence means that the Escape character (CHR$(27)) is sent, followed by a letter or number, which tells the printer what to do.

I don't know Forum 80 and so do not know how to do what you ask from that piece of software.

The RX printer manual has some Basic examples in it which might help.

Books are hard to recommend because they cost a lot and so much depends on what you actually want to do with your Apple. Perhaps some of those advertised regularly in *Apple User* will help.

Max Parrott

Thanks, Max

THANKS for the Forth article by Max Parrott (Apple User, March 1984). I have found his CATALOG quite a useful addition to my dictionary.

I would appreciate more Forth applications (not articles about Forth and RPN – there are enough of them). – **Paul Hartley, Wirral, Merseyside.**

Visicalc solution

NICK Levy posed a question in the last sentence of his article on pages 18 and 19 of the April issue of Apple User. I don't have TK!Solver so I don't know just how you would formulate the equations for it, but I guess they would be very similar to those used in the Visicalc solution below.

Visicalc can be used as a powerful equation solver and has the advantage of showing how a system homes in on, or oscillates about, equilibrium as well as what that equilibrium is.

It is first rate, for example, at solving fluid flow problems. Indeed, it is not evident what equations TK!Solver will be able to solve that Visicalc cannot.

For my money and for maths, finance, engineering and the rest, none of the Visicalc clones comes near the elegance and clarity of the original, which runs a treat on the equally splendid Apple. – **John C. Robertson, Castlecraig.**

I WAS pleased with the presentation of my article and program On your marks! in the June 1984 issue of Apple User.

However, I wish to apologise for the fact that I omitted to replace the sign £ appearing in lines 1040 and 1240. This is the character used by the French keyboard instead of # . I hope your readers won't have any difficulty in finding the bug.

I was disappointed that Miss Jeanette Cau, a professional translater who kindly agreed to adapt herself to computer jargon, was not given due credit. – **Gilbert Dispaux, Varèse.**

Weekday program

AS printed, the Weekday program quoted by Max Parrott in Apple User, February, 1984, Page 47, does not yield an integer for the day of the week, but a decimal remainder. Presumably a final line is missing.

Unfortunately there is still an

Nano's hideaway

I WAS very interested in Max Parrott's article on how to love assembly language via Nano 6502 (Apple User, Jan 1984). Where can I get Nano 6502 and how much does it cost?

Also could you indicate how the delete key on the Apple IIe keyboard is used, since reference to the use of this key in the Apple IIe manuals is very vague. – **M.A. El-Kalay, Renfrew, Scotland.**

● The Nano 6502 program, written by Malcolm Whapshott, is available from Honeyfold Software, Standfast House, Bath Place, High Street, Barnet, London EN5 1ED. Tel: 01-441 4130.

The Delete key on the Apple IIe generates an Ascii code (127) which can be used by software like any other character.

Thus many word processors use it to delete the character to the left of the cursor. Likewise a printer will either not print the character to the left or print an ← or do nothing.

Max Parrott

MARY & ANNE'S AGES — SOLUTION IS FOUND WHEN C7=C12

YOUNG ANNE	5.5	2	3	4	5	6	7
YOUNG MARY C3 X 3	16.5	6	9	12	15	18	21
YRS ON 44-C3-C4/2	11	18	16	14	12	10	8
ANNE NOW C3+C5	16.5	20	19	18	17	16	15
MARY NOW C4+C5	27.5	24	25	26	27	28	29
ANNE OLD C4 X 3	49.5	18	27	36	45	54	63
MARY = C9/2	24.75	9	13.5	18	22.5	27	31.5
ANNE = C10+C3-C4	13.75	5	7.5	10	12.5	15	17.5
MARY NOW = C11 X 2	27.5	10	15	20	25	30	35

THUS ANNE IS NOW 16.5 YEARS OLD.
AND MARY IS NOW 27.5 YEARS OLD.

J.C. Robertson's Visicalc solution

error somewhere in the calculations, as three of the last six years come out wrong. Could Max have a look at the original program "in many books" and publish a correction?

The program will not be valid in this country before 1753 for two reasons:

- *Up to, and including, 1752 the year began on March 25, the feast of the Annunciation of the Virgin Mary.*
- *Eleven days were taken out of the year 1752 to correct errors which had accumulated. Wednesday, September 2, was followed by Thursday, September 14.*

No Chancellor of the Exchequer has had the courage to do without the tax payable for those days, which is why our tax year still starts 11 days after Lady Day. – **Mike Bass, South Croydon.**

- Another reader also spotted the error. His letter, together with a reply from Max Parrott, appeared on Page 66 of the May issue of *Apple User.*

Plotting histograms

PETER Gorry's article in the March edition of Apple User is, as he rightly implies, a very flexible means for plotting histograms – either from data entered from the keyboard or read in from files.

