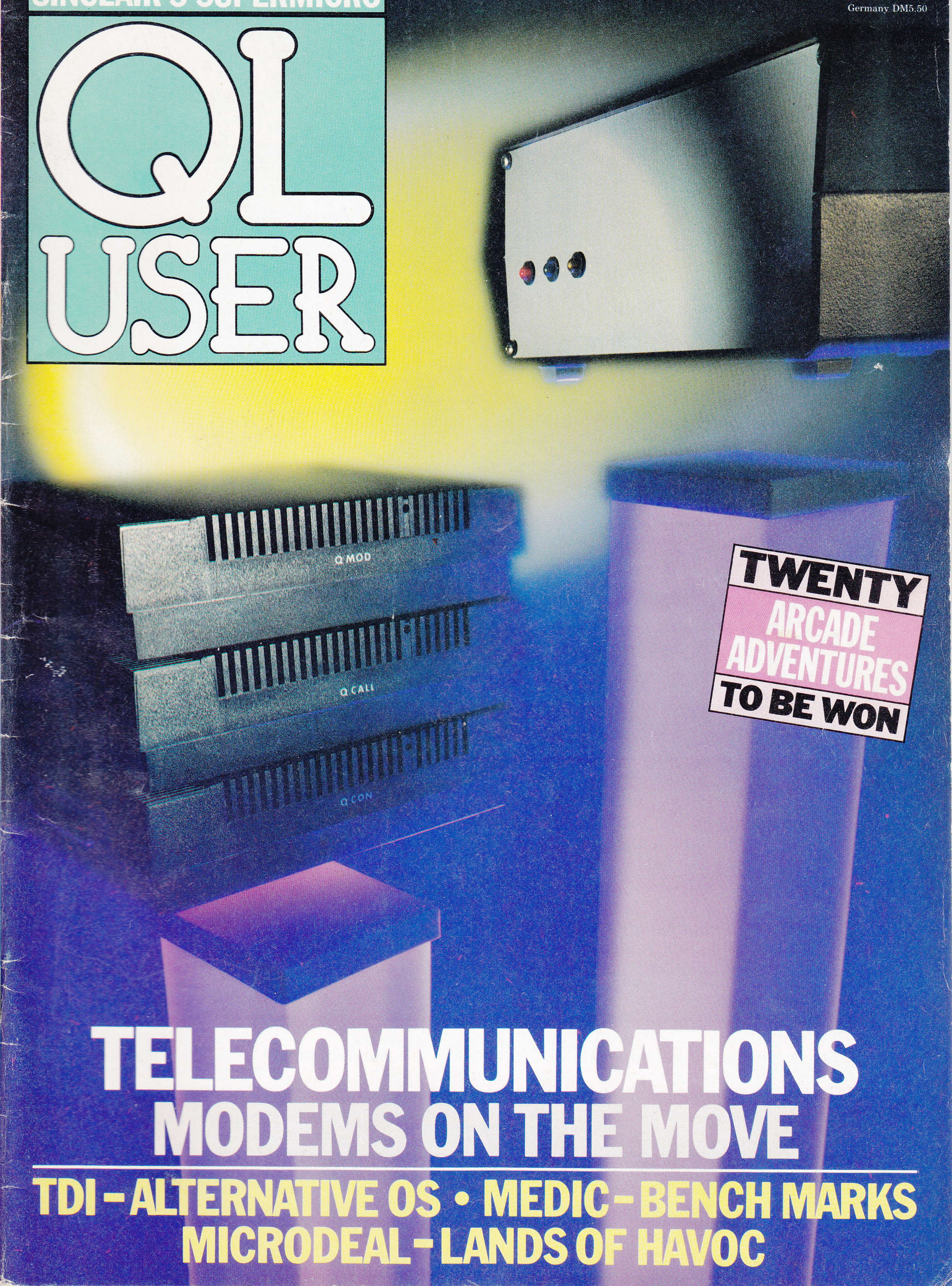
An EMAP Publication

SINCLAIR'S SUPERMICRO

QL USER

Every Month 95p August 1985

Germany DM5.50



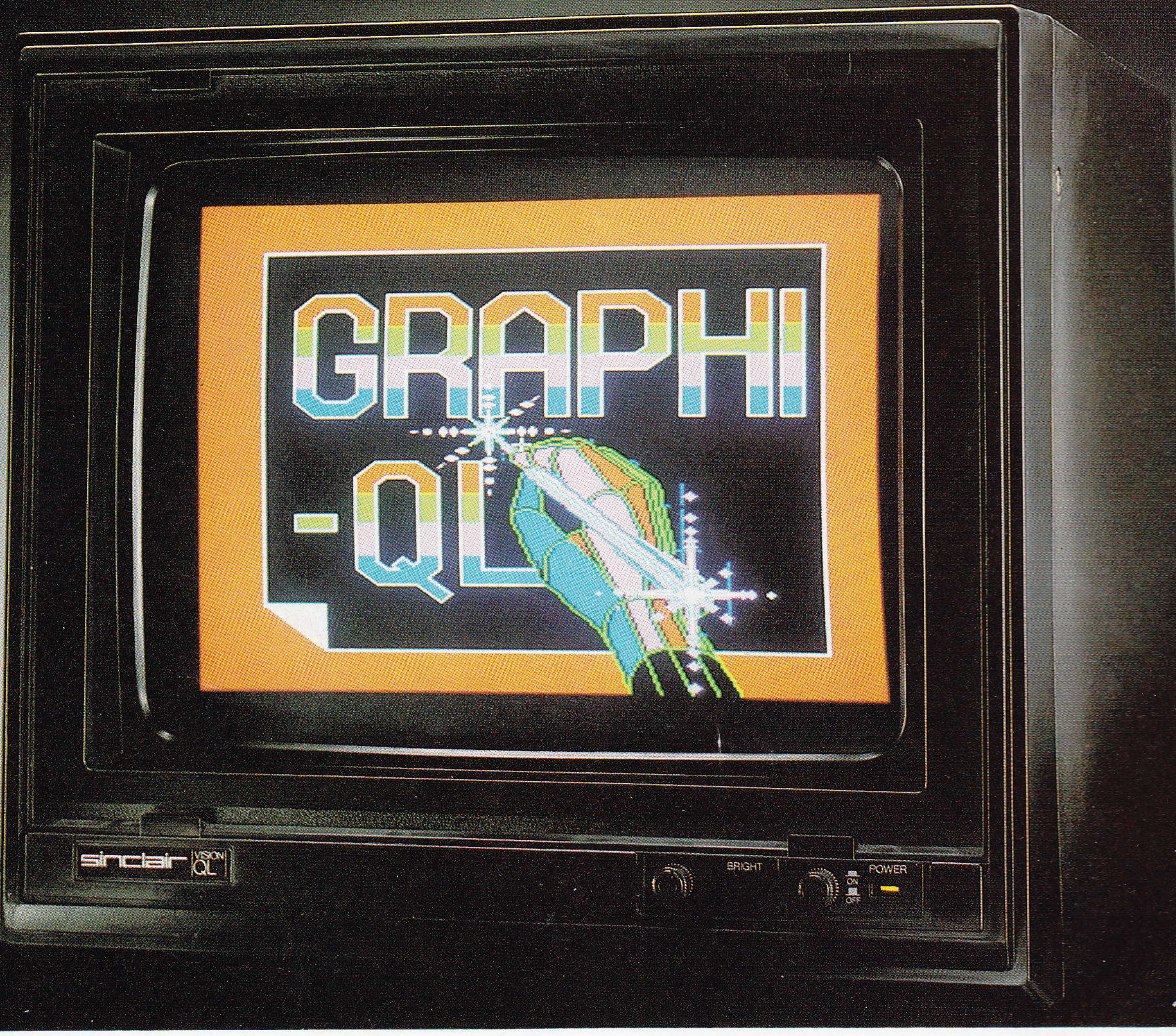
TWENTY ARCADE ADVENTURES TO BE WON

TELECOMMUNICATIONS MODEMS ON THE MOVE

TDI - ALTERNATIVE OS • MEDIC - BENCH MARKS

MICRODEAL - LANDS OF HAVOC

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.

QL USER

August 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

Cover photography Terry Beddis

Contributors

Ivor Kent, R K Lowrie, Andy Carmichael, Nicky Trevett, Adam Denning,. Gary Evans, Q, S Ackers, Roger Vernon, Ian Stewart, Marcus Jeffery, Richard Green, Andy McIlroy, Mary Sargent, Mike James, Irwin Joffe, James Lucy, David Green, Simon Craven and Mathew Cappe.

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

Telephone 01-251 6222

Readers may have been pleased to find a partial index for the QL manual in last month's edition, compiled by one J S Ellis. We were fairly pleased with it too and would like to show our appreciation with the customary contributor's payment. Unfortunately, somebody walked off with the address and we would therefore ask Mr Ellis to telephone these offices.

Readers following our Games Programming series may NOT have been too pleased about being unable to read line 62 of the program. The line in question should have read *006A 5888 62 addq.l #4,a0* and hopefully Paladin-coders can now zap away to their heart's content.

