An EMAP Publication

SINCLAIR'S SUPERMICRO

Every Month 95p JULY 1985
Germany DM5.50

QL USER

BROTHER TC600 COMMUNICATIONS

THE EDUCATED QL

FAMILY TREES ON ARCHIVE

READERSHIP SURVEY
+PRIZE DRAW

EXCLUSIVE
SOFTWARE-ON-MICRODRIVE
SERVICE

BREAKING THE SURFACE
SAGESOFT ACCOUNTS AND SINCLAIR CAVERNS

THE 1ST. QUALITY ARCADE GAME FOR THE SINCLAIR QL

Hopper

£14·95 POST & PACKING 75p

ON MICRODRIVE

Microdeal have taken the QL and produced an arcade game worthy of the high quality graphics the machine is capable of. Can you guide Hoppy safely home from the spawning grounds across the busy 6 lane motorway and the maelstrom of the cascading river to his home safe on the riverbank? As your skill increases so does the volume of traffic and the snake and crocodile in the river get angrier and angrier whilst the logs get less and less. Written in 100% machine code with full colour graphics and sound. Joystick or keyboard compatibility this is a must for your software collection.

Credit Card Orders
0726 73456

Mail Order
To Micropost 41 Truro Road
St. Austell Cornwall PL25 5JE

MICRODEAL

QL USER

July 1985

Editor	Paul Coster BSc
Assistant Editor	Paolo Baccanello
Editorial Assistant	Shirley Eborn
Art Editor	Mike Spiller
Technical Consultant	Adam Denning
Associate Editor	Peter Rodwell
Advertising Manager	Phil Baker
Advert Production	Yvonne Moyser
Publisher	Terry Pratt

Contributors

Sid Smith, Andy Carmichael, Nicky Trevett, Adam Denning, Rob Miles, Roger Vernon, P J Derlien, Ian Williams, Jane Robins, Peter Williams, Mary Sargent, Mike James, Ian Stewart, J S Ellis, and Stephen Hollywood

QL User, Priory Court,
30-32 Farringdon Lane,
London EC1R 3AU.

Telephone 01-251 6222

Cover photography Terry Beddis

Competition

There were four outright winners to our Categorical Computing competition in the May edition – M Gottlieb (Edgware, Middx), Ralf Biedermann (Langwedel, W Germany), David Carlin (Bathgate, W Lothian) and Neil Gordon (Hull). They all guessed the correct order to be B, C, A, F.

Nine other entrants had the correct attributes but in the wrong order. Of these, the first six out of the hat were: Jane Litman (Wembley, Middx), N Alexander (Margate), V H Hashmi (London E17), A M Rees (Uckfield, E Sussex), J R D Lehane (Aberystwyth, Wales) and M H Henderson (Biggin Hill, Kent).

All ten winners will be sent a Computer One language or utility of their choice once they have written to us stating which package they want.

Medicaments!

Frustration reigns supreme this month as despite being visited by 'Mr Medic' *and* promised a disk system for review, nothing arrived and we must again report that the Medic system remains somewhat of an enigma at present. Hopefully something will turn up next month.

CONTENTS

News 5
QLN
News and new products recently introduced onto the QL scene.

Feature 9
Satellite Stations II
The follow-up to our preview of QL communications, plus connection details for the Brother TC600.

Letters 12
Open Channel
A slight diversion this month with *your* answers to some past questions.

Preview 13
Next Month
A look into the future with the contents of our exciting August edition.

Series 14
Graphics On Display
Recursive curves and other wondrous creations are the subject for part two.

Series 18
Machine Code Tutorial
A brief overview of what's been covered so far.

Books 21
Bookmarks
Another collection of literary gems and other works of excellence.

Series 23
Games Programming
The final part of our intro to low level coding brings life to Paladin.

Feature 26
Sound And Vision
A double feature investigating the QL's sound and mapping potential.

Questionaire 29
Survey & Prize Draw
Our chance to discover the kind of people reading *QL User* – your chance to win one of ten prizes.

Review 34
Integrated Accounts
Sagesoft's latest accounts package bared for scrutiny.

Series 36
Articulating With Archive
The first in this new series of specialist Archive applications deals with family trees and other so-called relational databases.

Series 40
Terminal Emulator
Could it be the last part of our communications terminal? Mayhap!

Series 45
AI on the QL
Making decisions is a pretty subjective business at the best of times, but try explaining that to your micro.

Feature 48
QL At School
Believe it or not, the QL is second only to the BBC micro as an educational computer.

Programs 51, 54
'The Progs'
A tear-out-and-insert index for your QL manual, plus the second set of competition results.

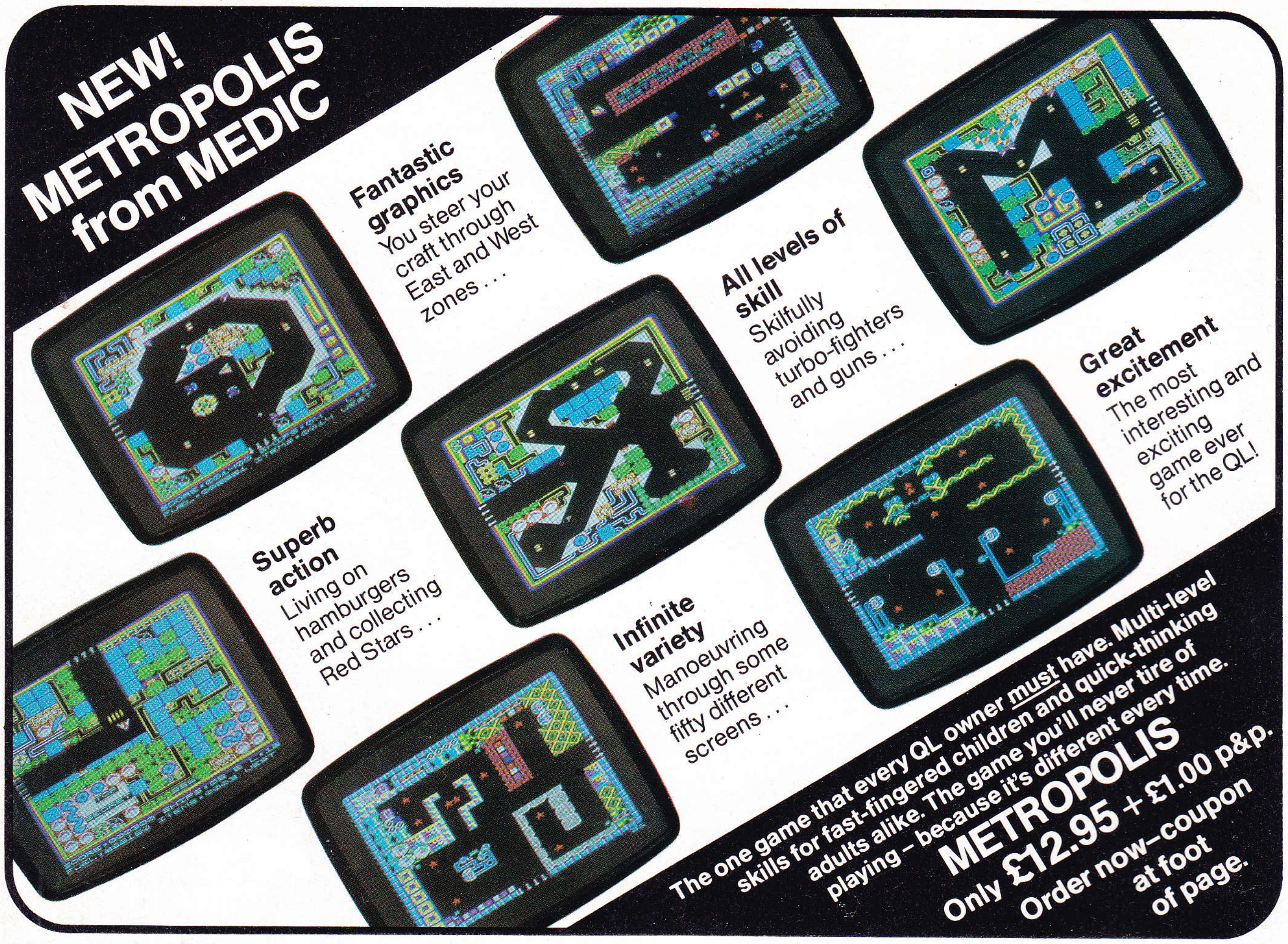
Review 57
QL Caverns
At long last a *real* game for the QL – was it worth the wait?

Readers Service 60
Microdrive Exchange
Send us your drives and we'll fill 'em with bootie!

Information 60
Instant Access
Completely updated and revised from the last time it was completely updated and revised.

Published and Distributed by EMAP
U.S. Sales Agent: 'Motorsport', RR1, Box 200D, Jonesburg, MO, 63351.
Typesetting by Contemporary Graphics Ltd., 20 Bowling Green Lane, London EC1.
Printing Riverside Press.
© Copyright QL USER, 1985

NEW! METROPOLIS from MEDIC



Fantastic graphics You steer your craft through East and West zones . . .

All levels of skill Skilfully avoiding turbo-fighters and guns . . .

Great excitement The most interesting and exciting game ever for the QL!

Superb action Living on hamburgers and collecting Red Stars . . .

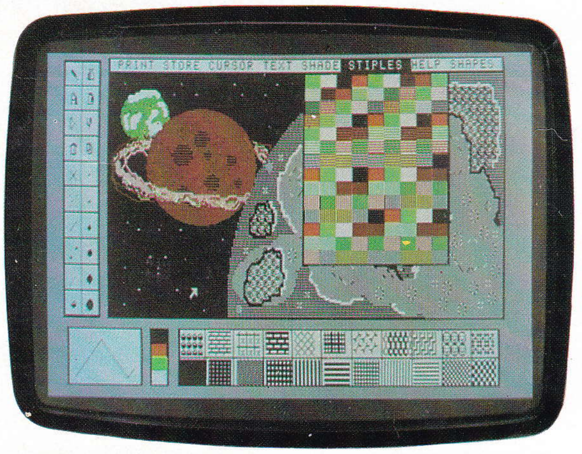
Infinite variety Manoeuvring through some fifty different screens . . .

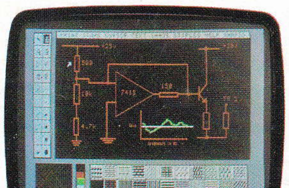
The one game that every QL owner <u>must</u> have. Multi-level skills for fast-fingered children and quick-thinking adults alike. The game you'll never tire of playing – because it's different every time.

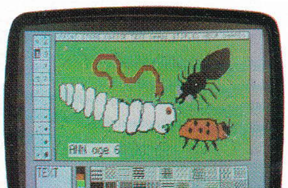
METROPOLIS

Only £12.95 + £1.00 p&p.

Order now–coupon at foot of page.







NOW–what the Macintosh does in monochrome – the QL does in colour!

M-PAINT — an amazing new program from MEDIC gives you total freedom to draw, design, sketch or paint. You can create any image, from a freehand scribble to flowcharts and technical drawings. Thousands of different effects, stipples, shadings, using a full palette of options, cursor-key or mouse-driven. Runs on the standard (unexpanded) Sinclair QL. Easy permanent print-outs. M-PAINT brings you complete design freedom – in brilliant colour. Order your copy now for immediate delivery.

New M-Paint from Medic Datasystems – £30.00 inc. VAT (Plus £1 p&p)

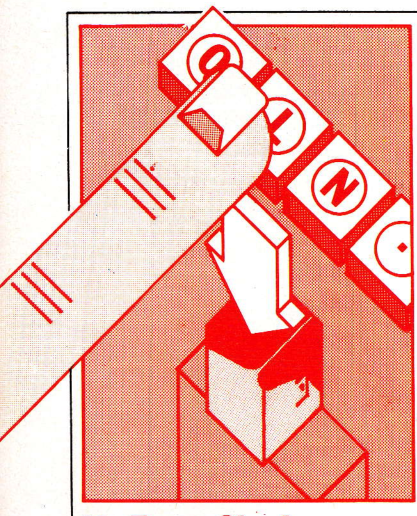
Please send me ☐ copies of MEDIC METROPOLIS on microdrive at £12.95 ea. inc. VAT, plus £1.00 p&p.
Please send me ☐ copies M-PAINT colour graphics program on microdrive at £30.00 ea. inc. VAT, plus £1.00 p&p.
I enclose cheque/P.O. value £ . . . . (add £1.00 for p&p, irrespective of number of programs ordered).

To: Medic Datasystems Limited, 76 Grainger Close, Basingstoke, Hants RG22 4EA
NAME . . . .
ADDRESS . . . .
. . . .
(Please write clearly)

NEWS

The latest information on the QL and associated products.

Compiled by Sid Smith.



No Fears Shed

"I'm afraid that most of the people who ordered and paid for QCOM from OE Ltd will have lost all their money," says Roy Pendleton, whose company — Tandata — has acquired rights to, and stocks of, the QL comms package.

And the mood of these OE debtors will not be improved by news that an ex-director of the company is convinced that it need not have collapsed, blaming its demise on the premeditated actions of two fellow directors.

Indeed, the only bright side of the affair is that Tandata is acquiring a list of QCOM orderers from OE's receiver and will be offering them a 20% discount on fresh orders.

Says Roy Pendleton, "The receiver is responsible for OE's debt, and if he's able to realise enough money then everybody will eventually get their money back. I would very much doubt if this will happen, because there are always preferred creditors such as the debenture holders Warburtons."

Only people who bought by credit card can hope for a refund — by claiming against the card issuers.

Pentleton explained his purchase of QCOM on the grounds that, "the QL is a very major micro, and likely to have significant sales in 1985/6. We paid a lump sum for QCOM, not a royalty arrangement — which shows our great faith in the QL."

Tandata doesn't make any of its products, and OE's receiver will be manufacturing a few units to tide things over for the first few weeks. Production will then be moved to one of Tandata's regular subcontractors in South Wales.

The only obvious uncertainty over the future of QCOM is that of BABT approval — essential for any device which is to be connected to the phone system. Approval was about to be issued when the receivers went in, and Pendleton is confident it can soon be resurrected.

Although shy about revealing the exact price paid for QCOM Pendleton confirms that the sum was well over £100,000.

"QCOM is crucial for correctly placing the QL in the business market — and Sinclair know that. They haven't underwritten the purchase price and they aren't subsidising our advertising, but they have given us some very powerful reassurances about market support."

According to Pendleton, "The original shareholders have no role in current developments at OE, these being in the hands of the receiver. However, Warburtons have a dual role at the moment; they're OE shareholders, but also debenture holders, because they paid off OE's bank overdraught and therefore have a preferential claim on any assets of OE that the receiver may realise."

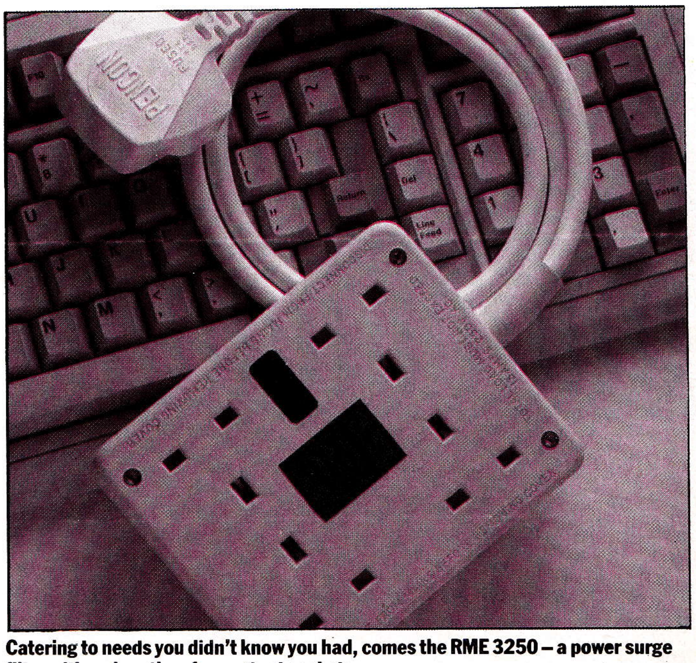
But it's precisely the role of Warburtons in the OE affair — and that of fellow shareholders Pulkinghorne Industries — about which OE ex-director Fred Ansell is perturbed.

Mr Ansell has said that the receiver was sent into OE by Warburtons and Pulkinghorne in the face of repeated assurances by him and his son, OE's ex-Managing Director Martin Ansell, that the company could trade itself out of its debts — which he puts at around £276,000.

Ansell believes that events subsequent to the receivership have proved his point. The QCOM purchase alone, he affirms, establishes that OE had products sufficiently profitable to assure its future.

Describing the actions of Warburtons as "scandalous" and "totally and absolutely wrong", a furious Mr Ansell claimed, "The only losers in this affair are the £276,000 worth of creditors — who have been utterly ignored."

Meanwhile, Commpak Data's modem is looking more and more like the front-runner — we'll be reviewing it next month so long as the News Editor will part with it.



Catering to needs you didn't know you had, comes the RME 3250 – a power surge filter with no less than four output sockets.

Now you can protect not just your microdrives from premature formatting, but the disk drive from wiped files, the TV from revolving rasters and still leave a socket to prevent a hiccupping printer.

The filter is available from Strong Computer Systems for £36 all inclusive.

Proteus Prospects

It's possible that the Sinclair portable, due for launch next year, will be based on QL technology.

According to Sinclair MD Nigel Searle, "The portable could have a high degree of compatibility with either Spectrum or QL software. It would obviously be crazy to set off on a third course."

Sir Clive has always said that the device will be Z80-based and Spectrum-compatible, even though this undercuts his simultaneous assertion that it will also be a "no compromise machine". In the background, meanwhile, a low chorus of Sinclair spokespeople have compared the Spectrum to the perennial Apple II, implying that the micro will likewise preserve its identity through many metamorphoses – a concept which squares very nicely with the Sinclair portable's code-name of Proteus.

However, Searle told your reporter that the Z80 option was still 'iffy', and implied that a final decision will depend on the relative strengths of the Spectrum and QL software bases – though Sinclair will doubtless bear in mind the smooth up-grade path offered to a QL-derived Proteus through other members of the 68000 family.

And Searle was adamant that the portable would use the Sinclair flat TV tube: "That display is our key advantage".

Olé QL

Sinclair remains bullish about the possibility of large QL sales abroad, and has begun its assault on non-English speaking countries with a Spanish version of the machine.

Apart from demonstrating the purpose of those funny letters accessible via the CTRL key, the twelve forthcoming local language versions of the QL (according to Charles Cotton, Sinclair's overseas Business Manager), form "the cornerstone of Sinclair's programme to create a strong presence in professional and business computing markets throughout the world."

Sinclair claims 75% of the Spanish market, selling the Spectrum, Spectrum+ and (even) ZX81 through some 800 outlets.

Due to shortage of space, User Group News has been omitted. It will return as usual next month.

The MEDIC QL E

Two well-known, independent computer journalists compare the MEDIC system with the other systems on the market.

“Disk drives are now becoming a standard add-on for the QL, and to supply the growing demand a number of companies have produced equipment. One of these, Medic Datasystems, asked us to assess its product against all the systems currently available. To do this we looked at three different areas; price, performance and features.

The price of the Medic interface on its own is £129.95, which puts it in the middle of the range. Adding a single drive brings the price to £249.95, which seems to be about the average, except that Medic supply a 720k (formatted) 3½-inch drive as opposed to the more usual 200k ones.

The table shows that the Medic system with extra memory added offers superior performance, especially if the RAM-disk facility (included in the Medic free software) is utilised. CST and Silicon Express interfaces offer similar timings, while those from Quest and MicroPeripherals are rather slow. The Quest system also requires the disk operating system to be loaded from microdrive, which adds another 10 seconds.

The software features of each system differ greatly. Silicon Express has just one extra command, from the Tony Tebby Toolkit, while Medic has most of the Toolkit commands and a large range of free software. Hardware features also vary considerably, Medic being the only system that does not require a further expansion board to add extra memory or I/O ports.

The choice seems to be between Silicon Express or CST, and Medic. The former are physically smaller than the Medic, but Medic has many extras such as built-in RAM disk software and an interface for Medic's forthcoming modem.”

Roger Thomas **John Lambert**

TEST RESULTS	1	2	3	4
Microdrive	2.5	6.5	13	167
Disk systems:				
CST	6	2	7	41
MEDIC	6	2	5	40
MEDIC + extra memory	6	2	5	25
MEDIC without extra memory, using free RAM-disk software	0.5	1	*	*
MEDIC + extra memory, using free RAM-disk software	0.5	1	2	6
MicroPeripherals	18	3	12	69
Silicon Express	7	2	8	36
Quest	19	35	39	**

Notes: 1 – save 32k code. 2 – load 32k code. 3 – load Quill program file. 4 – save Quill file, approx 10½ pages (3288 words, unpatched). For 1 and 2, directory booted into memory before timings taken.
*Timings for all Psion programs except Quill are similar to those shown below. Quill requires extra memory.
**Quill does not run on Quest without extra RAM.

MEDIC offer you immediate delivery of a complete expansion system for your QL. Choose the amount of extra memory you need, plus options of disk interface with parallel port and printer spooler, and modem – all contained simultaneously inside the MEDIC single-unit systemcartridge.

MEDIC unleashes the full power and performance of your Sinclair QL for business.

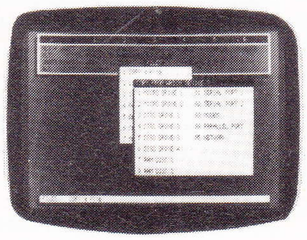
Extra memory, for speed and easier program and file handling, starts at £99.95. Complete systems, consisting of 1Mb disk drive, disk interface, parallel port, integral power supply and all cables, from £249.95. Even higher performance can be achieved by adding extra MEDIC memory to your disk system, utilising RAM-disk procedures available with MEDIC free software (see table for speeds).

You can upgrade your systemcartridge at any time, adding up to 512k extra memory, modem, and up to 4 disk drives.

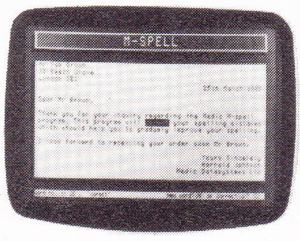
MEDIC – the reliable QL expansion system that *guarantees* you more than any other system.

FREE SOFTWARE

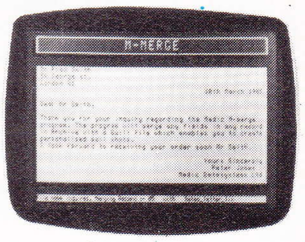
Nine performance-raising programs, free with every MEDIC disk drive system.
(Extra memory recommended.)


M-DESK

M-TRANSFER



M-SPELL



M-MERGE

M-DESK Macintosh-type single-key depression user interface, for instant program switching and utilities selection **M-BASE** turns Archive into a menu-driven database with single-key commands **M-ACCOUNTS** fully integrated sales, purchase, nominal ledgers, and stock control **M-KEY** single-key entry of user-defined text in any program **M-SPELL** spelling checker **M-MERGE** personalised mailshots **M-SQUEEZE** file compression **M-BOOT** sets up RAM disks in memory then automatically loads pre-defined files and programs **M-TRANSFER** microdrive – disk routine.

xpansion System.

...beats all other systems on speed, on disk capacity, on versatility, on extras... and on sheer value!

MEDIC

Medic Datasystems Limited
76 Grainger Close, Basingstoke,
Hants RG22 4EA
Telephone 0256 460748

16th ZX MICROFAIR
22nd June 1985
Royal Horticultural Hall
STAND Nos.
83,84,85 & 90

Order form

☐ Please send me the MEDIC brochure and copies of press reviews, free.

To order, please tick the items you require.
All prices include VAT. Add £10 p&p.
All orders despatched in strict rotation, delivery guaranteed within 28 days.

Extra memory
MEDIC systemcartridge with extra memory

64K	£99.95 ☐
128K	£129.95 ☐
256K	£169.95 ☐

Disk interface
(if you already have disk drives) £129.95 ☐

Dust cover
+ 3 adjustable feet £14.95 ☐

Disk pack
Ten 3½in disks in rigid plastic storage box £60.00 ☐

Modem*
Add to disk interface or complete system price (not sold separately, since connects through systemcartridge) £120.00 ☐
*not yet BT approved

Complete systems
1Mb disk drive, MEDIC systemcartridge inc. disk interface, parallel port, integral power supply, all cables
+ FREE SOFTWARE £249.95 ☐

2Mb double disk drive, MEDIC systemcartridge inc. disk interface, parallel port, integral power supply, all cables
+ FREE SOFTWARE £399.95 ☐

Complete systems + extra memory
Add to complete system price

64K	£50.00 ☐
128K	£80.00 ☐
256K	£110.00 ☐
512K	£200.00 ☐

I enclose cheque/P.O. payable to Medic Datasystems Ltd. in payment for the items ticked above, plus £10 p&p, total £

Signed ____________________

Name ____________________

Address ____________________

____________________ Telephone

Please write clearly

The Sinclair Vision QL. Designed to perform.

For sheer good looks, superb design and high performance there is nothing to touch the Sinclair Vision QL colour monitor.

Developed in conjunction with Sinclair Research, this beautifully compact unit with its 12″ non-glare tube, 85 column text display and high resolution colour graphics is the perfect partner to the Sinclair QL computer. Comes complete with lead and connector for instant use with the QL.

Take a closer look and you'll see that it also represents amazing value for money.

You'll find the brilliant Sinclair Vision QL at your local dealer, supplied exclusively by DDL. The data distribution company with a product range that is second to none.

Ring us on Ascot (0990) 28921 for the name of your nearest dealer.

Then you can call in and see for yourself what makes the Sinclair Vision QL colour monitor such a well-designed little performer.

DDL

5 King's Ride Park,
Ascot, Berks. SL5 8BP
Tel: 0990 28921
Telex: 846303 DD LTD G.

THE NEW FORCE IN DISTRIBUTION

SATELLITE STATIONS 2

Back in February we previewed the QL's potential as a communicator. With the arrival of Brother's TC600 it seemed appropriate to turn theory into practise – Adam Denning and David Green take up the story . . .

The QL is blessed with two RS232 ports, which should mean it connects to almost anything. Reality is a different matter, because so many manufacturers interpret the 'RS232 standard' in different ways.

Some of the more popular machines people try to connect to the QL are made by Brother, such as the EP44 and the TC-600. These can be used as printers, or in the case of the TC-600, as a portable typewriter whose contents can later be dumped to the QL and re-edited.

The model 600 (as the instruction manual refers to it) is a rather mixed bag of potentially useful facilities and functions. The latest in a line which spans back to the humble, yet crude, EP22 electronic typewriter, the TC600 can boast a proper typewriter keyboard (none of those annoying plastic 'buttons' for keys) and an expanded LCD display – single line only, however.

As a machine which you can switch ▶

on and use straight away, everything is fine so long as you've selected the correct mode (normal), which gives standard typewriter operation. There are, however, several refinements (more of these later) making use of the display window and permitting corrections before print-out. The big difference here is that no information is stored and once a line has been printed it cannot be altered later.

Most of the key functions in normal mode are self explanatory. There are the usual tab set and release, centring and flush left and right as well as justified printing. Added to this are some nice touches such as super and sub-script and auto-underlining. However, all these features are better used within WP mode.

The TC600 as a word processor is a good example of a machine floored by its weakest component – the LCD display. Considering the high print quality (albeit at a slow 16 cps), the wide range of assorted WP enhancements and the extended storage capacity when using the optional disk drive, it's such a pity that stored text can only be edited one line at a time. Other standard WP operations also suffer as you need to work out a specific method for achieving what would normally be a fairly simple procedure (moving blocks of text, adding/deleting para indents or running text back to the previous line). Despite being considerably improved from the EP22 and EP44, further development in these areas would certainly reap rewards.

As mentioned earlier, and partly making up for the above, the TC600 does have a wide range of set-up facilities enabling text to be printed out virtually anywhere on the 80-column line. Utilising the margin set and tab keys it's possible to whisk through a host of different formats in any of the justification modes and produce a high quality document printed in a superb typeface – unfortunately this is only at its best when performed using the thermal paper supplied.

The third TC600 mode (terminal) provides the most exciting aspect of the machine for QL owners – as an RS232 terminal and serial printer. Here Brother are to be congratulated on producing a printer/terminal that communicates freely with the QL without requiring additional interfacing hardware or two years experience as a Telecom cable fitter. All the settings for standard serial communication are controlled using keyboard commands (no fiddly DIP switches here) and within a few seconds, QL and Brother files become interchangeable.

To get the two talking to each other, we have to alter the connections on the standard Sinclair Research RS232 lead. This has a 6-pin BT plug at one end and a standard 25-pin D male connector at the other. The Brother also has a 25-pin D connector, but it's female. This means they'll plug into each other quite happily (though no communication will take place if the standard lead is used unchanged).

To get things going it's a good idea to buy two RS-232 leads from Sinclair, as the standard one works with various other bits of peripheral equipment, while the altered one won't.

Three of the wires can remain exactly where they are, as these are standard on both the QL and the Brother. These connections are the RxData, TxData and Signal Ground lines, on pins 2, 3 and 7. The RxData and TxData lines are used for sending and receiving information.

Next we need to know whether the peripheral is 'Data Terminal Equipment' or 'Data Communication Equipment'. If it's DTE, then it sends data out along pin 2, TxData, and receives it on pin 3, which is RxData. If the peripheral is a DCE, however, it sends data out from pin 3, which is still called RxData, and receives it on pin 2, which remains TxData. In the trade, we call this 'thoroughly confusing' *(Funny . . . I thought we called it a cockup – Ed!)*.

The RS232 ports on the TC-600 and EP44 are both configured as DTEs, which means that we can overcome most of the wiring difficulties if we use the socket on the QL which is configured as a DCE. This is SER1. Doing this makes pins 2, 3 and 7 correct at both ends.

Avoiding the XON/XOFF handshaking protocols (please!), the next best method uses 'device drivers'. This is done by sending voltages along other wires connected to more pins on the RS232 connector. It sounds fairly trivial, if it wasn't for the fact that no-one seems to be really sure which selection of handshake lines to provide. Sinclair provides two on its QL serial ports, known as 'Data Terminal Ready' (DTR) and 'Clear To Send' (CTS).

The DTR line is used by the DTE (who said RS232 terminology was confusing?!) to tell the DCE when it is ready. If this line is not in a certain state (either ON or OFF, depending on whether the equipment uses negative or positive logic), the DCE will not send anything to the DTE.

The CTS line is used by the DCE to tell the DTE when it is ready to receive information – in other words, when the line is clear to send. It almost makes sense, doesn't it?

Pin 4 on the Brother machine is 'RS', which is Japan's version of everyone else's 'RTS' – standing for 'Request to send' – and is an output from the Brother, and at logic level 1 (ie, ON) when the machine is in Terminal mode. This isn't the standard RTS action, though, as it is supposed to be ON only when the sender wants to send something. As the Brother's RTS is permanently ON, and as the QL doesn't use it, this is one pin which we can ignore.

The next pin worth considering is pin 8 on the Brother, which is the 'Carrier Detect' pin. This is related to a modem function and sets the line to logical 1 when the telephone line connected to the modem is holding the carrier signal. Unfortunately, although the QL is not a modem, the Brother expects this line to be ON before it will receive anything.

Now, as we've seen that the Brother's pin 4 is always ON, and this pin 8 always needs to be ON, we could solve our problem by directly con-

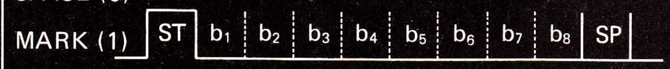
Transmit speed	75, 110, 300, 600, 1200 (Baud) full duplex
Data length	7-bit +parity, or 8-bit (no parity)
Parity	N.....NONE parity (parity bit is always 1) O.....ODD parity E.....EVEN parity Z.....ZERO parity (parity bit is always 0)
Data style	10 bits/character SPACE (0) MARK (1)  Start of character is the first shift (start bit) from MARK to SPACE. If without Line data, it is MARK mode. ST.....start bit b_1–b_7 .....data bit (LBS is b_1) b_8.....parity bit (MSB in 8-bit data) SP.....stop bit

Table 1. Communication parameters for the Brother TC600.

Pin number	Signal	Code	TC600←→Other	Details
1	Frame Ground	FG	←→	Ground line for protection
2	Send Data	SD	→	Data line sent from
3	Receive Data	RD	←	Data line received from
4	Request to Send	RS	→	ON: carrier output OFF: carrier stop
5	Clear to Send	CS	←	ON: data transmission possible OFF: data transmission not possible
6	Data set Ready	DR	←	ON: transmission/reception possible at connected device OFF: transmission/reception not possible at connected device
7	Signal Ground	SG	←→	Provides basic ground potential
8	Carrier Detect	CD	←	ON: receiving access signal OFF: not receiving access signal
20	External Ready	ER	→	ON: unit preparation completed OFF: unit preparation not completed

Table 2. Important pins on the TC600's RS232 connector.

necting pin 4 to pin 8. So that's what we'll do.

The final pin worth looking at is pin 20 on the Brother, which is normally called 'Data Terminal Ready', but not by Brother. They call it 'External Ready' (that's what is so good about international standards – no-one follows them). This pin is set to logic 1 by the Brother when it is ready.

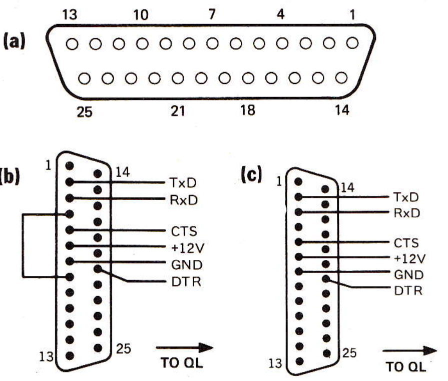


Figure 1. RS232 pins (a) and wiring details – before (b) and after (c).

Rather than go through all the connections again, we'll refer to two diagrams. *Fig 1b* shows the Sinclair Research lead as it is supplied, and *Fig 1c* shows what it should look like when you have finished with it.

Now, set up the Brother by moving the 'mode' switch to 'TERMINAL', and then press the button marked 'MODE'. Up comes a display saying 'SET-UP' followed by a few numbers and letters. Move the cursor until it's under the first number (usually '4'), and press '6'. This sets the baud rate to 1200. Now move the cursor right one position, and press 'D'. This sets the internal code used by the Brother for characters to be just about the same as the QL's. Now we need to alter the next letter, which is the 'parity'. This parity is a sort of double-check, to ensure that data really has been sent properly. However, we don't want any of this, so press 'N'. The next number tells the Brother how many of the pins on its RS232 interface it should take notice of. Press '1' here, as this stands for 'all of them'. Now we can select what sort of handshaking we want by setting the next setup character to 'E', which causes it to use hardware handshaking rather than XON/XOFF.

The final character in the setup determines what happens when the Brother receives a carriage return or a line feed. Set this number to '1', so that the Brother returns its carriage and feeds a line when it receives a carriage return. Now press RETURN. The Brother is ready.

To set up the QL depends on whether you're going to use the Brother as a printer for the Psion programs or as a general purpose printer. In the latter case, you simply type baud 1200 and press ENTER. Whenever you want to print something out, you must send it to ser1c (the 'c' tells the QL to convert all outgoing line feeds to carriage returns and vice versa – that's what you want). For example, to copy a file called 'myfile' on microdrive 1 to the printer, type:

copy_n mdv1_myfile,ser1c

To list a SuperBasic program, you must first open a channel to the printer:

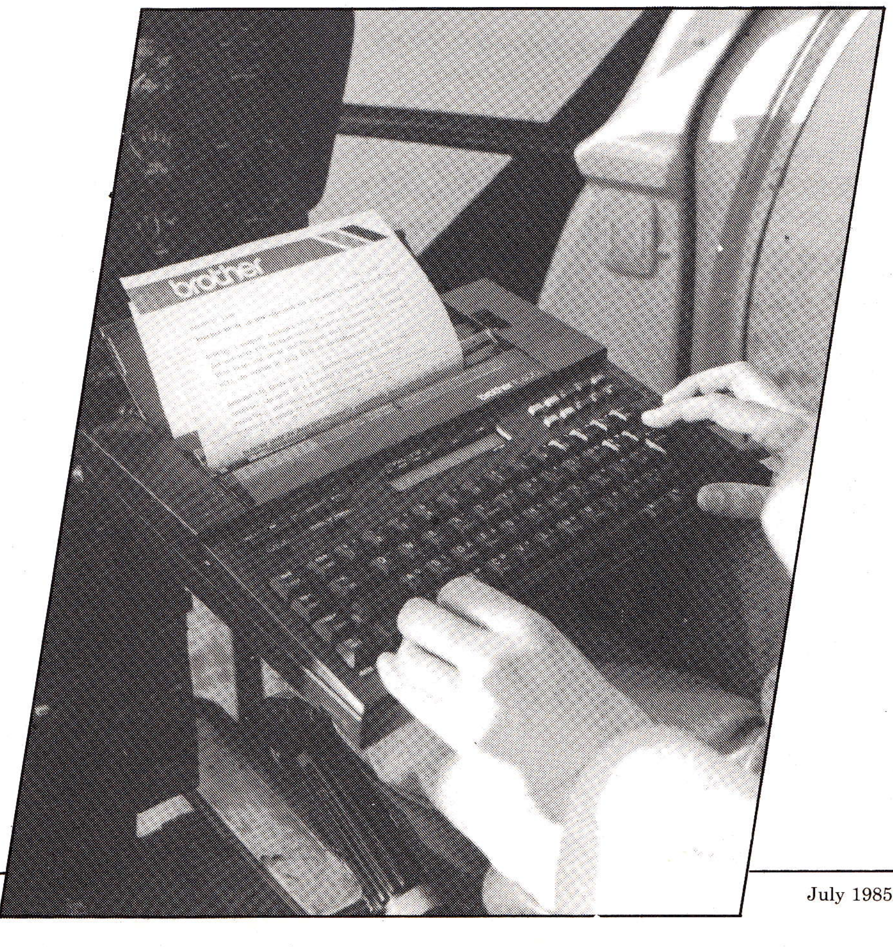
open#3,ser1c

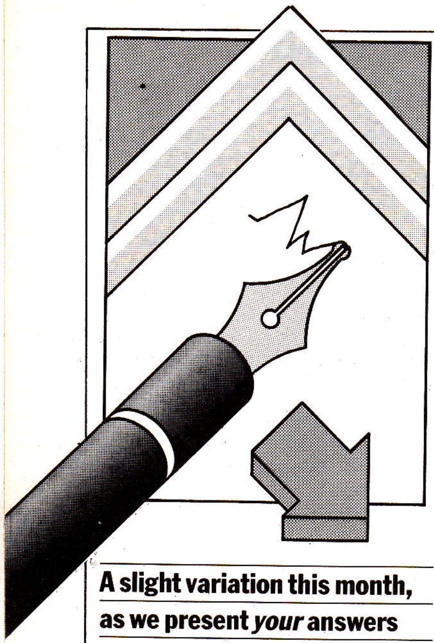
and then list the program to your chosen channel:

list#3

If you want to use the Psion programs with the Brother, you must run the install_bas program to set up the printer drivers in the correct way. The Sinclair manuals explain this reasonably well, but remember that you are using a baud rate of 1200, port SER1, and that the end of line code is CR.

So, what advantages are there for the QL owner having just forked out around £400 (or £575 with portable disk drive) for the TC600. Well, as a typewriter the machine is fine, but expensive, and as a word processor it is light years behind the QL/Psion combination. However, as an RS232 terminal and printer for the QL with portable typewriting and word processing capabilities thrown in, the TC600 represents an excellent proposition for people on the move.





A slight variation this month, as we present *your* answers and comments to past letters and articles. Next month, it's back to normal so send your letters to: Open Channel, QL User, Priory Court, 30-32 Farringdon Lane, EC1R 3AU.