As printed however, there is an error in line 40950 which selects the shade option. For the program to work as described this line should read:

40950 ON ZS+1 GOTO 40990,40960,40970,40980

This corresponds to the open, solid, horizontal and vertical shading options listed in line 40840. – **Doug Shaw, Huntingdon, Cambs.**

More adventure

I AM the proud owner of an Apple II computer. I get your magazine monthly and I must say it is very good.

The layout is much improved from Windfall, especially the contents page. Apple User contains many useful tips and ideas for programmers but it doesn't seem to be very adventurous.

Every issue contains the latest Apple success stories, which are very good but a little dull.

I want to do some practical things with my micro like rigging it up to sensors etc, and I am sure many other readers would like to hear about such things.

Please could you contain a section in your magazine explaining how you can do such things and what equipment you need. – **G. Paisley, Aylesbury, Bucks.**

Input dilemma

I REFER to your note to the Jaromir Smeic letter in Apple User, May 1984 page 67 regarding the immediate mode input dilemma.

This difficulty is caused purely by one attempting to use the input buffer twice at once.

When a line is input in immediate mode it is passed by the Basic routine at $D56C and re-written (compressed) back into the input buffer at $200. That's fine until you enter data which is also placed in the input buffer until it eventually overwrites the immediate mode line – hence syntax errors.

Some input will be accepted where the line is redundant.

Input can be maximised by using other statements before the input statement.

Alternatively GET can be used but this makes the line more complicated.

```
VTAB1
POKE118,0:C$="":
    FORM=1T0255:GET
    A$:C$=C$+A$:?A$
    ;:M=(ASC(A$)=13)
    *300:NEXT:?:?
```

This line will allow you to enter a full string length of 255 characters from the keyboard. – **Dave Ward, Uttoxeter, Staffs.**

Gremlins strike

UNFORTUNATELY in publishing my solution to your TK!Solver problem in the July edition of Apple User, the gremlins slipped in two misprints in paragraph two and three which would render the equations insoluble.

The equations should have read:

$$X + Y = 44$$
$$X - Z = 3*(Y-Z)$$
$$X - A = 3/2*(X-Z)$$
$$2*(Y-A) = X$$

I was pleased to see that Mr Bradley came up with the same answer, although Alastair Thompson's solution seems to have hit a gremlin of its own.

By putting A1 = 3, one must come up with M3 = 15 and A3 = 9 which do not satisfy his equation 5. – **James M. Coles, Caspe Research, Bayswater, London.**

Teletext adapter for Apple?

I WOULD be pleased if you could tell me where I could get an adaptor which would allow me to use the keyboard of my Apple II Europlus to receive Teletext (Oracle, Ceefax).

Not too long ago I saw such an adaptor being demonstrated on a lesser computer (named after some obscure radio station – I believe). – **J. Howell, Peckham, London.**

- We haven't come across this one. Can any reader help?

Snags with assemblers

DURING the course of developing my understanding of the Epson MX100 printer I read an article which included an assembly listing for initialising the printer.

One of the instructions was to load CSWL with the lo-byte address of the printer slot + 2 ($C102) which it printed as follows:

```
A9 02 LDA #<PSLOTAD ; set output to printer
85 36 STA CSWL
```

However when I attempted to use the same notation using the DOS Toolkit, the hexadecimal code showed that I had assembled the hi-byte instead ($CI). The use of the greater than sign (>) took the lo-byte in the DOS Toolkit assembler although the article showed the reverse.

Is there a standard on the use of these operators and if so which assembler conforms to it?

I don't know whether to thank or curse Mike Glover for his articles in Apple User – for while he rekindled my interest in learning more, he has left me no time at all to look at anything else!

Incidentally, does anyone know how to print data from DBMaster onto an Epson MX100 using the 8132 Epson interface card?– **Harold Binley, Bristol.**

- Unfortunately there are no real standards where assemblers are concerned, and the user has to be very careful when typing in other people's listings.

Always check the actual printed code from the listing where doubt exists. Many serious Apple programmers now use the Merlin assembler and we would recommend this product.

With regard to your printer problem, the Mike Glover/Chris Roper articles on the Epson printer (Windfall May and June 1983) show how to alter CP/M in order to make it compatible with the 8132 Epson interface card. This change can be made permanent by altering CP/M on disc.

Psion pack for Mac

AN enhanced version of software originally developed by Psion for the Sinclair QL will become available for the Macintosh by late autumn.

The company says the Xchange package will effectively end American dominance of the micro software market.

The four programs involved – which can be bought separately – are Quill (word processing), Archive (database management), Abacus (financial planning and Easel (business graphics).

As multi-task retrieval is built into the software, this allows up to eight tasks to be resident in the machine at the same time. The user can switch from one to the other by pressing two keys.

Xchange, according to Psion, also incorporates "a radical and innovative virtual memory system which gives very large effective memory to the users' applications".

Dr David Potter, managing director of Psion, said: "Xchange is configured for the machines of the next few years – the big desk top micros characterised by 16 bit processors, large disc capacities and high-processing power.