Microdrive Exchange

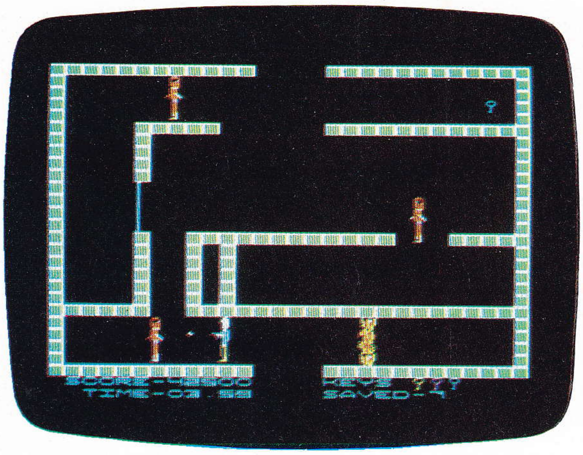
Due no doubt to their excitement over our Microdrive Exchange service, a few readers have placed orders without sending us their addresses. If they could rectify this situation we'll process their orders in the usual way. They are: R L Wells (Kingston Upon Thames), J W P Hazell (London W1), P M Sander (Wanstead), R Faulkner (Edinburgh) and P C Collins (Uxbridge).

CONTENTS

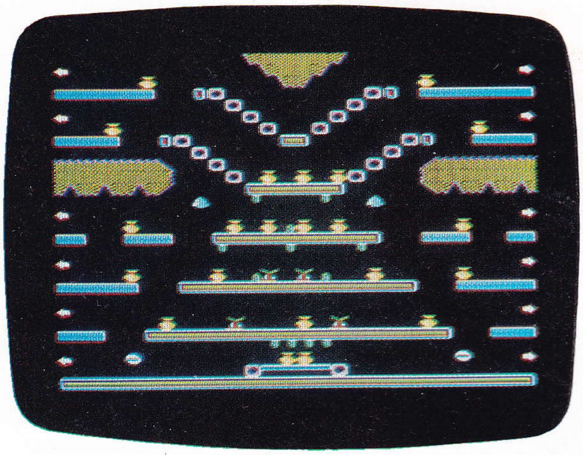
Published and Distributed by EMAP
U.S. Sales Agent: 'Motorsport', RR1, Box 200D,
Jonesburg, MO, 63351.
Typesetting by Crawley Composition Ltd.,
Stephenson Way, Three Bridges, Crawley, Sussex.
Printing Riverside Press.

FREE!

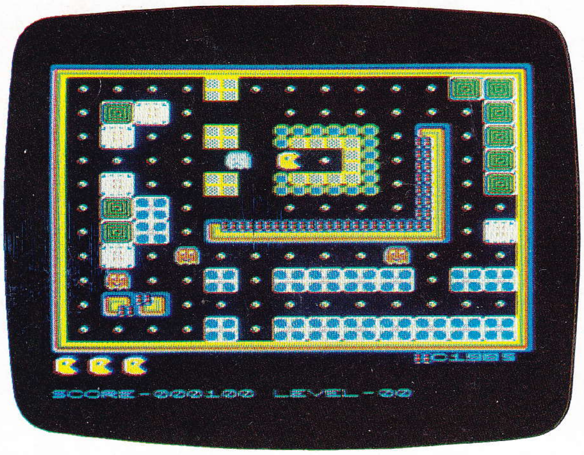
Four free games for every QL owner!



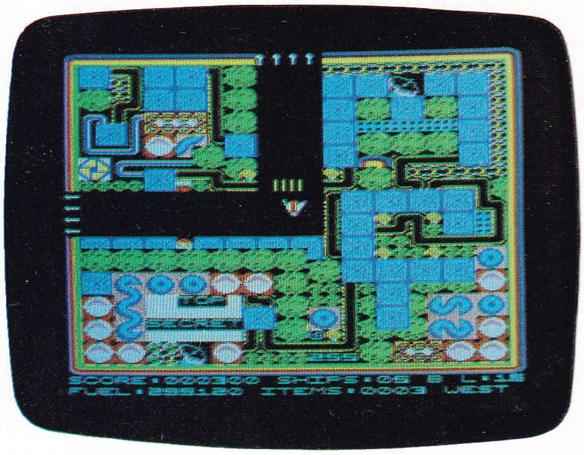
M – COSMIC. Your job is to save your friends, deep-frozen in an enemy space-ship. A mission of cool nerve and firm courage.



M – TREASURE. Deep underground there are **bags of gold** for you to collect – and man-**eating spiders** to stop and slay you.



M – CRUNCHER. Stay clear of the jaws that chase you through the maze, avoiding hazards and building points as you go.



M – METROPOLIS. Fast-moving excitement as you steer through screen after screen of incredible dangers and rich rewards.