Automatic ASCII

As SuperBasic programs are stored on microdrive as just ASCII files, it is easy to manipulate them. The programs are tokenised after loading, before they can be executed or edited so it is possible, for example to write direct commands to this file and they will be executed when you just Load them rather than Lrun. You can use this method to run a SuperBasic program automatically when loaded. After loading the program if you want to make it autorum type:

```
  OPEN_NEW = 3, MDV1_
file–name
  PRINT # 3, "RUN"
  LIST # 3
  CLOSE # 3
```

This will write the command RUN to the file, then list the program to the file so on loading the program it will RUN automatically. You can write all the commands, functions, etc, of SuperBasic in this way — even whole programs.

Philip Teakle
Bristol

City Success

I've had my QL for nearly a year (JM version) and am now really getting good value out of it. I have concentrated on using the Psion software packages and have had great success on transferring data between Abacus and Easel. I am a management consultant and recently used Easel to make a presentation to the main board of a City financial group. This involved 30 or so graphs, bar and pie charts and text screens, put together to form a 90 minute seminar. Both the content and the presentation method (a 26″ TV) were very well received.

My only problem with the QL — the left hand shift key doesn't work!!

Paul Thomas
Cardiff

Enigmatic Beep

I was interested to see Rob Miles' "Sound Experimentor" program in the May issue. We have been using a similar program to try to produce 'useful sounds. The trouble is that, with 8 arguments, each able to take a wide range of values, there are something in excess of 7E19 combinations possible, which makes a complete investigation rather daunting. Might we suggest that QL User should start a "Beep Corner", to which readers could contribute interesting sounds? We would be particularly glad of a door creaking, and of a good 'laser gun'!

Meanwhile here are some of our own discoveries.

BEEP 0,0,20,0,3 — Like a ship's hooter (?)

BEEP 0,0,30,20,12 — Like a helicopter

BEEP 0,0,20,–5000,–200 — "Scary music" — we use it for the opening of a 'Haunted House' game

BEEP 0,0,20,5000,100 — Rather jolly little tune (?)

BEEP 0,0,20,5000,0 — sonar

The Frayn Family
Manchester

Slicing The Cake

I have found that if an array is sliced a large amount of memory is used. If for example the first program is run, the report 'out of memory' will be given at about the 200th repeat (unexpanded RAM). With this report the array is lost. This can be circumvented by assigning the complete element to a simple variable and then slicing this. Try the second program with run 1000, it will complete normally.

I have a Canon PW-1080A printer. To stop having to change the DIP switch settings I have set it to give an auto line feed. The printer driver program for Easel will then need to be changed to stop graphs being double spaced, by deleting the line feed code at position 418 from $0A to $00 (10 to 0). This can be done by reading the driver into an array, altering the array and then printing to a new file or overwriting the existing driver file.

```
100  REMark FIRST
     PROGRAM
110  DIM A$ (800, 100)
120  FOR X = 1 TO 800: A$
     (X) = FILL$ ('AB', 50)
130  FOR X = 1 TO 800
150  PRINT X!A$ (X,1 TO 10);
     A$ (X,12 TO 20)
160  END FOR X

1000 REMark SECOND
     PROGRAM
1010 DIM A$ (800,100)
1020 FOR X = 1 TO 800: A$
     (X) = FILL$ ('CD', 50)
1030 FOR X = 1 TO 800
1040 A1$ = A$ (X)
1050 PRINT X!A1$ (1 TO 10);
     A1$ (12 TO 20)
1060 END FOR X

100  REMark ALTER EASEL
     PRINTER DRIVER
110  DIM A$ (520)
120  OPEN_NEW#4, MDV1_
     CANON_prt
130  OPEN_IN#5, MDV1_
     gprint_prt
140  FOR X = 1 TO 520: A$
     (X) = INKEY$ (#5)
150  A$ (418) = CHR$ (0)
160  FOR X = 1 TO 520:
     PRINT#4; A$ (X)
170  CLOSE#4: CLOSE#5
```

D J Jenkins
Dursley, Glos.

Sinclair Snub

I bought my QL in October 1984 on the understanding that during my first year of membership of QLUB (at a cost of £35) I would be entitled to receive one free update of each of the four QL programs. Sinclair Research has now gone back on that promise and, in effect, has said that the updates will now cost me £50.

In writing to Sinclair I have received a rebuff, in that they have not dealt with my enquiry but merely sent me a standard stereotyped letter.

I am pursuing the matter by writing direct to Sinclair's MD (Mr Searle) but, in the meantime, I wonder whether any of your readers who purchased a QL before Sinclair's "cut-off" date on the same understanding as I, have received similar treatment.

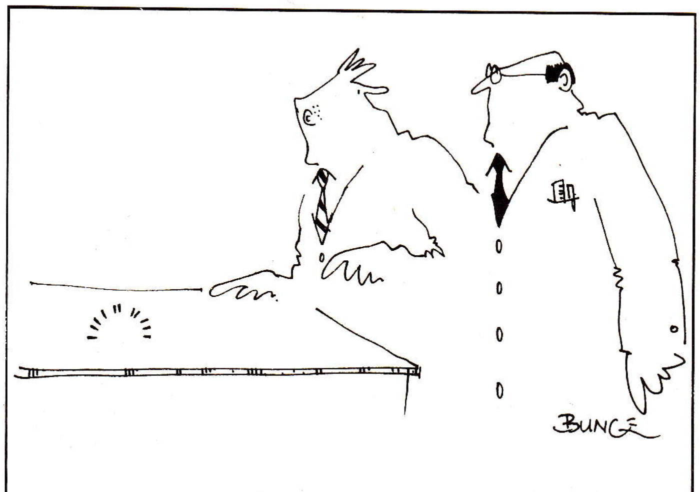
G Boxer
Tintagel, Cornwall

Worthy Note

Having recently received the new version of Quill I re-read your review in the April issue. I agree with the reviewer that the improvements are outstanding. The speed of the commands would make it worthwhile alone, but being able to write quite lengthy documents without mdv2_ whirring away means I no longer have to keep my fingers crossed. I don't think I have had a corrupted letter in the new version.

One thing in the review that doesn't seem to happen with me is when I try to print without the Quill cartridge in place. I do get the error message but it quite happily prints out, obviously without the special codes for underline and so on.

Ian McRobert
Peterborough



"It claims to be a fifth generation micro!"

Fit Of Peek?

With reference to a couple of peek and poke addresses that you have published in the last two months:

Anyone who tried Alan Turnball's poke to turn the caps lock on from within a program and found it did not work should try POKE_W 163976, 256.It must also be pointed out that this disables the caps lock button, which can be brought back into use by POKE_W 163976, 0 and brings the QL back into lower case.

David Nowotnik sent in a tip on how to display memory. If like me you could not get that to work either try PRINT PEEK_L (163860) –PEEK_L (163856) –4096. This will give you the exact amount of memory you have used in your program. Obviously this peek can be included during the running of a program to give a constant readout of memory used, and avoid the problems of running out of memory, which apparently clears out all the variables! Returning you back to square one to start all over again.

Tim Fuller
Southampton,
Hants

Give It A Plug

In Psion Probe on p. 17 of the May issue of *QL User* Angus Ross raised the problem of QL freezes associated with mains spikes. I have experienced the same kind of crash when using my QL. The freezes occurred most days except Saturday and Sunday. The crash times were usually, but not necessarily, between 8 am and 9 am, 11 am and 12 am, 4 pm and 5 pm. There were sometimes periods of several minutes when reset was followed repeatedly by a crash.

Switching tests with equipment in the house failed to identify anything which could be associated with a crash. The house is in a rural area and is on the end of a PME line.

Power International, at the address given on p. 14 of the same issue, were consulted and suggested that mains spikes were a likely cause. I purchased The Plug from P & P Micro Distributors Ltd for a total of £20.41 including packing, postage and VAT. I don't usually pay such a large sum for something which is not a proven necessity, but I could not continue with the system as it was and was prepared to give it a last chance, largely because of the speed and excellence of Psion version 2 software but also because of the power and flexibility of Sinclair Basic.

I have been using The Plug since April 25 and it is now May 13. The QL does not crash! I leave the QL switched on and loaded with an Archive database night and day. It is always ready for use when I return to it. My work, chiefly with Archive and Quill, proceeds without hitch or hindrance. Furthermore, I no longer find small corruptions when the database is printed.

To establish that The Plug is the cause of the dramatic change in performance I should run controlled trials with and without The Plug, to provide data for statistical tests of significance, but I am sure you will not ask me to do that. I am a simple QL user and not part of the QL design team. I have had my share of crashes and cannot afford more time on them. I think however that it is worth recording that since installing The Plug they no longer occur and that since then my QL system has been reliable.

H R B Hack
Hebley-On-Thames

Commpak Comeback

I would like to pick up on a few points in your Modem Moves article (June issue).

The Bright Star will be available from both Commpak Data and Modem House.

The Modem and associated control software was principally designed by David Byrne.

The price of the Bright Star is now £179.95 including software and serial cable, as opposed to £160 previously for the Modem alone. Whilst every attempt has been made to keep the price down, manufacturing costs for Intelligent Modems are high, and the only way to ensure proper distribution and support, is by offering competitive trade discounts.

First deliveries are scheduled for early July, and an Auto-Dial, Auto-Answer upgrade board will be available soon after. Another model incorporting these features as standard will also be available.

Keith Webb
COMMPAK DATA

NEXT MONTH

TELECOMMUNICATIONS

"QL, phone home . . ."

Forget the previews and prevarifications, we've got hold of the two main contenders in the QL modem stakes and will be putting them through their paces.

Readers' Programs Special

Whilst we've been presenting competition results, it seems that you've been beavering away on some really top flight programs. So, we've extended 'The Progs' to cope!

The QL Business Triangle

Three examples of commercial applications software and their development, as expounded by one company involved in the production of specialist QL software.

The Odd Couple

Ever heard of a company called *Hantarex*, or how about *C-Itoh?* Well, we hadn't either, so it was with some surprise that we received a monitor from one and a printer from the other – both QL compatible . . .

PLUS THE RETURN OF:

Psion Probe – your letters answered by Psion's head of Software Support.

Software File – with in depth reviews.

Contents subject to late revision

AUGUST '85 EDITION ON SALE 21st JULY

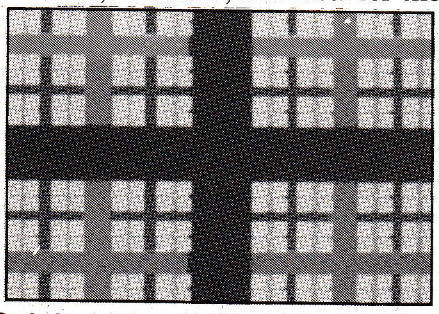
GRAPHICS ON DISPLAY

Turtle Graphics needn't be just child's play. With the aid of SuperBasic, Ian Stewart shows how to achieve simple but stunning effects.

Computer graphics are usually based on some kind of co-ordinate grid, specifying positions by rows and columns. But there is a quite different approach, invented by Seymour Papert as part of the language Logo – Turtle Graphics. In Turtle Graphics you have at your disposal a small number of simple commands with which you direct the motions of a *turtle* The turtle can be anything from a pixel on a television screen to a motorised robot equipped with a felt-tipped pen. The advantage of the former is that programming errors do not lead to unusual designs on the living-room carpet.

The turtle can be told to raise or lower its pen, turn to or through a particular angle, and move a specified distance. You can combine sequences of these moves into named procedures (Square, Spiral, Olympic_emblem, Mona_Lisa), effectively adding new commands to the language. By building up a library of such procedures, you can develop your own powerful graphics language.

QL SuperBasic includes built-in Turtle Graphics commands. I'm going to explain roughly how they work, and then use them to explore one of the current frontiers of scientific research, the curious world of Fractals. Fractals are curves or surfaces that possess detailed structure on a wide range of scales. They are intricate, beautiful, and ideal for the QL's structured SuperBasic.


Producing ever-decreasing blocks is a prime example of recursive programming.

TURNING TURTLE

Suppose the QL is in its standard state, with the screen scale at its default value SCREEN 100,0,0. Let d represent a distance on the screen and a an angle in degrees, measured (as usual) anticlockwise from 3 o'clock. The basic Turtle Graphics commands are: MOVE d (move forward distance d); TURN a (turn through angle a); TURNTO a (turn to angle a); PENDOWN (start leaving a trail); PENUP (stop leaving a trail). So to persuade the turtle to traverse a square, use something along the lines of *listing 1*.

Listing 1

```
100 TURNTO 0
110 PENDOWN
120 MOVE 50:TURN 90:MOVE 50
130 TURN 90: MOVE 50
140 TURN 90: MOVE 50
150 PENUP
160 STOP
```

The turtle starts off at position (0,0), bottom left of screen, unless you start it elsewhere using POINT x,y.

Drawing regular polygons usually means a lot of fiddling with sines and cosines, but our turtle is much more clever. All he does is move a fixed distance, turn through a fixed angle, and repeat! *Listing 2* gives a general procedure for this, and uses it to draw polygons with 2,3, . . . 16 sides.

Listing 2

```
100 PAPER 1:INK 7:CLS
110 SCALE 20,-10,-4
120 POINT 0,0
130 FOR k=2 TO 16
140   polygon k,3
150 END FOR k
500 DEFine PROCedure polygon(n,d)
510   LOCal k
520   PENDOWN
530   FOR k=1 TO n
540     MOVE d
550     TURN 360/n
560   END FOR k
570   PENUP
580 END DEFine polygon
```

The usual curves you encounter in geometry are nice, smooth objects, like a circle. Magnify a circle enough, and it looks flat. In the last ten yers or so, scientists have discovered a quite different type of curve (or surface), called a *Fractal* best described as wiggly. Under magnification the original wiggles grow very large so you can't see them any more, but new, tiny wiggles show up. This goes on for some time, as you keep making the fractal bigger. Fractals are important because many natural phenomena have this kind of structure. The wiggles in a coastline, the bark of a tree, the air-passages in a lung, the slopes of a mountain, the eddies in a turbulent river.

Using Turtle Graphics, procedures, and the QL's ability to handle arbitrarily long strings, we can explore some of the delights of the World of Fractals.

THE SNOWFLAKE

This curve was invented by Helge von Koch to show that an infinitely long curve could enclose a finite area. It starts with a triangle, adds smaller triangular peaks to each side, and then keeps repeating the process. The result is a beautiful symmetrical shape, rather like a snowflake *(Listing 3)*.

A similar program *(Listing 4)* plots a 'snowflake' based on a pentagon rather than an equilateral triangle.

DRAGON CURVES

Dragon curves are built up from a straight line segment by putting a right-angled bend into it, with the

Listing 3

```
100 snow$="111"
110 SCALE 100,0,0
120 MODE 4
130 FOR g=1 TO 5
140   POINT 40,30
150   TURNTO -120
160   string g
170   draw_string g
180 END FOR g
1000 DEFine PROCedure string(g)
1010 LOCal a$,t
1020 a$=""
1030 FOR t=1 TO LEN(snow$)
1040   IF snow$(t)="l" THEN a$=a$&"lrlr"
1050   IF snow$(t)="r" THEN a$=a$&"rrlr"
1060 END FOR t
1070 snow$=a$
1080 END DEFine string
2000 DEFine PROCedure draw_string(g)
2010   LOCal t,d
2020   d=81/(3^g)
2030   CLS
2040   PENDOWN
2050   FOR t=1 TO LEN(snow$)
2060     IF snow$(t)="l" THEN TURN 120
2070     IF snow$(t)="r" THEN TURN -60
2080     MOVE d
2090   END FOR t
2100 END DEFine draw_string
```

Listing 4

```
100 SCALE 100,0,0
110 MODE 4
120 z=2+.5*(1+SQRT(5))
130 pent$="11111"
140 FOR g=1 TO 5
150  POINT 50,28
160  TURNTO -72
170  draw_string g
180  string g
190 END FOR g
1000 DEFine PROCedure string(g)
1010   LOCal t,a$
1020   a$=""
1030   FOR t=1 TO LEN(pent$)
1040     IF pent$(t)="l" THEN a$=a$&"lrllr"
1050     IF pent$(t)="r" THEN a$=a$&"rrllr"
1060   END FOR t
1070   pent$=a$
1080 END DEFine string
2000 DEFine PROCedure draw_string(g)
2010   LOCal t,d
2020   d=150/(z^g)
2030   PENDOWN
2040   FOR t=1 TO LEN(pent$)
2050     IF pent$(t)="l" THEN TURN 72
2060     IF pent$(t)="r" THEN TURN -72
2070     MOVE d
2080   END FOR t
2090 END DEFine draw_string
```

Listing 5

```
100 SCALE 100,0,0
110 MODE 4
120 drag$="rr"
130 FOR g=1 TO 12
140   string g
150   CLS
160   AT 0,0:PRINT "Dragon curve"\"Stage"!g
170   draw_string g
180 END FOR g
1000 DEFine PROCedure string
1010   LOCal t,k,a$
1020   a$=""
1030   FOR t=1 TO LEN(drag$)
1040     k=t MOD 2
1050     IF drag$(t)="r" AND k=1 THEN a$=a$&"rl"
1060     IF drag$(t)="r" AND k=0 THEN a$=a$&"rr"
1070     IF drag$(t)="l" AND k=1 THEN a$=a$&"ll"
1080     IF drag$(t)="l" AND k=0 THEN a$=a$&"lr"
1090   END FOR t
1100   drag$=a$
1110 END DEFine string
2000 DEFine PROCedure draw_string(g)
2010   LOCal t,d
2020   d= 65/(1.618)^g
2030   POINT 35,35
2040   TURNTO 108-36*g
2050   PENDOWN
2060   FOR t=1 TO LEN(drag$)
2070     IF drag$(t)="l" THEN TURN 72
2080     IF drag$(t)="r" THEN TURN -72
2090     MOVE d/2+d*RND
2100   END FOR t
2110 END DEFine draw_string
```

Listing 6

```
100 SCALE 100,0,0
110 MODE 4
120 drag$="rr"
130 FOR g=1 TO 12
140   string g
150   CLS
160   AT 0,0:PRINT "Dragon curve"\"Stage"!g
170   draw_string g
180 END FOR g
1000 DEFine PROCedure string
1010   LOCal t,k,a$
1020   a$=""
1030   FOR t=1 TO LEN(drag$)
1040     k=t MOD 2
1050     IF drag$(t)="r" AND k=1 THEN a$=a$&"rl"
1060     IF drag$(t)="r" AND k=0 THEN a$=a$&"rr"
1070     IF drag$(t)="l" AND k=1 THEN a$=a$&"ll"
1080     IF drag$(t)="l" AND k=0 THEN a$=a$&"lr"
1090   END FOR t
1100   drag$=a$
1110 END DEFine string
2000 DEFine PROCedure draw_string(g)
2010   LOCal t,d
2020   d= 65/(1.414)^g
2030   POINT 35,35
2040   TURNTO 135-45*g
2050   PENDOWN
2060   FOR t=1 TO LEN(drag$)
2070     IF drag$(t)="l" THEN TURN 90
2080     IF drag$(t)="r" THEN TURN -90
2090     MOVE d
2100   END FOR t
2110 END DEFine draw_string
```

bends on alternating sides of the line. *Listing 5* draws dragons (named for their convoluted shape, rather like a Chinese dragon if you have a good imagination!) using a variation on the 'left/right sequence' technique.

Although the dragon curve runs into itself every so often, it never retraces the same segment twice. At advanced stages, it's quite remarkable how the dragon curve tucks itself snugly into various 'bays' left in what's been drawn so far.

As mentioned fractals can resemble coastlines. *Listing 6* shows some quite convincing examples using this method. Like real coastlines, there are not only promontories and bays but smaller bays in the sides of the bays, and smaller promontories in the bays, and so on.

NOW, buy a CARTRIDGE TIDY and you're a free! member of CARTRIDGE CADDY SYSTEM

Cartridges available separately (with labels), at £1.90 each. Guaranteed availability and 24-hour delivery

- Holds 16 Microdrive Cartridges
- Interlocking design feature allows Horizontal and Vertical Stacking –Space Saver

What Cartridge Caddy membership offers you

GUARANTEED STOCKS

Microdrive cartridges at £1.50 delivered to your door! within 24 hours

Please send me . . . . (Qty) Cartridge Tidy's, ☐ZX ☐QL, at £6.99 each, inclusive. and . . . . blank microdrive cartridges at £1.50 each I enclose a cheque for £ . . . . Add £1 p&p

Make cheques payable to: 4 SYSTEMS

4 SYSTEMS, M.O.D. supplier to leading U.K. businesses and universities

Mr/Mrs/Miss

Address

4 Systems

68 FOXWOOD CLOSE, FELTHAM, MIDDLESEX TW13 7DL TEL: 01-844 1399

APPROVED for use with telecommunication systems run by British Telecommunications in accordance with the conditions in the instructions for use

Prestel and Telecom Gold are registered trade marks of British Telecommunications plc.

TANDATA FOR THE

Q-CONNECT

Complete RS232 output for general communications from 75-9,600 baud with full two-way buffering and flow control. Plus all the software on a micro drive to support all 3 modules. The software includes Prestel Viewdata/Videotex AND VT100 emulation, as well as user-to-user communications with error-corrected file transfer (with encryption for security) and many other features including a telesoftware downloader.

Connects to the QL's SER 2 port allowing the micro to be used with virtually any asynchronous modem and for general data communications.

Q-MOD

Manual dial V23 modem operating at 1200/75 baud and 1200/1200 baud half duplex.

Q-CALL

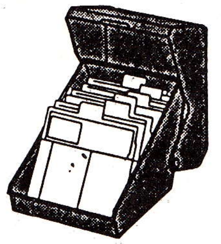
Provides traditional pulse/loop disconnect auto-dial and auto-answer.

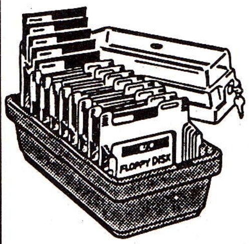
Disk Storage No. 1

This diskette storage holds 70 diskettes

Our **Super Low** price

£14.95 including VAT





Disk Storage No. DX45

This lockable diskette box holds 45 diskettes and displays them as shown

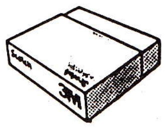
Super Low price

£12.00 incl. VAT

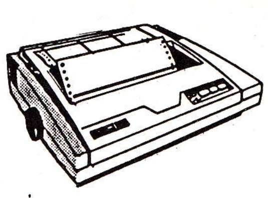
3M Scotch Disc

5¼″ diskettes single sided

	1-2 boxes	3-9 boxes	10+ boxes
SS 40 TRACK	£15.00	£13.50	£13.00
DS 40 TRACK	£20.00	£18.40	£17.70
SS 80 TRACK	£23.00	£20.70	£20.00
DS 80 TRACK	£26.70	£24.00	£23.00



10 DISCS PER BOX



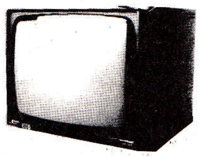
Dot Matrix Printers

Taxan Kaga KP810 **£279**
Canon PW1080A **£279**
Epson RX80/FT **£269**
Epson FX80 **£389**
Shinwa CPA80 (100cps) **£219**
Mannesman Tally MT80 **£209**
All prices include VAT

Daisy Wheel Printers

Juki 6100 **£339**
Quen Data **£239**
All prices include VAT

Please add £25 for printer lead (serial to parallel)



Microvitec Cub Monitor

Price

£275 including VAT

Please add for carriage:
A — £2, B — £3, C — £8

Delivery usually less than 7 days

Post to: VIGLEN COMPUTER SUPPLIES, UNIT 7, TRUMPERS WAY, HANWELL W7 2QA. Telephone: 01-843 9903. QL7

Please send me ______

I enclose Cheque/P.O. for £ ______ Incl. carriage
Cheques payable to Viglen Computer Supplies

Name ______

Address ______

Credit cards valid only if signed by card holder. Address must be the same as card holder.

I prefer to pay by ACCESS/BARCLAYCARD
(Delete whichever is not applicable)

Card No. ______

Signature ______

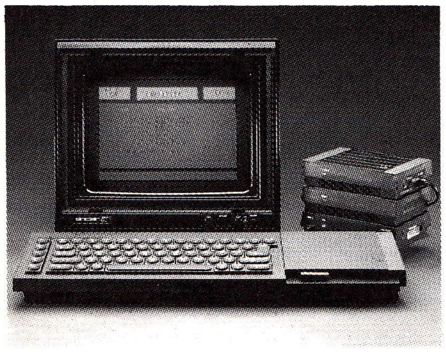
COMMUNICATIONS SINCLAIR QL

Thanks to Tandata you can now convert your QL into a powerful and comprehensive communicating terminal. You can contact distant databases such as British Telecom's Prestel system, private viewdata systems, traditional ASCII databases and electronic messaging/mail services such as Telecom Gold. You can even replace your existing VT100 terminal and enjoy secure communications with other QLs.

The three smart modules have been designed to match the QL in style and for added convenience they stack together using vertical bus connectors without the need for interconnecting cables.

The modules are available separately, but by using all three as a complete matched system full advantage can be taken of the integrated features of the Q-CONNECT's software.

If you'd like to know more, simply complete the coupon.



Tandata

Tandata Marketing Limited

Albert Road North, Malvern, Worcs. WR14 2TL.
Telephone: 06845 68421. Telex: 337617 Tandat G.
Prestel *799# Telecom Gold 81: TAN001
A subsidiary of Tandata Holdings plc.

QL7

I'd like to know more about Tandata communications for the QL.

Name ______

Address ______

______

______

______

Tel No ______

Send to:
Tandata Marketing Ltd., Albert Road North, Malvern, Worcs. WR14 2TL.

MACHINE CODE TUTORIAL

Adam Denning guides us through some of the 68000's more advanced features.

Last month we discovered how to add numbers using simple 68000 machine code instructions. We also discovered one method of returning the results to us via the SuperBasic interpreter, but there's an awful lot more that we can do with a microprocessor as sophisticated as the QL's 68008.

To be able to get to grips with the more advanced features, we need to find out more about addressing modes and instruction types. We'll do this by example, as it's a more pleasant way to learn than pure theory!

To make things more interesting, we'd obviously like to see what we're doing; we have to be able to read the keyboard and write onto the screen, and the only sensible way of doing this is through the QL's operating system, QDOS.

As programmers familiar with SuperBasic will already know, we can only read and write from input and output ('i/o') devices by opening *channels* to them. We do this from SuperBasic by using the OPEN keyword with a channel number and device specifier (or name), as in OPEN#3,mdv1_myfile.

This statement causes the specified device or file ('mdv1_myfile' here) to be opened and *associated* with channel 3. The SuperBasic OPEN keyword has OPEN_IN and OPEN_NEW variants to allow us to specify that a device is to be opened for either input (OPEN_IN) or output (OPEN_OUT) only. When a channel has been opened for input, we may read data from the device or file attached to the channel using the SuperBasic INKEY and INPUT statements. When a channel has been opened for output, we may write data to the device using the PRINT sttement.

Things are much the same in machine code, except tht words like INKEY or PRINT can't be used, as the 68000 has no idea what they mean. Also, we can't specify which channel number to use; QDOS tells us. But this is more sensible than it sounds, as we'll find out.

To open a channel to a device, we use one of the 68000's exceptions – the TRAP. TRAP #2 is reserved by QDOS for this purpose (amongst others), and we tell it to open a channel by putting a special number into register DO before executing the TRAP instruction. We must also give the routine some more information. It needs to know the name of the device or file to open and which method to use (ie, for input only, output only, or both at the same time). Finally, it needs to know which *job* the channel is being opened for – which will take us into the advanced realm of multitasking.

When the routine invoked by the TRAP #2 exception has finished, it returns to the point in our program following the TRAP instruction, with a few values in certain registers to tell us if it succeeded. We'll examine these shortly, but first we'll write a short and complete program to print a string (a sequence of characters) onto the screen *(Listing 1)*.

Then this machine code program can be loaded and executed from SuperBasic using *Listing 2*

What we're doing here is rather more straightforward than it seems. The first instruction *loads the effective address* of the name of the device FNAME into A0. Now, although it may not look like it, we're using program counter relative addressing here, so the LEA instruction loads the current value of the PC into A0 and then adds the offset from this point in the program to the start of the message. This ends up with the absolute address of FNAME being in A0. We then move some values into data registers ready for the QDOS routine. We put 1 in D0 to signify 'IO.OPEN', which is the name of the file opening routine, and –1 in D1 to say 'for this job'. The code we put into D3 is 2, which means 'open as a new file'. We then call the routine with TRAP #2. Normally, we would then check that everything worked by examining D0 after the trap. If it is zero, all went well; if not, it holds the relevant QDOS error code. To make things easier we'll assume that it has succeeded.

```
IO.OPEN   EQU     1            Routine to open a channel
IO.CLOSE  EQU     2            Routine to close a channel
UT.MTEXT  EQU     $D0          Routine to print a message

          LEA.L   FNAME,A0
          MOVEQ   #IO.OPEN,D0
          MOVEQ   #-1,D1
          MOVEQ   #2,D3
          TRAP    #2
          LEA.L   MESSAGE,A1
          MOVE.W  UT.MTEXT,A2
          JSR     (A2)
          MOVEQ   #IO.CLOSE,D0
          TRAP    #2
          RTS

FNAME     DC.W    4            This is the length of the name
          DC.B    'SCR_'       and these are the characters

MESSAGE   EQU     *            The message goes here
```

Listing 1. Printing characters to the screen.

```
100 addr=RESPR(512)
110 RESTORE
120 FOR x=0 TO 16:READ p:POKE_W addr+(x*2),p
130 REPeat loop
140   INPUT#0;'What is your message?'!m$
150   POKE_W addr+34,LEN(m$)
160   FOR x=1 TO LEN(m$):POKE addr+x+35,CODE(m$(x))
170   CALL addr
180 END REPeat loop
1000 DATA 16890,26:REMark LEA.L FNAME,A0
1010 DATA 28673,29439:REMark MOVEQ #1,D0  MOVEQ #-1,D1
1020 DATA 30210,20034:REMark MOVEQ #2,D3  TRAP #2
1030 DATA 17402,20:REMark LEA.L MESSAGE,A1
1040 DATA 13432,208:REMark MOVE.W UT.MTEXT,A2
1050 DATA 20114,28674:REMark JSR (A2)  MOVEQ #2,D0
1060 DATA 20034,20085:REMark TRAP #2  RTS
1070 DATA 4:REMark Length of device name
1080 DATA 21315,21087:REMark 'SCR_'
```

Listing 2. Loader for the above program.

After opening the channel, we need to print our message to it. The message starts at the MESSAGE label, so we need to load its address into A1, using LEA once more. There are numerous ways of printing out information on the QL, but the easiest way to print a message is to use the utility routine 'UT.MTEXT'. The address of this routine is held in address $D0, so we use MOVE to put it into A2.

To actually call the routine, we use address register indirect addressing, which in the case of JSR and JMP finds the address held in the specified register and loads it into the program counter. This is slightly different from the usual address register indirect system, which features data from the address pointed to by the register.

Once UT.MTEXT will print out the message, we must close the channel. A further TRAP #2 exception routine does this, and all we need to do is put 2 into D2 and the *channel ID* into A0. The channel ID is the unique number returned by IO.OPEN when it's opened the channel. Thankfully, it returned this number in A0, and it's still there. The final instruction, RTS, takes us back to BASIC. Normally we would need to clear D0 to O before returning to BASIC, otherwise an error would be generated. But here, we can take advantage of the fact tht IO.CLOSE will do this for us if successful, which it did.

Next issue, we finish the series off by writing a large program using several QDOS routines.

AT LAST!

THE QL COMMUNICATES

ANNOUNCING THE QL BRIGHT STAR

The only multi-standard intelligent modem, suitable for use with the QL.

MANUFACTURED BY MODEM HOUSE — WHO ELSE!

1985 is The Year Of Communications, so . . . link your QL to Prestel, Cityservice, Homelink, Farmlink, BT Gold, Knowledge Index, Easylink, One to One etc.

The QL Bright Star multi-standard modem is a completely new concept in modem design, and is supplied with communications software and built in centronics printer interface.

The Bright Star is truly a Quantum leap for the QL.

The microprocessor controlled modem allows you to spool incoming data to microdrive or printer, save data to microdrive or printer, or link your QL to a standard centronics printer.

Data Encryption facilities and fully comprehensive help menu provided as standard.

Auto-Answer, Auto-dial version coming soon!

Main specifications: CCITT V21/V23, BELL 103/113

Speeds: 300/300, 1200/75 75/1200, 1200/1200

The QL Bright Star is truly the last word in communications.

Special note: This product is not, and has not been connected in any way, with the Q-CON or O.E. Ltd, and is British designed and manufactured.

At an amazing special price of £179.95 complete (RRSP £199.95)

MODEM HOUSE

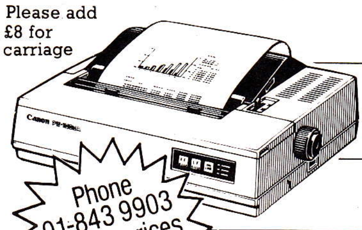
Iolanthe Drive, Exeter, Devon EX4 9EA

Telephone: (0392) 69295

The best service, the best value in printers from Viglen

Dot Matrix Printers

Please add £8 for carriage



Canon PW1080A/ Taxan Kaga KP810 **£279**

Epson RX80/FT **£275**

Epson FX80 **£389**

Phone 01-843 9903 for best prices on Monitors

Daisy Wheel Printers

NEW LOW PRICES

Juki 6100 **£369**

Quen Data **£247**

Please add £8 for carriage

Please add £25.00 for the Serial to Centronics Interface Cable for the QL.

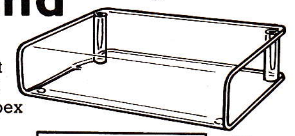
Viglen are also major suppliers to educational and government establishments and welcome further enquiries and orders.

Visit our showroom
Weekdays 9.30 – 6pm.
Saturday 9.30 – 4pm.

All prices correct at time of going to press

ALL PRICES INCLUDE VAT

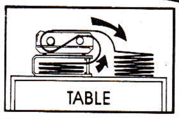
Printer Stand



- 80 column dot matrix printers
- Raises printer high enough to put continuous stationery underneath
- Beautifully finished in clear perspex
- Viglen quality every time
- Will accept paper up to 12½" wide
- Non-slip rubber pads

Dimensions:
15" (380mm) wide
12½" (320mm) deep
4" (90mm) high

Please add £3 for carriage



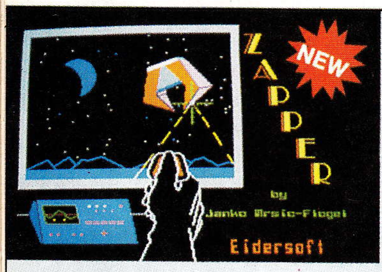
Our price **£14.95** inc. VAT

Also available 136 column printer stand £27.00

Post to: Viglen Computer Supplies.
Unit 7, Trumpers Way, Hanwell W7 2QA.

Credit Card Holders may order by phone. 01-843 9903.

Please send me	
I enclose Cheque/P.O. for £	Incl. carriage.
Cheques payable to Viglen Computer Supplies.	
Name	
Address	
I prefer to pay by ACCESS/BARCLAYCARD (Delete whichever is not applicable)	Credit Cards valid only if signed by card holder. Address must be the same as card holder.
Card No.	
Signature	

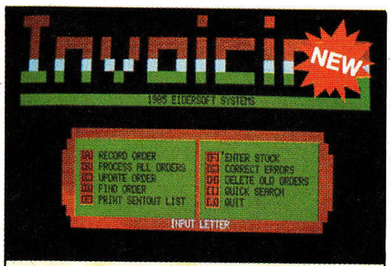


ZAPPER £10.95

ALSO ON 40 TRACK DISK

100% MACHINE CODE GAME. FUN FOR ALL THE FAMILY. FEATURES 8 LEVELS OF MULTICOLOUR GRAPHICS; SUPER-SMOOTH HIGH SPEED ACTION; SOUND; HIGH SCORE; VERY ADDICTIVE!!

LENGTH 65K APPROX. LOAD TIME 1 MIN
TV/MONITOR JOYSTICK/CURSOR.



ARCHIVER £16.95

ALSO ON 40 TRACK DISK

QUALITY BUSINESS PROGRAMS FOR PSION ARCHIVE VERS 2.

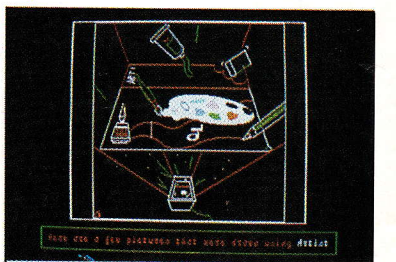
A MASSIVE PACKAGE OF TEMPLATES AND **READY TO RUN** PROFESSIONAL PROGRAMS THAT CAN BE ADAPTED TO SUIT YOUR OWN PURPOSES.

STOCK CONTROL INVOICING MAILING / ADDRESS DATABASE APPOINTMENTS SYSTEM

MANY ROUTINES INCLUDED FOR DATA ENTRY, BACKUP, PRINTER DRIVING, SEARCH, DATING ERROR TRAPPING AND MUCH MORE!!

FULL 48 PAGE USER AND TECHNICAL MANUAL

LOAD TIME : VARIOUS LENGTH : 100K+ STOP PRESS - A FULL RANGE OF EXTRA TEMPLATES AVAILABLE. SEE PACK FOR DETAILS.



QLART £14.95

ALSO ON 40 TRACK DISK

THE ULTIMATE GRAPHICS AID FOR SUPERBASIC 4 GREAT PROGRAMS TO MAKE QL GRAPHICS EASY

★ Cursor Drawing with advanced features
★ Character Designer (all sizes)
★ Window Designer
★ File Manager
★ + High Res. Printer Dumps

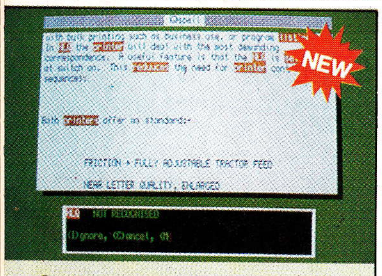
LOAD TIME 26 SECS. LENGTH 100K APPROX. MANUAL 13 PAGES

EIDERBOX

WITH 4 MICRODRIVE CARTRIDGES

An attractive video style storage box with 4 cartridges, 50 labels, full index book and microdrive guide

ONLY £7.95! + 50p P&P ONLY FROM EIDERSOFT!


Qspell
NEW
FRICTION + FULLY ADJUSTABLE TRACTOR FEED
NEAR LETTER QUALITY, ENLARGED
NOT RECOGNISED
(I)gnore, (C)ancel,

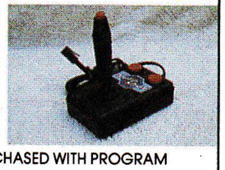
QSPELL £19.95

ALSO ON 40 TRACK DISK

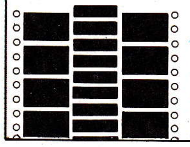
A SPELLING CHECKER FOR QUILL WITH MANY ADVANCED FEATURES!
★ 25,000 WORD DICTIONARY
★ RESIDES IN MEMORY!
★ USER DICTIONARIES
★ AUTO LEARNS WORDS
★ SOLVES CROSSWORDS ANAGRAMS & PUZZLES

LOAD TIME 1·28. MANUAL 8 PAGES. LENGTH 90K. A4 PAGE SPEED 24 SECS.

QL JOYSTICKS



SURESHOT QL microswitched steelshaft £25.95
QUICKSHOT QL world beater £13.95
QL CONVERT LEAD converts Atari Joysticks £5.49
£2 DISCOUNT IF PURCHASED WITH PROGRAM



MICRODRIVE LABELS
on printer tractor feed
WHITE/RED/BLUE/GREEN/YELLOW
500 WHITE £4.95
500 COLOUR £5.95
KIT OF 2000 ALL COLOURS PLUS QL DRIVER SOFTWARE £24.95

EIDERSOFT

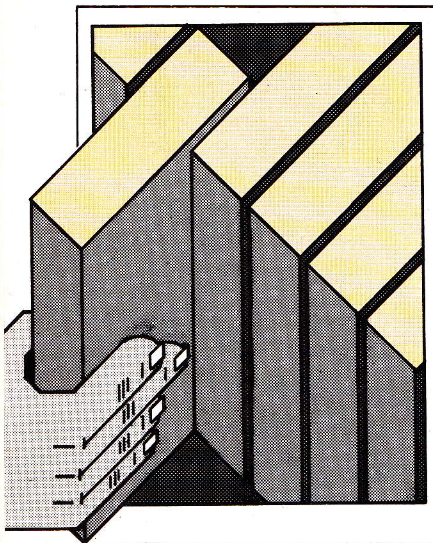
Please note all photographs appearing in this advertisement are actual screen shots of our programs.