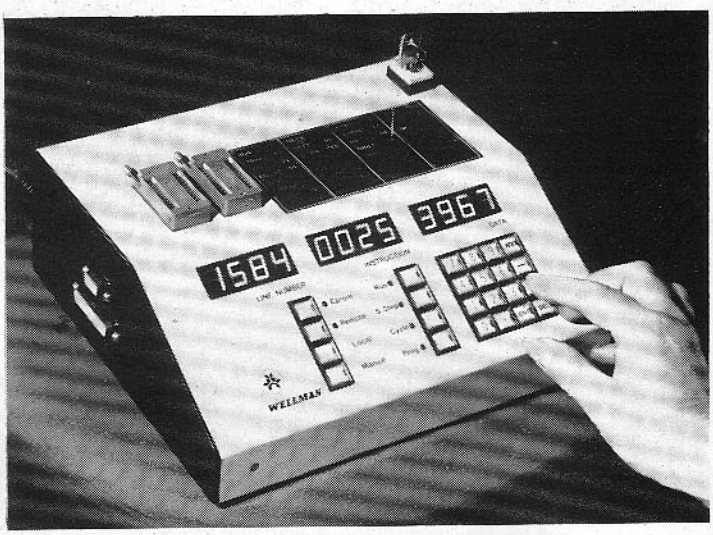
"The minimum requirements are 256k RAM and 320k disc capacity".

Xchange is to cost in the region of £500.

● *Psion Systems Ltd, 22 Dorset Square, London NW1 6QG. Tel: 01-723 9408.*

Link with industry

WELLMAN Microtechnology has developed an interface that enables an Apple II to be used to program industrial programmable logic controllers, so reducing costs for the user and simplifying programming procedures.



Wellman's interface module for use with the Apple II

Initially, the interface is being marketed with Mullard's PC20 and MC20 controllers.

A subsidiary of Wellman plc, Wellman Microtechnology estimates that more than 90,000 Apples are in use in British companies, making a huge potential market for the product.

General manager Paul Taylor believes that computers like the Apple can provide better programming facilities than dedicated units sometimes two or three times the price.

● *Wellman Microtechnology Ltd, Roderts House, Cornwall Road, Smethwick, Warley, West Midlands B66 2JU. Tel: 021-565 2766.*

Enter Macforth

APPLE dealer P & P Micro Distributors is offering Macforth, an interactive programming language for the Macintosh.

Forth has been well established since 1979, and Pete and Pam brought this Macintosh version back from Comdex in Atlanta.

It features 32 bit stacks and default data structures, separated vocabulary heads and direct/token debug compiler.

There are extensive trace and debug features. Included in the £119 price is a computer aided instruction course.

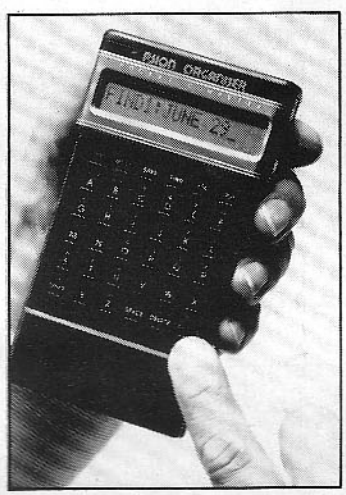
● *P & P Micro Distributors, New Hall Hey Road, Rossendale, Lancs. Tel: 0706 217744.*

Pocket library

PSION has launched what it claims to be the world's first practical pocket computer, the Organiser – and it can be linked to an Apple.

Housed in a protective sliding case, it measures 142 x 77 x 30mm, weighs just 225 grams and costs £99.95.

The machine features an auto switch off and low power consumption components which allows it to run for six months on a standard PP3 battery.



Psion's Organiser

The Organiser has a built in database facility in the operating system which enables the user to create a large personal and permanent information base on 8k and 16k datapaks – which is Psion's name for eproms.

Each machine comes with an 8k datapak, but it can access up to 44,000 characters of information when using two 16k datapaks simultaneously. This is equivalent to 1,000 names, addresses and telephone numbers.

Its plug-in data and program packs play the same role as discs in desk top micros providing open-ended, fail safe data storage and ultra fast retrieval.

A 200 character record can be scrolled through on the display, which has adjustable contrast. All data is stored permanently in the datapak and there is no danger of it being lost, even if the battery is disconnected.

Data can be fed directly to an Apple via an RS232 interface.

The Organiser has its own language – POPL. Built around a set of four commands, it includes a range of mathematical, scientific and generalised functions which can be used within programs and are automatically added to the Organiser's editable calculator functions.

A time and date clock are also available at the touch of a button.

Enhanced light pen

THE Gibson Light Pen, now made under licence by Koala Technologies, is now available with additional software.

The system includes an advanced light pen and five professional software systems. It turns the Apple into an imaginary sheet of paper with the light pen used like a fountain pen to draw directly on the screen.

● *P & P Micro Distributors, New Hall Hey Road, Rossendale, Lancs. BB4 6JG. Tel: 0706 217744.*