MEDIC now offer every Sinclair QL owner four brilliant new games – absolutely free! Just send two formatted blank microdrive cartridges for these four best-ever games for the QL, testing your skill, speed and concentration with colourful screens and inventive displays.

All games 100% machine code, with flicker-free multi-colour multi-screen graphics. Two full cartridges of fast-moving instantly-responsive action.

WHY? Medic offer you this fantastic games value FREE, because they want you to see the quality of Medic software, and as their way of showing their committment to this superb machine.

Fill in and post the coupon TODAY to make sure of your MEDIC free games.

To: Medic Datasystems Limited,
76 Grainger Close, Basingstoke,
Hants RG22 4EA.

Please send me my four free MEDIC games. I enclose two blank formatted cartridges, plus £2.50 to cover postage, packing and recording. (Cheque or P.O.).

Name______________________________

Address______________________________

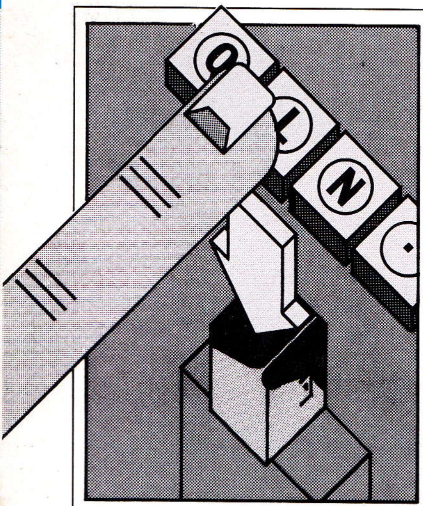
____

Telephone Code______Number____________

Please complete all details and write clearly.

NEWS

Our regular roundup on the QL and associated products



IEEE-488 Bus Arrives

Cambridge Systems Technology are the first to supply an IEEE-488 bus for the QL. The bus permits very high speed communications between computers and peripherals and is commonly used to link up laboratory equipment.

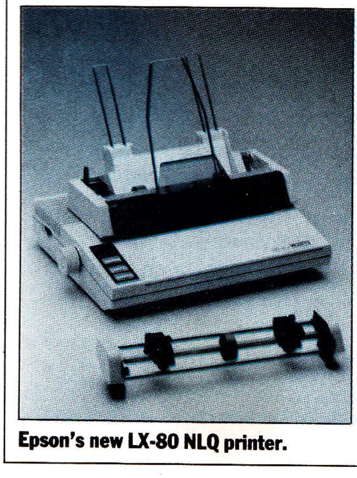
The interface conforms with GPIB and HPIB standards and consists of a single unit with onboard ROM. The firmware includes a bus analyser to help with debugging 'awkward' devices. Access to IEEE-48 facilities may be through SuperBasic and other high-level languages, as well as low-level assembler.

The Q-488 interface costs £195.50 including VAT.

NLQ Deluge

Epson have acquired a reputation for producing robust and durable printers. However without facilities to produce near-letter-quality print (NLQ), their popular RX and FX ranges now appear long in the tooth. Their new LX-80 100 cps 9 pin dot matrix should therefore be extremely welcome. The printer incorporates an easy "selectype" font selection to give emphasised, double strike, condensed and elite styles in regular and NLQ (16 cps) fonts. The unit costs £293 (including VAT) and optional extras include tractor feed at £20 and cut sheet feeder at £63.

In direct competition, German printer manufacturers Juki have released their new 5500 NLQ range. With a similar specification but only faster (180 cps draft and 30 cps NLQ) and with a built-in tractor feed, the units retail at £349 (black and white) and £449 (colour).



Epson's new LX-80 NLQ printer.

BSI Approved

Metacomco's ISO Pascal has been validated by the British Standards Institution as fully conforming to ISO 7185 – the international standard for Pascal. This is a first for a 68000-based compiler and should ensure that Pascal programs will run on many different machines, provided, of course, that the standard has been implemented on them.

Maxwell Steps In

Estimates last year would have valued Sinclair Research somewhere between £80 and £120 million. Robert Maxwell's 75% controlling interest bought for a paltry £12 million shows how far the company has fallen since then.

Some of Sinclair's problems can be laid at their own door. The runaway success of the Spectrum led management to believe that any Sinclair product would sell without the need for a coherent marketing strategy and adequate quality control. The QL launch is a case in point. Delays, dongles and bugs in the software undermined confidence and alienated press and public alike. A year's head start on *all* the competition was frittered away on patches and upgrades.

Complacency alone, however, does not explain Sinclair Research's collapse. A 68000 machine for under £1000 should be a winner. A view that Commodore, with the Amiga and Atari with the ST, both share. However, not only do both these companies have access to a far larger and richer market but the funding that both enjoy reflects a positive attitude to new technology.

On this side of the Atlantic little other than lip service has been paid to the new technology and it's doubtful that Mr Maxwell's initiative reflects a general shift in attitudes. However, the acquisition of Sinclair Research brings him a step closer to his stated aim of establishing a worldwide communications company before the decade is out and, provided that he and Sir Clive see eye to eye, may well provide Sinclair products with sufficient impetus to carry them through an uncertain market.

Sir Clive and the elusive WSI.

Wafer Breakthrough

Research into wafer scale integrated circuits at Sinclair's Metalab would appear to be proceeding apace. Tests indicate that volume manufacture is possible. As previous attempts at WSI have floundered with wafers being either too costly to produce or yielding too few working chips, this represents a fundamental breakthrough.

According to Sir Clive WSI "represents the next logical development in the manufacture of increasingly complex electronic components. It could prove comparable in significance to the invention of radio valves, transistors and, most recently integrated circuit semiconductor chips."

With WSI, a silicon wafer will replace the relatively large PCB's where chips have to be individually mounted, wired and packaged in plastic. The wafer will comprise of many hundreds of chips. Some will be faulty, others will not. Using a technique invented by the British scientist (Ivor Catt, a member of the Metalab team), switching circuits on the wafer itself will identify and isolate faulty chips.

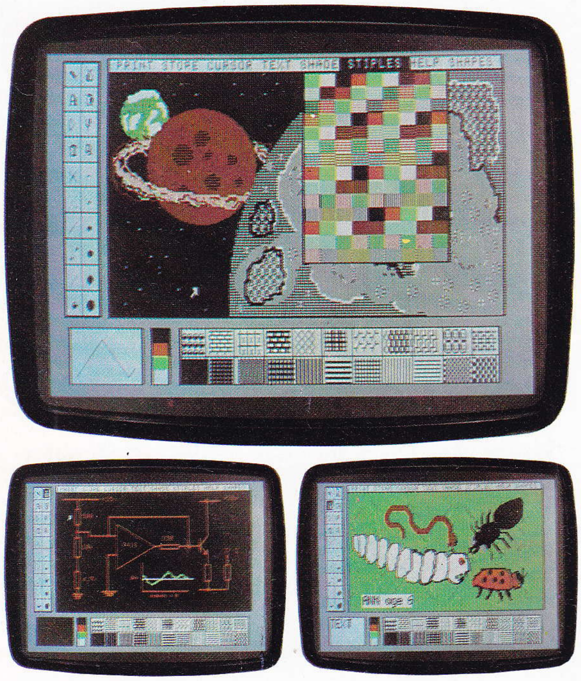
The first WSI product will be a 500K memory/mass storage device for the QL.