QL SYSTEMS 01-478 1291 TLX.8951807

DEALERS REQUIRED PLEASE PHONE FOR DETAILS

All prices are subject to alteration.

CREDIT CARDS 01-478 1291 DESPATCH WITHIN 10 DAYS OR MONEY BACK
SEND TO EIDERSOFT, THE OFFICE, HALL FARM, N. OCKENDON, UPMINSTER, ESSEX RM14 3QH
NAME.......................
MICRODRIVE/DISK
ADDRESS.......................POSTCODE.......
DISCOUNT £.......
PLEASE SUPPLY (QTY) ZAPPER £10.95 ( )
ARCHIVER £16.95 ( ) QLART £14.95 ( )
QSPELL £19.95 ( ) QDUMPS £9.95 ( )
QUICKSHOT QL £13.95 ( ) LABELS ( )
SURESHOT QL £25.95 ( )
LABEL KIT £24.95 ( )
I ENCLOSE CHEQUE FOR
£.......



BOOKMARKS

Authors pander to the needs of dedicated programmers this month, as Nicky Trevett finds out.

From Sunshine comes Andrew Pennell's *The Sinclair QDOS Companion,* price £6.95. This book is, as the author points out, very definitely for those who know 68000/8 machine code and who wish to put their knowledge to practical use. It is worth bearing in mind that the book was written as a sequel to the author's previous work, *Assembly Language Programming on the Sinclair QL*.

The book starts with a fascinating history of QDOS (and, incidentally, SuperBasic), which includes the obligatory 'Domesdos' anecdote, and finishes up with a brief look at the various versions of QDOS, culminating in version 1.03, the 'current' version on which the book is based. If your machine contains 1.02, however, the book should still be relevant.

From there it's straight into memory organisation, with a useful diagram illustrating the hardware memory map and the usage of each port – the real-time clock, the write-only and read-only ports, and so on. There's another diagram showing the RAM memory map, a listing of the system variables (byte space reserved for use by the system, some of which is unallocated and available for user programs), and a look at the ways QDOS can be called up via system traps and system vectors.

Having flitted about QDOS in this way, the book proceeds to examine in detail multi-tasking, input/output, the device drivers, interrupts and external ROMs and device drivers. There's a chapter devoted to the 8049 second processor, the slave processor which is not directly accessible from the 68008, and another which looks at the QDOS utilities.

Considering the heavily technical nature of the content, the style is informal, and mostly avoids being a text book. Much of the material seems based on personal experience with QDOS, as indeed the preface claims, and the result is a practically-orientated guide, recommended to those with a serious interest in machine code programming, who don't so much want a quick reference as a narrative introduction to the subject.

Full Steam Ahead

If, on the other hand, you are after a comprehensive reference manual which contains pretty well all the QDOS routines you are ever likely to need, Adrian Dickens' *QL Advanced User Guide* could be for you.

Published by Adder at £12.95, this is a heavyweight volume, about twice as thick as Andrew Pennell's work, and covering, to an extent, much the same ground.

The introduction makes the author's intention clear. The QL User Guide is woefully inadequate as far as "the more interesting aspects of the QL as a new generation of computer" are concerned. To make the most of your QL, you must forget about limited old BASIC and go straight for assembly language programming. Only then can you expect to take full advantage of the QL's potential – in multi-tasking, and in producing your own advanced programs. Above all, you will be able to take full control.

It's aimed at the same type of user as *The Sinclair QDOS Companion,* but its approach is quite different. The book starts with an introduction to the 68008 microprocessor, a useful summary but hardly a beginner's guide to machine code. You will need basic knowledge of machine code to make the most of the book as a whole.

Most of the rest of the book is presented as the sort of reference guide that Andrew Pennell's book *isn't* but first comes an overview of QDOS, covering concepts like multi-tasking and system control, and a chapter entitled "Experimenting with QDOS". This looks at SuperBasic keywords "not properly covered in the Sinclair QL User Guide", like Call, Exec, Peek, Poke and so on, and the Experimentor program, which allows you to get into QDOS from BASIC and is invaluable if you don't possess an assembler. The chapter includes a full Experimentor listing, and some example programs written in assembler to get you started.

The remaining chapters cover the sections of QDOS in detail, starting with a few pages of introduction followed by an extensive reference section. Thus the chapter on manager traps starts with a discussion of the part they play within QDOS, looks at the different types of memory allocation, then moves on to the nitty gritty: first a list summarising the traps and what they do, then a closer look at each trap in turn.

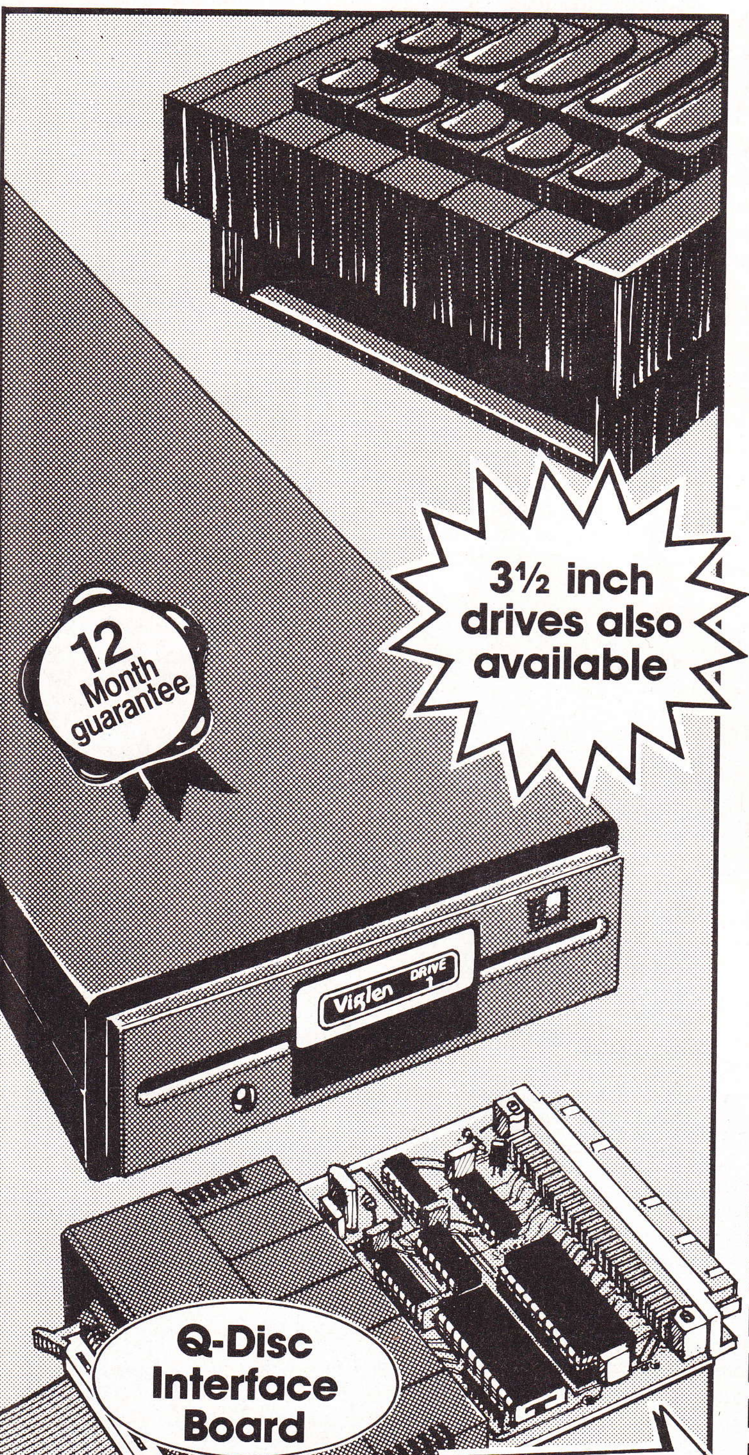
AI Academia

QL users with a serious interest in artificial intelligence, unsatisfied by the 'hobbyist' books that have so far appeared on the subject, should welcome *Artificial Intelligence: Tools, Techniques and Applications,* published by Harper and Row at £12.95.

Edited by Tim O'Shea and Marc Eisenstadt, this is a 500-page compilation of articles on matters pertaining to mainstream AI research, including Prolog and Lisp, robot control and natural language, text processing and expert systems. It is not geared to any particular type of hardware, or any specific reader; these are papers written by experts across a broad range of subjects, for academics, industrialists, or quite simply for anyone who is interested.

PHOTO BY TERRY BEDDIS

Viglen offer the best alternative to the microdrive


3½ inch drives also available
12 Month guarantee
Viglen DRIVE 1
Q-Disc Interface Board
ORDERS are usually despatched within 24 hours

Speeding up your data access with Viglen disc drives means instant reliable access every time

Specially designed to fit neatly into the case of the Sinclair QL, the Q-Disc Interface Board and companion disc drives are colour-matched to compliment your QL computer.

Q-Disc Interface + 200k single drive (Formatted capacity)	**£199 inc VAT**
Q-Disc Interface +800k single drive (Formatted capacity)	**£249 inc VAT**
Q-Disc Interface + 1.6M dual drive (Formatted capacity)	**£329 inc VAT**

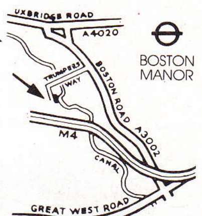
The above systems are complete and ready to plug in and use, they include a Utilities disc to convert Psion exchange software from microdrive to disc.

Q-Disc Interface on its own INC. VAT **£149**

Prices correct at time of going to press. Offers subject to availability. Even lower prices with other QL interfaces.

Visit our showrooms
Open Monday – Friday 9.30 – 6.00 Saturday 9.30 – 4.00

Unit 7, Trumpers Way
Hanwell W7 2QA



Carriage
Please add £12.00 for carriage. Orders are usually despatched to you within twenty-four hours of receipt.

Viglen are also major suppliers to educational and government establishments and welcome further enquiries and orders.

Post to: VIGLEN COMPUTER SUPPLIES.
Unit 7, Trumpers Way, Hanwell W7 2QA.
Credit card holders may order by telephone on 01-843 9903.

Please send me ____________________

I enclose Cheque/P.O. for £ __________ incl. carriage.
Cheques payable to Viglen Computer Supplies.

Name: ____________________

Address: ____________________

____

____

I prefer to pay by ACCESS/BARCLAYCARD (delete one).

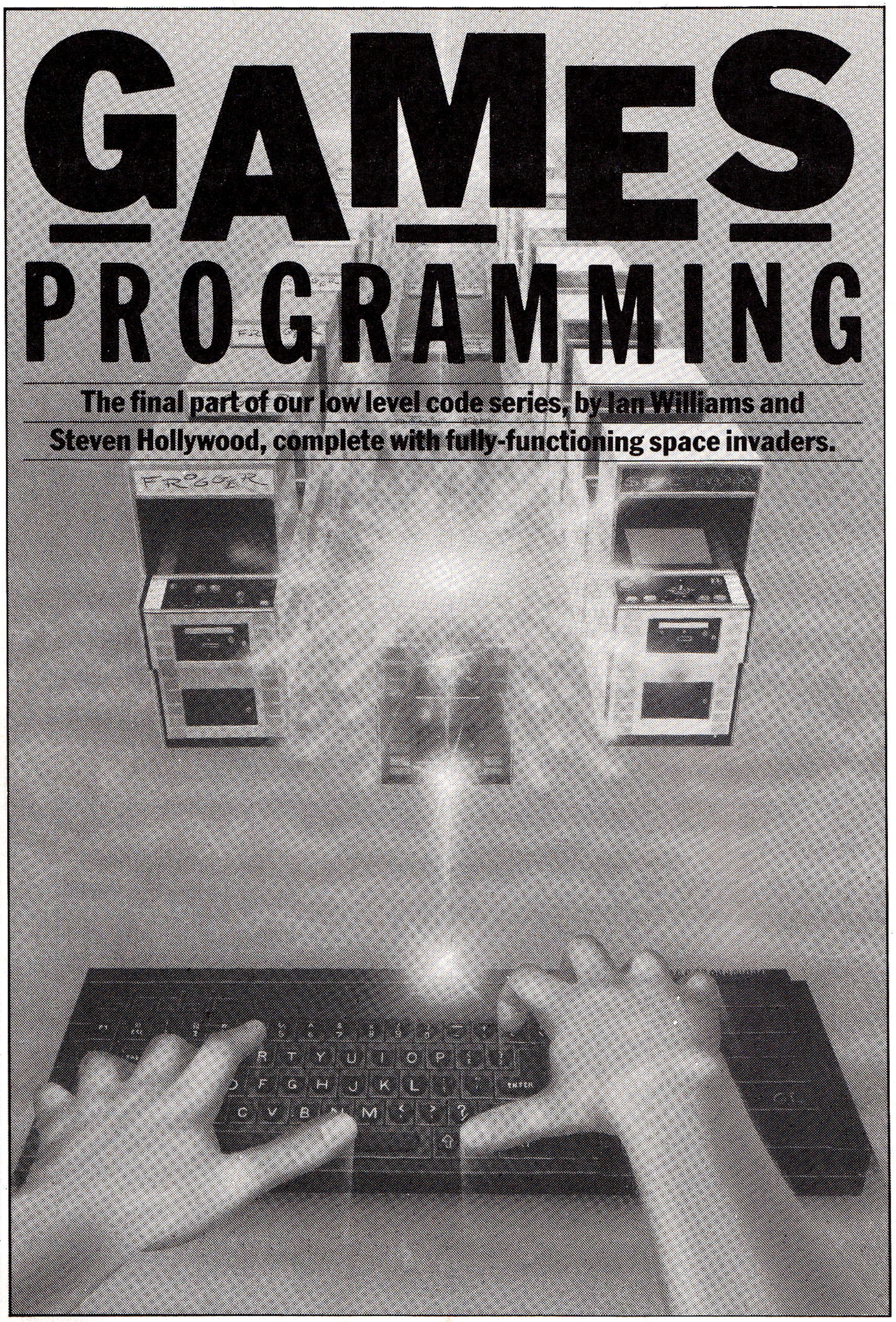
Card No: ____________________

Signature: ____________________

Credit Cards valid only if signed by card holder. Address must be the same as card holder.

GAMES PROGRAMMING

The final part of our low level code series, by Ian Williams and Steven Hollywood, complete with fully-functioning space invaders.



Here is the last part of the series and, when entered, will give you fully working, high speed arcade Space Invaders which you can zap to your heart's content between those long hard hours of sweat, toil and tears.

This month we'll look at how to print the scores, drop the bombs from the invaders, detect the resultant collisions, how to make a mother ship appear and reincarnate brand new bases from the ravaged ashes of their forebearers.

The mother ship is constructed as two sprites in the data and moved across the screen in 'baddybus'. This routine is extremely straightforward and simply replots the two sprites horizontally, as explained in the program comments.

The score is managed by four routines; Print_score, print_hi_score, print_num and inc_score. The scores are stored in Binary Coded Decimal (BCD) in the variables Score and Hi_score. In BCD each nybble of the byte, word or longword in memory is only allowed to cover the range of values 0 to 9. This effectively means that each nybble will represent a decimal digit.

When addition is needed a new problem arises. In the game we use six digit scores but the 'add' version of the command will only add one byte, or two BCD digits at a time. It's therefore necessary to have a buffer (already defined) called 'mess' (defined as a long word) into which we move $100 (line 61). This is then added, byte by byte using the 'abcd' command to the long word containing the scores.

Although the above might seem a little complicated it actually makes life a lot easier when we come to printing out the digits on screen (the command 'Print_num' does this).

The bomb dropping routines are fairly uncomplicated consisting of two sections, 'Bombs' and 'drop_bomb'. The first seeks out the vertically closest invader to the gunbase and initialises it as the one which drops the bomb. The second actually drops the bomb and checks for collisions.

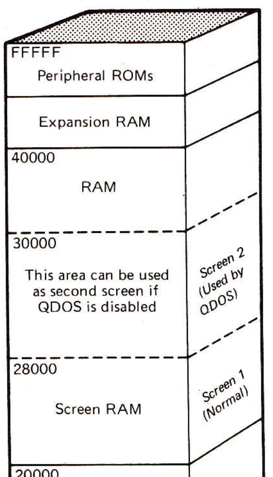
Collision detection is a similar process to that used in 'bul_mov' which was explained last month. When the bomb hits the barriers the process of gradually tearing chunks out of the barriers is very similar to that used in 'bul_hit_bar', also in last month's issue.

One of the interesting things about magazine programs is trying out modifications and improvements. There are a number of ways to play about with this program, one of which might be the speed at which the bombs drop; the set speed is rather fast making it very difficult to start the game and still retain any self-respect!

As the program stands, there always has to be a bomb on screen. Slowing down the bomb is relatively easy through the simple expedient of calling the 'drop_bombs' loop fewer times from the main loop (starting at line 33 published in the first part).

The remainder of this month's program is called by the routine 'base_ded'. When the gun base from which you are firing is struck by a bomb, this is detected in lines 134 – 145 (drop_bomb) and the program branches accordingly. After making the appropriate explosion noise the routine performs, in effect, a 'tidying up' job on the screen, removing any old bombs or bullets in flight and completing all the explosion sequences necessary. The number of lives is then examined and, if some re-

Fig 1



main, 'waitkey' is then called, a simple routine which awaits a keypress; in this case the space bar or fire button.

If there are no lives left, however, the scores are examined to see if yours is higher than the current highest score. If so, then it is placed into the high score. The 'Play again (Y/N)' message is then loaded. At first sight the way in which this is done might seem a little odd, especially as the method uses over 700 bytes! The reason, however, is fairly simple but does involve an understanding of how the QL uses its screens.

Many arcade type games on computers use the technique of screen flipping; the process by which the areas in memory used by the screen are alternately displayed at high speed. This allows alterations to be made to the 'concealed' screen whilst displaying the 'active' screen. Thus, high speed animation effects can be achieved. The QL is easily capable of high speed and smooth action without this method but screen flipping offers even faster and more versatile graphic displays. The one disadvantage in the QL's case is that QDOS must be disabled as the area which could be used as screen 2 is occupied by the management system variables, buffers and channel definitions. *Fig 1* shows the layout in memory and the location in which the second screen resides.

Throughout the whole of the program QDOS is used as little as possible and the loading of the 'Play again' message permits us, in future modifications, to enable the screen flipping process.

If you've been faithfully entering everything each month you will have noticed a problem with part three. Somehow, gremlins detached part of our program in the first month and you will have had difficulties without that part, which is given in full below. We do apologise for that and hope the fun you'll have playing on the completed version makes up for the frustration of not being able to get all of it working perfectly as you went along.

Finally, remove the remaining rems (;) from all the program lines. The full program is only 9k long in code but over 25k in assembler without the explanatory rems – Happy zapping!

```
0000           ;
0000           ;       ************************************************
0000           ;       *                                              *
0000           ;       *                << PALADIN >>                 *
0000           ;       *                                              *
0000           ;       *                     by                       *
0000           ;       *                                              *
0000           ;       *  STEVEN HOLLYWOOD     &       IAN G WILLIAMS *
0000           ;       *                                              *
0000           ;       ************************************************
0000           ;
0000           ;
0000           ;
0000           ;
0000           print_score                                 ; This routine prints the score
0000 48E70380          movem.l d6-d7/a0,-(a7)              ; at the top RH side of the
0004 3C3C0000          move.w  #score_pos,d6               ; screen at score_pos(x co-ord)
0008 7E00              moveq   #0,d7                       ; y co-ord
000A 41FAFFFE          lea     score,a0                    ;
000E 610E              bsr.s   print_num                   ; see below
0010 4CDF01C0          movem.l (a7)+,d6-d7/a0              ;
0014 4E75              rts                                 ;
0016           ;
0016           ;
0016           ;
0016           print_hiscore                               ;Same as above but for HI Score
0016 7C00              moveq   #score_poshi,d6             ; (x co-ord)
0018 7E00              moveq   #0,d7                       ; (y co-ord)
001A 41FAFFFE          lea     hiscore,a0                  ;
001E           ;
001E           ; This prints a six digit decimal number from a binary coded decimal
001E           ; long word pointed to by a0 (See text for a sensible explanation!)
001E           ;
001E           print_num                                   ;
001E 48E7C400          movem.l d0-d1/d5,-(a7)              ;
0022 3A06              move.w  d6,d5                       ; Store x pos in d5
0024 7205              moveq   #5,d1                       ; These four lines
0026 6100FFFE  next_w  bsr     blank_out                   ;  clear the position to
002A 04460008          sub.w   #8,d6                       ;   be occupied by the
002E 51C9FFF6          dbf     d1,next_w                   ;    number
0032 3C05              move.w  d5,d6                       ; Restore x pos
0034 7A00              moveq   #0,d5                       ;
0036 7205              moveq   #5,d1                       ; Set digit counter
0038 2010              move.l  (a0),d0                     ; Moves BCD long word to d0
003A 1A00      nexdig  move.b  d0,d5                       ; Move lowest 2 nybbles into d5
003C E888              lsr.l   #4,d0                       ; Shifts BCD nybbles down
003E 0205000F          and.b   #15,d5                      ; Masks off highest ny. of d5
0042 06050000          add.b   #num_base,d5                ; Adds sprite base (num_defs)
0046 6100FFFE          bsr     plot                        ; Put digit  on screen
004A 04460008          sub.w   #8,d6                       ; Shifts pointer back
004E 51C9FFEA          dbf     d1,nexdig                   ; loop for next digit
0052 4CDF0023          movem.l (a7)+,d0-d1/d5              ;
0056 4E75              rts                                 ;
0058           ;
0058           ; Procedure to increase score by whatever you wish (100 in this case)
0058           ;
0058           inc_score                                   ;
0058 023C00E7          and.b   #$e7,sr                     ; Clears the extend & neg.flag
005C 41FAFFAC          lea     score,a0                    ;
0060 43FAFFFE          lea     mess,a1                     ;
0064 22FC0000          move.l  #$100,(a1)+                 ; Moves 100 (BCD) into mess
0068 0100
006A [illegible]
006C 7003              moveq   #3,d0                       ; Sets up loop to add bytes
006E C109      sub_dig abcd    -(a1),-(a0)                 ; Adds the BCD bytes
0070 51C8FFFC          dbf     d0,sub_dig                  ; Next add operation
0074 618A              bsr.s   print_score                 ; Prints new score
0076 30280002          move.w  2(a0),d0                    ; These lines check to see if
007A 02400FFF          and.w   #$fff,d0                    ; last 3 digits are noughts
007E 662A              bne.s   odd                         ; If not, end
0080 41FAFFFE          lea     numflag,a0                  ; Increase number of flags
0084 5210              addq.b  #1,(a0)                     ;  by one
0086 0C100003          cmp.b   #3,(a0)                     ; Test to see if 3 present
008A 66000180          bne     print_flags                 ; If not, update & return
008E 4210              clr.b   (a0)                        ; If yes, reset no. of flags
0090 41FAFFFE          lea     num_bas,a0                  ; These lines provide an extra
0094 1C10              move.b  (a0),d6                     ;  base if more then three
0096 5210              addq.b  #1,(a0)                     ;   flags (3000 points)
0098 E94E              lsl.w   #4,d6                       ; Calculates pos. of new base
009A 5A46              addq.w  #5,d6                       ;
009C 1E3C00F8          move.b  #248,d7                     ; At bottom of screen
00A0 7A08              moveq   #8,d5                       ;
00A2 6100FFA2          bsr     plot                        ; Place a new base on screen
00A6 610001B2          bsr     rub_flag                    ; Removes obselete flags
00AA 4E75      odd     rts                                 ;
00AC           ;
00AC           ; This routine chooses the nearest invader to your gun  base and
00AC           ; intitiates the bomb drop
00AC           ;
00AC           bombs                                       ;
00AC 3C3AFFFE          move.w  gunpos,d6                   ; read gun pos. into d6
00B0 7600              moveq   #0,d3                       ;
00B2 343C1388          move.w  #5000,d2                    ; Set closest yet to 5000
00B6 41FAFFFE          lea     xpos,a0                     ; (a lonnng way away.....)
00BA 43FAFFFE          lea     ypos,a1                     ;
00BE 7000              moveq   #num_sp-1,d0                ; Loop counting off invaders
00C0 0C50FFFF  thing   cmp.w   #-1,(a0)                    ; If invader is dead
00C4 6712              beq.s   grogy                       ; ignore it
00C6 3206              move.w  d6,d1                       ; These lines compute distance
00C8 9250              sub.w   (a0),d1                     ; (hor) between inv. and gunbas
00CA 6A02              bpl.s   itspos                      ; If Dist. negative then
00CC 4441              neg.w   d1                          ; make it positive
00CE B242      itspos  cmp.w   d2,d1                       ; If closer than closest so far
00D0 6206              bhi.s   grogy                       ; then
00D2 3401              move.w  d1,d2                       ; Set closest
00D4 3810              move.w  (a0),d4                     ; Save x and y pos. of
00D6 1611              move.b  (a1),d3                     ; closest so far
00D8 5289      grogy   addq.l  #1,a1                       ; Moves pointers to next
00DA 5488              addq.l  #2,a0                       ; invader
00DC 51C8FFE2          dbf     d0,thing                    ; Check next invader
00E0 06030008          add.b   #8,d3                       ; moves one line down
00E4 41FAFFFE          lea     x_bomb,a0                   ; Initialise x pos. bomb var
00E8 3084              move.w  d4,(a0)                     ; to closest invader  x pos
00EA 41FAFFFE          lea     y_bomb,a0                   ; Same again for y pos
00EE 1083              move.b  d3,(a0)                     ;
00F0 1E03              move.b  d3,d7                       ; Initialise bombs onto
00F2 3C04              move.w  d4,d6                       ; Screen
00F4 7A09              moveq   #9,d5                       ;
00F6 6000FF4E          bra     plot                        ;
00FA           ;
00FA           ; This routine moves the bomb, checks for collisions etc...
00FA           ;
00FA           drop_bomb                                   ;
00FA 43FAFFEE          lea     y_bomb,a1                   ; Load y pos of bomb
00FE 1E11              move.b  (a1),d7                     ;
0100 3C3AFFE2          move.w  x_bomb,d6                   ; Same for x pos
0104 7A09              moveq   #9,d5                       ;
0106 6100FF3E          bsr     plot                        ; Un plot old bomb
010A 5207              addq.b  #1,d7                       ; Move it down one pixel
010C 0C0700F0          cmp.b   #240,d7                     ; Test for bottom hit
0110 6700FF9A          beq     bombs                       ; If yes - drop new bomb
0114 1287              move.b  d7,(a1)                     ; Store new bomb y pos
0116 5E07              addq.b  #7,d7                       ; moves to tip of bomb
0118 5846              addq.w  #4,d6                       ;
011A 6100FFFE          bsr     id_col                      ; Tests colour - (May issue)
011E 6724              beq.s   ex_drop                     ; If black plot new bomb
0120 0C0700E9          cmp.b   #233,d7                     ; Test for base collision
0124 6200005A          bhi     base_dead                   ; If yes kill base
0128 0242FF00          and.w   #$ff00,d2                   ; Masks off red byte
012C 6606              bne.s   joe                         ; If not red then No barr. col
012E 0C0700C7          cmp.b   #199,d7                     ; Is it low enough for barrier
0132 6218              bhi.s   hit_barrier                 ; Yes - kill a bit of barrier
0134 6100FFFE  joe     bsr     rev_bul                     ; Kill any bullet on screen
0138 41FAFFFE          lea     bulposx,a0                  ;
013C 30BCFFFF          move.w  #-1,(a0)                    ;
0140 6000FF6A          bra     bombs                       ; Drop new bomb
0144 5F07      ex_drop subq.b  #7,d7                       ; Put new bomb on screen
0146 5946              subq.w  #4,d6                       ;
0148 6000FEFC          bra     plot                        ;
014C           ;
014C           ; This routine takes a chunk out of the top of the barrier when hit
014C           ; by a bomb
014C           ;
014C           hit_barrier                                 ;
014C 5946              subq.w  #4,d6                       ;
014E 1607              move.b  d7,d3                       ; Save y pos of collision
0150 1E3C00C8          move.b  #200,d7                     ; Top of barrier
0154 45FAFFFE          lea     rndpos,a2                   ;
0158 7207              moveq   #7,d1                       ; Sets the count to zap 8 lines
015A 3252      scrub   move.w  (a2),a1                     ; These lines get a nearly
015C 3A11              move.w  (a1),d5                     ; random number for the
015E 5452              addq.w  #2,(a2)                     ; barrier destruction sequence
0160 02450007          and.w   #7,d5                       ; Masks off the upper bits
0164 DA03              add.b   d3,d5                       ; lower y co-ords of erasee
0166 040500C8          sub.b   #200,d5                     ; Calc. No. of dots from top
016A 6100FFFE          bsr     vline                       ; Rubs 'em out
016E 5246              addq.w  #1,d6                       ; Move to next column
0170 51C9FFE8          dbf     d1,scrub                    ; Erase next column
0174 47FA09E0          lea     crunch,a3                   ; Makes noise....
0178 7011              moveq   #17,d0                      ;
017A 4E41              trap    #1                          ;
017C 6000FF2E          bra     bombs                       ; Drops new bomb
0180           ;
0180           ; Deals with destruction of the gun base by a falling bomb and saves
0180           ; partially destroyed barriers for re-creation on screen after
0180           ; replacement of new base
0180           ;
0180           base_dead                                   ;
0180 47FA09E4          lea     base_bang,a3                ; Makes a noise...
0184 7011              moveq   #17,d0                      ;
0186 4E41              trap    #1                          ;
0188 588F              addq.l  #4,a7                       ; Removes return add. off stack
018A 1E3C00F0          move.b  #240,d7                     ; Sets up explosion position
018E 103AFFFE  waitexp move.b  seq_num,d0                  ; Completes any explosion
0192 6708              beq.s   okexp                       ; sequence in the process
0194 6100FFFE          bsr     draws_bang                  ; of actually
0198 6000FFF4          bra     waitexp                     ; exploding
019C 6100FFFE  okexp   bsr     explode                     ; Initialises explosion for gun
01A0 6100FF92          bsr     rev_bul                     ; Remove bullet
01A4 6100FFEE  grows   bsr     draws_bang                  ; draw frame of explosion
```

```
01A8 303C0D05              move.w  #3333,d0                ; Pause between frames
01AC 51CBFFFE     phut     dbf     d0,phut                 ;
01B0 103AFFDC              move.b  seq_num,d0              ; until end of sequence
01B4 66EE                  bne.s   grows                   ;
01B6 6100FFFE              bsr     savebarriers            ; Save barriers in buffer
01BA 41FAFFFE              lea     baddyx,a0               ;  Kill Mothership
01BE 30BCFFFF              move.w  #-1,(a0)                ;
01C2 41FAFECC              lea     num_bas,a0              ; Decrease no. of lives
01C6 5310                  subq.b  #1,(a0)                 ;
01C8 6B08                  bmi.s   endgame                 ; If none left then end
01CA 61000146              bsr     waitkey                 ; ELSE wait for key press
01CE 6000FFFE              bra     restart                 ; Restart
01D2 41FAFE46     endgame  lea     hiscore,a0              ;
01D6 203AFE32              move.l  score,d0                ;
01DA B090                  cmp.l   (a0),d0                 ; Compare score/hi-score
01DC 6302                  bls.s   not_hi                  ; If score>hi-score
01DE 2080                  move.l  d0,(a0)                 ; then hi-score=score
01E0 61000100     not_hi   bsr     load_message            ; Move message to screen
01E4 7207         none_p   moveq   #7,d1                   ;
01E6 6100FFFE              bsr     keyrow                  ;
01EA 08010006              btst    #6,d1                   ;
01EE 6610                  bne.s   reset                   ; If n key pressed then reset
01F0 7205                  moveq   #5,d1                   ;
01F2 6100FFF2              bsr     keyrow                  ;
01F6 08010006              btst    #6,d1                   ;
01FA 67E8                  beq.s   none_p                  ; If not y key then wait
01FC 6000FFFE              bra     warm                    ; ELSE start new game
0200 4E40         reset    trap    #0                      ; Set supervisor mode
0202 3E780000              move    $0,a7                   ; Set stack to reset position
0206 48780004              pea     $4                      ; Push reset addr
020A 4E75                  rts                             ; RESET
020C              ;
020C              ; Prints flags at top of screen (1 per 1000 points - collect three
020C              ; for a bonus gun base!!!)
020C              ;
020C              print_flags                              ;
020C 611C                  bsr.s   rub_flag                ; Remove old flags
020E 3C3C00F0              move.w  #240,d6                 ; Set x-pos
0212 7A25                  moveq   #37,d5                  ; Set sprite number
0214 7800                  moveq   #0,d4                   ; Clear d4 ready for
0216 183AFE68              move.b  numflag,d4              ; number of flags
021A 600B                  bra.s   flaged                  ;
021C 6100FE28     nexflag  bsr     plot                    ; plot flag
0220 0646000C              add.w   #12,d6                  ; Move pos for next flag
0224 51CCFFF6     flaged   dbf     d4,nexflag              ; loop
0228 4E75                  rts                             ;
022A              ;
022A              ; Removes old flags
022A              ;
022A              rub_flag                                 ;
022A 7E00                  moveq   #0,d7                   ;
022C 3C3C00F0              move.w  #240,d6                 ;
0230 6100FDF4              bsr     blank_out               ;
0234 06460010              add.w   #16,d6                  ;
0238 6000FDEC              bra     blank_out               ;
023C              ;
023C              ; Prints spare lives (gun bases) at bottom RH corner of screen
023C              ;
023C              print_bases                              ;
023C 4244                  clr.w   d4                      ;
023E 183AFE50              move.b  num_bas,d4              ; No. of bases into d4
0242 5304                  subq.b  #1,d4                   ; -1
0244 6B14                  bmi.s   ex_prb                  ; If none then return
0246 7A08                  moveq   #8,d5                   ; Bases sprite number
0248 7C00                  moveq   #bas_pos,d6             ; x pos. of bases on screen
024A 1E3C00F8              move.b  #248,d7                 ; y pos. of bases on screen
024E 6100FDF6     pr_loop  bsr     plot                    ; plot base
0252 06460010              add.w   #16,d6                  ; move on
0256 51CCFFF6              dbf     d4,pr_loop              ; loop to next base
025A 4E75         ex_prb   rts                             ;
025C              ;
025C              ; Moves the mother ship furtively across the top of the screeen
025C              ;
025C              baddybus                                 ;
025C 41FA0918              lea     zeke,a0                 ; Count down zeke
0260 5310                  subq.b  #1,(a0)                 ;
0262 4A10                  tst.b   (a0)                    ;
0264 6A44                  bpl.s   retbad                  ; If <0 then return
0266 10BC0002              move.b  #2,(a0)                 ; ELSE reset zeke
026A 41FAFF4E              lea     baddyx,a0               ;
026E 3C10                  move.w  (a0),d6                 ; Ship's x-pos into d6
0270 7E00                  moveq   #nemalt,d7              ; Ship's y-pos into d7
0272 610A                  bsr.s   saver                   ; unplots ship
0274 5B46                  subq.w  #5,d6                   ;
0276 0C4601EF              cmp.w   #495,d6                 ; If edge of scr. then
027A 6212                  bhi.s   endbad                  ; kill ship
027C 3086                  move.w  d6,(a0)                 ; Saves new position
027E 7A23         saver    moveq   #35,d5                  ; Replots ship sprite pt.1
0280 6100FDC4              bsr     plot                    ;
0284 06460008              add.w   #8,d6                   ;
0288 7A24                  moveq   #36,d5                  ; Replots ship sprite pt.2
028A 6000FDBA              bra     plot                    ; and return
028E 30BCFFFF     endbad   move.w  #-1,(a0)                ; Kill mother ship
0292 41FAFFFE              lea     badwait,a0              ; Set random time delay
0296 43FAFEBC              lea     rndpos,a1               ;  between appearances
029A 3451                  move.w  (a1),a2                 ;   of the mother ship
029C 1092                  move.b  (a2),(a0)               ;    and prior to
029E 0210003F              and.b   #63,(a0)                ;      subsequent appearances
02A2 00100010              or.b    #16,(a0)                ;
02A6 5210                  addq.b  #1,(a0)                 ;
02A8 5451                  addq.w  #2,(a1)                 ;
02AA 4E75         retbad   rts                             ;
02AC              ;
02AC              ; Establishhes window + borders and places message in centre
02AC              ;
02AC              save_message                             ;
02AC 43FA0596              lea     con_block,a1            ; Define window
02B0 347AFFFE              move.w  ut_con,a2               ;
02B4 4E92                  jsr     (a2)                    ;
02B6 43FA0598              lea     message,a1              ; print message
02BA 347AFFFE              move.w  ut_mtext,a2             ;
02BE 4E92                  jsr     (a2)                    ;
02C0 7002                  moveq   #2,d0                   ; Close window
02C2 4E42                  trap    #2                      ;
02C4 207C0002              move.l  #$2332e,a0              ; Start addr. of window in RAM
02C8 332E
02CA 43FA02A8              lea     message_buff,a1         ; Message buffer pos.
02CE 7013                  moveq   #19,d0                  ; No. of rows in window
02D0 7211         mesave1  moveq   #17,d1                  ; No. of columns in window
02D2 32D8         mesave2  move.w  (a0)+,(a1)+             ; move to buffer
02D4 51C9FFFC              dbf     d1,mesave2              ; Next column
02D8 D0FC005C              add.w   #92,a0                  ; move to next row
02DC 51C8FFF2              dbf     d0,mesave1              ; Next row
02E0 4E75                  rts                             ;
02E2              ;
02E2              ; Moves message from buffer to screen
02E2              ;
02E2              load_message                             ;
02E2 227C0002              move.l  #$2332e,a1              ; start pos. of window in RAM
02E6 332E
02E8 41FA028A              lea     message_buff,a0         ; Message buffer
02EC 7013                  moveq   #19,d0                  ; No. of rows
02EE 7211         meload1  moveq   #17,d1                  ; No. of columns
02F0 32D8         meload2  move.w  (a0)+,(a1)+             ; move to screen
02F2 51C9FFFC              dbf     d1,meload2              ; Next column
02F6 D2FC005C              add.w   #92,a1                  ; move to next row
02FA 51C8FFF2              dbf     d0,meload1              ; Next row
02FE 4E75                  rts                             ;
0300              ;
0300              ; Unplots bomb and drops a new one
0300              ;
0300              endbomb                                  ;
0300 1E3AFDE8              move.b  y_bomb,d7               ;
0304 3C3AFDDE              move.w  x_bomb,d6               ;
0308 7A09                  moveq   #9,d5                   ;
030A 6100FD3A              bsr     plot                    ;
030E 6000FD9C              bra     bombs                   ;
0312              ;
0312              ; Waits for a the space bar (Fire button) to be pressed and
0312              ; then released
0312              ;
0312              waitkey                                  ;
0312 48E7FFFE              movem.l d0-d7/a0-a6,-(a7)       ;
0316 7201         no_spc   moveq   #1,d1                   ;
0318 6100FECC              bsr     keyrow                  ; Read keyrow 1
031C 08010006              btst    #6,d1                   ; If key not pressed
0320 67F4                  beq.s   no_spc                  ; Then Wait again
0322 7201         no_spc2  moveq   #1,d1                   ; Read keyrow 1
0324 6100FEC0              bsr     keyrow                  ;
0328 08010006              btst    #6,d1                   ; If key pressed
032C 66F4                  bne.s   no_spc2                 ; Then wait again
032E 4CDF7FFF              movem.l (a7)+,d0-d7/a0-a6       ;
0332 4E75                  rts                             ;
0334              ;
0334              ; This section was inadvertantly missed off the first issue
0334              ; and should be entered as shown at the very end of the program
0334              ; or immediately following the definition of "rnd_pos".
0334              ;
0334              base_buf
0334 00000240              ds.b    576
0574              message_buff
0574 00000168              ds.w    360
0844              con_block
0844 03020007              dc.w    $0302,$0007,144,20,184,102
0848 00900014
084C 00B80066
0850 0014         message  dc.w    20
0852 2020504C              dc.b    "  PLAY  AGAIN  (Y/N)"
0856 41592020
085A 41474149
085E 4E202028
0862 592F4E29
0866                       align
0866 00000168              ds.w    360
0B36                       align
0B36              zap
0B36 0A08                  dc.b    $0a,8
0B38 0000AAAA              dc.l    $0000aaaa
0B3C 1932                  dc.b    25,50
0B3E 02007017              dc.b    2,0,112,23
0B42 121201                dc.b    18,18,1
0B45                       align
0B46              zam
0B46 0A08                  dc.b    $0a,8
0B48 0000AAAA              dc.l    $0000aaaa
0B4C 1450                  dc.b    20,80
0B4E 2003B80B              dc.b    32,3,184,11
0B52 100001                dc.b    16,0,1
0B55                       align
0B56              crunch
0B56 0A08                  dc.b    $0a,8
0B58 0000AAAA              dc.l    $0000aaaa
0B5C FAFF                  dc.b    250,255
0B5E 0100B80B              dc.b    1,0,184,11
0B62 23FF01                dc.b    $23,$ff,1
0B65                       align
0B66              base_bang
0B66 0A08                  dc.b    $0a,8
0B68 0000AAAA              dc.l    $0000aaaa
0B6C 3237                  dc.b    50,55
0B6E 0A00204E              dc.b    10,0,32,78
0B72 A1FF01                dc.b    $a1,$ff,1
0B75 00           nexburp  dc.b    0
0B76 00           zeke     dc.b    0
0B77 26272728     bar_tab  dc.b    38,39,39,40,39,39,39,39,39,41,41,39
0B7B 27272727
0B7F 27292927
0B84                        end


SYMBOLS :

bar_tab                  R00000B77  nexburp                  R00000B75
zam                      R00000B46  zap                      R00000B36
base_buf                 R00000334  no_spc2                  R00000322
no_spc                   R00000316  endbomb                  R00000300
meload2                  R000002F0  meload1                  R000002EE
mesave2                  R000002D2  mesave1                  R000002D0
message_buff             R00000574  ut_mtext                 U00000000
message                  R00000850  ut_con                   U00000000
con_block                R00000844  save_message             R000002AC
badwait                  U00000000  endbad                   R0000028E
saver                    R0000027E  nemalt                   U00000000
retbad                   R000002AA  zeke                     R00000B76
baddybus                 R0000025C  pr_loop                  R0000024E
bas_pos                  U00000000  ex_prb                   R0000025A
print_bases              R0000023C  nexflag                  R0000021C
flaged                   R00000224  warm                     U00000000
reset                    R00000200  keyrow                   U00000000
none_p                   R000001E4  load_message             R000002E2
not_hi                   R000001E0  restart                  U00000000
waitkey                  R00000312  endgame                  R000001D2
baddyx                   U00000000  savebarriers             U00000000
phut                     R000001AC  grows                    R000001A4
explode                  U00000000  draws_bang               U00000000
okexp                    R0000019C  seq_num                  U00000000
waitexp                  R000001BE  base_bang                R00000B66
crunch                   R00000B56  vline                    U00000000
scrub                    R0000015A  rndpos                   U00000000
bulposx                  U00000000  rev_bul                  U00000000
hit_barrier              R0000014C  joe                      R00000134
base_dead                R00000180  ex_drop                  R00000144
id_col                   U00000000  drop_bomb                R000000FA
y_bomb                   U00000000  x_bomb                   U00000000
itspos                   R000000CE  grogy                    R000000D8
thing                    R000000C0  num_sp                   U00000000
ypos                     U00000000  xpos                     U00000000
gunpos                   U00000000  bombs                    R000000AC
rub_flag                 R0000022A  num_bas                  U00000000
print_flags              R0000020C  numflag                  U00000000
odd                      R000000AA  sub_dig                  R0000006E
mess                     U00000000  inc_score                R00000058
plot                     U00000000  num_base                 U00000000
nexdig                   R0000003A  blank_out                U00000000
next_w                   R00000026  hiscore                  U00000000
score_poshi              U00000000  print_hiscore            R00000016
print_num                R0000001E  score                    U00000000
score_pos                U00000000  print_score              R00000000
```

Sound and...

Though limited to a single sound generator, the QL can still produce some interesting effects. Robert Miles pitches in.

The QL differs from the BBC in that it does not have a special chip to handle the sound, it's produced instead by the 8049 second processor which is also in charge of the keyboard. This means the amount of control available is more restricted (eg, there is no control over the loudness of the note). Even so some very interesting sounds can be produced on the QL.

The command used to produce sound on the QL has the same name as on the Spectrum. However, the duration and pitch parameters are represented in quite a different way on the QL. The duration is specified in steps of 72 microseconds, ie, 13889ths of a second, up to a maximum length of 32767 steps (2.36 seconds). If the duration is given as 0 the note will continue until terminated by another BEEP command which has an end point. The SuperBasic manual mysteriously says of the pitch value that 1 gives a high note and 255 gives a low note. What the guide does not say is that the pitch value changes the more complex you make the BEEP sound statement.

For example, if you set a note playing with a duration of 0, a pitch value of 35 corresponds roughly to middle C, but if you give a different duration the pitch drops so that middle C is now nearer 34. The most drastic effects can be achieved by giving values for the "fuzzy" parameter (see later) which really push the pitch through the floor! Combined with the fact that gaps between the notes are not semitones, this means playing tunes is a rather fraught process. You can however produce some very interesting sound effects by using just a single BEEP command.

Second Pitch

Following the pitch parameter, a second pitch value can be specified. This causes the pitch of the beep to move between two pitches at the specified rate. The rate of change is given in terms of the number of steps between each change in pitch, and the size of each change. For example, here is the command to specify a start pitch of 10, a second pitch of 20, a time interval of 100 and a step size of 2: BEEP 0,10,20,100,2. The note starts playing with pitch 10. After 100 steps the pitch value goes up to 12, after 100 more steps it rises to 14 and so on until it reaches 20. Here the value starts to fall in steps of 2 until it reaches 10 and then the whole thing starts again. Since a time interval of 100 is very small you will hear a rapid trilling sound. If you increase the interval to 2000, ie, change the 100 above to 2000, you will hear each individual change in pitch.

The number of steps between each pitch change is called grad_x and it is given in the same units as the length of the sound. The manual incorrectly says that this can only vary between –32768 and 15, whereas it can be used with values between –32767 and 32768.

The size of each change in pitch is called grad_y and is given in the same units as the pitch (from –8 to 7). Remember that the higher the pitch value, the lower the pitch and a negative value means the pitch gets higher initially.

An interesting thing to note is that the pitch value 'loops round', that is, if you add 1 to 255 (low note) you get a pitch of 0 (high note). This can result in some amazing effects by making the pitch 'change' in the wrong direction.

Wrapping Up

The set of second pitch information can be followed by a value giving the number of 'wraps' the pitch change will undergo. Normally the movement in pitch changes direction each time it reaches its end point. Thus our scale above first climbs to the top and then climbs back down again, giving an up and down effect.

You can stop this happening, and make the sound go back to its initial pitch and start climbing up again, by giving a value for the number of 'wraps' to be performed. For example, in the scale above if we added a 'wrap' value of 1 when the first scale was completed, instead of climbing back down again the scale would start at the low end and climb upwards, performing 1 wrap. The scale would then go up to the top, and since the required number of wraps has been performed, go back down again. Once it has reached the bottom it would then start at the top again, performing the same wrap only upside down! This is considerably more difficult to explain than it is to understand.

The number of wraps can be given from 0 to 15, with 15 meaning *wrap all the time*. You can use the wrap effect along with frequent changes in pitch to produce some very complex effects.

The wrap parameter can be followed by a 'fuzz' value which dramatically affects the sound produced. It very rapidly changes the pitch of the note as it is played. It can also divide the speed at which things happen by a factor of about 10, so that if you add fuzz to a note it will become deeper and any pitch changes will take place more slowly.

Random Value

The final parameter is the 'random' one. This can be used to produce mind-boggling sounds. When a 'random' value is given, a random offset is added to the pitch value. The amount of randomness added depends on the value of this parameter, it seems to start having an effect at around 5 and by the time you reach the maximum value of 15 it has made the original sound completely unrecognisable!

Here are a few intgeresting sounds to try out. See if you agree with the names!

Chickens:
BEEP 0,10,10,170,–8,0,0,8
Sub-space radio:
BEEP 0,255,1,200,–1,5,0,9
Racing car:
BEEP 0,100,250,1000,–1,2,10,8
Police siren:
BEEP 0,30,37,9000,7,0,0,0
Tuneful:
BEEP 0,1,1,4500,0,5,0,9

Finally, for further experimentation – readers should refer back to the sound program in our May issue (page 37).

Vision

A very famous problem in mathematics is the following: How many colours do you need to colour a map so that no two countries sharing a border are shaded with the same colour? (countries meeting at a point are not regarded as sharing a border). This problem has been around since the mid-nineteenth century. The answer has long been suspected to be four, but it was only in 1976 that proof was given for this, and a rather complicated one it turned out to be, requiring the help of a computer for the exhaustive analysis of all possible cases (see Appel and Haken – Scientific American Oct. 1977 if you are interested in complicated mathematical proofs).

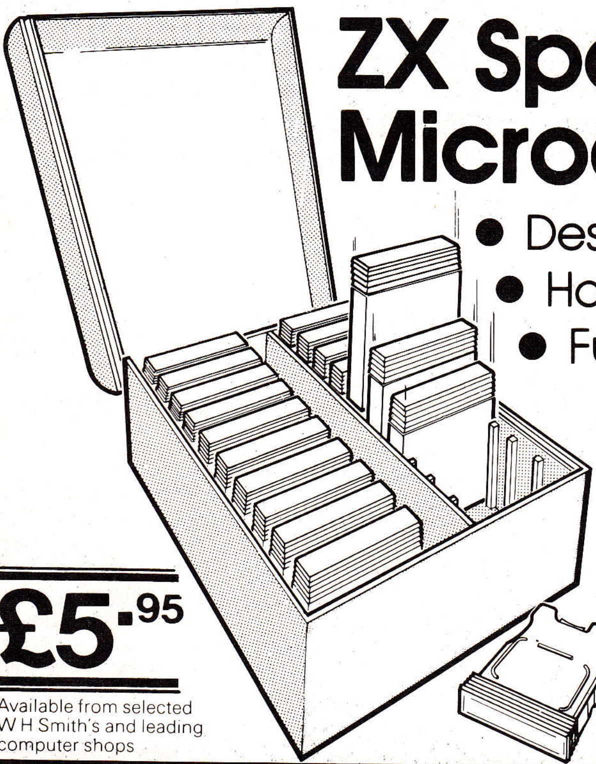
This article is about a special case of the 4-colour map problem.

```
1 REMark * QL User 1985 **
4 REMark *** MAPCOL ******
5 REMark ** P.J.Derlien **
6 REMark *****************
20 SCALE 100,0,0:INK 7:PAPER 0
30 OPEN #3,scr_310x220a44x6
40 MODE 4:INPUT "How many countries on the map? ";
maxC
50 DIM addr(maxC),size(maxC),newpt(2,1),vert(1000)
100 REPeat main
110   initialize
120   FOR times=1 TO maxC-1
130    split largest
140   NEXT times
150   x=50:y=50
160   whichregion
170   AT #0,3,0:PRINT #0,"press SPACE for another
map, ESC to finish":k$=INKEY$(-1)
180   IF CODE(k$)=27 THEN EXIT main
200 END REPeat main
499 :
500 DEFine PROCedure initialize
505 RANDOMISE
510 MODE 4
```

```
515 colour4=106
517 BORDER #3,10,colour4
520 colours
530 RESTORE 660
540 addr(1)=0
550 READ vert(0)
560 k=2*vert(0)
570 FOR j=1 TO k
580   READ vert(j)
590 NEXT j
600 nextfree=k+1
605 size(1)=400:totalsize=400
610 Nregions=1
620 polygon 1,vert(0)
650 END DEFine initialize
660 DATA 4, 12,0, 112,0, 112,100, 12,100
699 :
700 DEFine FuNction largest
710 LOCal max,n
720 max=size(1):n=1
730 FOR j=1 TO Nregions
740   IF size(j)>max THEN n=j
750 NEXT j
760 RETurn n
770 END DEFine largest
999 :
1000 DEFine PROCedure markerspot (x,y,k)
1010 LOCal n
1020 REPeat blob
1030   k=CODE(INKEY$)
1040   n=KEYROW(1)
1050   IF k>47 AND k<53 THEN EXIT blob
1060   INK 0:POINT x,y: INK 7
1070   x=x+((n&&16)=16)*2-((n&&2)=2)*2
1080   y=y+((n&&4)=4)*2-((n&&128)=128)*2
1090   POINT x,y
1100 END REPeat blob
1110 END DEFine markerspot
1199 :
1200 DEFine PROCedure split (region)
1210 start=addr(region)
1220 nv=vert(start)
1230 newedge nv,V1choice,v1plus
1235 cutline=ABS(newpt(2,0)-newpt(1,0))+ABS(newpt(
2,1)-newpt(1,1))
1240 oldv=(V1choice+1) MOD nv
1245 source=start+1+oldv*2
1340 store Nregions+1,1,v1plus,2
1470 nextfree=p+2
1475 store region,2,nv-v1plus,1
1640 Nregions=Nregions+1
1650 nextfree=p+2
1660 END DEFine split
1699 :
1700 DEFine PROCedure newedge (nv,V1choice,v1plus)
1720 REPeat tester
1730   V1choice=RND(nv-1)
1740   breakline V1choice,1
1745   IF NOT(pass) THEN NEXT tester
1750   v1plus=2
1770   v2choice=(V1choice + v1plus) MOD nv
1780   breakline v2choice,2
1783   IF pass THEN EXIT tester
1785 END REPeat tester
1790 LINE newpt(1,0),newpt(1,1) TO newpt(2,0),newp
t(2,1)
1800 END DEFine newedge
1999 :
2000 DEFine PROCedure breakline(v,n)
2010 LOCal a,px,py,qx,qy,frac
2015 pass=0
2020 a=start+1+v*2
2030 px=vert(a):py=vert(a+1)
2040 a=start+1+(v+1)MOD nv*2
2050 qx=vert(a):qy=vert(a+1)
2060 frac=.25+RND*.5
2080 newpt(n,0)=(1-frac)*px+frac*qx
2090 newpt(n,1)=(1-frac)*py+frac*qy
2095 IF ABS(px-qx)+ABS(py-qy)>10 THEN pass=1
2100 END DEFine breakline
2999 :
3000 DEFine PROCedure store (newR,a,n,b)
3010 p=nextfree
3020 addr(newR)=p
3030 vert(p)=n+2
3040 p=p+1
3050 vert(p)=newpt(a,0)
3060 vert(p+1)=newpt(a,1)
3070 size(newR)=cutline
3080 p=p+2
3090 previousx=newpt(a,0):previousy=newpt(a,1)
3100 FOR j=1 TO n
3110   vert(p)=vert(source)
3120   vert(p+1)=vert(source+1)
3130   size(newR)=size(newR)+ABS(vert(p)-previousx
)+ABS(vert(p+1)-previousy)
3140   previousx=vert(p):previousy=vert(p+1)
3150   p=p+2
3160   oldv=(oldv+1) MOD nv
3170   source=start+1+oldv*2
3180 NEXT j
3190 vert(p)=newpt(b,0)
3200 vert(p+1)=newpt(b,1)
3210 size(newR)=size(newR)+ABS(newpt(b,0)-previous
x)+ABS(newpt(b,1)-previousy)
3220 totalsize=totalsize-size(region)+size(newR)
3230 END DEFine store
11999 :
12000 DEFine PROCedure whichregion
12010 PRINT #0,\"Move spot with cursor keys"\"Pres
s 1,2,3, or 4 to colour, 0 to escape"
12020 REPeat ask
12040   markerspot x,y,hue
12050   IF hue=48 THEN EXIT ask
12060   AT #0,3,0:CLS #0,3:PRINT #0,"region ";
12070   FOR area=1 TO Nregions
12080     inside=seeifin(area,x,y)
12090     IF inside THEN PRINT #0,area:shade area,
hue-48:EXIT area
12100   END FOR area
12110 END REPeat ask
12120 END DEFine whichregion
12199 :
12200 DEFine FuNction leftfromedge(v,x,y)
12210 LOCal i,j
12220 i=spos+v*2
12230 j=spos+ (v+1) MOD npts *2
12240 RETurn ((vert(j)-vert(i))*(y-vert(i+1))-(ver
t(j+1)-vert(i+1))*(x-vert(i))>0)
12250 END DEFine leftfromedge
12499 :
12500 DEFine FuNction seeifin(r,x,y)
12510 LOCal i,k,ans
12520 npts=vert(addr(r))
12530 spos=addr(r)+1
12540 FOR i=0 TO npts-1
12550   IF NOT(leftfromedge(i,x,y)) THEN ans=0:EXI
T i
12560 NEXT i
12570 ans=1
12580 END FOR i
12590 RETurn ans
12600 END DEFine seeifin
13999 :
14000 DEFine PROCedure shade (region,tint)
14010 LOCal spos,n,j,p
14020 spos=addr(region)+1
14030 n=vert(spos-1)
14040 INK paint(tint):FILL 1
14050 polygon spos,n
14060 FILL 0
14070 INK 0:polygon spos,n:INK 7
14130 END DEFine shade
14999 :
15000 DEFine PROCedure colours
15010 DIM paint(4)
15020 paint(1)=7:paint(2)=5:paint(3)=2:paint(4)=co
lour4
15030 FOR c=1 TO 4
15040   BLOCK 20,20,400,c*40-5,paint(c)
15050   AT c*4,64:PRINT c
15060 NEXT c
15070 END DEFine colours
15999 :
16000 DEFine PROCedure polygon(spos,n)
16010 LOCal j,p
16050 LINE vert(spos),vert(spos+1)
16060 FOR j=1 TO n
16070   p=spos+ j MOD n * 2
16080   LINE TO vert(p),vert(p+1)
16090 NEXT j
16190 END DEFine polygon
```

TRANSFORM LTD. Computer Accessories

ZX Spectrum and QL Microdrive Storage Box



- Designed to match QL and Spectrum
- Holds 20 cartridges and index cards
- Fully interlocking

£5.95

Available from selected W H Smith's and leading computer shops

Spectrum and QL

QL Centronics Interface	£29.95
QL Dust Cover	£5.00
QL RS 232 Lead	£10.00
QL Monitor	£299.00
Spectrum monochrome monitor connector	£11.50
Spectrum '+' Dust Cover	£4.00
Microdrive Cartridges	£1.99
Spectrum R.G.B. Connector	£45.00
Spectrum repairs	**£18.95**

•We also supply printers, monitors, labels, ribbon and listing paper.

Access TRANSFORM LTD. (Dept. QL) 089 283 4783 VISA
'Swatlands', Lucks Lane, Paddock Wood, Kent TN12 6QL.

AN IMPORTANT ANNOUNCEMENT TO ALL QL OWNERS

bad or changed medium

Your data is worth a lot to you. Source programs, text, scientific information, records, all represent hours of your time and effort. Losing a file could at worst cost you money — and it's **always** infuriating!

No storage medium is 100% reliable. That's why TALENT has developed the CARTRIDGE DOCTOR. It's a sophisticated machine code program which will, in most cases, enable you to:

- recover files from a bad medium
- recover files which have been accidentally deleted
- recover files with lost or damaged blocks using the 'block patch' utility.

It's very easy to operate and no knowledge of BASIC or machine code is required.

Can YOU afford to be without the CARTRIDGE DOCTOR?

Available only from:

£21.95
+ 50p postage & packing

QL is a registered trademark of Sinclair Research

TALENT COMPUTER SYSTEMS
Curran Building,
101 St. James Road, Glasgow G4 0NS
Tel: 041-552 2128 (ACCESS & VISA accepted)
SOFTWARE FROM SCOTLAND

SUBSCRIPTION OFFER

£1.60 OFF

one year's subscription

FOR A LIMITED PERIOD ONLY

For the next three months only, we are offering a full year's subscription to QL User for just £11.40 – that's £1.60 off the usual price of £13.

Just complete and return the form below and we will deliver QL User direct to your home for the same as it costs to buy in the shops!

BUT THAT'S NOT ALL BECAUSE . . . as a subscriber to QL User you'll also enjoy the benefits of two other great deals:

- 10% discount on the Microdrive Exchange
- Access to the QL User Technical helpline

The **Microdrive Exchange** is a special readers' service exclusive to QL User and in the back of this magazine.

Just send us a blank microdrive and for a small administration charge (£1-£5 including the author's royalty) we'll copy onto it any listings from the magazine.

As a subscriber to QL User you can deduct 10% from the total price on all your orders just by quoting your subscription number — it's an easy way to save money.

The QL User **Technical Helpline** is available outside working hours (before 9.30am and after 5.30pm, or at weekends).

Subscribers can leave a short message detailing any problem concerning the QL, its software or hardware (one question per call).

Within 48 hours an answer will be winging its way to you by first class post.

To take advantage of this special offer just fill in the form below and return it no later than 30th September 1985:

Return completed forms to: Carl Dunne, Magazine Services (Dept QL), Priory Court, 30-32 Farringdon Lane, London EC1R 3AU

CUT HERE ✂ - - - - - - - - - - - - - - - - -

I would like to subscribe to QL User at the specially reduced rate of £11.40 (12 issues).

☐ I enclose a cheque/PO for £11.40 made payable to QL User

☐ Please debit £11.40 from my Access/Visa* card account

Account Number____________ Card Expiry Date________

Signature____________________

*(*delete as applicable)*

NAME

ADDRESS

NOTE. THIS OFFER IS ONLY OPEN TO RESIDENTS OF THE UK.

Readers' Survey

+PRIZE DRAW

Here it is – your chance to influence the way we put *QL User* together and (perhaps!) the QL market as well.

Just complete the questionnaire below, tear out the page and fold it as shown – we've paid the postage. And as an extra incentive we'll be giving away 10 QL books to 10 lucky readers in a free prize draw to be held on 31st July 1985 – so send your completed surveys now!

	YES	NO
1. Do you own a QL	☐	☐

If so, what is its serial number ....

2. Which of these computers (if any) do you own

☐ Amstrad	☐ Mackintosh
☐ Apricot	☐ Memotech
☐ Atari	☐ Oric Atmos
☐ BBC B	☐ Sharp
☐ CBM 64	☐ Sinclair ZX81
☐ Dragon	☐ Sinclair Spectrum
☐ Electron	☐ Spectravideo
☐ IBM PC	☐ Tandy
☐ Lynx	☐ Texas

3. Which, if any, peripherals do you have

☐ Monitor (not TV)	☐ Printer
☐ Disk drive	☐ Parallel convertor
☐ Joystick	☐ Expansion card

4. About how many hours per week do you spend using your computer ....

How much of that time (approximately) is spent on each of the following

	ALL	75%	HALF	25%	NONE
Leisure	☐	☐	☐	☐	☐
Writing programs	☐	☐	☐	☐	☐
Learning/education	☐	☐	☐	☐	☐
Business	☐	☐	☐	☐	☐
Working at home	☐	☐	☐	☐	☐

5. What are the two most likely (in order) peripherals you will buy in the next 12 months (select from the list in question 3)

1st ....

2nd ....

6. How many microdrives with programs on (eg, Toolkit) have you bought in the last 6 months

>10	6-9	3-5	1-2	0
☐	☐	☐	☐	☐

7. Which issues, if any, of *QL User* did you buy in the last 6 months

DEC/JAN	FEB	MAR	APR	MAY	JUN
☐	☐	☐	☐	☐	☐

8. How many other people, if any, regularly read your copy of *QL User* ....

9. How often do you buy any of the following magazines

	EVERY MTH/WK	OFTEN	SOME-TIMES	NEVER
Computer & Video Games	☐	☐	☐	☐
Electronics & Computing	☐	☐	☐	☐
What Micro	☐	☐	☐	☐
Your Computer	☐	☐	☐	☐
Popular Computing Weekly	☐	☐	☐	☐
Sinclair User	☐	☐	☐	☐
Personal Computer World	☐	☐	☐	☐

Personal Details: If you wish to be included in the prize draw please complete this section.

Sex	MALE ☐	**Age**	under 13 ☐	14-16 ☐	17-18 ☐	19-21 ☐	22-24 ☐
	FEMALE ☐		25-30 ☐	31-35 ☐	36-44 ☐	45-55 ☐	over 55 ☐

Occupation ....

	<£5000	£5-8000	£8-10,000	£10K-£15K	£15K-£20K	>£20K
Salary	☐	☐	☐	☐	☐	☐

Name ....

Address ....

....

....

Readers' Survey + PRIZE DRAW

FIRST FOLD

THIRD FOLD (TUCK IN)

Postage
will be
paid by
licensee

Do not affix postage stamps if posted in
Great Britain, Channel Islands, Northern Ireland
or the Isle of Man

2

BUSINESS REPLY SERVICE
LICENCE NO. 79/4

QL USER
PRIORY COURT
30/32 FARRINGDON LANE
LONDON EC1R 3AU

SECOND FOLD

Write programs to unlock the multitasking power of your QL!
Choose the language you need from Metacomco's unrivalled range of high-quality languages for the QL: The best selling ***ASSEMBLER****; The language of the future,* ***LISP****; and two professional compilers,* ***BCPL*** *and* ***PASCAL****.*

RELEASE THE POWER OF THE QL

QL PASCAL £89.95 INC. VAT

Newly available, a true Pascal compiler for the QL! Developed by Metacomco: leaders in Pascal - the first software house to achieve full ISO validation of a Pascal compiler for the 68000. QL Pascal is a high specification implementation of this popular language, endorsed by Sinclair Research for use on the QL.

> True compiler producing native 68000 code: compact and efficient.
> Fast, single pass compilation without intermediate stages.
> Complete implementation of ISO 7185 - the international standard for Pascal - making it ideal for commercial and educational use.
> Direct addressing of the full QL address space.
> Key routines provided on ROM for extra speed and convenience.
> No run time licences required for commercial applications written using QL Pascal.
> Any length variable names and full 32-bit integers.
> Easy to use interfaces to QL windows, graphics, traps, screen handling, file operations, etc.
> Will handle very large sets and arrays.
> Approved by Sinclair Research for use on the QL.

NEW

ASSEMBLER £39.95 INC. VAT

A high specification macro assembler supporting the full Motorola instruction set.

Features: external references; absolute, position independent, and relocatable code; linker; precise error messages; formatted listings; macro expansions; conditional assembly; and a large range of directives.

BCPL £59.95 INC. VAT

A true compiler, ideal for systems programming - writing utilities, games and applications.

Features: generates native 68000 code; run time library includes easy QDOS interfaces; link loader links separately compiled segments; modules can be linked with Pascal or assembler.

LISP £59.95 INC. VAT

A LISP interpreter for exploring "The language of artificial intelligence".

Features: Turtle graphics; compatible with LISP for the BBC micro; full support of QL features; structure editor; prettyprinter; garbage collector and tracer.

Every DEVELOPMENT KIT includes Metacomco's popular screen editor, and a detailed manual.

The Metacomco Development Kits are available from W.H. Smith, John Lewis, HMV, Menzies, Boots and other leading retailers, or direct from Metacomco.

METACOMCO

26 Portland Square, Bristol BS2 8RZ.
Tel: Bristol (0272) 428781

QL and QDOS are the trade names of Sinclair Research Ltd.

NAME ____________
ADDRESS ____________
____
POSTCODE ____________
TEL. NO. ____________
SIGNATURE ____________

Phone today or post this coupon to: Metacomco, 26 Portland Square, Bristol BS2 8RZ. Please send me:

Assembler Development Kit	£39.95	___
LISP Development Kit	£59.95	___
BCPL Development Kit	£59.95	___
QL PASCAL Development Kit	£89.95	___

I enclose a cheque for £________ or debit my ACCESS/VISA Account No.

CARD EXPIRY DATE: ____________

Price includes VAT, postage and packing UK mainland only. Delivery, allow up to 28 days. QL7 silicon express

INSIDER DISK INTERFA

☆ Full microdrive emulation facility.

☆ Incorporates more Toolkit commands than any other system.

☆ Insider disk interface only £99 + VAT.

☆ Compatible with any two 3″, 3½″ or 5¼″ independently powered disk drives.

☆ Supports 40 or 80 track, single or double sided drives offering up to 1.4 megabytes formatted storage capacity.

☆ Fits snugly inside the case of the QL, connecting to the main expansion port.

☆ Advanced design incorporates custom chip technology.

☆ Fast access times eg. boots 'Quill' in under 10 seconds.

☆ Software designed by the author of QDOS to ensure full compatibility with all Sinclair hardware and software revisions.

☆ Full two year guarantee.

☆ Supplied with comprehensive user manual.

☆ Complete disk systems available from £217 + VAT.

For immediate despatch of goods you can order now by phoning us on Leicester (0533) 374917 quoting your Access card number.

Or fill in and post the coupon – no stamp required – and we will despatch your goods within 14 days.

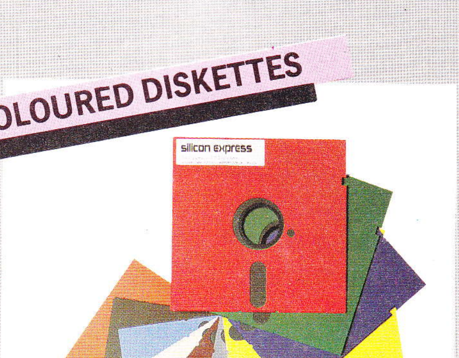
Sinclair, QL and QDOS are registered trade marks of Sinclair Research Ltd.

☆ **3½″ Disk Drives now available.**

☆ **All cards contain complete ROM-based toolkit utility.**

QL DISK DRIVE

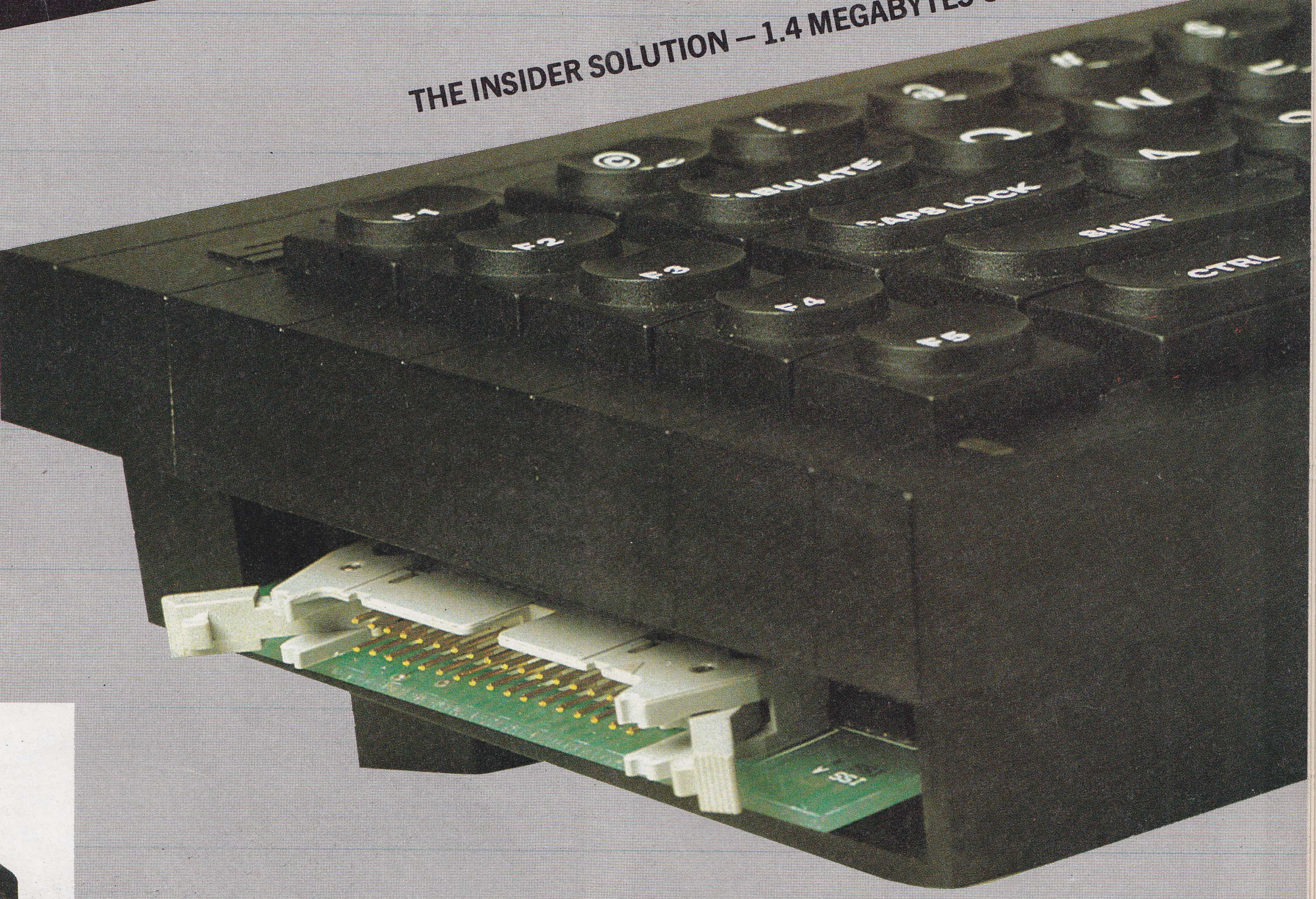
COLOURED DISKETTES



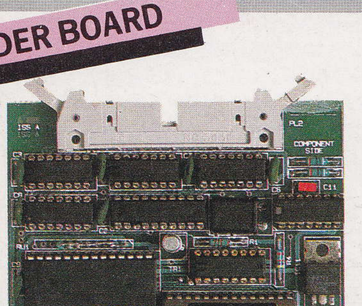
IN

ACE CARD....

THE INSIDER SOLUTION — 1.4 MEGABYTES UNDER COVER



INSIDER BOARD



To order please phone Leicester (0533) 374917 or complete and post this coupon.

ORDER FORM No stamp required if posted in the UK

Post to: Silicon Express Ltd., FREEPOST, Rothley, Leicester, LE7 7QZ.

Please allow 14 days for delivery of postal orders; telephone orders, immediate despatch.

______(Qty) Insider Board at £113.85 ....£

______(Qty) Insider Board with single, 80 track, double sided 5¼" drive at £249.00 ....£

______(Qty) Insider Board with dual, 80 track, double sided 5¼" drive at £399.00 ....£

______(Qty) Insider Board with single, 80 track, double sided 3½" drive (incl PSU) at £296.70 ....£

______(Qty) Second 80 track double sized 3½" drive at £159.85 ....£

______(Qty) Box of 10 DSDD 96TPI coloured diskettes at £35.00 ....£

______Further details on the INSIDER BOARD

I enclose cheque/postal order for £ made payable to Silicon Express Ltd.

Name ______________________

Address ______________________

Prices include VAT, post and packing. QU5

SAGESOFT ACCOUNTS

Paolo Baccanello examines an integrated accounts package and decides it's another winner in the profesional software stakes.

professional software stakes.

Business software has been a long time coming on the QL. With Cash Trader (reviewed May) and now Sagesoft's accounts the waiting would appear to be over. The former, easy to use though with limited credit balance facilities caters to the small business and sole trader. The latter, with its integrated ledger system and extensive reporting facilities is quite capable of supporting large scale company accounts.

Packaged in the usual black plastic Sinclair livery, Sagesoft's product consists of an extensive manual and some 220K of code (written in C) spread across three microdrive cartridges and broken down into four programs dealing with REPorts, POSTings, UTILities and INSTALLation procedures. Programs are run by typing the command 'CRUN' followed by the program name. Explicit instructions are given in the manual as to which program to run and when, however, a few accompanying on-screen prompts would not have gone amiss.

Software set up

Getting Sagesoft's system up and running requires a good deal of preparation and takes the better part of a day. The first task is to decide upon the best possible storage medium bearing in mind the needs of your business and the following allocations (in bytes) made by the programs:

Reserved to run the programs	2366
+	
Each sales ledger a/c	133
Each purchase ledger a/c	133
Each nominal ledger a/c	33
+	
Each transaction	76

From these figures you don't have to be a genius to work out that a business requiring 100 Sales, 75 purchase and 60 nominal accounts leaves room for around 1000 transactions per microdrive cartridge, 10,000 per single sided (360K) floppy and 20,000 for the double-sided (720K) variety.

However, before rushing out to buy disk drives it should be noted that the programs keep close watch on remaining storage and prompt the user to run a special reconfiguration program when 90% capacity has been reached. The routine first requires the user to print period analyses and then carries forward to blank a cartridge or floppy all 'live' transactions (ie. those relating to balances outstanding) along with the aged debtors, audit trial and extended trial balance.

Reconfiguring (on a monthly basis) then, not only provides a convenient way to organise printed records but means that relatively large accounting systems may be efficiently maintained on a QL with only limited tape or disk capacity. In our example, provided the monthly volume of transactions falls below 1000 it would be possible to spread a year's accounts over 12 microdrives thereby obviating the expense of a disk drive.

Once device designators have been set (ie mdv, flp, fdk etc) and the data disk or cartridge configured the user moves on to allocate six digit numeric account codes to the Debtor, Creditor & Tax Control accounts as well as Bank, Cash and Discount accounts. All other codes (Sales-*alphanumeric*, Purchase-*alphanumeric* or Nominal-*numeric*) are entered through the POSTings module. Linking account creation and transaction entry so closely together results in considerable savings in time and effort as the operator needs only switch menus to move from one operation to the other.

The final stage in setting up the system involves installing a printer driver and defining the various tax rates to be levied on the business's

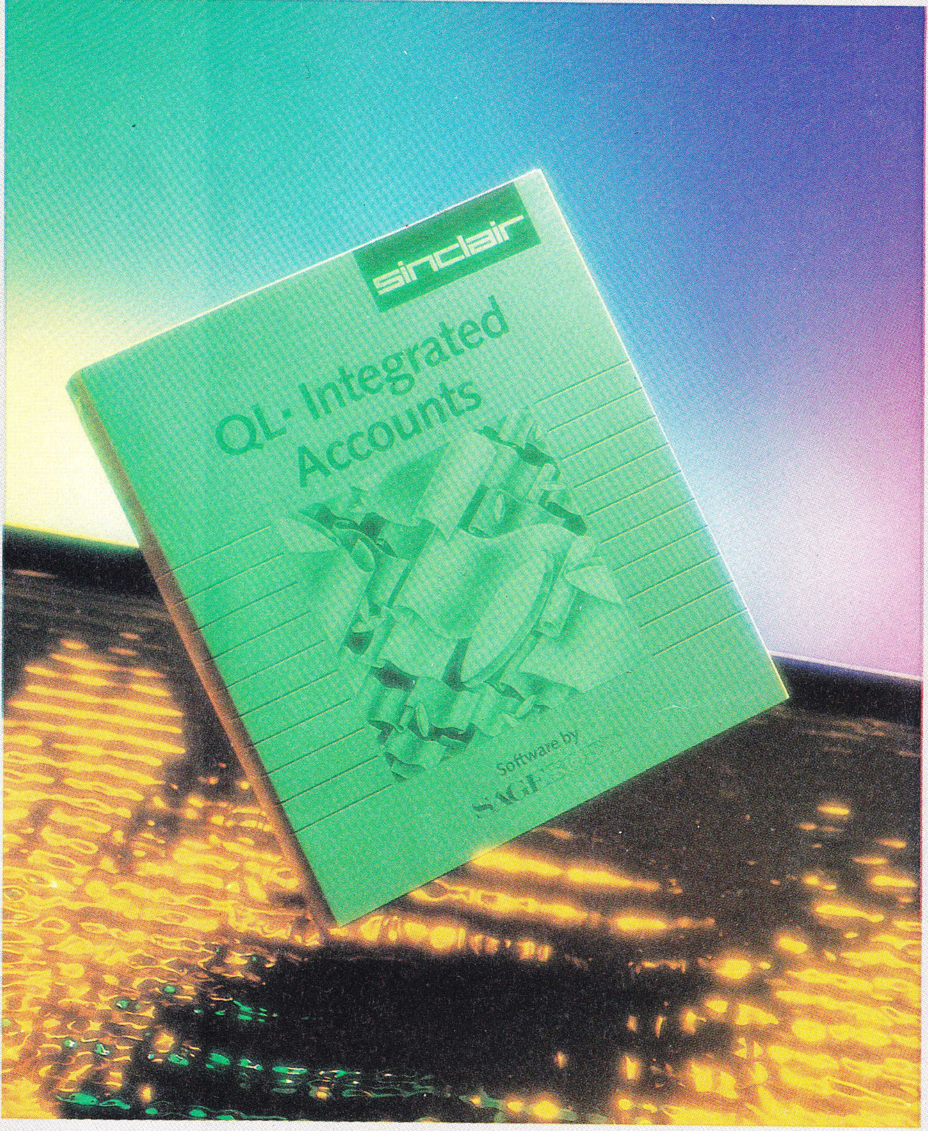


PHOTO BY TERRY BEDDIS

transactions. Up to 10 rates may be defined and are referenced by a two digit alphanumeric code (ie, T1 = 15%). Transactions must be entered nett during posting and then followed by an appropriate tax code. The program will then automatically gross up the transaction and debit or credit the tax amount to a pre-defined tax account.