All in One System

Technology Research, the latest entrant in the QL peripheral market, launched their Delta Disk interface at the recent ZX Microfair, June 22. The company have opted for an all in one configuration similar to that adopted by Medic Datasystems. The interface will support up to four Shugart compatible disk drives (5.25″, 3.5″, 3″), has a built in centronics printer interface and may be upgraded to contain additional RAM. Prices extend from £130 for the unexpanded version to £249.50 for the top of the range 128K version.

The MEDIC QL E

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
£30.00 inc. VAT (Plus £1 p&p)

(Please write clearly)

To: Medic Datasystems Limited,
76 Grainger Close, Basingstoke, Hants RG22 4EA

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).

NAME . . . .

ADDRESS . . . .

. . . .

TELEPHONE Code. . . . . No . . . .

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.

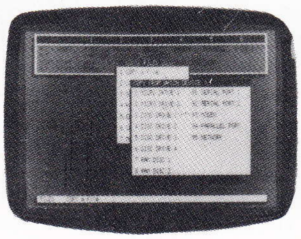
Extra memory, for speed and easier program and file handling, starts at £99.95. Complete systems, consisting of 1Mb disk drive, disk interface, parallel port, integral power supply and all cables, from £249.95. Even higher performance can be achieved by adding extra MEDIC memory to your disk system, 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.

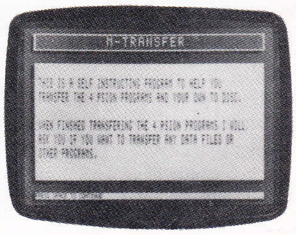
MEDIC – the reliable QL expansion system that *guarantees* you more than any other system.

FREE SOFTWARE

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



M-DESK

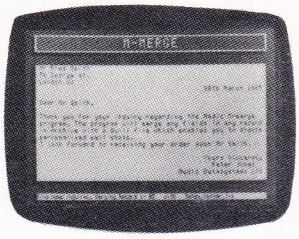


THIS IS A SELF INSTRUCTING PROGRAM TO HELP YOU TRANSFER THE 4 PSION PROGRAMS AND YOUR OWN TO DISC.

WHEN FINISHED TRANSFERING THE 4 PSION PROGRAMS I WILL ASK YOU IF YOU WANT TO TRANSFER ANY DATA FILES OR OTHER PROGRAMS.

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.

FREE – microdrive dust covers.
Simply fill in and post the coupon ↓

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

QL User Reviews the Medic System!

This issue Pages 20/21

MEDIC

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

System now available from stock for immediate delivery.
Dealer enquiries invited.

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

AT LAST!

THE QL COMMUNICATES

ANNOUNCING THE QL BRIGHT STAR

The 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 LTD

Iolanthe Drive, Exeter, Devon EX4 9EA
Telephone: (0392) 69295

MODEMS · ON · THE · MOVE

Michael Graham examines two dedicated QL modems and speculates on the communications potential they facilitate.

It is widely predicted that one of the major applications for home computers in the near future will be as communications terminals between individual micro-users and between micro computers and large mainframe databases. At present, services such as electronic mail boxes and tele shopping/banking are only available on a limited basis, but the range of such services is set to grow rapidly over the next few years. In order for the owner of any micro to make use of the most obvious form of communications link – the public telephone system network – it is necessary to use a special interface called a modem (MOdulator DEModulator).

Sadly, QL owners have had a longer wait than most to obtain a dedicated modem, but now there are at least two designs around – Modem House's Bright Star and Tandata's QL Communications Package – the subject for this review. First, however, some background to the way these devices function.

SERIAL SINS

Why is a modem necessary in order to communicate over the telephone system? The QL has a wealth (well two), of serial ports which can output data over a two wire link. The phone system also consists of a two wire system. In ordinary use the link is used to carry audio frequency signals (ie, the human voice), so it might seem that there should be no reason why the system should not be able to carry the data output by one of the QL's serial ports. As usual in the world of computers, however, things are not as simple as they may seem.

The first problem is the incompatibility of the DC conditions at the telephone jack and those existing at the QL's serial ports. The QL outputs data at signal levels defined by the RS232 standard, these being +12V to represent the mark condition (logic 1 or *high*) and −12V for space (logic 0 or *low*). At the telephone end there is a nominal 50V between the A and B exchange lines, which rises to around 75V AC when a ringing current is being transmitted. Like chalk and cheese, the DC conditions at the QL and the BT exchange line just don't mix. A simple isolating circuit could readily overcome the DC incompatibility, but that's not the only problem.

The QL's data output switches between +12V and −12V (24V in total). In contrast the amplitude of a typical speech signal is at line level, around 0.7V. Obviously some form of attenuation is required. Additionally, however, there are restrictions placed on the frequency of signals that may be passed down a telephone line. Some of the most commonly used baud rates (300 baud for example) produce frequencies that fall outside those allowed and it is therefore necessary to process the raw data output by the QL in order to avoid such transgressions.

Other items of concern to BT when it comes to third party's connecting equipment up to the 'phone network include such things as the level of insulation between the equipment and mains voltages. BT have a duty to protect both their staff and other users of the network.

Taking all the above into account, even a basic modem has to be fairly complex. So, the best way to look at it is broken down into two distinct blocks, the transmit and receive sections. The transmit circuitry must first take the output from the QL and convert the RS232 level data signal into a +5V level for use by the rest of the circuit. This signal will then be applied to a special section that converts the logic 1's and 0's of the data into two distinct frequencies that fall within the band allowed by BT.

The receive circuit is responsible for amplifying the signal received from the line interface to a level suitable for processing by the rest of the circuitry. After filtering to remove any noise introduced by the line, this is fed to a circuit that reconstitutes the 1's and 0's of the data stream from the amplified audio tones. The output of this section is then fed to a level converter in order to match the signal to the requirements of the RS232 interface.

The majority of this hard work, interfacing a computer to the 'phone system, can now be undertaken by a single silicon chip. Having said that though, design of a modem is not to be undertaken lightly. Assuming that a computer had RS232 ports capable of operation at split baud rates, few have and that includes the QL, there would still be the problem of writing the terminal emulation software, that is the software responsible for controlling the flow of data between the host and remote computers.

THE BAUDOT SPLIT

The problem of split baud rates is one to which most modem designers must address themselves. This is because in order to receive data from any Prestel format database it is necessary for the computer to receive data at 1200 baud and to transmit at 75 baud *at the same time.* While the baud rate of the QL's two serial ports may be set to either 1200 or 75 baud, it is not possible to use both rates simultaneously. Hardware must therefore be built into the modem to support this form of operation.

The terminal software supplied with a modem is another important part of a modem package. The way in which computers handle the display of information on screen varies greatly and in order to facilitate communication between micros, terminal emulation software has to configure the QL as a Prestel display terminal.

TANDATA QCOM

The QCOM modem has, in a short space of time, had rather a chequered history. The modem was designed by and was due to be marketed by OEL. That company, though, went into receivership and there was a danger that the modem would never see the light of day.

At one stage it was thought that Sinclair might put up the money to rescue the design as it was the QCOM modem that was to be the Sinclair approved product for the QL. This, though, was before Sinclair ran into their own money problems. The rights to the QCOM design have now been acquired by Tandata who are now to market and support the unit.

QCOM breaks down into three component units. Each is housed in an identical ribbed black plastic box and stack one on top of the other. The base unit is the Q-Connect which is a multi-purpose serial interface that connects to the SER-2 port of the QL. In addition to forming the link to the rest of the system this unit provides an RS232 output via a standard 25-way plug. The second unit in the stack is the optional Q-Call auto-dial/auto-answer unit while the top of the stack is formed by the Q-Mod unit itself.

Connecting the units to the QL is a straightforward operation. First, the lead from the QL's power supply should be removed from the QL and plugged into the power socket on the Q-Con unit. Needless to say the mains power should be switched off throughout the process. Next the lead marked Ser 2 on the unit should be plugged into the corresponding terminal on the QL. The Q-call unit is simply stacked on top of the base unit, connection being by way of a mating plug and socket arrangement. Finally Q-Mod is stacked on top of the modem unit.

Q-Mod has a lead marked PSTN (Public Switched Telephone Network – the proper term for the BT phone system). This lead is terminated in a standard BT plug and should be plugged into one of the new style BT phone sockets. The unit also has a socket marked phone. Into this it is possible to plug a 'phone terminated with the appropriate plug.

The tower of units is designed to stand to the right of the QL computer and due to the short length of the leads, there is little choice over this matter. A series of LEDs indicate the status of the units. The Q-Con unit's LED is marked power and indicates that the mains power is on. The LED on the Q-Call unit is marked 'Ring' and shows that the auto-dial function is operative. The 'Seize' LED of the modem indicates that the unit has successfully established a link with a remote computer.

The instructions supplied with the review modem were very much of a temporary nature consisting of just three photostat pages giving the bare details of operation of the system. The final version of the manual is however in preparation and should be available in the near future.

CHARGING UP

With the QCOM software installed in microdrive 1, applying power to the QL will boot up the terminal software. The first thing to greet the user is a screen concerned with initialising the real time clock, built into the system. Such a facility is very useful when trying to keep track of the cost of using the modem. Line charges can mount up if extended use is made of the facilities offered by various databases.

After entering the date, you are asked to type in the user's name. Pressing return at this stage will request a file name. This refers to a phone book file, a file of numbers that can be used in conjunction with the auto-dial unit in order that 'phone numbers may be dialled by simply inputting the name of the system being called (more on this later). If no ' 'phone book' file exists, entering return will take the user to the default file supplied on the QCON microdrive.

The user is now presented with the main system menu through which the rest of the system may be accessed. Selecting option 3, switchboard, and then 5 will take the system into the mode in which new phone numbers may be stored. In response to the page name prompt a suitable name, (eg, test) may be entered. This will cause a

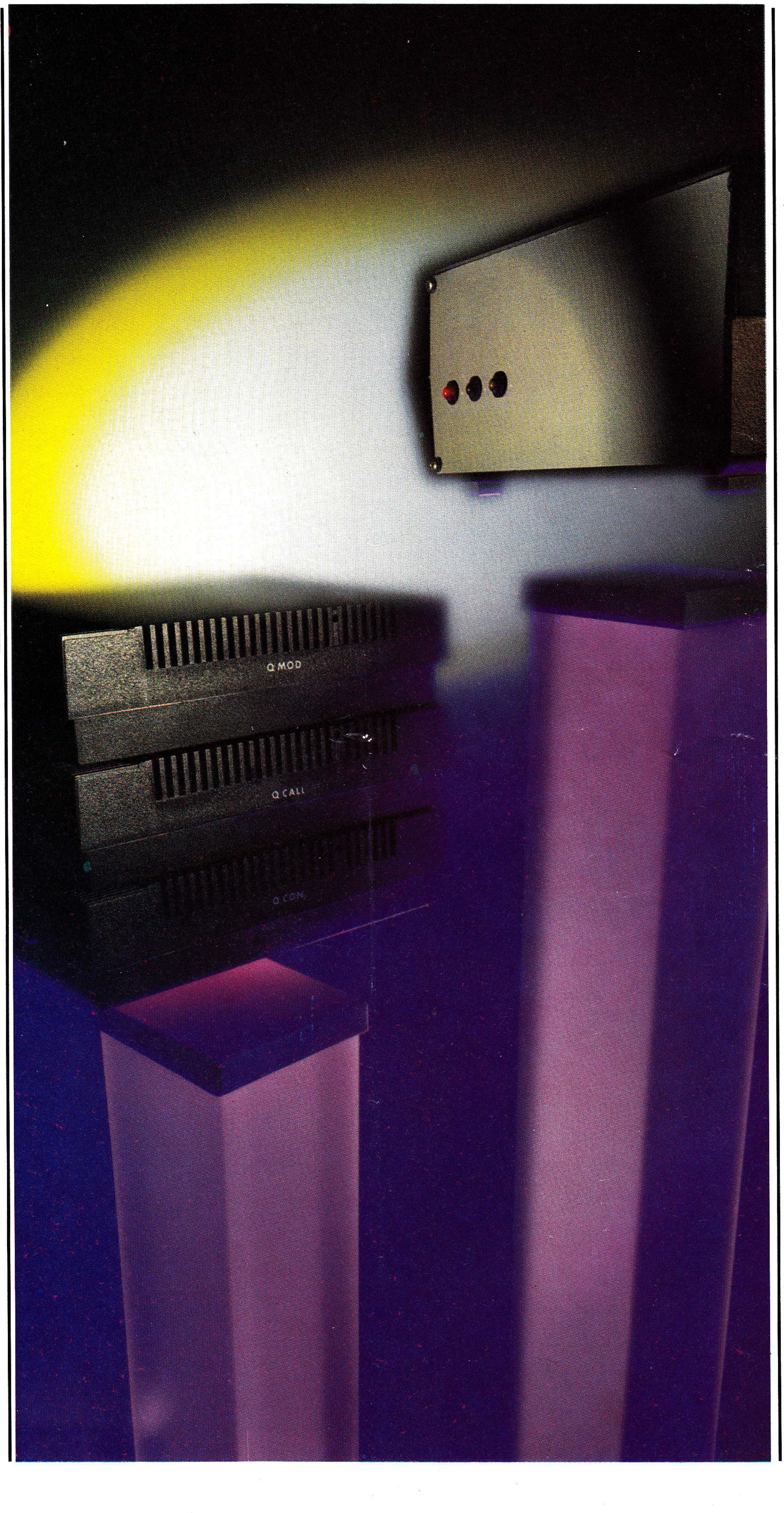


PHOTO BY GEORGE LOGAN

blank phonebook page to be displayed. The first column on the page is headed ID and the string entered here will be the name by which the 'phone number will be accessed by the auto-dial option. Into the next column is entered the 'phone number of the database while the last column can be used for two purposes. Either for a short comment on the database (ie the times at which it is on-line) or for any code word string that might have to be entered for the system to allow the user to log on. In the latter case the password string should be enclosed by <>.