Record keeping

All data entry takes place in the POSTings module. Here each ledger appears a separate option on the control menu. Further sub-menus divide data entry into a number of clearly recognisable operations. For example, having decided to work on the Sales or Purchase Ledgers the user may choose between creating a new account or entering invoices, credit notes or receipts. Interestingly, Cash and Petty Cash entries are classified as nominal ledger operations alongside account creation and journal entries.

The actual process of entering the data is standard throughout the module. Numbers, accounts codes or Text are entered one at a time in a small box at the foot of the screen. Each entry is then validated. For example, if a nominal account code is expected then the program checks to see whether that code in fact exists. If it doesn't exist the entry is disallowed, otherwise it is displayed at a position highlighted by the cursor on the main screen. Additionally, the operator may use the cursor keys to highlight previous entries and so amend them. Pressing the ESC at any time will terminate entry and gives the operator the choice of whether to POST all transactions displayed or abort. Altogether then, the process is simple, easily remembered and virtually idiot proof.

When it comes to dealing with receipts and payments the package allows for both manual or automatic matching against customer Sales or Purchase invoices. The former method provides an easy way to record part or discounted payments. The latter simply matches off any balance against transactions in the order in which they appear on the clients account.

Finally, as regards journal entries, batch postings may be easily entered though, obviously, the operator will not be allowed to leave the option until both sides of the double entry are complete.

Reports generation

The REPorts module is where Sagesoft's package comes into its own. The various reports that may be produced are listed below:

Sales or Purchase Ledgers
List of account names
Summaries and aged breakdowns of accounts
Account history
Statements or remittance* advices

Nominal Ledger
List of account names
Trial balance
Account history
Control account breakdowns

Management Reports
Day books
List of journal entries
Tax return*
Audit trial
* Printer only

The reports are fairly standard to most businesses and need no explanation. However, a few points should be noted. First, with respect to statements and remittance advices, there are no facilities to alter the print layouts. User's will either have to obtain the pre-printed stationary from Sagesoft themselves or else have their own made up to suit the package. Bearing in mind the package's low cost and the fact that defining one's layout is often complex and time consuming this is only a minor niggle.

Secondly, on the plus side, the tax return not only summarises total tax and nett for inputs and outputs for each of the ten tax codes but also goes to analyse each transaction type.

Finally, the audit trial report which lists every single posting made to the accounts (and forms the basis of all other reports) is organised in such a fashion that invoice or payment details appear alongside the transaction to which they relate. This makes it particularly easy for an auditor to trace back the settlement of outstanding balances and identify mispostings.

On a more general level, most reports may be displayed to the screen as well as printed out. Also, wherever appropriate, Sagesoft have permitted a range of accounts to be specified. As regards the Sales and Purchases this means that the operator can use the REPorts module to deal with customer enquiries. The ability to use an abbreviation of customer's name for his account code further enhances this 'psuedo' facility.

Finally, to produce a Balance Sheet or Trading and Profit and Loss account the user must return to the UTILities module first used to set up the system. Here he is required to identify groups of nominal ledger accounts either as assets or liabilities to appear on the balance sheet or as an items of income or expenditure on the profit and loss account. The exact classifications are:

Balance Sheet	Profit & Loss A/C
Fixed Assets	Sales
Current Assets	Purchases
Liabilities	Direct Expenses
Financed by:	Overheads (1)
	Overheads (2)
	Overheads (3)

At this juncture the importance of having carefully planned your nominal ledger codes from the outset becomes all too apparent in that no more than five ranges of accounts codes may be assigned to any one classification. However, provided that codes are logically organised then this device will permit complex and comprehensive financial reports to pieced together and even amended with comparative ease.

Finally, we should add that though adequately explained in the manual, the job of selecting account codes and organising accounts into their various classifications should not be taken on lightly by users with limited accounting knowledge. Distinctions tend to blur and misclassifications may produce wholly unrepresentative financial statements. Here a few hours in the company of an accountant whilst setting up the system may well save many hours of corrective postings at a later date.

As a whole, Sagesoft's accounts come across as a professional and well finished product. A simple and logically organised system of menus common to all the programs makes it easy to move from one operation to the next. Additionally, screen layouts throughout are attractive and easily legible, making full use of the QL's windowing capabilities. The manual supplied is well written and replete with useful examples. Here we were pleased to see that not only had much lip service been paid to the importance of making backup copies on a regular basis but also a workable system for doing so had been expounded. If the software is to be used to keep reliable records on any magnetic medium, let alone microdrive cartridges, such a system is a must. Also, for those with queries not covered by the manual Sagesoft have opened up a 'hot line' service. This will be available free of charge for 90 days after purchase and thereafter for a year at a cost of £40. The fee will also entitle users to free updates of the software as well.

Sagesoft's accounting software is something of a breakthrough on the QL. The ease with which accounts may be maintained and reports extracted belies its sophistication. Priced at £89.50 it represents exceedingly good value, especially against other dearer packages.

Available from leading computer stores and Sinclair direct.

ARTICULATING WITH ARCHIVE

THE THEORY OF RELATIVITY

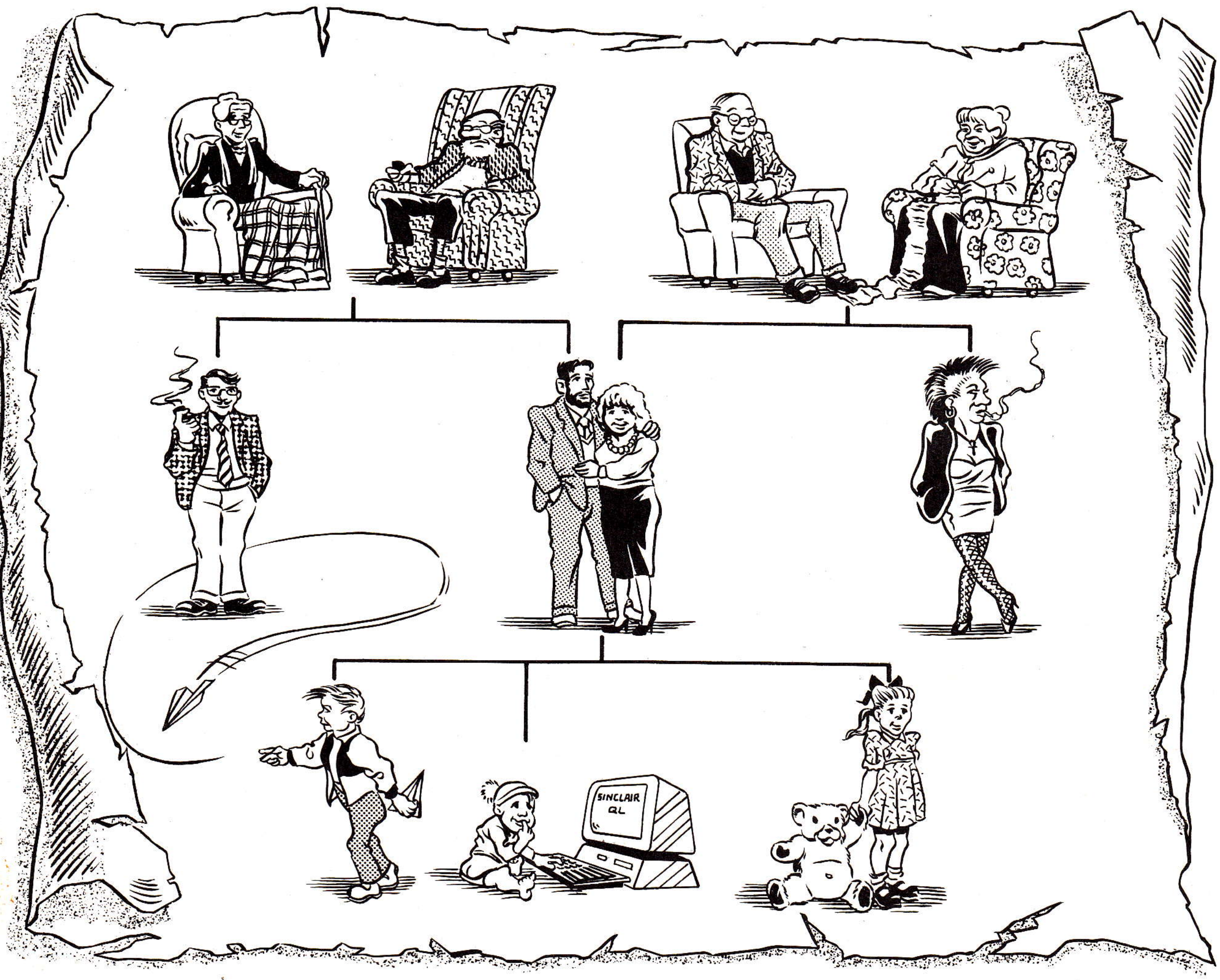


ILLUSTRATION BY DAVID HINE

A new series of applications for Psion's Archive database – Andy Carmichael kicks off with the construction of a family tree.

If you've had your QL and the Psion software since before the days of the version 2.0 upgrade, you will no doubt have discovered that the Archive package is a most useful tool, suitable for simple database applications like an address list, club membership records, keeping track of a collection of stamps or photographs and many other such tasks. The Archive commands *insert, alter, order, search, find* and so on, may well be enough for creating and using the database without the need to set up your own commands with procedures, except perhaps for printing reports, labels or whatever. On the other hand if you tried to use procedures in earnest with version 1.0, or attempted to set up medium to large data files (over 500 records) you would probably have hit snags similar to those that dogged the development of this 'family tree' database. Happily version 2.0 changed all that and if you have the upgraded software, now is the time to get down to some more ambitious applications! Some problems remain of course and we would advise readers to refer to the section entitled "wise words" before constructing their own database along the lines suggested in this article. Basically, make sure you back everything up.

Many of the more sophisticated databases currently available are described as "relational". This tag refers to a database in which data is arranged in "relations" (or tables) which are then manipulated algebraically using complex formulae and esoteric symbols. The overall effect is to simplify the structure of the data and so make that information more manageable.

Whilst Archive is not a "relational" database in the strict definition of the word, it does make use of some relational principles. Notably it is all based on tables of data. In practise this has meant that unlike simple 'cardbox' databases, Archive is capable of manipulating multiple files of information and permits the user to build up linked list records. Our "family tree" database illustrates this capability. Additionally it provides a useful tool for those interested in tracing their lineage.

RISKY BUSINESS

There are two main points of interest in a family tree: the individual people in it, and the relationships between them. The purpose of this package is to display details about an individual, showing his immediate family as a small tree, and then to move around the database from person to person constructing further trees on the screen according to the various relationships between them. Thus to go to a person's father you would first press ↑ (up arrow), to find their wife or husband you would press =, to find the eldest brother or sister you would press ← (left arrow), and so on.

Furthermore an important consideration has been to minimise the risk of ending up with a totally inconsistent database which if you looked at one record would tell you that so-and-so was someone's father but if you looked at another it would tell you he was his brother. This has meant that the structure of the data files has had to be well thought out, with a minimum of redundant or duplicated data, and has required a way of entering the data so as to maintain the consistency throughout.

The solution given below uses four data files to store the information: *People_dbf*, *Families_dbf*, *Marriage_dbf* and *Parents_dbf*. The *People* file just stores information about individuals – not relationships. Thus there are fields to contain a reference number, the name, the dates, the sex and general comments about the person.

The *Families* file contains the relationship information. Each record in this file represents one family – a couple and their children. Thus there are fields in this file to contain the reference numbers of the couple, their children and the dates of the marriage (if any).

The remaining two files are strictly redundant in that they only contain data which could be found from the *Families* file. However, they are required since without them a complete search of the *Families* file would be needed every time the program wanted to find who was the parent or spouse of a particular person. By having these extra two files we can use the much faster command 'locate' which will go straight to the appropriate record on the microdrive cartridge or in memory, thereby saving a great deal of time. Thus the *Marriage* file will contain, for each reference number, a family in which the person referred to is either father or mother. The *Parents* file will contain for each reference number, the family in which the person referred to is one of the children. The procedures which create the four data files and order them appropriately are *NewPeople*, *NewFamilies*, *NewMarriage* and *NewParents* respectively (*listing 2*).

In order that the *Marriage* and *Parents* files do not end up with data that conflicts with that found in the *Families* file, the procedure *UpSubs* (*listing 2*), which is called (when inputting a family) updates the two files from the given record in the *Families* file. In this way, although we have redundant data in the database, we may be reasonably confident that it won't conflict, at least across files.

Multiple Files

There are a number of reasons for using multiple files rather than trying to fit all the data into one file. As we have already seen with the two extra files *Marriage* and *Parents*, the files are needed to give faster access

Listing 1. Save this in a file called *Relation_prg*. It is executed by typing: run "Relation".

```
proc p10;s$
  rem *** Prints line leaving coloured border ***

  print tab 10; paper 0;"   ";S$+rept(" ",50-len(S
$));"   "
  endproc
proc Start
  rem *** This is the first procedure to be called
.  Displays ***
  rem *** details of program and loads procedures
for phase 1 ***
  paper 2: ink 7: mode 0,4
  cls : print at 5,14; paper 0;" ""RELATIONS"" "
  print at 10,3; paper 0;"   A suite of Archive pr
ocedures   "
  print tab 3; paper 0;"  for storing and displayi
ng your  "
  print tab 3; paper 0;"             family tree.
    "
  print at 18,12; paper 0;"çAndy Carmichael"
  print at 23,1;"Press any key to continue..."
  let a$=getkey(): mode 0,8: cls : print at 5,1
  p10;"":p10;"INSTRUCTIONS..."
  p10;"---------------"
  p10;"":p10;"Loading procedures for phase I..."
  p10;"":p10;"During this phase the files are crea
ted (for a new"
  p10;"database) or opened for existing databases.
 You"
  p10;"must enter the file names."
  p10;"":p10;"For a new database details may be en
tered of the"
  p10;"people and their relationships.":p10;""
  print at 22,1;"Loading..."; at 21,1
  run "relat1"
  endproc
```

Listing 2. Save in file *Relat1_prg* from where it can be loaded and run by the procedures in listing 1.

```
proc Bye
  rem *** Close all files that may be open.  NB: A
n error ***
  rem *** will be caused by this proc and must be
trapped ***
  while 1: close : endwhile
  endproc
proc Check
  rem *** Checks if an error occurred opening file
s ***
  rem *** If so closes files and retries.
  ***
  if errnum(): error Bye: cls
    print ink 2;"Error in opening files"
    Start: endif
  endproc
proc Chop;X$,n,Delim$
  rem *** Sets C$ to given string with first n 'wo
rds' removed ***
  local Words,Dlim
  let Words=n: let C$=X$
  let Dlim=instr(X$,Delim$)
  while Words
    if not Dlim or Dlim>=len(C$): let C$="": retur
n : endif
    let C$=C$(Dlim+1 to )
    let Dlim=instr(C$,Delim$)
    let Words=Words-1
    endwhile
  endproc
proc DelFam;Fam
  rem *** Deletes the records in all files  associ
ated with ***
  rem *** family "Fam"   (Does not delete "people"
 records) ***
  use "F": locate Fam
  if Fam<>F.Family: return : endif
  DelMar;F.Father,Fam
  DelMar;F.Mother,Fam
  let C$=F.Children$
  while val(C$)
    DelPar;val(C$),Fam
    Chop;C$,1,","
    endwhile
  delete "F"
  print "*** Family DELETED ***"
  endproc
proc DelMar;Who,Fam
  rem *** Deletes marriage of "Who" if in family "
Fam" ***
  use "M": locate Who
  while M.RefNo=Who
    if M.Family=Fam
      delete "M"
      else : next : endif
    endwhile
  endproc
proc DelPar;Who,Fam
  rem *** Deletes parents of "Who" if in family "F
am" ***
  use "Pa": locate Who
  while Pa.RefNo=Who
    if Pa.Family=Fam
      delete "Pa"
      else : next : endif
    endwhile
  endproc
proc FileNames
  rem *** Gets names of data files (Defaults given
) ***
  input "     Name of  'people'  file (Peo_dbf): "
;Peo$
  input "     Name of 'families' file (Fam_dbf): "
;Fam$
  input "     Name of 'parents'  file (Par_dbf): "
;Par$
  input "     Name of 'marriage' file (Mar_dbf): "
;Mar$
  if Peo$="": let Peo$="Peo": endif
  if Fam$="": let Fam$="Fam": endif
  if Par$="": let Par$="Par": endif
  if Mar$="": let Mar$="Mar": endif
  endproc
proc GetPerson
  rem *** Finds record of person (or inserts one)
***
  let yes=0
  while not yes
    getrec;"P"
    if not yes
      YesNo;"Do you want to enter new person? "
      if yes:InPerson: endif
      endif
    endwhile
  endproc
proc GetRec;lfn$
  rem *** Finds record, displays it, searches for
next if not OK ***

  local n$,I
  if lfn$="": let lfn$="Main": endif
  use lfn$: let yes=0
  input "who? ";n$
  if n$="": return : endif
  find n$
  while not yes
    if not found(): print n$;" not found": return
: endif
    let I=0
    while I<numfld(lfn$)
      print fieldv(I);":"
      let I=I+1
      endwhile
    print :YesNo;"OK (y/n)? "
    if not yes: continue : endif
    endwhile
  endproc
proc InFamily;Df$,M,D
  rem *** For inputting a family. Default surname
is Df$. ***
  rem *** RefNo of Mother (M) or Father (D) may be
 given. ***
  cls
  while 1
    use "F": last
    let F.Family=F.Family+1
    print "Father is ";
    rem (FOR PHASE II) if D:prin;D,0,0: else
    let Sx=1: let Def$=Df$:GetPerson:REM endif
    let F.Father=P.RefNo: let ChN$=P.Surname$
    print "Mother is ";
    rem (FOR PHASE II) if M:prin;M,0,0: else
    let Sx=0: let Def$="":GetPerson:REM endif

    let F.Mother=P.RefNo
    input "Date married? ";F.DateMarried$
    input "If divorced, give date... ";F.Dissolved
$
    let F.Children$=""
    YesNo;"Any children? "
    while yes
      let Def$=ChN$: let Sx=1:GetPerson
      let F.Children$=F.Children$+str(P.RefNo,2,0)
+","
      YesNo;"Any more children? "
      endwhile
    use "F": append :UpSubs
    rem (FOR PHASE II)let Subject=F.Father:Details
;Subject
    print :YesNo;"Satisfactory? (y/n) "
    if yes: return : else
      YesNo;"Delete family just entered?"
      if not yes: return : else
        DelFam;F.Family: endif : endif
    endwhile
  endproc
proc InPerson
  rem *** For inputting details of a person (setti
ng defaults) ***
  cls : use "P": last
  let P.RefNo=P.RefNo+1
  let P.Surname$=Def$
  let P.FirstNames$=""
  let P.Male=Sx
  let P.DateOfBirth$=""
  let P.DateOfDeath$=""
  let P.Comments$=""
  append : sprint : alter : cls
  endproc
proc Instructions
  rem *** Display instructions for next phase whil
e loading ***
  cls : print ink 2;"Instructions for phase II."
  print : print "During this phase family trees ma
y be displayed and changes made to"
  print "the database (if files were opened for mo
difications)."
  print : print "The following key presses have th
e effect shown:"
  print : print tab 5;"↑"; ink 4; tab 25;"Subject
moves to father (if known)"
  print tab 5;"<SHIFT> ↑"; ink 4; tab 25;"Subject
moves to mother (if known)"
  print tab 5;"←"; ink 4; tab 25;"Subject moves to
 eldest sibling (if any)"
  print tab 5;"→"; ink 4; tab 25;"Subject moves to
```

continued overleaf

to the data. Another important reason is that a file such as the *People* file may be used in a number of applications not at all related to this particular case. In order to avoid duplicating the data for each application – with the accompanying risk of inconsistency through not updating all the data files when the information changes – the information should be separated so that different application programs can access common files. This is linked to the most important reason why multiple data files arise, which is the natural structure of the data. In the stock control case for example, given in the Users' Manual (Archive p.34), the three data files, Stock, Suppliers and Orders, are three naturally distinct areas. Only by distorting the structure of the data, and adding many more fields to each record, could these be fitted into a single data file. This is also the case with the present example. If the structure isn't right, family relationships can become very complicated to deal with.

The procedures in *listings 1* and *2* should be saved into two files (called *Relation_prg* and *Relat1_prg*). This speeds up loading and saves memory. Don't believe what it used to say in the Archive manual (page 54, paragraph six) – the size of your database *is* limited by the amount of memory in the computer, and too many procedures cut down the available memory dramatically. Having entered and saved the procedures the method for using the package is simply to type: run "relation"

This runs the first procedure which displays the instructions while loading the next file. The program then gives you the option of starting an entirely new database, modifying an existing set of files, or looking at a set of existing files.

Should you opt to start a new database a series of prompts will lead you through the various steps of data input, family by family. The amount of memory remaining is displayed regularly as when this falls below a certain amount Archive may crash without warning. Reference numbers will be automatically incremented. Entries should be made in the remaining fields, namely Surnames, First Names, Male(1 or 0), Date Of Birth, Date of Death and Comment.

WISE WORDS

The root of most evils associated with Archive is a chronic lack of memory. Record indices, procedures and user-defined screen layouts all competed for a limited 12K on version 1.01 ADB. The additional 8K available under version 2.0 was therefore most welcome.

This is amply illustrated on our relational database where the Archive program routines occupy almost 12K by themselves leaving 8K for the data file index. The latter occupies space along the following lines:

No ordering	– 6 bytes/record
1 ordered field	– 14 bytes/record
2 ordered fields	– 22 bytes/record
3 ordered fields	– 30 bytes/record
4 ordered fields	– 38 bytes/record

As our four files are ordered on only one field there should be room for 600 records or about 150-200 people in our family tree on an unexpanded QL. However, in practice, Archive may start falling over before this upper limit is reached and data could be lost. Our advice, therefore, is to back-up your data and program files regularly. Never OPEN a *_dbf* file unless you have at least one back-up copy on a separate cartridge. As regards procedures, keep two copies of the *_prg* files and after each EDIT save the procedures alternately to one of the files. That way if a change doesn't work properly you can go back to the previous version.

Next Month: We conclude this topic with procedures for displaying the trees.

```
 next sibling (if any)"
  print tab 5;"="; ink 4; tab 25;"Subject moves to
 spouse (if any)"
  print tab 5;"<SHIFT> = (ie. +)"; ink 4; tab 25;"
Subject moves to NEXT spouse. (Only valid after ""
="")"
  print tab 5;"↓"; ink 4; tab 25;"Subject moves to
 first child (if any)"
  print tab 5;"?"; ink 4; tab 25;"Search for new s
ubject"
  print : print tab 5;"<SPACE> or <ENTER>"; ink 4;
 tab 25;"Display family tree of subject"
  print tab 5;"E or <ALT>+any_key"; ink 4; tab 25;
"Edit the subject's family"
  print tab 5;"I"; ink 4; tab 25;"Insert a new fam
ily"
  print : print tab 5;"<ESC>"; ink 4; tab 25;"Exit
 from program."
  print at 22,1; ink 2;"Loading..."; at 21,1
  endproc
proc Looks
  rem *** Opens the files for read only ***

  print "     Opening files (read only)..."
  look Fam$ logical "F"
  look Mar$ logical "M"
  look Par$ logical "Pa"
  look Peo$ logical "P"
  endproc
proc MakeNew
  rem *** Start a new database ***

  cls : print "     Starting a new database..."
  error NewFamilies;Fam$:Check
  error NewMarriage;Mar$:Check
  error NewParents;Par$:Check
  error NewPeople;Peo$:Check
  YesNo;"     Do you want to start inputting data?
"
  while yes
    error InFamily;"",0,0
    if errnum(): cls : print "***WARNING*** Error
";errnum();" occurred during input of family"
      YesNo;"Delete last family?": if yes:DelFam;F
.Family: endif : endif
    YesNo;"Another family?"
    endwhile
  Mem
  endproc
proc Mem
  rem *** Prints remaining memory and No. of recor
ds used ***

  cls : print "Total memory remaining =  ";
  print memory();" Bytes"
  print "RECORDS USED:";
  use "P": print tab 15;"People:    ";count()-1
  use "M": print tab 15;"Marriages: ";count()-1
  use "Pa": print tab 15;"Parents:   ";count()-1
  use "F": print tab 15;"Families:  ";count()-1
  endproc
proc NewFamilies;Name$
  rem *** New 'Families' file ***

  create Name$ logical "F"
    Family
    Father
    Mother
    DateMarried$
    Dissolved$
    Children$
    endcreate
  let F.Family=0
  append
  order Family;a

  endproc
proc NewMarriage;Name$
  rem *** New 'Marriage' file ***

  create Name$ logical "M"
    RefNo
    Family
    endcreate
  append
  order RefNo;a
  endproc
proc NewParents;Name$
  rem *** New 'Parents' file ***

  create Name$ logical "Pa"
    RefNo
    Family
    endcreate
  let RefNo=0
  append
  order RefNo;a
  endproc
proc NewPeople;Name$
  rem *** New 'People' file ***

  create Name$ logical "P"
    RefNo
    Surname$
    FirstNames$
    Male
    DateOfBirth$
    DateOfDeath$
    Comments$
    endcreate
  let RefNo=0
  let P.Surname$="#NULL#"
  append
  order RefNo;a
  endproc
proc Opens
  rem *** Opens files with write access ***

  print "     Opening files..."
  open Fam$ logical "F"
  open Mar$ logical "M"
  open Par$ logical "Pa"
  open Peo$ logical "P"
  endproc
proc Start
  rem *** This procedure is called first          *
**
  rem *** Opens the files - (new, write or read) *
**
  print "Press any key to continue..."
  let a$=getkey(): paper 0: cls : print at 5,1
  YesNo;"     Do you want to start a new database?
 "
  FileNames
  if yes:MakeNew: error Bye: endif
  YesNo;"     Do you want to modify the database?
"
  if yes
    YesNo;"     Have you backed up the data files?
 "
    if yes
      print paper 6; ink 0;"WRITE privilege to fil
es - BEWARE! "
      error Opens:Check
      else
      print "     Back up files (";Fam$;",";Mar$;"
,";Par$;",";Peo$;") before continuing"
      print "     Press any key to continue..."
      let a$=getkey(): mode 1,8: stop
      endif
    else
    error Looks:Check
    endif
  let Subject=1
  Instructions
  run "relat2":rem *** The listing for these proce
dures will be given in next month's article ***

  endproc
proc UpSubs
  rem *** Update "Marriage" and "Parents" files **
*
  use "F"
  let M.Family=F.Family
  let M.RefNo=F.Father
  append "M"
  let M.RefNo=F.Mother
  append "M"
  let Pa.Family=F.Family
  let C$=F.Children$
  while val(C$)
    let Pa.RefNo=val(C$)
    append "Pa"
    Chop;C$,1,","
    endwhile
  endproc
proc YesNo;P$
  rem *** Gets Y or N from keyboard and sets varia
ble "yes"
  while 1
    print P$;: let Q$=lower(getkey())
    let yes=(Q$="y")
    if instr("ny",Q$): print " "+Q$: return : endi
f
    print : endwhile
  endproc
```

BACK IN TOWN FOR A GREAT NEW SHOW!

MICROFAIR

NEW HORTICULTURAL HALL
Saturday 22nd June 1985 10am – 6pm
Greycoat Street/Elverton Street
(Junction) London SW1
Admission: £1.50 (Adults)
£1.00 (Kids under 14 yrs)

If you have a Sinclair Computer this is the show you can't afford to miss!

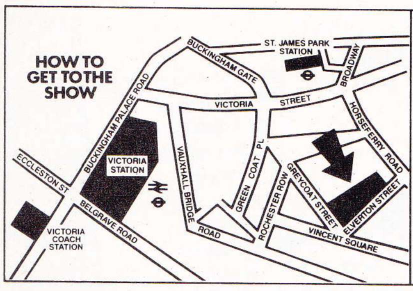
The great day is June 22nd.
One day you won't forget if you own or use a Sinclair Computer.

The Microfair is back in town.

See everything that's new and original for The Spectrum, Spectrum+ and the fantastic QL.

If it's just launched you'll find it at the MICROFAIR. If you want advice you can talk face to face with the manufacturers. If you can't find it in the shops you'll probably find it at the MICROFAIR.

Why not send today for cut-price advanced tickets.



Underground: Nearest stations are Victoria, and St James's Park.
British Rail: Victoria Station.
Bus Routes: 11, 24, 29, 70, 76 and Red Arrow 507.
Road: Signposted (RAC AA) Horticultural Halls.

POST TODAY

Send to Mike Johnston (Organiser), Dept **QL**, ZX Microfairs, 71 Park Lane, London N17 0HG.

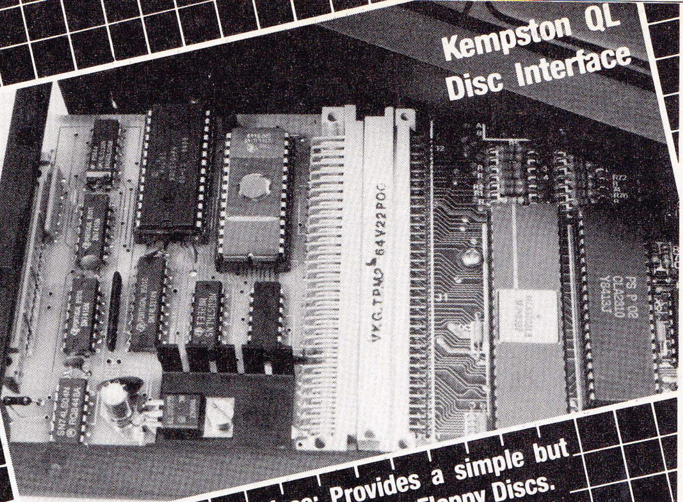
Please send me .... Advance Tickets (Adult) @ £1.25
Please send me .... Advance Tickets (Child under 14) @ 80p.
(Prices at the door are £1.50 and £1.00

16th ZX MICROFAIR

Name: ______________________
Address: ______________________

Please include a stamped self-addressed envelope and make cheques/POs payable to ZX Microfair.
EXHIBITORS: CALL MIKE JOHNSTON ON 01-801 9172 FOR STAND AVAILABILITY.

Kempston QL Disc Interface

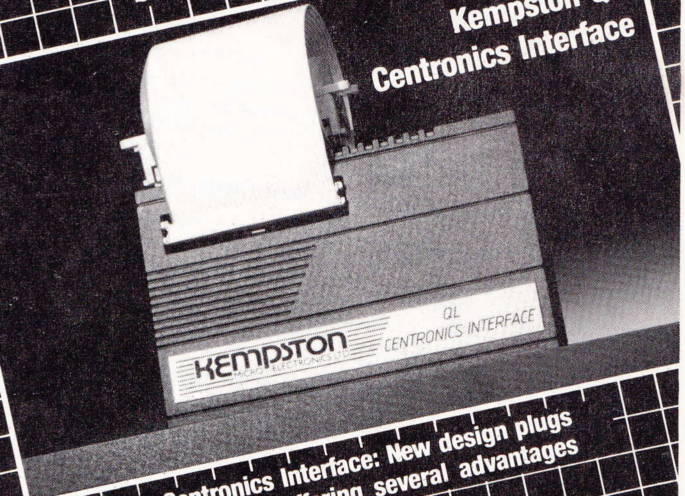


Kempston QL Disc Interface: Provides a simple but powerful upgrade from Microdrives to Floppy Discs.

- Microdrive filing system operations are supported.
- QL Toolkit commands included in 8K operating system. Microdrive extentions OPEN, OVERWRITE, RENAME, TRUNCATE. File handling commands for direct access. Wild name definition of files. WSTAT, WDIR, WDEL, WDEL-F. Job control commands for multitasking under QDOS.
- Compatible with standard independent powered disc drives, 3", $3^1/_2$" and $5^1/_4$".
- Single or multiple Disc Drives.
- Fits compactly inside the QL.
- Complete with a fully documented manual.

£115.00 inc VAT & PP

Kempston QL Centronics Interface



Kempston QL Centronics Interface: New design plugs directly into ROM port, offering several advantages over the serial drivers provided.

- Large buffers for print spooling.
- Automatic form creation.
- File holding.
- Driving software contained in an on-board ROM.
- Allows screen dump on nine different printer formats.
- Comes complete with a metre of cable, centronics connector and a concise manual.

£39.95 inc VAT & PP

Available directly from Kempston Micro Electronics Ltd. Singer Way, Kempston, Bedford MK42 7AF. Tel. (0234) 856633 Please send me;

☐ **Kempston QL Disc Interface £115.00** ☐ **Kempston QL Centronics Interface £39.95**

Access/B.Card ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Signature ______________

I enclose a cheque/P.O. for £ ______________

Name/Address ______________________

All prices include VAT and P&P
Overseas add £4.00 P&P

TERMINAL EMULATION

Continuing our communications terminal, Adam Denning introduces some special options.

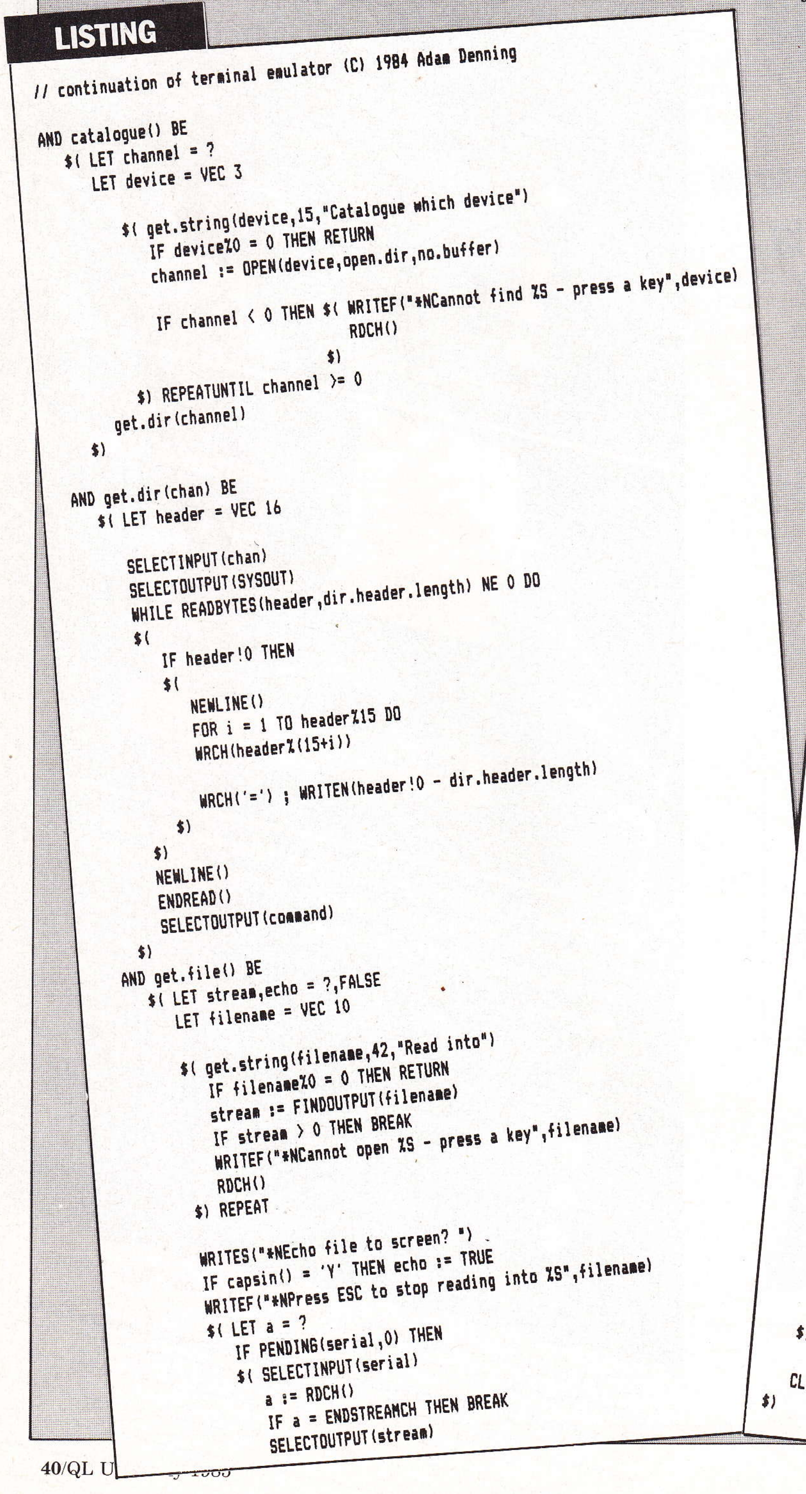
We left things in mid-air last month with the most interesting options menu still to come. This menu is selected when you press function key F3, and includes wonderful things like job control and file transmission. Not surprisingly, it's one of the hardest parts of the program to understand, as it has to access QDOS routines at a very basic level.

The main controlling routine is **poptions()**, which is called from **action()**. It follows the normal path of selecting the command window for input and output, displaying the menu and waiting for a valid keypress. The options available (apart from ESC) are:

C:	Catalogue drives
J:	Job control
R:	Read file
S:	Send file
W:	Re-draw

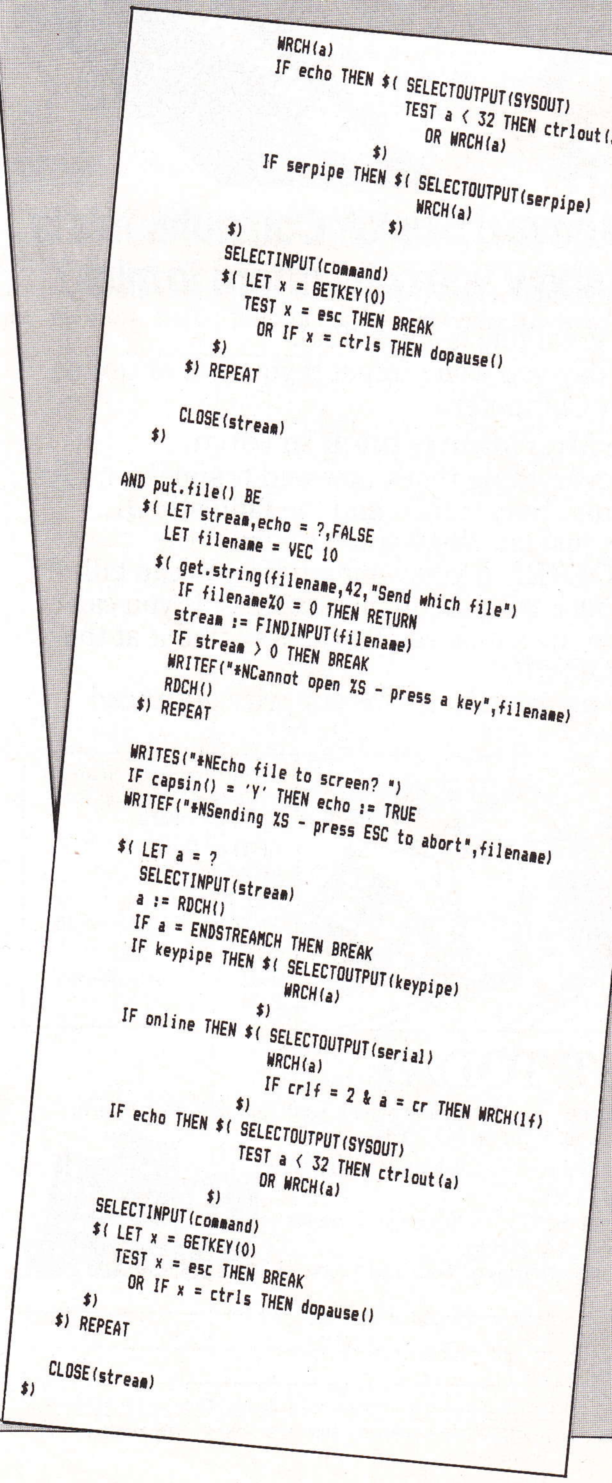
We'll start off with the easiest, which is the 'W' option to re-draw the windows. All it actually does is reinstate the colours and borders for the terminal and status windows, as well as re-drawing the clock window while it isn't looking. This isn't really happening of course, but by opening a screen device with identical parameters and clearing it, it reinstates the clock's window.

All this is done within the SWITCHON block, while all the other options call their own routines. If we take a look at the catalogue drives option, we see that it calls a procedure called **catalogue()**. This repeatedly prompts for a device to catalogue ('mdv1_', 'flp2_', etc) until



LISTING

```
// continuation of terminal emulator (C) 1984 Adam Denning

AND catalogue() BE
   $( LET channel = ?
      LET device = VEC 3

         $( get.string(device,15,"Catalogue which device")
            IF device%0 = 0 THEN RETURN
            channel := OPEN(device,open.dir,no.buffer)

            IF channel < 0 THEN $( WRITEF("*NCannot find %S - press a key",device)
                                   RDCH()
                                $)
         $) REPEATUNTIL channel >= 0
       get.dir(channel)
    $)

  AND get.dir(chan) BE
     $( LET header = VEC 16

        SELECTINPUT(chan)
        SELECTOUTPUT(SYSOUT)
        WHILE READBYTES(header,dir.header.length) NE 0 DO
        $(
           IF header!0 THEN
           $(
              NEWLINE()
              FOR i = 1 TO header%15 DO
              WRCH(header%(15+i))

              WRCH('=') ; WRITEN(header!0 - dir.header.length)
           $)
        $)
        NEWLINE()
        ENDREAD()
        SELECTOUTPUT(command)
     $)
  AND get.file() BE
     $( LET stream,echo = ?,FALSE
        LET filename = VEC 10

        $( get.string(filename,42,"Read into")
           IF filename%0 = 0 THEN RETURN
           stream := FINDOUTPUT(filename)
           IF stream > 0 THEN BREAK
           WRITEF("*NCannot open %S - press a key",filename)
           RDCH()
        $) REPEAT

        WRITES("*NEcho file to screen? ")
        IF capsin() = 'Y' THEN echo := TRUE
        WRITEF("*NPress ESC to stop reading into %S",filename)
        $( LET a = ?
           IF PENDING(serial,0) THEN
           $( SELECTINPUT(serial)
              a := RDCH()
              IF a = ENDSTREAMCH THEN BREAK
              SELECTOUTPUT(stream)
```



```
              WRCH(a)
              IF echo THEN $( SELECTOUTPUT(SYSOUT)
                              TEST a < 32 THEN ctrlout(a)
                                 OR WRCH(a)
                           $)
              IF serpipe THEN $( SELECTOUTPUT(serpipe)
                                 WRCH(a)
           $)                 $)
           SELECTINPUT(command)
           $( LET x = GETKEY(0)
              TEST x = esc THEN BREAK
                 OR IF x = ctrls THEN dopause()
           $)
        $) REPEAT

        CLOSE(stream)
     $)

AND put.file() BE
  $( LET stream,echo = ?,FALSE
     LET filename = VEC 10
     $( get.string(filename,42,"Send which file")
        IF filename%0 = 0 THEN RETURN
        stream := FINDINPUT(filename)
        IF stream > 0 THEN BREAK
        WRITEF("*NCannot open %S - press a key",filename)
        RDCH()
     $) REPEAT

     WRITES("*NEcho file to screen? ")
     IF capsin() = 'Y' THEN echo := TRUE
     WRITEF("*NSending %S - press ESC to abort",filename)

     $( LET a = ?
        SELECTINPUT(stream)
        a := RDCH()
        IF a = ENDSTREAMCH THEN BREAK
        IF keypipe THEN $( SELECTOUTPUT(keypipe)
                           WRCH(a)
                        $)
        IF online THEN $( SELECTOUTPUT(serial)
                          WRCH(a)
                          IF crlf = 2 & a = cr THEN WRCH(lf)
                       $)
        IF echo THEN $( SELECTOUTPUT(SYSOUT)
                        TEST a < 32 THEN ctrlout(a)
                           OR WRCH(a)
                     $)
        SELECTINPUT(command)
        $( LET x = GETKEY(0)
           TEST x = esc THEN BREAK
              OR IF x = ctrls THEN dopause()
        $)
     $) REPEAT

     CLOSE(stream)
$)
```

it can successfully open the directory on this device with the OPEN function. It then calls a further procedure, **get.dir,** to extract the directory information and display it. **get.dir** simply reads each directory entry, whic is 64 bytes long, into a vector. The first word of this vector will then hold the file's length. If the length is zero, the file has been deleted, so we ignore it. Otherwise, we extract the filename from the vector, print it out on the screen along with an '=' sign, and then follow this with the decimal file length in bytes. A directory entry contains the length of the file + 64, so we must subtract 64 from this length before printing it out.

File transmission starts with the **put.file()** procedure, which is fairly simple. It asks you for the name of the file to send, and opens it. It then asks if you want to echo the file to the QL's screen as it is being sent. Pressing 'Y' or 'y' at this stage sets a local variable called **echo** to TRUE. A message is printed onto the command window telling you which file is being sent, and how to stop it being sent (by pressing ESC). The routine then enters a REPEAT loop which reads a character from the file into **a,** and breaks out of the loop if EOF has been reached (when **a** will equal ENDSTREAMCH). If the **keypipe** global variable is set, the byte is sent down the keyboard pipe, and if the terminal is online we also send this down the serial line. Notice how we take special provision to send an extra linefeed character if **crlf** is 2 and **a** is a carriage return. This keeps the host happy. If local echo is requested, we output the character to the screen, using **ctrlout** to convert it if necessary.

At the end of each character transmission, we test the keyboard to see if ESC has been pressed. If it hasn't, we loop and process the next character, otherwise we exit the loop. We also check for CTRL-S, so that the user can pause transmission temporarily.

Reading of files follows much the same process, but in reverse of course, using the **get.file** procedure. Pressing ESC will abort file reading, and CTRL-S will suspend it until CTRL-Q is pressed.

The final option is job control, and this calls the **jobcontrol** procedure. This tests a global variable called **jobnua** to see if you have accessed the job control routines before. This is so the pipe accessing routines can tell if they can validly be called, as it

```
AND jobcontrol() BE
  $( LET ans = ?
     jobnua := TRUE
     WRITES("*N  Jobs:  A)ctivate  C)reate  K)ill  S)uspend  R)elease  P)riority  I)nformation")
     ans := capsin() REPEATUNTIL ans = 'A' | ans = 'C' | ans = 'K' |
                                 ans = 'S' | ans = 'R' | ans = 'P' | ans = 'I'

     job.exec(ans)
  $)

AND job.exec(option) BE
  $( LET myjob = 0
     TEST 'C' NE option NE 'I' THEN $( LET numbervec = VEC 2
                                       $( get.string(numbervec,11,"Which job?")
                                          myjob := get.hex(numbervec)
                                          IF ISJOB(myjob) THEN BREAK
                                          WRITEF("*NJob %X8 does not exist - press a key",myjob)
                                          RDCH()
                                       $) REPEAT
                                       SWITCHON option INTO
                                       $(
                                          CASE 'A': doactiv(myjob)
                                                    ENDCASE
                                          CASE 'K': KILLJOB(myjob)
                                                    ENDCASE
                                          CASE 'P': doprior(myjob)
                                                    ENDCASE
                                          CASE 'R': RELEASE(myjob)
                                                    ENDCASE
                                          CASE 'S': dosusp(myjob)
                                                    ENDCASE

                                       $)
                                    $)
     OR TEST option = 'C' THEN
        $( LET stream,addr = ?,?
           LET filename = VEC 10
           LET header = VEC 3
           $( LET datalen = 0
              $( get.string(filename,42,"Filename")
                 IF filename%0 = 0 THEN RETURN
                 stream := OPEN(filename,1,0)
                 IF stream > 0 THEN BREAK
                 WRITEF("*NCannot open %S - press a key",filename)
                 RDCH()
              $) REPEAT
              READFILEHEADER(stream,header,16)
              FOR i = 6 TO 9 DO datalen := (datalen << 8) + header%i
              addr := CREATEJOB(header!0,datalen)
              IF addr NE 0 THEN BREAK
              WRITES("*NCannot create job - press a key")
              RDCH()
           $) REPEAT
           READFILE(stream,addr,header!0)
           CLOSE(stream)
           myjob := RESULT2
           WRITEF("Job %X8 is now created - press a key",myjob)
           SCREEN(screen.cursor)
           RDCH()
           SCREEN(screen.nocursor)
```

```
                 $)
              OR get.info()
         $)

   AND get.hex(hexvec) = VALOF
     $( LET count = 0
        FOR i = 1 TO hexvec%0 DO
        $( LET ch = hexvec%i
           TEST '0' <= ch <= '9' THEN ch := ch - '0'
              OR ch := ch + 10 - 'A'
           count := (count << 4) + ch
        $)
        RESULTIS count
     $)

   AND doprior(thisjob) BE
     $( LET prior = ?
        LET numvec = VEC 2
        get.string(numvec,11,"Priority")
        prior := get.dec(numvec)
        PRIORITY(thisjob,prior)
     $)

   AND doactiv(thisjob) BE
     $( LET prior,timeout = ?,?
        LET numvec = VEC 2
        get.string(numvec,11,"Priority")
        prior := get.dec(numvec)
        get.string(numvec,11,"Timeout")
        timeout := get.dec(numvec)
        ACTIVATE(thisjob,prior,timeout)
     $)

   AND dosusp(thisjob) BE
     $( LET timeout = ?
        LET numvec = VEC 2
        get.string(numvec,11,"Timeout")
        timeout := get.dec(numvec)
        SUSPEND(thisjob,timeout)
     $)

   AND get.info() BE
     $( LET nextjob = 0
        $( LET base = JOBINFO(nextjob)
           WRITEF("*N Job %X8 Owner %X8 Priority ",nextjob,RESULT2)
           nextjob := jinfo1
           TEST (jinfo2 >> 8) > #X7FFF THEN WRCH('S')
              OR WRCH(' ')
           WRITEF("%I3 Name ",(jinfo2 & #XFF))
           IF base%6 = #X4A & base%7 = #XFB THEN
              FOR i = 1 TO base%9 DO WRCH(base%(9+i))
           WRITES(" - press a key")
           RDCH()
           IF nextjob = 0 THEN BREAK
        $) REPEAT
        WRITES("*NThere are no more jobs in the tree - press a key")
        RDCH()
     $)

   AND poptions() BE
```

makes little sense to open a pipe without a job to send the data to!

Following this, it writes out a new menu, which lets you select an aspect of job control. You may select

A:	Activate a job
C:	Create a job
K:	Kill a job
S:	Suspend a job
R:	Release a job
P:	Change a job's priority
I:	Get information on a job

If we could just type in all the BCPL code and run it, all would be great. Unfortunately, we need another section of machine code to get the effects we require, and this introduces three new routines to the library (**ISJOB, JOBINFO** and **PUTPIPE**), which completes our set of machine code extensions. You must remember to change libhdr to incorporate the **ISJOB** routine in its global list, and to alter UG. The other routines are specific to this program so their global declarations appear in the source. The code for each routine is fairly straightforward: **ISJOB** asks QDOS if the specified job exists, and returns TRUE if it does, **JOBINFO** scans the job tree and returns the results one at a time, and **PUTPIPE** puts a QDOS channel ID on top of a job's stack. Remember to link the assembled version of this file when you compile the whole program.,

Please note: June's addition to the terminal emulation program *should* have made it into a working copy. Unfortunately, a couple of things went wrong! Apart from leaving out the **get.string** procedure (which was reproduced last month), there was a typing error on the dummy **action()** routine – the set of empty parentheses were left out after the named, which confuses the compiler into thinking you're declaring a variable in an illegal place!

Finally, a manifest constant called **err.ef,** was used which is the value of the internal code for QDOS error 'end of file'. I did not include this in the program's MANIFEST declaration list because it was in my copy of libhdr. I find it useful to keep *all* QDOS constants in libhdr, but totally forgot that not everyone knew them. The value of **err.ef** is −10, so include **err.ef** = −10 in the MANIFEST declarations at the beginning of the program.

```
$( LET option = ?
   SCREEN(screen.nocursor)

   SELECTINPUT(command)
   SELECTOUTPUT(command)

   WRITES("*N  C: catalogue drives  J: job control  R: read file  S: send file  W: re-draw")
   option := capsin() REPEATUNTIL option = 'C' | option = 'J' | option = 'R' |
                                  option = 'S' | option = 'W' | option = esc

   SWITCHON option INTO
   $(
      CASE 'C': catalogue()
                ENDCASE
      CASE 'J': jobcontrol()
                ENDCASE
      CASE 'R': TEST online THEN get.file()
                   OR $( WRITES("*NYou are not online - press a key")
                         RDCH()
                      $)
                ENDCASE
      CASE 'S': put.file()
                ENDCASE
      CASE 'W': SELECTOUTPUT(FINDOUTPUT("SCR_150x12a12x244"))
                SCREEN(screen.border,white,1)
                SCREEN(screen.paper,18)
                SCREEN(screen.clear)
                ENDWRITE()
                set.status()
                SCREEN(screen.border,red,1)
                SCREEN(screen.ink,white)
                SELECTOUTPUT(command)
                ENDCASE
   $)
   set.comm()
   SELECTINPUT(SYSIN)
   SCREEN(screen.cursor)
$)
```

```
* A routine to return job information to a BCPL program
* baseaddress := JOBINFO(jobID) Global 126
* Returns BCPL base address, next job in tree in JINFO1, owner in RESULT2 and
* priority / suspension status in JINFO2
* A routine to put a channel ID onto a job's stack
* PUTPIPE(base,stream) Global 129
* The first parameter is the byte base address of the job and the second is the
* BCPL stream identifier
* A routine to return TRUE if a given job exists
* jobexists := ISJOB(jobID) returns TRUE if job exists and FALSE if not.
* Global 115

RESULT2   EQU     10
ISJOB     EQU     115
JOBINFO   EQU     126
JINFO1    EQU     127
JINFO2    EQU     128
PUTPIPE   EQU     129

MT_JINF   EQU     2
```

```
JOB_AREA  EQU       $68
SAVE_USP  EQU       $5C

FIRST     DC.L      (ENDMOD-FIRST)/4

          CNOP      0,4

JINFOHERE

          MOVEM.L   A0/A1,-(A7)             Save registers
          MOVEQ     #0,D2                   make Basic top-of-tree job
          MOVEQ     #MT_JINF,D0             get job info
          TRAP      #1
          MOVE.L    D1,(JINFO1*4)(A2)       store next job ID in JINFO1
          MOVE.L    D2,(RESULT2*4)(A2)      owner job in RESULT2
          MOVE.L    D3,(JINFO2*4)(A2)       and priority/susp in JINFO2
          MOVE.L    A0,D1                   move base address into D1
          LSR.L     #2,D1                   convert to BCPL address (>>2)
          MOVEM.L   (A7)+,A0/A1             retrieve registers
          JMP       (A6)                    and return

          CNOP      0,4

PIPEHERE

          MOVEM.L   A0-A2,-(A7)             Save registers
          MOVEA.L   D2,A1                   Put SCB address in A1
          MOVEA.L   (A1),A0                 ...and channel ID in A0
          MOVEA.L   D1,A1                   job base address in A1
          SUBA.L    #(JOB_AREA-SAVE_USP),A1 make A1 point to saved USP
          MOVEA.L   (A1),A2                 and USP in A2
          MOVE.L    A0,-(A2)                channel ID on stack
          MOVE.L    A2,(A1)                 and resave USP
          MOVEM.L   (A7)+,A0-A2             retrieve registers
          JMP       (A6)                    return

          CNOP      0,4

ISJOBHERE

          MOVEM.L   A0/A1,-(A7)
          MOVEQ     #0,D2
          MOVEQ     #MT_JINF,D0
          TRAP      #1
          MOVEQ     #-1,D1                  TRUE is -1
          MOVEM.L   (A7)+,A0/A1
          TST.L     D0
          BEQ.S     YES_JOB
          MOVEQ     #0,D1
YES_JOB   JMP       (A6)                    and FALSE is 0

          CNOP      0,4

          DC.L      0
          DC.L      JOBINFO,(JINFOHERE-FIRST)
          DC.L      PUTPIPE,(PIPEHERE-FIRST)
          DC.L      ISJOB,(ISJOBHERE-FIRST)
          DC.L      JINFO2

ENDMOD    END
```

More Memory to your QL

- **¼ MEGABYTE MORE.** This new memory Expansion Board from Simplex Data Ltd. gives your existing QL a further 256k worth of extra memory.
- **EVEN MORE EXPANSION.** In addition to the ever popular ¼ Megabyte board, you now have the unique opportunity to purchase a ½ Megabyte Memory Expansion board–512k! Even more memory for your QL
- **DESIGNED FOR THE SINCLAIR QL.** Most users of the QL will realise the limitations of a 128k system. Simplex Data have not only expanded the memory potential, but have given you, the user, a wider field of operation.
- **EASY TO USE.** Simply remove the port cover located at the end of the QL keyboard and slot in your NEW Simplex Memory Expansion board–It's as simple as that!
- **COMPACT SIZE.** There's nothing like it on the market. Measuring approximately 97mm x 77mm, this little board is a major advance in Micro technology.
- **HOW IS IT DONE?** Utilising the latest 256k x 1 bit Dynamic RAM'S.

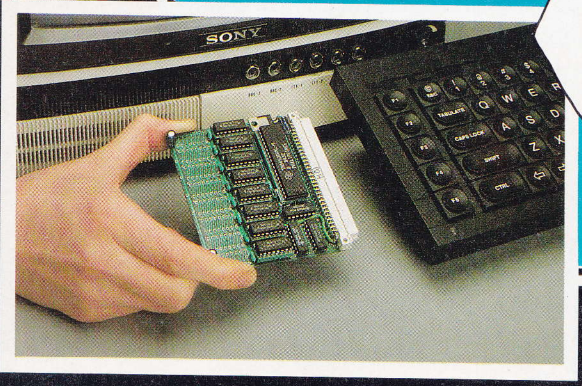
Only £198.00*

Inc. VAT, postage and packing. 1 year warranty included. Please allow 28 days for delivery.

64K	— £99.00
256K	— ★
512K	— £396.00

Inc VAT, P&P

day of April.



Exclusively available from

SDL Simplex Data Limited

432 Greenford Road, Greenford, Middlesex UB6 8SG. Tel: 01-575 7531

To: SIMPLEX DATA LIMITED, FREE POST, 432 GREENFORD RD, GREENFORD, MIDDLESEX UB6 8SG.

Please send me: ______________________(Qty) Memory Expansion Boards at £198.00 (inc. VAT, Post and Packing).

I enclose Cheque/Postal Order for £ ______________________

Name: ______________________

Address: ______________________

____

Telephone No.: ______________________

Payment by ACCESS or BARCLAYCARD phone: 01-575 7531

The Sinclair Vision QL

The only OFFICIAL monitor for the QL

Designed to perform

The Sinclair Vision QL colour monitor represents amazing value for money. Just look at these features:

- Developed in conjunction with Sinclair Research
- 12" high resolution RGB colour monitor
- Designed for QL's 85 column text and colour graphics display
- Ideal for professional business applications
- Full 12 months warranty
- Complete with integral lead and connector for the QL

Mail order and retail shop, trade and export enquiries welcomed. Government and educational establishments official orders accepted.
Carriage: £7.00 via Securicor.

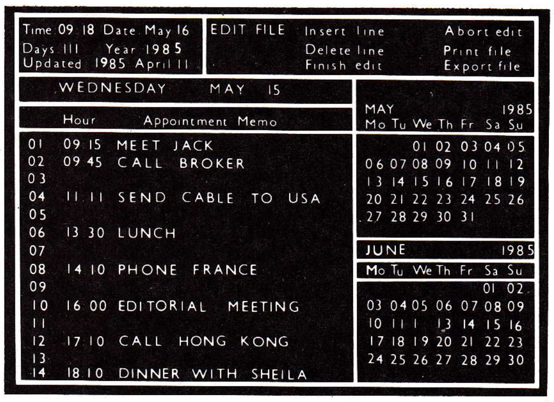
Access orders: simply phone Watford (0923) 50234 (24 hours)

Prices subject to change without notice and availability.

Our price – just £249 incl. VAT (save £50 on recommended prices)

Watford Electronics
Department QL, 250 High Street, Watford, Herts.
Tel. 0923 40588/37774 Telex 8956095

Enhance the Power of the QL



```
Time 09 18 Date May 16    EDIT FILE  Insert line    Abort edit
Days 111   Year 1985                 Delete line    Print file
Updated 1985 April 11                Finish edit    Export file

   WEDNESDAY      MAY   15

       Hour      Appointment Memo            MAY                 1985
                                             Mo Tu We Th Fr Sa Su
01   09 15  MEET JACK                              01 02 03 04 05
02   09 45  CALL BROKER                      06 07 08 09 10 11 12
03                                           13 14 15 16 17 18 19
04   11 11  SEND CABLE TO USA                20 21 22 23 24 25 26
05                                           27 28 29 30 31
06   13 30  LUNCH
07                                           JUNE                1985
08   14 10  PHONE FRANCE                     Mo Tu We Th Fr Sa Su
09                                                          01 02
10   16 00  EDITORIAL MEETING                03 04 05 06 07 08 09
11                                           10 11 1  13 14 15 16
12   17 10  CALL HONG KONG                   17 18 19 20 21 22 23
13                                           24 25 26 27 28 29 30
14   18 10  DINNER WITH SHEILA
```

AGENDA The program that does everything your pocket or desk diary will not do. Agenda is a calendar-based program to remind you of appointments, phone messages, birthdays, holidays, bills and many more.

- ⋆ Make a print-out of any display
- ⋆ Alarm facility to warn you of an appointment
- ⋆ Real time clock
- ⋆ Facility to export to Quill
- ⋆ Zoom in to any day facility
- ⋆ and more

Introductory offer £19.95 (fully inclusive price) delivered in the U.K.

Q-SOFT
P.O. Box 90
Barnet
Hertfordshire
EN5 5RN
Tel 01-449 7417

Phone or post this coupon. I enclose a cheque for £______
Debit my Access/Visa Acc. No. Card expiry date:______

Name:______
Address:______
______Postcode:______
Tel. No.:______Signature:______

Fully inclusive price delivered in the U.K. Europe add £1.00. Allow 14 days for delivery.

Dealer enquiries welcome.

AI ON THE QL

Expert Systems are quickly proving themselves invaluable in many fields — Mike James looks at the reasons behind the reasoning.

ILLUSTRATION BY PAUL ALLEN

AI ON THE QL

Reasoning something out is a particularly human activity. Given a faulty computer, motor car or any other of today's modern applicances then an appropriate expert can (usually!) be found to correct the fault. Such fault finding is typical of the way that humans reason about the world and now computers are starting to get in on the act. So called 'expert systems' are being used on a regular basis to aid such diverse activities as medial diagnosis, electronic fault finding, mineral exploration etc. Expert systems are probably AI's most successful product to date and for this reason alone they are worth examining.

Raison D'Etre

Computers reason using exact logic and this is why they are so good at obeying the programs that we write for them. However, humans rarely think logically in this sense, unless they are forced to by the nature of the problem. This is not to say that humans think illogically but that they have a different approach. A human will not analyse the situation in minute detail but try to draw on past experience (ie, match some detail of the problem with something seen in the past). In other words, use is made of a wide ranging knowledge of the way that the world works, knowledge that is generally not included in programs that reason logically.

If this is the case, then to construct programs that reason about things in the way that humans do we should first look at the representation of knowledge inside a computer. Computers are thought to be good at collecting and storing vast amounts of information and indeed this is true, but they store it in a very simple way. A computer's collection of facts is more like the way that an encyclopaedia 'remembers' things rather than the way a human remembers (memory is the subject of next month's article). For example, a collection of facts is useless unless you know what the 'consequences' of a fact are. If you are trying to decide what the weather will be, both you and the computer might know that the sky is black and full of clouds but you alone can deduce that this means it is likely to rain. In other words, you know the possible consequences of a black, cloud-filled sky – the computer does not!

Knowing The Question

It's not difficult to think of ways of storing information along with the consequences as a collection of rules. For example, the weather 'knowledge' could be sorted inside a computer as: 'IF black cloudy sky THEN high possibility of rain'.

In general a piece of knowledge can be represented by a list of conditions and a list of consequences. For example: 'IF black clouds, high humidity, summer THEN thunder storm' might be a statement of what we know about thunder storms. The commas between each condition should be read as 'AND' because each of the conditions has to apply before we are willing to reach the conclusion with a degree of confidence. Notice that although we are using IF . . . THEN, which is so familiar from programming, this use is different. In this case IF . . . THEN isn't an instruction to do something if something else is true, it is a statement of the relationship between different facts.

If you wanted to construct a program that used such rules to solve problems then all you would do is collect as many rules as possible, in other words build a 'rule database', and then, to find out the meaning or consequence of a set of conditions, simply search the database for rules with the same conditions. There may be more than one rule for any particular set of conditions. For example, if you knew there was a black sky you might search and find both of the IF . . . THEN weather rules given above. The predicted consequences would be rain by the first rule and possibly a thunder storm from the second. To find out which was more likely you would have to supply more information.

Programs of this sort are usually referred to as 'knowledge-based expert systems' and they are receiving a great deal of attention from the computer community and the general public at the moment as the best thing that AI has ever produced (without being necessarily complicated).

Rather than continue with theory and explanation it is easier and more instructive to proceed straight to a simple SuperBasic expert system.

All this talk of knowledge, rules and expert systems may seem convincing, but does it work? To demonstrate how powerful the idea is, the program in *Listing 1* will *learn* to become an expert on types of animals. The reason why types of animal has been used is that this particular program has a long history in one form or another and has always been presented with an animal database. However, since the program learns the database rather than having it already built-in you could use it in other areas, such as fault diagnosis, just by changing the first question asked. The reason this is a very simple expert system is that it uses rules of the sort: 'IF list of animal characteristics THEN it is a . . . (name of animal)'

For example, you might have the rule: 'IF it has feathers, is a predator THEN it is an eagle' To make finding the correct rule easy and make the addition and modification of existing rules possible, the rules are represented by a tree. The use of a tree is not essential to the method but it has many practical advantages and seems to fit quite naturally into the program.

To see how a tree can be used to represent a number of rules look at *Fig 1*. Starting from the first question *'Does it have feathers'* you can work your way down the tree answering the questions until you come to an animal's name. Each time you answer a question you take the branch of the tree that corresponds to the answer, that is, the lefthand branch for *'yes'* and righthand branch for *'no'* and ask the next question you meet. For example, if the answer to *'Does it have feathers'* is *'no'* then the next question is *'does it have wool'*. If the answer to this question is *'yes'* then the animal is a sheep. The rules that are contained in this tree are:

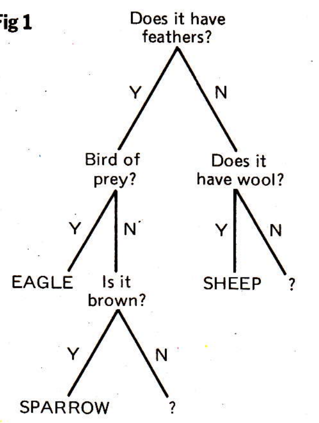
IF feathers, predator THEN eagle

IF feathers, not predator, brown THEN sparrow

IF not feathers, wool THEN sheep

You should be able to see that each of these rules is represented in the tree by the path taken to get to the animal's name. The advantage of storing the rules in this form is that you can match the conditions against the rules one at a time rather than all together. It also provides a way of asking the user to supply information when it is necessary rather than all at once.

Fig 1



Is It A Bird . . .

This is all very well but how do the rules and tree structure get there in the first place? The answer is that every time the *Aardvark* program reaches a '?' in the tree structure it doesn't know what the animal is. To correct this and gain some information it asks the user what the animal is called and for a question that it can ask next time to identify the animal. Once it has this information it inserts in into the tree structure for later use. For example, suppose after asking a number of questions the program finds itself at the '?' following the *'is it brown'* question. It then asks the user *'what is the animal'* to be told that it is a seagull and the question it should ask is *'Is it a seabird?'*. The result is that the new question replaces the question mark and the yes branch of the new part of the tree leads to *'seagull'* and the no branch to yet another question mark.

There is one other way that the *Aardvark* program can learn and that is by getting the answer wrong! If it follows a path down the tree and arrives at the answer 'sparrow' only to be informed by the user that the animal is in fact a *'wren'*, then it can avoid this mistake a second time by asking the user for another question in order to tell the difference between the two birds. For example, if the question is *'Is it the smallest brown British bird'* then the *'wren'* would be on the yes branch and the *'sparrow'* on the no branch. The question itself would of course replace the entry *'sparrow'* in the original tree. In this way the tree 'grows' and modifies itself to reflect what you tell it about animals. You will have to try it for yourselves to see just how quickly it learns.

Integral Arrays

The details of the program are not difficult to understand. The tree structure is represented by two arrays, L% and R%, corresponding to the left and right branch of the tree, following each question. If you reach an element of either the L% or the R% array that contains zero, then you have reached the end of the tree and don't know what the animal is. If you reach an element of either array that contains a negative number you have found a possible candidate for the animal. The names of the animals are stored in the array N$ and the index of the animal that you have found is stored in the L% or R% array as a negative number. So if you find that L%(X%) is negative, the animal's name is in N$(−L%(X%)). If, however, the value stored in R% or L% is positive and not zero, then this is the index of the next question you should ask and the index of the next element of either R% or L% to look at as a result of the answer to the question. Finally, the questions are stored in the array Q$.

The tree starts off with one question and no animal names stored. So Q$(1) contains the question and R%(1) and L%(1) both contain zero (you can think of zero as standing for the question mark in the tree diagram. If the answer to the first question is *'yes'* then R%(1) is examined and as it's zero the program asks for an animal name and a new question. The new name is stored in N$(1) and the question in Q$(2). R%(1) is changed to 2 so that question Q$(2) will be asked following a *'yes'* answer to question 1 and R%(2) and L%(2) are changed so that one of them holds −1 to indicate that the animal's name is in N$(1) while the other is set to zero to indicate that yet another question and name is needed.

Extending Aardvark

This crude program illustrates a method for working out a conclusion from a set of conditions and it learns from mistakes. The way in which it 'grows' the knowledge tree is haphazard but flexible – an animal can appear more than once in the tree to allow for different sets of conditions that *define* it. The biggest problem is that the order in which the questions are asked is governed purely by the order in which they are learned, and this is usually not the best order. If a human was playing Aardvark the first question chosen would be to provide the maximum information. For example, by asking *'does it have feathers'* you can immediately narrow down the rest of the search into one of the two categories – birds or non-birds.

As suggested earlier, by changing the first question in Aardvark to something else like *'Is the fault electrical?'*, the program could be turned into a fault finder. However, naming the fault is only part of the solution. You should also store instructions about what to do along with the name. Some other suggestions for facilities that to Aardvark are:

Add a routine to save and read in an existing tree

Find a way of printing the entire tree for analysis

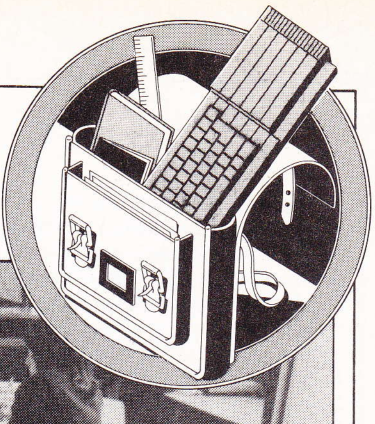
Allow the user to ask for a description of any animal in the tree.

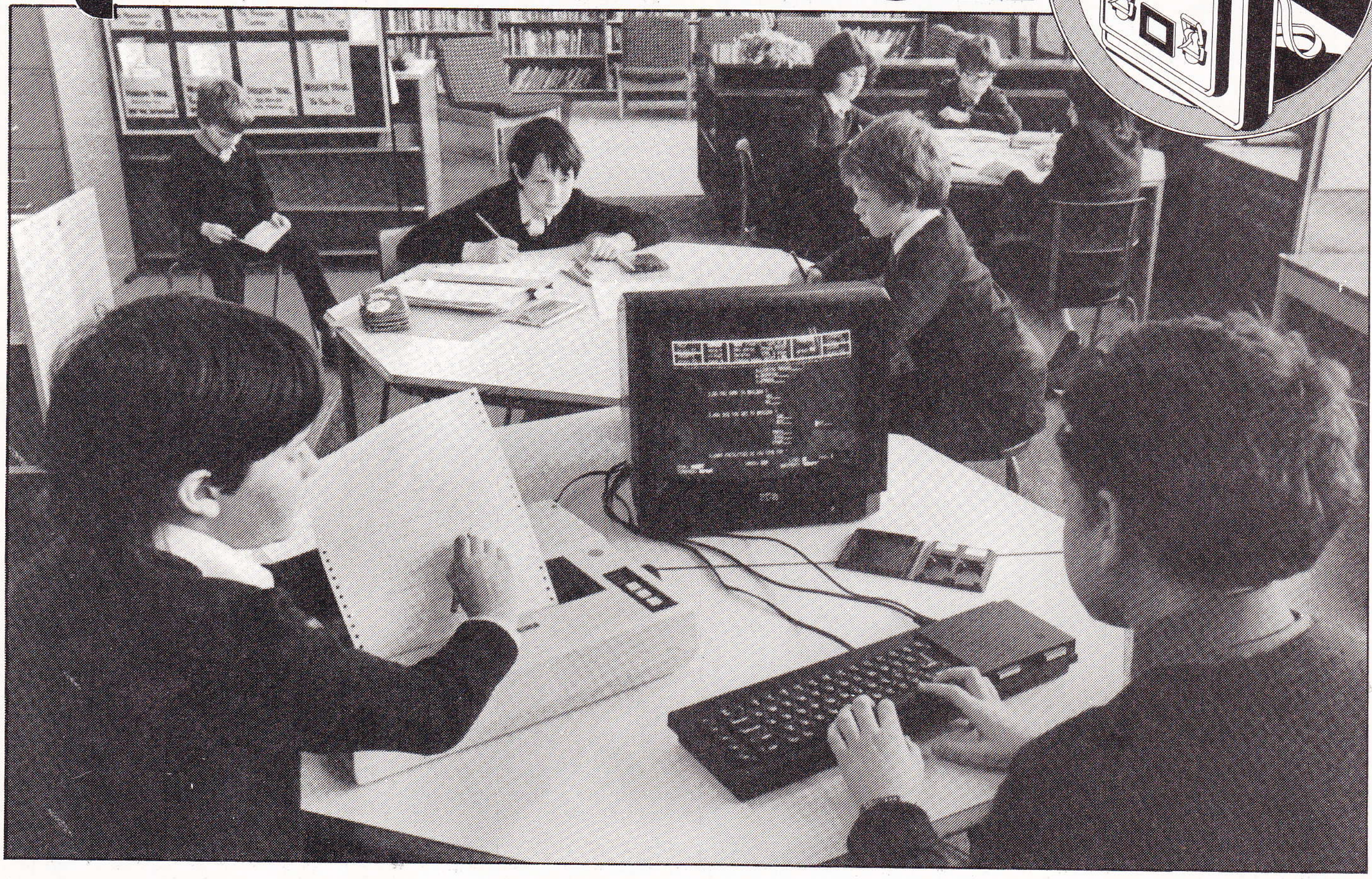
Next Month: Is what we know a *definite* certainty?

LISTING 1

```
10   REMark AARDVARK
20   WINDOW 452,236,60,20
30   MODE 8
40   init
50   title
60   REPeat game
70    question
80    PRINT \
90    INPUT "DO YOU WANT ANOTHER GO Y/N ";AGAIN$
100   IF AGAIN$="N" THEN EXIT game
110   PRINT "THINK OF A NEW ANIMAL"
120  END REPeat game
199  STOP
1000 DEFine PROCedure init
1010 DIM Q$(20,40),R%(20),L%(20),N$(20,30)
1020 Q$(1)="DOES IT HAVE FEATHERS"
1030 D%=1
1040 R%(1)=0
1050 L%(1)=0
1060 N%=0
1070 Q%=1
1499 END DEFine init
1500 DEFine PROCedure title
1510 PRINT \\\
1520 PRINT TO 10;"AARDVARK"
1530 PRINT \\
1540 PRINT "YOU THINK OF AN ANIMAL AND"
1550 PRINT "I WILL GUESS IT -"
1560 PRINT "ANSWER EACH QUESTION WITH"
1570 PRINT "YES OR NO"
1599 END DEFine title
2000 DEFine PROCedure question
2010 X%=1
2020 REPeat query
2030  answer
2040  IF A$="Y" THEN
2050   SORT$="BIRD "
2060   CASE=R%(X%)
2070   SELect ON CASE
2080    ON CASE=0
2090     new_animal
2100     an_am=1
2110    ON CASE=-999999 TO -1E-3
2120     A%=-CASE
2130     guess
2140     an_am=1
2150    ON CASE=REMAINDER
2160     X%=CASE
2170     an_am=0
2180   END SELect
2190  ELSE
2200   SORT$="ANIMAL "
2210   CASE=L%(X%)
2220   SELect ON CASE
2230    ON CASE=0
2240     new_animal
2250     an_am=1
2260    ON CASE=-99999 TO -1E-3
2270     A%=-CASE
2280     guess
2290     an_am=1
2300    ON CASE=REMAINDER
2310     X%=CASE
2320     an_am=0
2330   END SELect
2340  END IF
2350  IF an_am=1 THEN EXIT query
2360 END REPeat query
2999 END DEFine question
3000 DEFine PROCedure answer
3010 REPeat ans
3020  PRINT \\Q$(X%);" ";
3030  INPUT A$
3040  A$=A$(1 TO 1)
3050  IF A$="Y" OR A$="N" THEN EXIT ans
3060  PRINT "I DO NOT UNDERSTAND YOUR ANSWER"
3070  PRINT "PLEASE ANSWER YES OR NO TO MY"
3080  PRINT "QUESTIONS - THANK YOU"
3090  PRINT
3100 END REPeat ans
3999 END DEFine answer
4000 DEFine PROCedure new_animal
4010 PRINT \\"I DO NOT KNOW THE ";SORT$
4020 PRINT "THAT YOU ARE THINKING OF"
4030 INPUT "WHAT IS IT CALLED ";B$
4040 PRINT \\"WHAT EXTRA QUESTION CAN I ASK"
4050 PRINT "TO DISTINGUISH THIS NEW ";SORT$
4055 INPUT C$
4060 N%=N%+1
4070 N$(N%)=B$
4080 Q%=Q%+1
4090 Q$(Q%)=C$
4100 IF A$="Y" THEN
4110  R%(X%)=D%+1
4120 ELSE
4130  L%(X%)=D%+1
4140 END IF
4150 PRINT "WHAT IS THE ANSWER TO"
4160 PRINT C$
4170 PRINT "FOR A ";B$
4180 REPeat ans
4190  INPUT D$
4200  D$=D$(1 TO 1)
4210  IF D$="Y" OR D$="N" THEN EXIT ans
4220  PRINT "ANSWER YES OR NO PLEASE"
4230 END REPeat ans
4240 D%=D%+1
4250 IF D$="Y" THEN
4260  R%(D%)=-N%
4270  L%(D%)=0
4280 ELSE
4290  L%(D%)=-N%
4300  R%(D%)=0
4310 END IF
4999 END DEFine new_animal
5000 DEFine PROCedure guess
5010 PRINT \\"IS IT A ";N$(A%);" ";
5020 REPeat ans
5030  INPUT B$
5040  B$=B$(1 TO 1)
5050  IF B$="Y" OR B$="N" THEN EXIT ans
5060  PRINT "PLEASE ANSWER YES OR NO"
5070 END REPeat ans
5080 IF B$="Y" THEN
5090  PRINT "I THOUGHT SO !!"
5100 ELSE
5110  PRINT "I GIVE UP !"
5120  INPUT "WHAT IS IT ";B$
5130  PRINT "WHAT QUESTION WOULD TELL"
5140  PRINT "THE DIFFERENCE BETWEEN A ";N$(A%)
5150  PRINT "AND YOUR ";B$
5160  INPUT C$
5170  Q%=Q%+1
5180  Q$(Q%)=C$
5190  N%=N%+1
5200  N$(N%)=B$
5210  IF A$="Y" THEN
5220   R%(X%)=Q%
5230  ELSE
5240   L%(X%)=Q%
5250  END IF
5260  D%=D%+1
5270  PRINT "WHAT IS THE ANSWER TO "
5280  PRINT C$
5290  PRINT "FOR A ";B$
5300  REPeat ans
5310   INPUT D$
5320   D$=D$(1 TO 1)
5330   IF D$="Y" OR D$="N" THEN EXIT ans
5340  END REPeat ans
5350  IF D$="Y" THEN
5360   R%(D%)=-N%
5370   L%(D%)=-A%
5380  ELSE
5390   L%(D%)=-N%
5400   R%(D%)=-A%
5410  END IF
5420 END IF
5430 PRINT\\
5999 END DEFine guess
```

QL AT SCHOOL





Peter Williams, Headmaster of a comprehensive school in West Yorkshire, highlights the benefits of using the QL in his school.