Having entered this information, pressing function key F2 will display a menu at the bottom of the page. If the details entered are correct the tabulate option will allow the communication protocols required to be set up. The range of baud rates, parity options and other parameters provided should allow just about any databases' requirements to be met. Luckily a set of default options, conforming to the standard Prestel format, are built into the software and pressing F2 twice will return the user to the main menu.

If the auto dial unit is fitted all that is needed in order to call a remote computer is to select option 1 from the main menu and then 1 again to the log-on/log-off prompt. At this point the computer will request a name (this refers to the heading that the number was referred to in the ' 'phonebook' file). Once that's been entered the system will display the number that is being detailed and the Q-Call unit will seize the line. The faint clicking of the dial pulse relay should be heard.

If the Q-Call unit is not connected, the process is very similar. It is still necessary to set up a ' 'phonebook' file but instead of the dialling being automatic, having entered the name of the system to be dialled the user is prompted with the appropriate number and asked to dial manually. When the remote computer answers, pressing the space bar will establish the call.

In common with many other modern designs, the Tandata modem has a provision for 'user-to-user' mode. This was not documented in the temporary user guide, but such a facility gives QL users the ability to exchange information between themselves or to swap program files amongst a group of users. Almost networking but not quite!

BRIGHT STAR

In contrast to the modular design of the Tandata modem, the Bright Star modem consists of a single unit that is quite uninspiring to look at. This appearance, however, belies the fact that the modem offers a wider range of facilities from Automatic Scan and mode selection to the provision of a serial to Centronics printer interface complete with a 2K buffer.

Once again documentation for the modem was in a preliminary version (let's hope the actual products are ready – Ed!) and the software supplied was not in production form, though the modifications to the software that will be incorporated in the final unit concerned very minor modifications to that supplied with the review unit.

Connecting the modem to the QL is a simple process. As with the Tandata modem, connection of the unit to the 'phone system simply involves placing the modem 'in-circuit' between your telephone and the BT wall socket. Unlike the Tandata modem, the Bright Star has its own internal power supply and is supplied with a standard mains plug fitted to the mains cable.

The modem does not feature any front panel switches and this indicates that the unit is controlled entirely via software – a far more satisfactory approach than modems that require the user to select modes of operation from a series of switches. The front panel does, however, carry a series of LEDs. These indicate power, on-line and carrier detect status.

The modem is connected to the QL via the SER2 socket and having made the appropriate connections it is possible to confirm correct operation of the unit by invoking a self test mode. This is instigated by first sending a DLE code to the modem. The DLE (Data Link Escape) code indicates that the next character sent is to be interpreted as a command and not as a character for onward transmission. The DLE code is produced by pressing CTL P on the keyboard, the self test is started by following this with CTL E. If all is well the modem will respond with an answer – back message detailing its version number, mode and status.

Assuming that the modem is operating correctly, the QTALK software can be loaded into the QL. Once the program has loaded most of the screen will clear and a clock will be displayed in the top right-hand corner of the screen. Again this useful facility allows the user to keep track of the length of time that they are logged on to a system. QTALK assigns various commands to the function keys of the QL. F1 is a help key. F2 will display the modem's status, F3 is used to log on to a system in the auto scan mode. F4 selects the commands menu, while F5 is the log-off key.

Before describing the commands menu, just a brief word about the auto-scan mode of the Bright Star. This very useful facility means that it is not necessary for the user to know what protocols are adopted by any system that they are trying to use, the modem will scan the incoming signal and automatically select the right mode. As sorting out comms' protocols is sometimes a nightmare for even an experienced user of modems, this facility should be particularly useful for those new to the world of computer communications.

Pressing F4 to enter the command mode will reassign the functions associated with the QL's function keys. F1 will now allow QL files to be sent to the modem and any incoming data to be sent to a file. F2 will toggle the printer between its on and off condition, F3 is concerned with the modem's newline mode, F4 toggles the screen between 40 and 80 columns while F5 will kill the comms software and return the user to SuperBasic.

Another powerful feature of the Bright Star is a 'QL-to-QL' comms mode. This is a packet mode transmission system operating at 1200 baud.

The Bright Star unit supplied did not have an auto-dial unit fitted, but one will be available in the near future.

SIDE BY SIDE

Both the Tandata and Bright Star modems are well designed items of equipment and represent the top end of the modem market. Both allow the user full control of the functions of the modem by means of commands provided by powerful terminal software. The auto-scan mode of the Bright Star will be welcomed by many users, but for those whose main requirement is for communication with Prestel systems, the default modes of the Tandata modem will make establishing such links a straightforward process.

Having bought a modem you will obviously want some numbers of systems to dial. In order to use Prestel you will have to become a subscriber of the service. Information on joining Prestel can be obtained by dialling 100 and asking for freephone Prestel.

There are, however, a number of free databases up and down the country. Lists of these numbers can be found in some computer magazines but in order to get you started here are a couple of numbers to try (0702 546373, 0462 677177). These operate to standard Prestel protocols 24 hours a day.

Auto-dialler
Device for automatically dialling the number of a remote computer from a directory of numbers created by the user.

Baud
Measure of the speed at which data is transmitted over a communications link. Dividing the baud rate by 10 gives an approximate figure for the number of characters transmitted per second – for example a 300 baud modem will transmit 30 characters per second.

CCIT
International standard for the transmission of data between countries.

Database
Information stored on a computer in such a way that it can readily be accessed by registered users of the system, usually over a remote link.

Electronic Mail
The sending of messages from one user to another. Prestel allows users to deposit electronic mail in the mailbox of any other user.

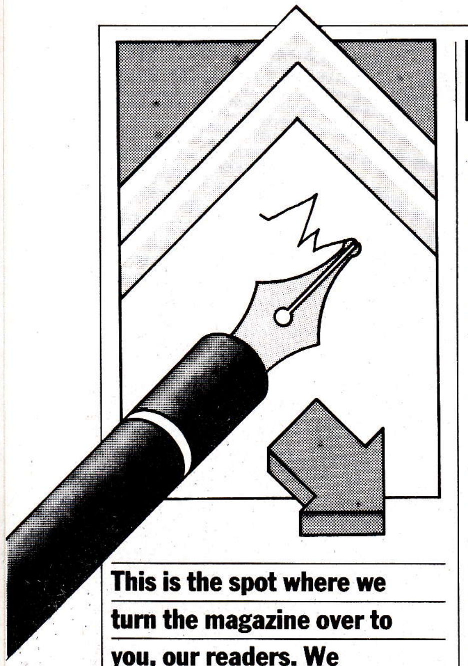
Half-Duplex
Data link that allows the transmission of data in only one direction at a time.

Full-Duplex
Data link that provides for the transmission of data in both directions at the same time.

RS232
Standard format for the transmission of serial data.

REN
Ring Equivalence Number. A standard telephone has a REN of 1. The total REN rating for equipment connected to an exchange line should not exceed 4.

OPEN CHANNEL



This is the spot where we turn the magazine over to you, our readers. We welcome any comments, criticisms or anecdotes about either the QL or QL User. The address to send your letters is: Open Channel, QL User, Priory Court, 30–32 Farringdon Lane, EC1R 3AU

Time Is Money

I have been using a QL since Sinclair finally delivered it to me on 31st July 1984 (it was ordered during the first week of April). It is used entirely for business purposes and I must say that with the improved Psion software and the addition of a Simplex Data 256K memory expansion (so that I can use more than the top right hand corner of the Abacus spreadsheet), I find it very satisfactory. So much so in fact that I have come to completely depend on it. It is in daily use already and when a proper accounting package and stock control software become available (we are told that they are "imminent" so I confidently expect them before the end of the decade), it will be in more or less continuous use.

There is however a problem. My microdrives, or at least one of them, appear to be faulty and inconsistent. The difficulty is, as yet, more of an inconvenience than a serious problem but I would nevertheless feel a lot happier, bearing in mind how much I depend on this machine, if it were put right.

Here we reach the real difficulty – Sinclair say that I must return the machine to them and that the repair will take "10 days". Now I know that in Sinclair language 28 days=four months so 10 days should be about six weeks, but even if it really means 10 days there is no way that I could manage without a computer for that long (even two days would be a serious problem) – I should have no option but to buy another one as a spare.

How can Sinclair continue to promote the QL as a "business" machine with this kind of back-up? It is one thing to give up shooting down aliens for two weeks while they repair your toy but Uncle Clive ought to realise that it is quite another to tell ones employees "Sorry no wages for two weeks – the computer is being repaired"!

David A Hawley
DAH Marketing

Screen Fitting

Can you please tell me the size of the windows on channels 0 and 1 in TV or Monitor mode? I need this information for my programs as anything larger than these windows will not fit my television screen. Also can you tell me anything about the system variables?
One last thing, a friend of mine was experimenting with the CALL command and caused system RESET. However, he has been unable to reproduce this. Was this a sheer fluke or is there a reset address?

G Mann
London

The sizes and colours for the screen windows attached to channels 0, 1 and 2 on switchon are:

Monitor Mode
Channel 0: 512,50,0,206
Paper: 0 Strip: 0 Ink: 4
Border: 0
Channel 1: 256,202,256,0
Paper: 2 Strip: 2 Ink: 7
Border: 1,246
Channel 2: 256,202,0,0
Paper: 7 Strip: 7 Ink: 2
Border: 1,246

TV Mode
Channel 0: 448,40,32,216
Paper: 0 Strip: 0 Ink: 7
Border 0
Channel 1: 448,200,32,16
Paper: 2 Strip: 2 Ink: 7
Border 0
Channel 2: 448,200,32,16
Paper: 1 Strip: 1 Ink: 7
Border 0

The system variables are rather more complicated than those on earlier Sinclair machines, as there are far more things for QDOS to keep track of. Although I could tell you that SV_BASE is nominally at $28,000, I doubt it would help very much! If you want to get this deeply involved in the machine, you will have to learn 68000 Assembly language. You will also need to buy the QL Technical Guide, which is available from Sinclair Research and costs £14.95. Likewise, numerous addresses can be CALLed and seem to produce the RESET effect, but it really isn't as simple as that. You would need to enter Supervisor mode, play around with the hardware by issuing a RESET instruction, load the system stack pointer from location 0 and then jump to the address held in location 4. Like this, in fact:

```
TRAP #0          puts the
                 system in
                 supervisor
                 mode
ORI.W #$700,SR   disable all
                 interrupts
MOVEA.L 0,A7     load SSP
                 from vector
RESET            ...just in
                 case!
JMP (A2)         jump to the
                 contents of
                 the reset
                 vector
```

As you can guess, a simple call is unlikely to do all this, especially as the 68008 in the QL starts off in supervisor mode when you switch it on anyway!
(Adam Denning)

Engineer's Standpoint

I write regarding some comments concerning a Fortran 77 compiler for the QL. Fortran 77 is a superset of standard Fortran which has all the advantages of a structured language such as Pascal while at the same time enabling the use of a vast amount of software already available in Fortran. It has been, is and will be the language for scientific and engineering software. This perhaps is the reason why Microsoft support MS-Fortran for all MS-DOS (and PC-DOS) micros. The runaway success of the IBM-PC is in a large way due to its use in scientific and engineering applications and the availability of a wide choice of Fortran 77 based software. As a new owner of a QL and a developer of engineering software, I am of the view that it cannot be seriously called a supermicro as at present it cannot even be considered for an area of applications in which supermicros will be widely used.

Dr G Amaratunga
University of Southampton

Obvious Overlooked

For business use, QL and accompanying software are all we expected. However, there is one exception which is slightly annoying. Having had no problem installing printer drivers on Quill and Abacus we cannot do so on Archive. According to the User Guide this should not be a problem. We returned the package to Psion asking for advice. A new copy was forwarded with no explanation. "Not found" remains the error message whenever we try to print.

H Brownrigg-Stevens
Peterborough

The solution to your problem is simple. You need to copy across a file called "printer_dat" from your Quill cartridge to your Archive cartridge. To do this place Quill in mdv1_, a copy of Archive in mdv2_ and enter:
COPY mdv1_printer_dat to mdv2_printer_dat

Professional Opinion

Doubtless you are as aware as anyone of the shortage of software for the QL. It was therefore all the more surprising to find your review (June edition) of the Triptych software so disappointing. The three pieces of serious, supposedly professional software of the sort crucial to the success of the QL had been eagerly awaited since being first advertised in late 1984. Your reviewer managed to dispense with all three in only one page – less if illustrations are excluded – and I would have welcomed serious comment on the value of the programs: for example –

a) To what extent do these programs have a serious use? Or are they just clever games?

b) How do the QL versions of these games differ from those available for other machines, and how effectively do they make use (if at all) of the potential extra memory of the QL's 128K?

c) With reference to Entrepreneur, is the treatment of tax and other financial matters both accurate and realistic? The opinion of an accountant would have enriched the value of the review considerably.

To conclude, I have noted with interest the many criticisms of Sir Clive Sinclair in your pages: yet you must have an almost equally strong interest in fostering sales of the machine. A little more professionalism on your part would not come amiss.