A problem facing every school at the moment is how to respond to the challenges and opportunities the technological revolution offers within a budget which is continually being squeezed by current government's restraint on the public sector. Paradoxically the government-sponsored 'Micros in schools' programme ensured that every school in the UK had the opportunity to buy a half-priced computer. The majority of schools plumped for the BBC micro, which at an offer price of under £200, represented excellent value for money. The problem which now faces schools, however, is how to fulfil their obligation to prepare up to 1300 pupils for the 21st century with one computer!

The truth is, of course, that most medium and large schools are finding ways of raising money to buy more machines. But with the Government offer now ended, and prices of machines with specifications equal to, or better than the 'Beeb' falling rapidly, the temptation is to look at some of these newer machines in terms of value-for-money.

There is a great deal of debate within education at present as to just how pupils should be prepared for an increasingly technological society. There is widespread disenchantment with the idea of separate 'computer studies' courses, and many schools which tried this approach are now looking for alternatives which stress the wide-ranging potential of information technology right across the school curriculum, and thus better equip their students for the future.

Most would agree that the school's role is to awaken in all its pupils an interest in, and an understanding of how the computer shapes all our lives. This means that computers should be widely available, *and used,* in a wide range of situations in school. Teachers and pupils should turn to the computer, in appropriate circumstances, just as easily as they use a pencil, piece of chalk, or text book.

To achieve this at my school (Belmont in Baildon, West Yorkshire) we are committed to providing a micro in every classroom – a tall order, and one which would be out of the question using the BBC with its high initial cost and expensive peripherals. The solution we have adopted has been to standardise on the Spectrum with microdrives, which provide a viable low-cost alternative, with the bonus of excellent commercial educational software support.

This is proving to be a good decision, and with the addition of reasonably priced graphics tablets, printers, a turtle and the excellent Sinclair Logo, the system provides a basis for the satisfaction of almost all our needs.

Parallel with these developments, however, has been the establishment of a 'Multi-Media Resource Centre' in the school, based on a large and well-equipped library area. The Centre provides access for pupils and staff to a wide range of information contained in books, on slides and filmstrips, in computer software, on maps charts and pictures, in schools' radio and TV broadcast recordings, on local and national viewdata systems . . . the list is almost endless. In addition, the Centre provides facilities for the printing and copying of materials produced within school – pupils' stories, poems and plays, graphs of survey findings, indexes of collections, letters and reports . . . again, the possible applications seem legion.

We are fortunate to already possess a business system, a DEC Rainbow with 10Mb Winchester hard disk, based in the school office for administrative tasks. In the Resource Centre, the need was for a smaller, less expensive system which was easy to use, and yet which would provide fairly sophisticated word processing, graph-production and database facilities for use by pupils and staff.

Whilst a number of software packages for the Spectrum are very respectable – indeed I regularly use Tasword Two and Masterfile in the production of letters, documents, records and reports – the latest generation of business software leaves these far behind in terms of flexibility, power and ease of use. These are the qualities which we were looking for.

Of all the programs currently available, the four business packages that come free with the QL appeared the most impressive. These seemed to provide just what was wanted – programs which would be easy to use with a minimum of training, and yet capable of development as our needs grew. I have made extensive use of the programs myself, both in school and in connection with my journalistic activities, and as a result the only real reservations I had were the slowness of the programs, and the infamous 'bugs', particularly in Quill.

Word processing seemed a good place to begin. We have set up a system, including computer, monitor and printer in the Resource Centre, to which pupils can bring handwritten drafts of their stories, poems, etc, type them in, print off a draft copy and save it onto microdrive. They can then work on the draft with their teacher, correcting spellings and grammar, deciding on page layout, etc, before returning to the Centre to make amendments.

This is fine, and the 'WYSIWYG' *(what you see is what you get)* operation of Quill lends itself beautifully to on-screen formatting and the production of really creative and attractive page layouts. However, easy though Quill is to use, particularly with the improvements of version 2, there was still a need to train all our 400 or so pupils (and staff) in the basics of its use – a tall order, until we hit on the idea of using a cassette-based audio-training programme as part of the pupils' introduction to the facilities of the Centre.

Using a simple cassette player and headphones, up to four pupils at a time can be talked through the use of Quill in about 40 minutes, covering program loading, creating and editing documents, loading and saving files, designing page and screen layouts, and using a printer. In a five-week period, most of our pupils have used the package and become conversant with the basic features of Quill.

The next step planned is to use Easel, the business graphics package. There are many occasions when pupils want to represent the results of their surveys graphically, and the flexibility and ease of use of Easel makes this an ideal choice for the purpose. Again, an audio-training cassette is being prepared, which could either be used to train pupils for their future use of the program, or enable them to bring a set of figures to the Centre and, without any previous training, produce a graph in one of Easel's standard formats.

Future plans include the creation of a simple-to-use database structure using Archive, and the addition of a modem to give access to local and national viewdata systems. Another possibility is a link with the office Rainbow via a VT100 emulator to provide access to the hard disk store with the QL as a terminal in the planned resource catalogue and information retrieval system.

A great deal of local interest has been shown in these developments, particularly in the audio-training packages, both for educational and business applications. Commercial versions of programmes covering setting up the QL and using the four applications packages are currently in production, and will be on the market in the near future.

PRINTERLAND

ORDERED TODAY – DELIVERED TOMORROW

VISA

Free... WITH ALL PRINTERS: QL USERS PRINTERS GUIDE · Getting the best from your printer is not easy
We supply a **free** booklet with all printers which explains how to obtain all the features available on your printer. Without this you could waste many hours. Buy elsewhere and you'll pay far more and get far less.

■ **DOT MATRIX**	EX VAT	INC VAT
SHINWA CP A80	£185-00	£212-75
EPSON RX80	£185-00	£212-75
EPSON RX80 F/T+ SPECIAL OFFER	£212-00	£243-80
EPSON FX80 F/T+	£314-00	£361-10
EPSON FX100 F/T+	£425-00	£488-75

■ **DOT MATRIX PLUS NEAR LETTER QUALITY**		
*CANON PW1080A SPECIAL OFFER	£245-00	£281-75
KAGA KP810	£245-00	£281-75
KAGA KP910 17″ CARRIAGE	£340-00	£391-00
CANON 1156 17″ CARRIAGE	£340-00	£391-00

■ **DAISY WHEEL**		
QUENDATA 1120	£225-00	£258-75
JUKI 6100	£325-00	£373-75
EPSON DX100 SPECIAL OFFER	£356-00	£409-40

■ **COLOUR PRINTERS**		
EPSON JX-80 SPECIAL OFFER	£450-00	£517-50
CANON PJ1080A	£425-00	£488-75

■ **PRINTER INTERFACES**		
MIRACLE SYSTEMS	£26-05	£29-95

■ **COMPUTERS**	EX VAT	INC VAT
SINCLAIR QL ONLY	£334-00	£384-10

■ **DISC DRIVES**
We stock and use ALL drives and interfaces available including MEDIC, COMPUTAMATE and MICRO PERIPHERALS.
For helpful impartial advice and best prices contact us.

Prices include power supply and interface.

MICRO PERIPHERALS		
3½″ SINGLE DRIVE SYSTEM (.75 MBYTE)	£241-00	£277-15
3½″ DUAL DRIVE SYSTEM (1.5 MBYTE)	£365-00	£419-75
COMPUTAMATE		
5½″ SINGLE DRIVE SYSTEM (.75 MBYTE)	£290-00	£333-50
5½″ DUAL DRIVE SYSTEM (1.5 MBYTE)	£399-00	£458-58
MEDIC		
PRICES ON APPLICATION.		

■ **MONITORS**		
PHILIPS 7502 GREEN	£75-00	£86-25
MICROVITEC CUB 1451/653	£220-00	£253-00

■ **RAM EXPANSION BOARDS**		
SIMPLEX DATA 256K	£165-00	£189-75

COMPLETE PACK SPECIAL OFFER

Package includes: SINCLAIRE QL, MICROVITEC CUB653, PRINTER & ALL LEADS AND INTERFACES

WITH EPSON RX80 F/T	£778-57	£898-86
WITH CANON PW1080A	£814-73	£940-44
WITH CANON PW1156A	£889-31	£1026-21
WITH EPSON DX100 F/T	£920-00	£1058-00

OTHER PRINTERS ON REQUEST

PRINTERLAND Although mainly mail order, our offices are open 9am to 5-30pm Mon-Fri and 9am to 1pm Saturday. Callers welcome for demonstration/collection. DELIVERY-PRINTERS/MONITORS (SECURICOR) £10-00 Other £1-00
Unit 27 Estate Buildings, Railway Street, Huddersfield HD1 1JP. Tel. 514105/512037

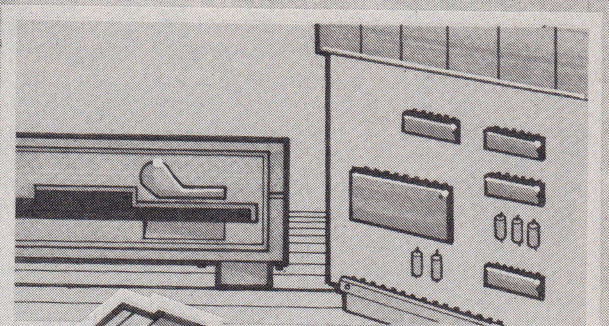
QL Future Growth

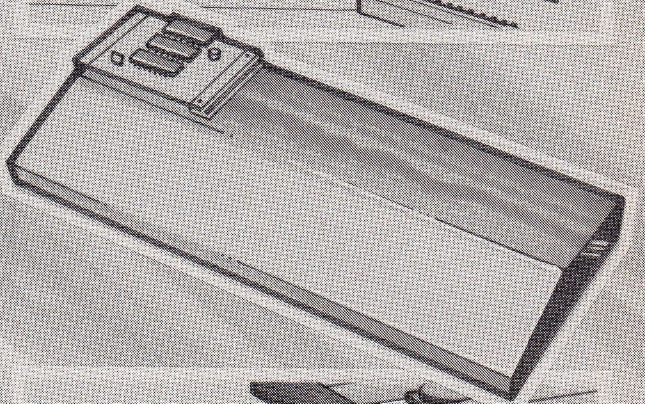
New horizons for your microcomputer from CAMBRIDGE SYSTEMS TECHNOLOGY, the dedicated specialists in expansion peripherals for the Sinclair QL.

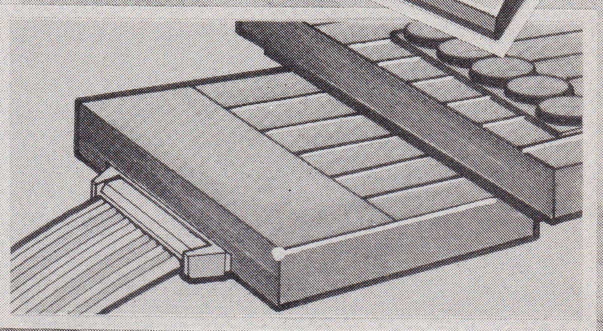
CST who were the first on the market with a disc drive controller, a Centronics port and a fully operational IEEE-488 interface, now offer the Q+4 multi-way expansion module. With four fully-buffered ports, the Q+4 is fully compatible with QL add-ons and features a controller ROM functioning with any version of the QL operating system. Built into a rugged matching case, the Q+4 is designed to sit beneath the computer.

The CST Q-disc is the first controller to allow standard disc drives to be connected to the Sinclair QL, via the QL expansion port. The Q-disc offers extensive file handling and random access facilities plus an essential utility disc and a comprehensive manual.

The Q-488 is a fully implemented IEEE-488 interface which permits the Sinclair QL to communicate with scientific and industrial equipment offering extensive help facilities plus comprehensive error checking.







CST
CAMBRIDGE
SYSTEMS
TECHNOLOGY

30 Regent Street Cambridge
Telephone: Cambridge (0223) 323302

Index Links

J S Ellis

An index for the 'QL User Guide', excluding the Psion packages.

b = Beginners Guide,
k = Keywords,
c = Concepts,
i = Information,
a = Introduction.

ABS	k1
ACOS	k3
ACOT	k3
ADATE	k1
ALPHA COMPARISONS	b47
ARCS	b81
ARC ARC_R	k2
ARRAYS TWO-DIMENSIONAL	b85
ARRAYS	b83 c1
ARRAYS SLICING	b86
ARRAYS STRINGS	b84 c49
ARRAYS AND LOOPS	b34
ARRAY PARAMETERS	b109
ASIN	k3
ATAN	k3
AT	k2
AUTO	k3
BASIC	c2
BAUD	k4
BEEPING	k5
BEEP	k4
BINARY DECISIONS	b91
BLOCK	b77 k5
BORDERS	b17 b77 k6
BREAK	c3
CALL	k6
CHANNELS	b76 c4
CHANNELS AND DEVICES	b76
CHARACTER SET AND KEYS	c5
CHR$	k7
CIRCLES AND ELIPSES	b79
CIRCLE AND CIRCLE_R	k7
CLEAR	k8
CLOCK	c9
CLOSE	k8
CLS	b75 k9
CODE	k9
COERCION	c10
COLOURS	b16 b50 b73 c11
COLOUR PARAMETERS	b75
COMMANDS DIRECT	c17
COMMUNICATION RS232C	c12
CONTINUE	k10
COPY AND COPY_N	k10
COS	k11
COT	k11
CSIZE	b78 k12
CURSOR	k12
DATA	k13
DATA TYPES AND VARIABLES	c14
DATE	k14
DAYS	k14
DECISIONS BINARY	b91
DECISIONS MULTIPLE	b92
DECISION MAKING	b54
DEFINE	k15 k16
DEG	k15
DELETE	k17
DEVICES	c15
DEVICES AND CHANNELS	b76
DIMN	k18
DIM	k17
DIRECT COMMAND	c17
DIR	k19
DIV	k18
DLINE	k19
EDIT	k20
ELSE	k26
END DEFINE	k15 k16
END FOR	k23
END IF	k26
END REPEAT	k49
END SELECT	k53
EOF	k21
ERROR HANDLING	c18
ERROR RECOVERY	c18
EXEC AND EXEC_W	k20
EXIT	k21
EXPORT AND IMPORT	i1
EXPRESSIONS	c20
FILES	b75
FILES CHARACTER	b105
FILES COPY	b106
FILES NUMERIC	b105
FILES READ	b106
FILES SEQUENTIAL DATA	b105
FILES SET UP DATA	b105
FILE TYPES	c22
FILL	b81 k22
FILL$	k22
FLASHING	b75
FLASH	k23
FORMAT	k24
FOR	k23
FUNCTIONS	b98
FUNCTIONS MATHS	c29
FUNCTIONS AND PROCEDURES	c21
FUNCTION	k15
GOSUB	k25
GOTO	k25
GRAPHICS	c23
GRAPHICS SCALE	b78
GRAPHICS TURTLE	c42
IDENTIFIERS VARIABLES & NAMES	b47
IDENTIFIERS	b12 c25
IDENTIFIERS STRINGS	b24
IDENTIFIERS AND VARIABLES	b57
IF	k26
IMPORT AND EXPORT	i1
INKEY$	k28
INK	b75 k27
INPUT	k28
INPUT AND OUTPUT	b52
INSTR	k29
INTRODUCTION	a1
INT	k29
JOYSTICK	c26
KEYROW	k30
KEYS AND CHARACTER SET	c5
KEYWORD	c27
LAYOUT SIMPLE	b73
LBYTES	k31
LEN	k31
LET	k32
LINES AND POINTS	b78
LINES AND RECTANGLES	b51
LINE COLOURED	b16
LINE EDITOR	a10
LINE LINE_R	k32
LINE NUMBER RENUM	b28
LIST	k33
LN	k34
LOAD	k33

LOCAL k34
LOG10 k34
LOGIC AND b63
LOGIC NOT AND BRACKETS b64
LOGIC OR b63
LOGIC PRIORITIES b65
LOGIC WHY b63
LOGIC XOR EXC.OR b65
LOOPS b18 b52 b88
LOOPS NESTED b90
LRUN k35

MATHS FUNCTIONS c29
MEMORY MAP c28
MERGE k36
MICRODRIVES c29 c30a6 a10
MODES b16 k37
MODE RELATIVE b79
MOD k36
MONITOR c31
MOVE k37
MRUN k35
NETWORK c32 a11

NET k38
NEW k38
NEXT k39
NUMERIC EXPRESSION b60
NUMERIC FUNCTIONS b58
NUMERIC OPERATIONS b59

ON..GOSUB k39
ON..GOTO k39
OPEN_IN k40
OPEN_NEW k40
OPEN k40
OPERATORS c33
OVER b78 k40

PAN k41
PAPER b75 k41
PARAMETERS ARRAY b109
PARAMETERS TYPELESS b102
PARAMATERS VALUE b96
PARAMETERS VARIABLE b97
PAUSE k42
PEEK PEEK_W PEEK_L k42
PENUP PENDOWN k43
PERIPHERAL EXPANSION c34
PIXEL COORDINATE SYSTEM c36
PI k43
POINTS AND LINES b78
POINT POINT_R k44
POKE POKE_W POKE_L k44
PRINTERS i2
PRINTING SAMPLE b72
PRINT k45
PROCEDURES b99 k16
PROCEDURES AND FUNCTIONS b96 c21
PROCEDURES AND SUBROUTINES b54
PROCEDURE ANALYSIS b41
PROCEDURE DESIGN b41
PROCEDURE MODULARITY b40
PROCEDURE PASSING INFORMATION b42
PROGRAM c37
PROGRAM STRUCTURE b88

QL PROGRAMS a8
QDOS c38

RS232C COMMUNICATION c12
RAD k45
RANDOMISE k46
RANDOM EFFECTS b17
READ k13
RECOL k47
REMARK k47
RENUM k48
REPEAT k49
REPETITION c41
RESPR k49
RESTORE k13
RETRY k10
RETURN k48
RND k46
ROM CARTRIDGE SLOT c42
RUN k49

SAVE k50
SBYTES k50
SCALE k52
SCREEN b16 b73 c43
SCREEN ORGANISATION b50
SCREEN OUTPUT b72
SCROLLING AND PANNING b82
SCROLL k52
SDATE k51
SELECT b92 k53
SELECT SHORT FORM b94
SETTING UP a3
SEXEC k54
SIN k51
SLICING c44
SORT INSERTION b106
SORT MDV FILE b108
SOUND c46
SQRT k54
START UP c45
STATEMENT c48
STIPPLE CHEQUERBOARD b74
STIPPLE OTHER PATTERNS b74
STOP k55
STRING$ ASSIGNING b67
STRING$ COERCION b68
STRING$ COMPARING b69
STRING$ COPY SLICE b67
STRING$ IDENTIFIER b24
STRING$ JOINING b67
STRINGS LENGTHS b22
STRINGS NAMES ETC b22
STRINGS OTHER FUNCTIONS b69
STRINGS REPL.SLICE b68
STRINGS SEARCHING b69
STRINGS TEXT b67
STRING ARRAYS b84 c49
STRING COMPARISON c50
STRING VARIABLES c49
STRIP b78 k55
SUBROUTINES AND PROCEDURES b54
SUPERBASIC INTRODUCTION b47
SYNTAX DEFINITIONS c51

TAN k56
TECHNIQUES b104
THEN k26
TUNING IN a4
TURN, TURN TO k56
TURTLE GRAPHICS c52

UNDER b78 k57
USING THE QL a5

VARIABLES b57
VARIABLES AND DATATYPES c14
VARIABLES FLOATING POINT b57
VARIABLES INTEGER b58
VARIABLES LOGICAL b61
VARIABLES SCOPE OF b102
VARIABLES STRING b61 c49

WIDTH k57
WINDOWS b77 c53

Available Now For The

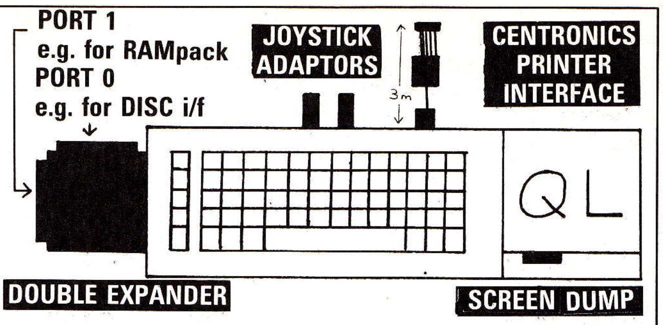
SINCLAIR QL

PURE MACHINE CODE
ARCADE ACTION
ONLY
£10:95

QL is a registered trademark of Sinclair Research Ltd

★ "Easily the best game for the QL to date — and the cheapest" PCW

Make cheques payable to:
WESTWAY LTD, 24 PRESTON ROAD, LYTHAM, LANCASHIRE FY8 5AA



CENTRONICS PRINTER INTERFACE £29.95 inc.
★ 12 month guarantee.
★ Connects QL to any CENTRONICS compatible printer.
★ fully self-contained with 3 metre cable.
★ Plugs directly into SER1 port and directly into printer.
★ Fully compatible with QDOS, SuperBasic, and PSION packages.

SCREEN DUMP £4.99 inc.
★ Copies QL screen image to printer.
★ Supplied on microdrive cartridge.
★ Contains other useful software.

DOUBLE EXPANDER £39.00 inc.
★ Connects up to 2 expansion cards to QL.
★ Ideal for RAMpack and DISC interface combination.

JOYSTICK ADAPTOR £4.99 inc.
★4 Plugs into either CTL1 or CTL2.
★ For use with Atari/Commodore style joysticks.

QL MOUSE — Call for details.

Please contact your computer dealer or send your order with cheque to:
**MIRACLE SYSTEMS LTD
Avondale Workshops
Woodland Way
Kingswood
BRISTOL BS15 1QL
Tel: (0272) 603871 × 210**

Sinclair and QL are Trade Marks of SINCLAIR Research Ltd

Access
ACCESS orders welcome

QL BARGAINS

★ QL COMPUTER – ONLY £379 ★ ALL SYSTEMS READY TO GO ★
★ PRINTERS (INTERFACE/CABLES INCLUDED) ★

BROTHER M1009 — 80 Col. 50 cps. Fully Epson compatable 9 dot. Full range of fonts....................**£199**
Trator feed **£18** Roll holder **£8** Paper rolls **£4**
MANNESMANN TALLY MT80+ — 80 col. 100 cps. Quality print for letters. Friction and tractor....................**£239**
TAXAN KP810 — 80 col. 160 cps (draft). 27 cps (NLQ 23 × 18 dot). Extra NLQ ROMS....................**£319**
DATAPRODUCTS 8010 (PAPERTIGER) — 80 col. 160 cps (draft). 30 cps (NLQ 36 × 9 dot)....................**£459**
QUEN DATA DAISYWHEEL — 18 cps. 13" printline. 10/12/15 cpi sheet/tractor options....................**£289**
Parallel interface.......... **£29** Serial cable..........**£11**

★ DIRECT CONNECT MODEM AT LAST ★
BRIGHTSTAR WITH ALL SOFTWARE FOR PRESTEL AND BULLETIN BOARDS....................**£179**

DISK DRIVES
ALL SYSTEMS 720K FORMATTED
COMPUTAMATE disk interface with extra QDOS commands....................**£149**
COMPUTAMATE single 5.25" or 3.5" drive....................**£340**
COMPUTAMATE dual 5.25" or 3.5" drive....................**£499**
MICROP disk interface with disk utilities....................**£113**
MICROP single 3.5" drive....................**£299**
MICROP dual 3.5" drives....................**£499**
MEDIC....................**Please call**

RAM & DISK EXPANSION
256K RAM .... **£169** **560K RAM**..........**£309**
Q+2 MOTHERBOARD WITH POWER SUPPLY..........**£69**

3250 WATT MAINS FILTER ....**JUST £39**
☆ Provides 4 protected mains sockets with plug
☆ Avoid Crashes and Damage to the QL's Components
☆ 30 dB Suppression 1 MHz to 30 MHz
☆ 130 Joules Spike Suppression
☆ Top quality and robust

SOFTWARE

METACOMCO:	Assembler	£36	BCPL	£54
SINCLAIR:	Cash Trader	£55	Tool Kit	£22
	Touch 'N' Go	£22	Project Planner	£36
	Entrepreneur	£36	Decision Maker	£36
	Integrated Account	£55	Chess (Psion)	£16
TALENT:	Zkul	£16	West	£16
	GraphiQL	£32		
DIGITAL:	Super Sprite Generator	£19	Super Backgammon	£13
	Monitor + Disassembler	£16		
TR COMPUTERS:	Payroll Package	£59	Accounts	£99

PLEASE CALL FOR LATEST SOFTWARE INFORMATION
4 cartridges ....**£8** Transform box for 20 cartridges..........**£6**
10 cartridges + Transform box **£24**
10 5.25" DS/DD disks .... **£23** 10 3.5" DS/DD disks..........**£49**

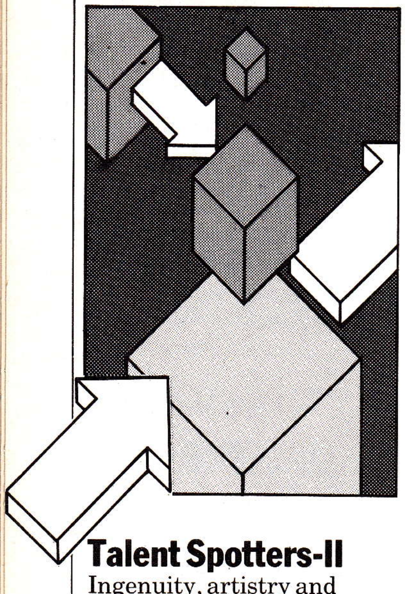
MONITORS (CABLES INCLUDED)
Microvitic 1451/DQ3 colour with swivel stand **£272**
Microvcitec 1451/DQ3 .... **£254** Swivel stand..........**£25**
Philips V7001 18Mhz Green....................**£92**
Philips BM7502 20Mhz Green **£102**
Philips BM 7522 20Mhz Amber **£108**
ALL MONITORS DISPLAY 85 COLOUMNS

★ Prices include VAT & carriage — next day delivery by Securicor from stock ★ BETTER SAFE THAN SORRY ★
★ PERSONAL HELP AT ALL TIMES FROM QL ADDICT ★

STRONG COMPUTER SYSTEMS
Bryn Cottage, Peniel, Carmarthen, Dyfed, SA32 7DJ
Telephone: (0267) 231246 for assistance ! ! !

THE PROGS

Five more winning entries from our Talent Spot competition complete this exhibition and make the QL the art gallery of the future.



Talent Spotters-II

Ingenuity, artistry and economy all combine to produce the winning entries in our Talent Spot Competition. Yet each solution to the problem of devising an attractive graphics screen in no more than 25 lines code is quite unique.

On the programming side, the diversity of methods is astounding. Simple loops range alongside multi-level recursive control structures. Points may be geometrically calculated, plotted at random or even 'hand drawn' from information contained within data statements (plain or encrypted).

On the pictorial side, styles are as varied as those to be found in a gallery of contemporary art. Here minimalist rubs shoulders with surrealist. Childlike *naifs* contrast with symmetric abstracts. In all its enough to make even the most pedogogic art critic give up the Tate and pick up a QL!

Third World

Nick Flowers

All that's missing from this screen to an 'off-world' adventure is the stardate in the captain's log! Space, the final frontier, has been effectively encapsulated within 25 of lines code. Stars appear as randomly generated pin-points against a black background.

```
1 REMark  **** QL User 1985 ****
2 REMark **** 3rd World  by Nick  Flowers
3 FILL 0:MODE 4
4 WINDOW 512,256,0,0:PAPER 0:CLS:SCALE 255,0,0
5 FOR f=0 TO 200:INK RND*7:POINT RND*511,RND*255
6 SCALE 110,-10,100:s:SCALE 512,20,0:s
7 SCALE 256,30,-40:s:FILL 1:INK 183:CSIZE 3,1
8 PRINT ," THE":INK 59:CIRCLE 400,250,100:FILL 0
9 AT 3,20:INK 183:PRINT "PLANET":INK 2:FILL 1
10 AT 1,18:ARC 230,190 TO 230,165,-2 TO 230,140,-2
11 PRINT "RD":LINE TO 190,140 TO 190,150
12 LINE TO 220,150:ARC 220,150 TO 220,160,PI
13 LINE TO 200,160 TO 200,170 TO 220,170
14 ARC  TO 220,180,PI
15 LINE TO 190,180 TO 190,190 TO 230,190:PAUSE
16 DEFine PROCedure s:FILL 1:INK 4
17 CIRCLE 125,124,48:INK 224:POINT 135,76
18 ARC  TO 115,172,-3.1 TO 135,76,-2.1:RETurn
```

Helm's Deep

Chris Bower

A virtuoso programming performance and a fine example of lateral thinking to boot. Encrypting his code and using characters for data the author crams three bits of information into one, so that 75 lines of code now become 25. As if this isn't enough, he departs from the norm, using turtle graphics to draw the detail and a SELECT statement to direct the flow of control within a procedure.

```
101 DATA "ùùcµ!çπ!x*ûφùùù;9!ki","ùùùQùqÅÑ+1  Ä(@",a$
102 a$="ő9+ )Y+1,901; [IA++(3A+9+9éAé+9 ¢":b$="Œ1+ 1;"
103 DATA b$&"133;39+ã+I+9+19A+1I+13131(3++1+1+Q+1A"&c$
104 c$="11w1=+9+Q+9+9+19II3131S1ő+ I+A+q+1+1A31I+90"
105 DATA "ÅùßùL ùQµ$hφ&Di!kk2¢µ$éőπöπ(ô@§&Çb !y«"
106 DATA "KátæMbµDtπ","!dDé804 D!k","é¤B","ÅùL, ù!"&f$
107 DATA "Hè4ë47Å4ä","8è4ü,/è+,«","!fc+K+d*!dÖéµ2 $4V"
108 DATA "ő,","ő31 $Q;≤1;1+1+9AcDùi;YI1ck3k8+9Ső+9"&d$
109 e$="f¤(i!j«":DATA "+333,I","ő8","Å+3T '"
110 d$=" »p:!jµé!fEφ!jEφ 'L!fµ!d%;/i(9¿932A[§ûÄK)!"&e$
111 DATA "ő13 3SCé9+ -é+9 IY9ő31"," ,S=[+"," ,C=K+"
112 DATA "Ç+","i+A+1A+1ăő+9 Aő+1","é1+ 9+;1+1+≤31+9+£"
113 DATA ")!j<Y3I;IIa319+1D9+A+1I+1IQTú+1I31aAD1+1"&g$
114 f$="eµ*!îπ":g$="1<91k+.!fµπ!d9φiiyÄä¿çg≥≥§"
115 DATA "A+9+915+01+19E199]9++x;139+;C315q9Y599Iő"&h$
116 h$="+9 .":DATA "+[A3?1,+ ;0;19033X;38C30w"
117 DATA "D30","Y++-9W","m+1I+@","+1+3-9Q?","C9+++;9"
118 DIM x$(99,99):FOR i=0TO 30:READ z$:x$(i)=z$&"  "
119 MODE 8:OPEN#1,scr_512x256a0x0:CLS:t=0:d 0,0:PAUSE
120 DEFine PROCedure d(x,a):LOCal i,b,e
121 a=a+1:c=CODE(x$(x,a))-34:e=(t+c)MOD 2:SELect ON c
122 =-2:RETurn :=128TO 141:b=a:FOR i=127TO c:a=b:d x,a
123 =1TO 127:MOVE INT(c/8)*SQRT(1+t):TURN c*45:t=e:=-1
124 a=a+1:INK CODE(x$(x,a))+92:=1TO 156:PENDOWN:FILL 1
125 d x+c-141,0:PENUP:END SELect :GO TO 121
126 REMark **** QL User ****
127 REMark **** Helm's Deep by Chris Bower
```

Dr Why

Hugh McGovern

Lines on a video display appear straight only if they are drawn along the vertical, horizontal or diagonal. Otherwise, the computer finds the best fit, lighting some dots and leaving others unlit. With a single line, the result is an unsightly ragged edge. With 3600 lines radiating from the screen's centre it's an intricate and elaborate (moire) pattern. The only problem is that it takes ages to draw.

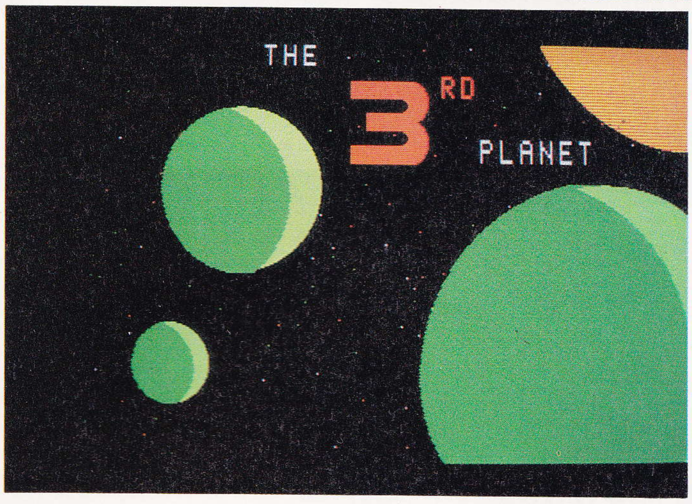
```
100 CLEAR:MODE 4:PAPER 0:OVER-1
110 WINDOW#1,512,256,0,0
120 WINDOW#2,512,256,0,0
130 PAPER#2,0:CLS#2:PAPER#1,0:CLS#1
140 SCALE 100,0,0
150 FOR i=0 TO 360STEP .1
160 x=72+COS(RAD(i))*76
170 y=50+SIN(RAD(i))*55
180 INK 2
190 LINE 72,50 TO x,y
200 NEXT i
210 AT 12,30:CSIZE 3,1:INK 4:PRINT"DR. W H Y."
220 STOP
230 REMark **** QL User 1985 ****
240 REMark **** Dr Why by Hugh Mcgovern
```

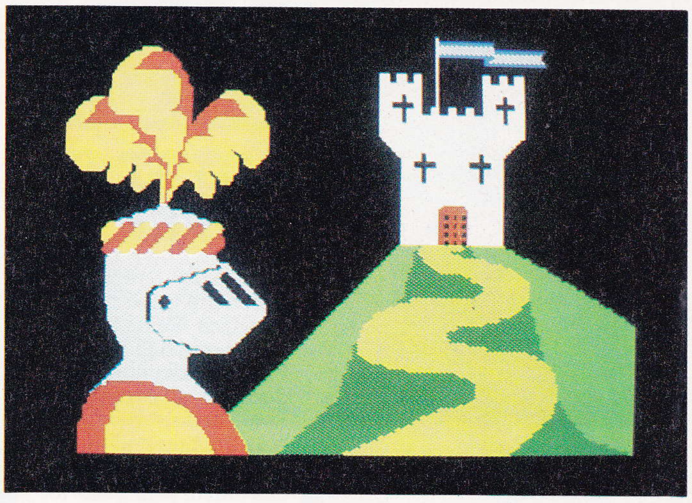
Rainbow Castle

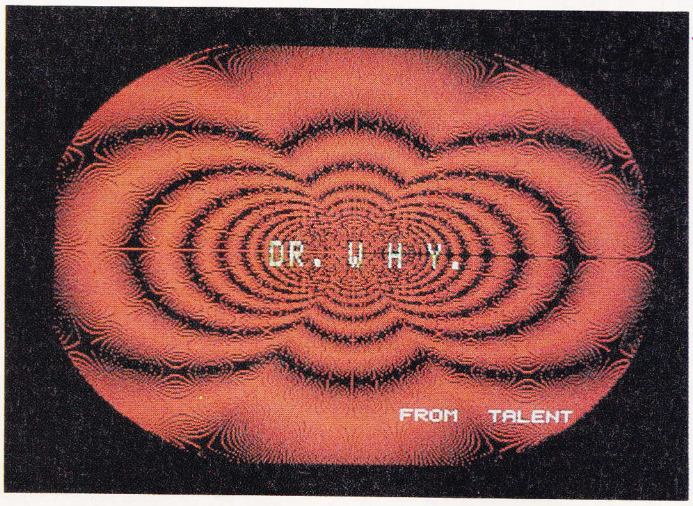
Richard Belsey

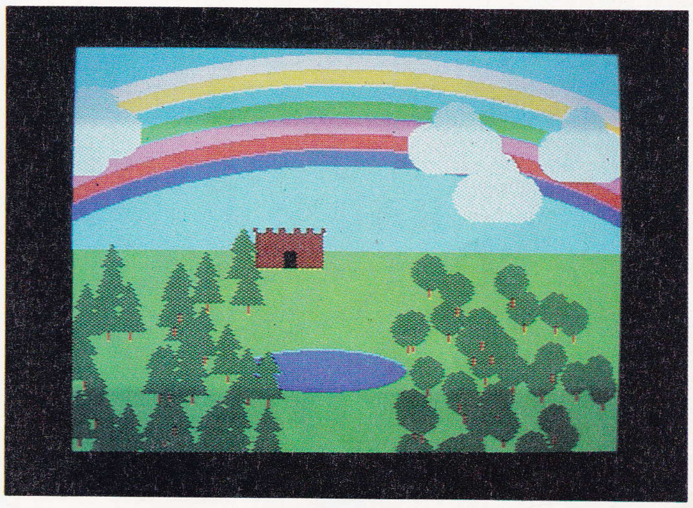
Simplicity is the keyword here. The program uses four FOR loops to build up a quaint fairytale landscape. Careful composition provides a partial perspective. The code should be easily understood as it makes use of conventional graphics commands and avoids any tortuous constructions.

```
1 MODE 8:WINDOW 448,240,32,16:PAPER 5:CLS
2 SCALE 240,0,0:RESTORE :DATA 7,6,5,4,3,2,1,5
3 FOR r=1 TO 8
4 READ i:INK i
5 FILL 1:ARC-50,190-r*10TO 380,190-r*10,-PI/3
6 NEXT r
7 AT 11,0:PAPER 4:CLS 2:INK 7,5
8 FOR c=1 TO 7
```

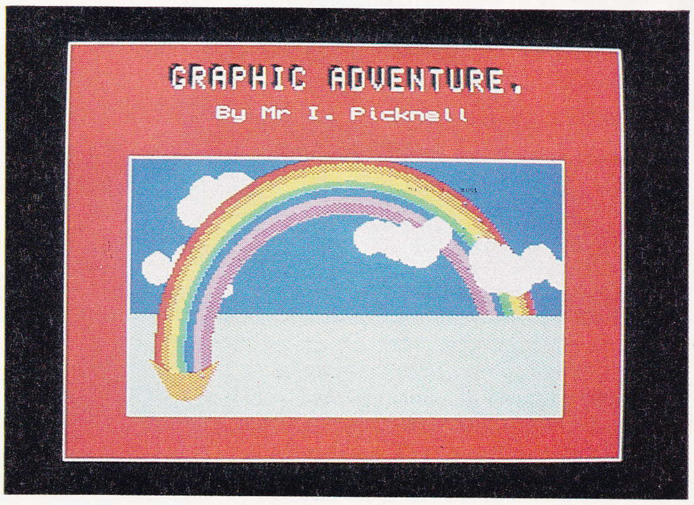








```
9 x=RND(300):y=RND(150TO 210):r=RND(5TO 15):FILL 1
10 CIRCLE x,y,r;x+r*2,y,r:FILL 1:CIRCLE x+r,y+r,r
11 NEXT c
12 INK 1:FILL 1:CIRCLE 150,50,50,.25,PI/2
13 INK 2,0:FILL 1:SCALE 240,-110,-110
14 LINE 0,20TO 0,0TO 40,0TO 40,20TO 0,20
15 FOR b=-2TO 40 STEP 8
16 FILL 1:LINE b,20TO b,23TO b+4,23TO b+4,20
17 NEXT b:INK 0
18 FILL 1:LINE 16,0TO 24,0TO 24,10TO 16,10TO 16,0
19 SCALE 240,0,0:FOR t=1 TO 40
20 LINE RND(-10TO 110),RND(-10TO 110):FILL 1
21 INK 4,0:LINE_R TO 20,0 TO -10,25 TO -10,-25
22 LINE_R 2,15 TO 16,0 TO -8,15 TO -8,-15
23 INK 2:FILL 0:LINE_R 8,-16 TO 0,-5:FILL 1
24 INK 4,0:CIRCLE RND(200TO 320),RND(-10TO 110),10
25 INK 2:FILL 0:LINE_R 0,-10 TO 0,-5:NEXT t:PAUSE
26 REMark **** QLUser 1985 ****
27 REMark **** Rainbow Castle by Richard Belsey
```



Rainbow's End

I Picknell

Adopting a pragmatic approach this entry provides all the elements that a commercial software house might look for in an introductory screen. Distinctive title and author's name ensure that credit goes where credit's due. Additionally, a pot of gold at the end of the rainbow reflects what must be foremost in the mind of anybody releasing competent software for the QL!

```
1 MODE 8:OPEN #3,scr_512x256a0x0:PAPER #3,0:CLS #3
2 OPEN#4,scr_460x250a26x3:PAPER#4,2:CLS#4:OVER#4,1
3 BORDER#4,1,7:INK#4,0:STRIP#4,2:CSIZE#4,3,1
4 z$='GRAPHIC ADVENTURE.':FOR i=1 TO 3
5 CURSOR #4,80+i,10+i : PRINT #4,z$:END FOR i
6 INK#4,7:CURSOR#4,84,14:PRINT#4,z$:CSIZE#4,2,0
7 CURSOR #4,120,40:PRINT#4,'By Mr I. Picknell'
8 PAUSE 50 : OPEN #5,scr_350x150a81x75 : RESTORE
9 RECOL#5,0,0,0,0,0,0,0,0:BORDER#5,1,7:s=75
10 SCALE #5,100,0,0:FILL #5,1:o=174:INK#5,4,7
11 LINE#5,0,0TO o,0TO o,40TO 0,40TO 0,0:INK#5,1
12 LINE#5,0,40TO 174,40TO 174,100TO 0,100TO 0,40
13 Cl 25,55,15:Cl 40,85,25:FOR col=1 TO 7
14 READ c,d:FILL#5,0:INK#5,c,d:FOR g=1 TO 40
15 ARC #5,87-s,10TO 87+s,40,-PI:s=s-8E-2:END FOR g
16 END FOR col:DATA 2,2,2,6,6,6,4,4,1,1,3,3,2,3
17 Cl 158,60,20:Cl 110,70,15
18 DEFine PROCedure Cl(x,y,n)
19 INK#5,7:e=RND(-2 TO 2):FILL#5,1:FOR i=1 TO n
20 CIRCLE#5,x+RND(-15 TO 15),y+RND(-5 TO 5),5+e
21 e=RND(-2 TO 2):FILL#5,1:END FOR i:END DEFine Cl
22 INK#5,6,2:FILL#5,0:a=0:FOR u=22 TO 8 STEP -.5
23 a=a+.25:ARC#5,8+a,u TO 37-a,u ,PI/2:END FOR u
24 INK#5,4,7:FILL#5,1:LINE#5,30,8 TO 37,22 TO 37,8
25 FILL#5,1:LINE#5,8,22 TO 16,8 TO 8,8 TO 8,22
26 REMark ****QL User 1985 ****
27 REMark **** Rainbow's End by I. Picknell
```

New

SINCLAIR **QL** SOFTWARE BY SHADOW GAMES

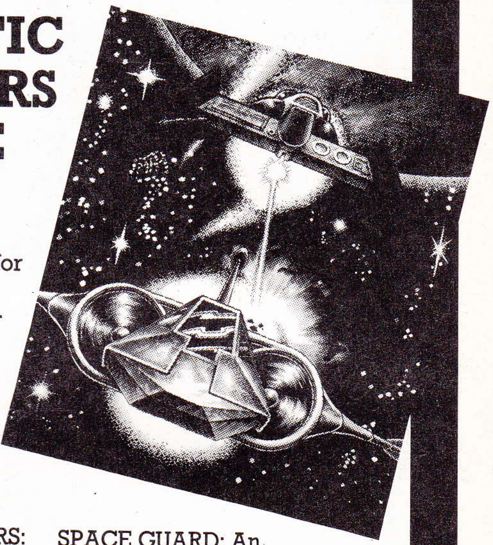
NIGHT NURSE

An arcade adventure with extra large detailed graphics. Guide Nurse Gray around St Spooks Hospital, avoiding randy doctors and Paranoid Pete, the forensic flu bug. Pick up and use whatever objects you can find in your quest for the Golden Prescription. Keyboard or joysticks.

£12.95

GALACTIC INVADERS & SPACE GUARD

Unbeatable value, 2 good arcade games for the price of one. Keyboard or joystick.



£14.95

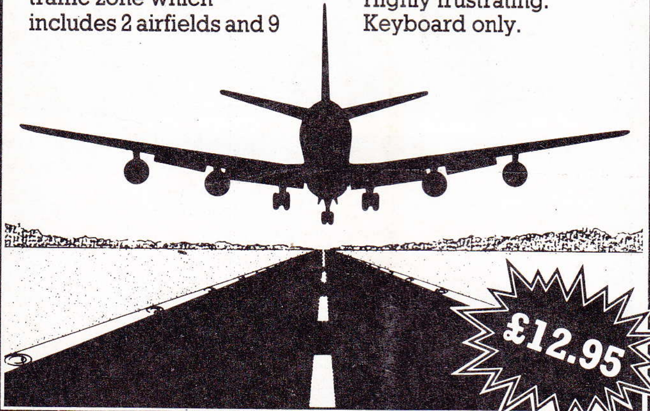
GALACTIC INVADERS: Shoot the aliens with your lazer before they advance down the screen. A very good version of an old favourite.

SPACE GUARD: An alien attack in 3D. Protect your cells from the aliens. Destroy them before they create the deadly pyramids.

TWIN PACK

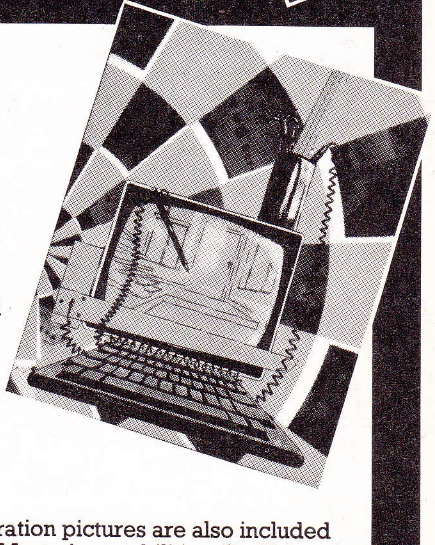
AREA RADAR CONTROLLER

A fascinating game and simulation. Control up to 69 aircraft in a busy air traffic zone which includes 2 airfields and 9 airways. Don't run out of fuel and avoid collisions. Very addictive — Highly frustrating. Keyboard only.



£12.95

PAINT MASTER



A Graphics Package and Mini-Adventure. Keyboard of joystick. This easy to use and well documented collection of programmes will assist you in creating full colour pictures which can be reproduced in your own programmes. Ideal for adding graphics to adventures. A mini text and graphic adventure is included to show how easy it is.

Three detailed demonstration pictures are also included to display some of Paint Master's capabilities. Facilities include:

● Use of all colours, stipples and text ● Automatic drawing of many geometric shapes ● Superimposed grid for guidance ● Powerful duplicate facility which reproduces any object in any size, at any position and as many times as you want ● Variable picture size ● Redraw facility ● Erasure of last command.

£14.95

All prices include VAT, postage and packing

Name ....

Address ....

....

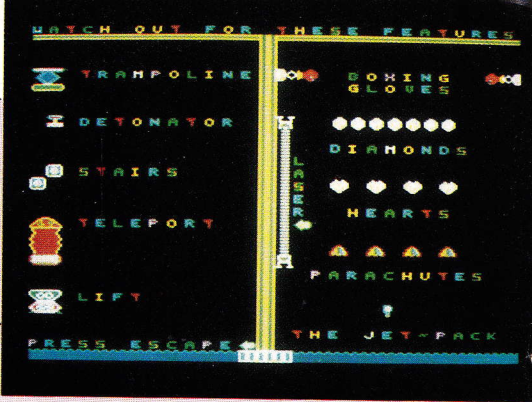
....

Please send me the following products on microdrive cartridge for the QL:

	cost
...... Night Nurse @ **£12.95**	....
...... Galactic Invaders & Space Guard @ **£14.95**	....
...... Area Radar Controller @ **£12.95**	....
...... Paint Master @ **£14.95**	....
TOTAL	....

Please make cheque or postal order payable to Shadow Games and send to:
Shadow Games, 70 Gooseacre, Cheddington, Near Leighton Buzzard, Beds. 0296 668740

QL PROGRAMMERS. Want someone to market your game? — send it to us for assessment; you will not be ripped off. At worst it will be returned uncopied, at best you will be offered a contract at good royalty rates.



QL CAVERNS

It has taken over a year for a full length arcade game to appear on the QL. Ralph Vernon finds out whether the waiting has been worthwhile.

"BJ has been caught on a spy mission by Drunx – the ruler of the 8th stellar system. Now BJ has to find 395 diamonds which are beneath the planet. The journey can be deadly so take care!"

These enigmatic instructions are all you get when it comes to playing the first all-action arcade game for the QL. Who is BJ? What's the Planet? Where on earth is Drunx? None of these questions are ever answered. Indeed, even the name of the game is in doubt for no sooner does one title screen come up introducing the game as *QL Caverns* from QCumber than another takes its place proudly announcing the same game to be *BJ in Space* from Sinclair Research. The fact that all the action takes place underground is, of course, immaterial. So too is the author's

"There are around 50 different locations to visit each affording its own unique screen full of deadly obstacles."

name. Are we really expected to believe that the game was written by a certain Janko Mrsic-Flogel? Or is this yet another elaborate anagram? Only the Evil Crialcnis could tell us and he was nowhere to be found.

Whatever its pedigree, *QL Caverns* is fun. The game may be played using a joystick or the cursor keys. The object is to guide the diminutive figure of BJ through a number of interconnecting underground chambers collecting diamonds as you go. Simple, you may think. Not so, however, as there are around fifty different locations to visit each affording its own unique screen full of deadly obstacles. Sticks of TNT litter the floors, razor sharp stalactites drip from the ceilings, homicidal spiders patrol the walkways and psychotic

"Sticks of TNT litter the floor, razor sharp stalactites drip from the ceilings, homicidal spiders patrol the walkways."

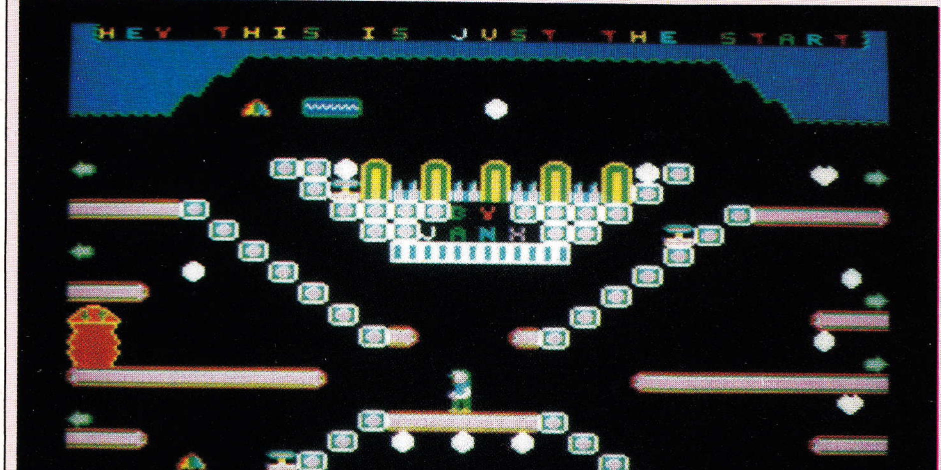
TV sets careen across the skies. All are to be avoided.

As for the diamonds themselves, these are usually located at the most inaccessible points on the screen and much of the game's enjoyment comes from devising a safe route to them. Unaided BJ is capable of crawling, walking or jumping over variety of different coloured terrains. However, more often than not to reach a diamond he will need to climb a staircase, hitch a lift on a raft, jump off a trampoline, put on a jet pack or even teleport to some far flung chamber. Most of these operations require split second timing and good co-ordination. In some instances it is even necessary to sacrifice a life. This may be recovered by collecting a flashing heart. On your journey you can also pick up rainbow coloured parachutes. Whilst these do not register on the scoreboard a constant tally is kept of them alongside the number of diamonds recovered and lives remaining.

Animation in the game is fluid and the graphics make full use of the QL's 8 colour pallet. Furthermore, despite the large number of screens, each one reflects an almost fanatical attention to detail. The sprites though considerably smaller than those used in Psion's chess are attractive and clearly distinguishable. Screen layouts are often planned with a visual quip in mind. In one, for example, the obstacles may be arranged so as to mimic an arcade classic such as QBert, whilst in another they may combine to form a giant QL logo. Whatever the theme, it is never repeated elsewhere.

Flashing arrows strategically located at the extremities of each screen indicate possible routes to adjoining locations. Moving BJ through one of these causes a new screen to instantly replace the existing one. On the whole, this device works well. It preserves the game's continuity and provides a workable alternative to the more elegant method of panning sideways which, with a massive 32K of screen to manipulate and no hardware scrolling on the QL, would have slowed the game to a crawl.

Overall then, *QL Cavern* (alias *BJ in Space)* shows considerable promise. As a game it is easy to play and yet at the same time engrossing and demanding. Furthermore, whilst the 'cavern flight' concept upon which it is based is not entirely original, the program's size (79K of machine code) and its complexity makes it the definitive implementation. A few more games like this and the QL could become the Rolls Royce amongst games playing machines. In the meantime, a game of this quality can be taken as a sure indicator that the 68000 games programmer has, at long last, come of age.



£12.95 – Available W H Smiths and leading computer stores or Sinclair Research direct.

NEW SYSTEMS PACKAGES FOR QL
NOW WITH
QDOS COMPATIBLE INTERFACE & DISK DRIVE

FRUSTRATED? **DON'T BE. OUR SYSTEMS *DO* WORK**

NO INTERFACE PROBLEMS | JUST PLUG IN AND GO! | FACTORY-TESTED BEFORE DESPATCH TO YOU | ALL CABLES SUPPLIED | PRINTERS SET UP READY TO USE | WE EVEN FIT MAINS PLUGS TO EVERYTHING

WE GIVE FULL TECHNICAL ADVICE
& AFTER SALES SUPPORT.
CAN YOU GET THIS KIND OF SERVICE
IN THE HIGH STREET?

NEW — **"Z" GUARD — Tidy up systems & SPIKES in 1 meat main socket block with built in filter and master switch**

1 way — £19.95
4 way — £31.95

PRINTER & MONITOR PACKAGE DEALS + QL OPTION

PRINTER CHOICE (Standard)	**WITH**	**MICROVITEC COLOUR MONITOR**	**OR**	**PHILIPS GREEN/AMBER MONITOR**
Smith Corona Fastext 80		£445		£295
Brother M1009		£445		£295
Epson RX80F/T+		£549		£399
Epson FX80		£725		£590
Epson FX100		£869		£735
Canon PW1080A NLQ		£649		£495
Daisystep 2000		£599		£445
Daisy Junior		£525		£385
Brother HR15		£715		£569
Epson DX100		£745		£590
Epson JX80 COLOUR (Superb with QL)		£895		

Printers are all supplied with RS232C interface. Monitors display full 85 columns of text.

WE MAKE IT ALL COMPATIBLE

'Help Sheet' supplied FREE with all printers.

SINCLAIR QL SUPPLIED WITH ANY SYSTEM — £389

NOW ADD DISK DRIVES!

Drive System including PSU & *Disk Interface*	**Single/Dual**	**Capacity (Formatted)**	**Price**
Computamate			
Teac 5.25" or 3.5" 40/80	Single	720K	
Teac 5.25" or 3.5" 40/80	Dual	2 × 720K	

'Q' Disk Interface alone
Micro P 3.5" 720K 2 × 720K
Micro P interface alone £110
Disks 3.5" box of 10 £52
Disks 5.25" box of 10 £20
Printer cables for QL (Serial)
MEMORY EXPANSIONS
2 and 4 slot expanders 64K, 128K, 256K, 512K memory expansion cards
RING FOR DETAILS
Epson, Brother, Canon, Shinwa etc.

Price
PLEASE
RING
FOR
PACKAGE
DEAL
PRICING
£149
£21

£9.20

MONITOR FOR QL
Factory set for full 85 column display

COLOUR	
Microvitec 1451DQ3	£275
MONOCHROME	
Philips	
BM7502 Green	£92
BM7522 Amber	£95
ACCESSORIES	
Epson RS232C I/F	£30
Epson RS232C 2K	£65
Miracle Systems I/F	£29.95

ALL PRICES INCLUDE VAT
FOR EXPRESS DELIVERY ADD £10 (SYSTEMS/MONITORS/DRIVES)
ACCESSORIES: P&P £2

ZEAL MARKETING LIMITED
VANGUARD TRADING ESTATE, STORFORTH LANE,
CHESTERFIELD, S40 2TZ
Telephone: 0246 208555 Telex: 547697

NEW — FROM TALENT!

GRAPHIQL

Create superb colour pictures on your QL with TALENT's outstanding new graphics package. It's supplied on two microdrives — the first holds the master program and a printer dump utility, the second, three demonstration pictures. Backup copies can be made. GRAPHIQL comes with a detailed, clearly-written instruction manual, outlining the program's many facilities.

Features include:

- Freehand drawing, 8 colours, optional flash
- Rubber banding, rubber boxes, even rubber circles and ellipses
- Variable size texture definition
- Doodle pad
- Colour and texture fill of any shaped area
- User-definable paint brush — any colour or width
- Colour list for full colour control
- Re-colour facility
- Magnification with panning
- Mirroring and rotation of blocks of screen
- Air-brush effect
- On-line 'help' facility
- Full file-store access
- Printer dump utility

Text can be included in pictures. The characters can be single or double height with flash and underline.
GRAPHIQL pictures can be put into BASIC or assembler programs with the sample routines provided.
Available in March by mail order direct from:

£34.95
+ 50p postage & packing
QL is a registered trademark of Sinclair Research

TALENT
COMPUTER SYSTEMS
Curran Building,
101 St. James Road, Glasgow G4 0NS
Tel: 041-552 2128 (24-hour Credit Card Hot-Line)
SOFTWARE FROM SCOTLAND

AD INDEX

Business Micros and Comms Ltd 59
CST Ltd 50
DDL 8
Eidersoft 20
Four Systems 16
Kempston Micros 39
Medic Data Systems Ltd 4
Medic Data Systems 6/7
Metacomco 31
Microdeal ....IFC
Microvitec Plc ....IBC
Miracle Systems Ltd 53
Modem House Ltd 19
Printerland 49
Q-Soft 44
Q-Spell 56
Shadowsoft 56
Silicon Express 32
Simplex Data Ltd 43
Strong Computers Ltd 53
Talent Computers 59
Talent Software 28
Talent Software ....OBC
Tandata Ltd 16
Transform Ltd 28
Viglen Computers Ltd 22
Viglen Computers Ltd 20
Watford Electronics 44
Westway 53
Zeal marketing 58

QL for BUSINESS
OWL COMPUTER TRAINING
A PROGRAMME OF LINKED COURSES.
INTRODUCTORY & ADVANCED APPLICATIONS.
044 282 7302 Freepost, Tring, Herts. HP23 4BR.

DISK DRIVES, PRINTERS & RAM CARDS

DELIVERY AND VAT INCLUDED IN ALL OUR PRICES . . .
NO HIDDEN EXTRA COSTS

NEW . . . MICRO P's MP165 Near letter quality Centronics dot matrix printer
165cps (9 × 9 matrix), fast 75cps NLQ (17 × 17 matrix), friction AND tractor feed, 2k buffer as standard, Epson compatible, bi-directional, plus many other features ....only £285

NEW . . . SHINWA CPA80/P Centronics dot matrix printer
100 characters per second, bi-directional, friction AND tractor feed, 2k buffer as standsard, up to 142 chars per line, Epson compatible, plus many other features ....only £219

NEW . . . MICRO P's DAISY JUNIOR Centronics daisywheel printer
14 characters per second, Qume compatible, friction AND pin feed, bi-directional, 10 inch platen, 7k buffer as standard, multi-copy, auto underline, plus many other features ....only £249

NEW . . . MIRACLE SYSTEMS QL to Centronics printer interface
psion, Qdos, and SuperBasic compatible, self-contained, 3 metre cable, with 12 months guarantee ....only £28.95

SPECIAL OFFERS . . . PRINTER COMPLETE WITH QL TO CENTRONICS INTERFACE
Daisy Junior . . . £269.95 CAP80P . . . £239.95 MP165 . . . £305.95

NEW . . . PCML (QLPLUS) Ram Cards
Boxed, 80% faster than QL's memory, no extra power supply needed, totally compatible with anay QL expansion system.
64k . . . £84.50 128k . . . £124.50 256k . . . £166.50 512k . . . £278.50

NEW . . . MIRACLE SYSTEMS Remodelled double expander
Connects up to 2 expansion units (eg a Ram card AND a disk interface). Please phone or write for further details.

NEW . . . MICRO P's Disk drives developed with Sinclair Research
THE high speed drive and interface for the QL (loads QUILL in 6 secs), 720k — 3.5in single drive, 7 extra commands, single/double sided — 80 tracks, with power supply, includes utilities disk and Rom, microdrive emulator.
Complete with interface . . . £264 Second drive . . . £149 Dual system . . . £409
Box of 10 3.5in DS/DD disks . . . £54
SPECIAL OFFER: INTERFACE + DRIVE 1 + DRIVE 2 + 10 DISKS £458
Box of 10 5.25in DS/DD disks (96TPI) . . . £28

WRITE OR PHONE FOR OUR LATEST SPECIAL OFFER PRICES
PLEASE SEND CHEQUES/MONEY ORDERS TO:
BUSINESS MICROS & COMMUNICATIONS LTD
(MAIL ORDER DEPT)
18 OSWALD ST., MILLFIELD,
SUNDERLAND SR4 6HN.
Tel: (0783) 674944

▼ MICRODRIVE EXCHANGE ▲

Each month, for a trial period, this column will contain details of readers' programs that we are able to offer on microdrive.

In return for a small administration charge (per program – including a royalty for the author), we will copy onto blank microdrives any or all of the featured programs.

Each program will be a direct copy of the published listing, or an extended version of that listing where the program in question was too long to print in full (programs for which an abridged version has been published are marked with an asterisk).

It must be stressed that we are not selling the software itself, nor providing any guarantee that it performs any particular function (though we do check every program that is to appear in *QL User)*, we are merely offering a service to readers who wish to obtain *QL User* programs on drive rather than by typing them in straight from the page.

HOW TO ORDER

Listed below are programs which have appeared as listings inside *QL User*.

To the right of each program entry is a small box, which you should mark with a bold cross if you want to order that program.

Once you have put a cross next to all the programs you wish to have copied onto microdrive, simply complete the rest of the order form and send it along with your PO/cheque AND BLANK FORMATTED DRIVE to:
QL User, MICRODRIVE EXCHANGE, Priory Court,
30-32 Farringdon Lane, EC1R 3AU.

If you wish us to supply the drive, please add an extra £2.50 for every drive required and mark the order form appropriately.

Please allow 28 days for delivery.

ORDER FORM

Author	Language	Program Name	Price	Issue	Size
Giles Todd	(B)	DIY Assembler	£5.00	Jun/Mar	120 ☐
Converts Assembler source into m/c object code					
Richard Cross	(MB)	Function Key Definer	£2.00	May	20 ☐
Programmable function keys just like on the BBC					
Rob Miles	(B)	3Dscapes	£1.00	May	15 ☐
Isometric Perspective representations of 3D Surfaces					
Shergold & Tose	(B)	* Golf	£2.00	May	35 ☐
From fairway to green on 50 different courses of varying difficulty					
Williams & Holliday	(AO)	Paladin	£5.00	Apr	70 ☐
The basis of our games programming series – a space invaders type game written entirely machine code					
Richard Cross	(MB)	Sprite Animation	£2.00	Apr	50 ☐
A subtle blend of machine code and SuperBasic that produces a versatile sprite designer and high speed animator					
Steve Deary	(B)	Pacman	£1.00	Mar	20 ☐
A reasonably fast rendition of the famous arcade favourite					
Adam Denning	(AO)	File Probe	£1.00	Mar	20 ☐
Machine code routine utility to dump the contents of a file to any device in hex. Reveals any hidden control characters for all to see					
QL User	(B)	Tape Utility	£3.00	Feb	20 ☐
File utilities which permit selective back-ups and deletions – in excess of a hundred files may be manipulated					
Andy Carmicheal	(B)	Hi-tech Sort	£1.00	Feb	10 ☐
Highly efficient recursive sort based on C Hoare's famous Quicksort method					
PJ Smith	(B)	* DIY Adventure	£1.00	Feb	60 ☐
A skeleton framework where you simply have to slot in the details to create your bespoke adventure					
Mike Newport	(B)	Psuedo Editor	£1.00	Feb	15 ☐
An ingenious half-way house between a full screen editor and the QL's somewhat primitive line editor					
Adam Denning	(AO)	Multitasking	£2.00	Dec/Jan	60 ☐
Two digital clock programs and one alarm. All will happily multitask on the QL					

B = SuperBasic, AO = Assembler + Object Code (ready to run), MB = Machine Code + Basic loader

Name ______________________

Address ______________________

____

No of programs ordered	....	Total cost	£............
Total sectors (max 200 per drive)	....		
No of drives sent	....		
No of drives required	.... @ £2.50 each		£............
		Plus postage & packing	£ 0.75
		Sub total	£............
		Subscribers reduction *(if applicable) @ –10%	£............
		Add VAT @ 15%	£............
		TOTAL TO BE SENT	£............

Please copy onto microdrive the programs above which I have indicated with a cross. I enclose a cheque/PO to the value of £............ (made payable to *QL User*). I understand that *QL User* only undertakes to SUPPLY these programs (copied onto microdrive) and accepts no liability for their operation as defined by the author. Neither can *QL User* supply additional information about any of the listings other than that originally printed in the magazine (any article reprints required must be ordered and paid for separately at £1 each inclusive of post and packing).

*Subscribers to QL User may deduct 10% off the Sub Total on their orders.

INSTANT ACCESS

Each month this directory is updated with new products and information.

If you or your company are currently manufacturing hardware or supplying QL software and would like to be included within this directory, just send details to 'QL User Reference Chart', Dept SE, QL User, Priory Court, Farringdon Lane, EC1R 3AU.

HARDWARE

Monitors

Citadel Products Ltd
01-951 1848
MBS Data Efficiency
0442 60155
Kaga, Sinclair Vision
Microvitec PLC
0274 390011
Microworld Computer & Video Centre
0273 671863
Microvitec, Philips, Vision
Opus Supplies Ltd
01 701 8668
JVC
Printerland
0484 514105/687875
Strong Computer Systems
0267 231246
Microvitec, Philips
Technomatic Ltd
01 208 1177
Microvitec, Kaga
Viglen Computer Supplies
01 843 9903
Zeal Marketing Ltd
0246 208555
Microvitec, Philips

Printers

Datasystems
01 482 1711
Star
MicroPeripherals
0256 473232
Canon
Microworld
0293 545630/0273 671863
Kaga, Epson, Smith Corona, Shinwa MicroPeripheral, Quendata, Juki
Printerland
0484 514105/687875
Epson, Brother, Kaga, Canon, Juki
Strong Computer Systems
0267 231246
Brother, Shinwa, Epson, Kaga, Mannesman Tally, Canon, Daisystep, Smith Corona
Technomatic Ltd
01-208 1177
Epson, Kaga, Juki, Brother
Twickenham Computer Centre
01-891 4991
Kaga, Ensign, Canon
Viglen Computer Supplies
01 843 9903
Zeal Marketing Ltd
0246 208555
Brother, Epson, Canon, Daisystep

Interfaces

Cambridge Systems Technology
0223 323302
Care Electronics
0923 777155
Computamate Data Products
0782 811711
Miracle Systems Ltd
0272 603871
Printerland
0484 514105/687875
Sigma Research
231 Coldhams Lane, Cambridge
Technology Research Ltd
0784 63547
Transform Ltd
089 283 4783
Zeal Marketing Ltd
0246 208555

Disk Systems

Computamate Data Products
0782 811711
Compware
0270 582301
CST
0223 323302
Medic Data Systems Ltd
0256 460748
MicroPeripherals
0256 473232
Printerland
0484 514105/687875
Quest
04215 66488
Silicon Express
0533 374917
Strong Computer Systems
0267 231246
Zeal Marketing Ltd
0246 208555

Modems

A>Line Computer Systems
0533 778724
Commpak Data
13 Beechwood Road, Uplands, Swansea
Microperipherals
0256 473232
Modem House
0392 69295
Strong Computer Systems
0267 231246
Tandata
06845 68421
(OEL)

Memory Expansion

Eprom Services
0532 667183
Medic Data Systems Ltd
0256 460748
PCML Ltd
0372 67282/68631
QL + RAM cards
Quest
04215 66488

Simplex Data Ltd
01 575 7531

Extras

A>Line Computer Systems
0533 778724
4-way mains filter/adapter

Action Computer Supplies
01 903 3291
Mains spike eliminator

Classified Product & Services
0930 52204
Leads etc.

Computer Supplies
146 Church Rd, Boston, Lincs
Joysticks

Eidersoft
01 478 1291
Quicksoft II Joystick

4 Systems
68 Foxwood Close, Feltham, Middx
Cartridges & box

Management Science Ltd
17 West Hill, London SW18
QL case

Power International
0705 756715
Mains spike eliminator

Sigma Research
231 Coldhams Lane, Cambridge
Joystick

Sinclair Research
0276 685311

Transform Ltd
089 283 4783
QL dust cover, microdrive storage box, RS232 lead

Viglen Computer Supplies
01-843 9903
Printer stand

Voltmace Ltd
Park Drive, Baldock, Herts

Zeal Marketing Ltd
0246 208555
Printer peripherals

SOFTWARE

Utility Programs

Accountancy Software
Sinclair Research
QL Cash Trader

Adder
0223 277050
Q-Doctor, Assembler

Bedsoft
30 Lansdown Rd, Bedford, Beds
Screen Editor

Computer One
0223 86216
Pascal, Forth, Assembler, Typing Tutor, Monitor

Co-op Soft Ltd
0272 22223
Civil/Structural Engineering

DA Bandoo
81 Mount Pleasant, Wembley
Assembler, Screen Editor

Data Management
0904 760351
SBUTIL, Mbackup, Terminal, Chargen, SBextras, FM

Digital Precision
01 527 5493
QL Super Sprite Generator, Games Designer, Monitor + Dissassembler

Flite Software Ltd
010 353 7423023
Equate (maths package)

GST Computer Systems
0954 81991
QL Assembler, 68K/OS

Harcourt
Sinclair Research
QL Touch 'n' Go

Hisoft
0793 26616
MonQL

J&D Software
3 Alfred Rd, Lowton, Warrington

Metacomco
0272 428781
Assembler, BCPL, Lisp

MicroAPL
01 622 0395

Micrologic Consultants Ltd
57 Station Rd

Micro Processor Engineering Ltd
21 Hanley Rd, Southwater, Horsham, E Sussex
QL Terminator Emulator

PCS
8 Oak Grove Way, Bridgewater, Somerset
PCS Utilities

Portfolio Software
PO Box No 15, London SW11

Positron Computing
0554 759624
Hi-res screen dump

Printerland
0484 514105/687875
Metacomco Assembler

Psion
01 723 9408/0553
Quill, Abacus, Easel, Archive

QCode
42 Swinburne Rd, Abingdon, Oxon
Terminal Emulation, 68000 Assembler/Editor

QJump
Sinclair Research
QL Monitor, QL Toolkit

QSoft
01 499 7417
Agenda

Quantum Mechanics
5 Twineham Green, London N12
Qspell

Quest
04215 66488
Business Accounts

Saltgrade Software
31 Royal Terrace, Edinburgh EH7
File Manager, File Editor

Sigma Research
231 Coldhams Lane, Cambridge
Sketchpad

Strong Computer Systems
0267 231246

Super Plant Software
097 231246
Plant & gardening software

TR Computer Systems
093 924 621
QL Payroll

TDI Software Ltd
0272 742796
USCD Pascal, USCD Fortran 77, Advanced Development Toolkit, USCD P System, USCD Prolog

TR Computer Systems
093 824 621
QL Payroll

Triptych
Sinclair Research
QL Decision Maker, QL Entrepreneur, QL Project Planner

WD Software
0534 81392

Games

Bedsoft
30 Lansdown Rd, Bedford, Beds
Gambler, Beat the Clock

Blain Software
8 Berkeley Close, Staines, Middx
Merry Muncher, Fire Tower, Advance Invaders

Brainstorm
4 Lindsey Close, Cramlington
Westminster Palace

CP Software
10 Alexandra Rd, Harrogate
Bridge Player

Digital Precision
01 527 5493
QL Super Backgammon

Eidersoft
01 478 1291
QL Art

Equate
2 Ffordd Denwyn, Penyffordd, Chester
Solar Invaders, Wall Breaker, Draughts, Mind Your Path, Statistical Averages, Calendar

Games Workshop
01 965 3713
D-Day

Intersoft
7 Richmond Rd, Exeter, Devon
Executive Adventure

New Horizon Software
Fourwinds, Cwn Lane, Rogerstone, Gwent
Pacman, QBERT, Gold

Peak Electronics
32 Clifton Ave, Hartlepool, Cleveland
QL Colour Quest

Printerland
0484 513105/687875
Psion Chess

Psion
01 723 9408
Psion Chess

Rodent Software
3 Brookend Cres, Henley-in-Arden, W Mids
Adventure Writer, QL Artist, 2 × 7 Games cartridges

S&B Software
20 St Nicholas St, Diss, Norfold
Fantasia Adventure

Shadowsoft
0296 669740
Area Radar Controller, Strategy

Snowsoft
6 Bousefield Cres, Newton Aycliffe, Durham
Hungry Harry in the Haunted House

Sinclair Research
0276 685311
Psion Chess

Summit Software
36 Wood Cres, Rogerstone, Newport, Gwent
Frogger, Dungeon

Swansoft
164 Vicarage Rd, Morriston, Swansea
Space Trek

Talent Computer Systems
041 552 2128
ZKUL, WEST, GraphiQL

WD Software
Hilltop, St Marys, Jersey
WD Morse Tutor

BOOKS

Publishers

Adder
0223 277050

Century
01 434 4241

Collins
01 493 7070

Duckworth
01 485 3484

Ellis Horwood Ltd
0284 789942

Granada
01 493 7070

Hutchinson
01 387 2811

Interface
9-11 Kensington High St, London W8

McGraw Hill
0628 23431

MicroPress
0892 39606

Melbourne House
01 940 6064

Prentice Hall
0442 58531

Sunshine
01 437 4343

Book Titles

Quill, Easel, Archive and Abacus on the Sinclair QL
McCallum Varey
(*Sunshine* £6.95)

Introduction to Simulation Techniques on the Sinclair QL
Cochrane
(*Sunshine* £6.95)

Mathematics on the Sinclair QL
Kosniowski
(*Sunshine* £6.95)

Programming in C
Traister
(*Prentice/Hall* £19.50)

Getting To Know Your Sinclair QL
Morris
(*Duckworth* £7.95)

The C Programming Language
Kernighan & Ritchie
(*Prentice/Hall* £22.95)

Learning to Program in C
Plum
(*Prentice/Hall* £15.95)

Logic, Algebra & Databases
Gray
(*Ellis Horwood* £9.95 p/back, £22.95 h/back)

68000 Assembly Language Programming
Kane, Hawkins, Levanthal
(*Osborne/McGraw Hill* £19.50)

QL Gamesmaster
Ewbank, James & Gee
(Collins £7.95)

Get More from the Epson Printer
Curran
(*Collins* £7.95)

A QL Compendium
Gandoff & Kinge
(*Addison Wesley* £7.95)

QL Handbook
Hartnell
(*Interface* £7.95)

Professional & Business Uses of the QL
Lewis
(*Collins* £7.95)

QL Abacus
Spottiswoode
(*Century* £8.95)

BASIC Programming on the QL
Cryer & Cryer
(*Prentice Hall* £7.95)

Quick QL Machine Language
Giles
(*Melbourne House* £7.95)

Now you can create new dimensions in Computer Graphics.

Create superb colour pictures on your QL with TALENT's outstanding new graphics package. It's supplied on two microdrives – the first holds the master program and a printer dump utility, the second, three demonstration pictures. Backup copies can be made. GRAPHIQL comes with a detailed, clearly-written 60 page instruction manual, outlining the program's many facilities.

• Freehand drawing, 8 colours, optional flash • Rubber banding, rubber boxes, even rubber circles and ellipses • Variable size texture definition • Doodle pad • Colour and texture fill of any shaped area • User definable paint brush – any colour or width • Colour list for full control • Re-colour facility • Magnification with panning • Mirroring and rotation of blocks of screen • Air-brush effect • On-line 'help' facility • Full file-store access • Printer dump utility.

Text can be included in pictures. The characters can be single or double height with flash and underline. GRAPHIQL pictures can be put into BASIC or assembler programs with the sample routines provided.

Available from selected branches of Boots & WH Smiths.
£34.95 + 50p postage & packing

TALENT
COMPUTER SYSTEMS

Curran Building, 101 St James Road, Glasgow G4 0NS. Tel: 041-552 2128 (24 hour credit card hot-line) Software from Scotland

QL and Microdrive are registered trade marks of Sinclair Research Ltd.