Peter J Fraser
Hertfordshire

Taking each point in turn. First, we feel that the review in question was succinct, concise and to the point. The author did not fall into the trap of writing a 'pseudo' operator's guide nor did he let the euphoria of discovering a commercial program for the QL go to his head. In a magazine that prides itself upon covering every aspect of the QL, quality and not quantity takes precedence.

Second, Tryptych's programs are built around a set of theories relating to decision-making, planning and financial analysis. The degree to which these may be applied to the user's work situation determines the extent to which the programs may be considered serious as opposed to entertaining. Clearly this varies from user to user.

Third, whilst we agree that a comparison with other machines may well have been of interest our principal concern remains the QL.

Fourth, as regards use of memory, this is a wholly inappropriate measure of a program's worth. It would, for example, be quite easy to construct a program that uses every bit of the QL's 90K free memory and yet does nothing.

Fifth, it may surprise you to know that our reviewer has three years' experience with a firm of chartered accountants and still retains close links within that profession. However, having described Entrepreneur as a "guided tour into the world of small business and limited finance" he feels that an acountant's opinion would have been as out of place as a commentary from a chartered surveyor on a sightseeing bus.

Finally, our writers are journalists of many years' standing in the computer field and all endeavour to maintain the magazine's independent stance.

The Significant Seven

Thank you for the review on Cash Trader. Is it correct that only seven digits may be entered? I would be interested to enter Ptas 10000000.

Ralf Oertzen
Alicante

QDOS will support no more than seven significant figures. If this limit is exceeded the QL will display a number in its exponential form.

All the business software so far reviewed (Cash Trader, Sagesoft Accounts included) is subject to this limitation.

PLEASE NOTE . . .

A slight anomaly occurred when we came to judge June's 'Inside Out' competition.

When it came to the second prize we realised that our ingenious Sub Editor (sweet thing that she is) had worded the competition in such a way as to make judging exceedingly difficult – the second prize would have to go to the entry with the longest word that *didn't appear in the first prizewinner's list!*

The prospect of wading through around 50,000 words to find the longest only to discover that it *was* included in the winners list didn't find great appeal, so we have decided on a new approach.

Sorting through the entries revealed that the longest words were of ten or more letters.

If you entered the 'Inside Out' competition and your list contained a word of ten or more letters, please telephone or write into us so we can check your entry against the winner's extensive compendium (in the event of a tie the winner will be chosen from out of a hat).

Back to the overall winner, who managed to generate 728 words. He is Peter Skiba from Blackpool and receives an Insider disk interface and single disk drive from Silicon Express. The runners up, who gain £20 vouchers against similar items, were: S L Hindley (Letchworth – 521), Peter Mockford (Southampton – 488), Chris Bowyer (Wokingham – 436), Jane Firth (Isleworth – 408), and V Wiffen (London E8 – 396).

NEXT MONTH

SOFT LANDING

We searched high and low
for companies selling, writing or just
designing QL software and here's
the result – a comprehensive survey
of tape and disk software, from games to
sophisticated business packages.

QL Hyperdrive

Hard on the heels of Microdeal, another well-known software company has come up with a really exciting game for the QL involving action on the race track. We've donned tracksuit and gloves to tackle a lap or two.

Taking a DIP

The first in a two part feature series on vital printer connection details and how to get something sensible out of them once they're hooked up!

System Variables

Don't panic, this new series has *nothing to do with machine code or QDOS,* it's all about different QL peripherals. Each month we'll focus on a particular QL set-up and highlight its salient features.

PLUS

QL User Technical Helpline – Full details and the number to ring for our independently operated telephone Q & A service.

ALL INSIDE THE

September Edition

ON SALE 21st AUGUST

Contents subject to late revision

COMPUTER ALLSORTS

Choosing the wrong 'sort' will strain patience and jeopardise your relationship with the QL. Marcus Jeffery shows how to avoid this by making the right selection.

Sorting operations are fundamental to computer programming. There are very few computers around that do not spend a significant portion of their time sorting items into ascending or descending order.

Here we compare different sorting techniques. The program below may be used to test each in turn by editing line 160 to read as follows:

160 insertion_sort number
160 heap_sort number
160 merge_sort 1, number
160 quick_sort1, number

```
100 CLS
110 number = 500
120 DIM s(number+1)
130 s(number+1) = 9999
140 :
150 start_time number
160 bubble_sort number
170 stop_time "BUBBLE SORT",number
360 STOP
400 DEFine PROCedure start_time (number)
410 :
420 initialise_elements number
430 SDATE 1985,1,1,0,0,0
440 :
450 END DEFine start_time
490 DEFine PROCedure stop_time (sort$,number)
500 LOCal time$
510 :
520 time$=DATE$
530 PRINT sort$;" took"
540 PRINT TO 5;time$(16 TO 17);" minutes ";time$(1
9 TO);" seconds"
550 PRINT TO 5;"to process ";number;" items."
560 PRINT
570 END DEFine time_sort
610 DEFine PROCedure initialise_elements (number)
620 LOCal element
630 :
640 FOR element = 1 TO number
650    s(element) = RND
660 END FOR element
670 :
680 END DEFine initialise_elements
720 DEFine PROCedure swap (a,b)
730 LOCal temp
740 :
750 temp = s(a)
760 s(a) = s(b)
770 s(b) = temp
780 :
790 END DEFine swap
```

Bursting The Bubble

Faced with a list of unsorted numbers which need sorting into ascending order, the most obvious approach would be to scan the list and put the smallest item at the front, followed by the next smallest item, and so on until the list is sorted.

Given the list:

5 4 6 2 3 1

This would result in the following steps:

(a) 1 5 4 6 2 3
(b) 1 2 5 4 6 3
(c) 1 2 3 5 4 6
(d) 1 2 3 4 5 6

The Bubble Sort, as this method is called, appears below. The outer loop, *i*, scans the elements *number–1* times, and within this, *j* scans the remaining unsorted elements. At each stage in this loop, the *j*th and *(j+1)*th elements are compared and, if necessary, swapped into ascending order. At the end of the first iteration of the outer loop, the largest element will have filtered through to the end of the list. The elements up to this element are now handled in the same way, and so on.

```
830 DEFine PROCedure bubble_sort (number)
840 LOCal i,j
850 :
860 FOR i = number-1 TO 1 STEP -1
870    FOR j = 1 TO i
880       IF s(j)>s(j+1) THEN swap j,j+1
890    END FOR j
900 END FOR i
910 :
920 END DEFine bubble_sort
```

Though this algorithm can be significantly improved by only performing the swap when the largest element is found and stopping as soon as the list is sorted, it is still very slow. It is said to have 'exponential time increases'. This means that given *N* items, the time will be a factor of N^2. Though this will not matter much for small data lists, larger lists could take a prohibitively long time.

Instant Inserts

Considerably more efficient than the Bubble Sort is the Insertion Sort given here. The algorithm is fairly straightforward. The outer *i* loop scans through the list once, from the second element onwards. The list is then checked backwards, from the element *i* until *j* is found, such that the *i*th element is greater than or equal to the *j*th element. All the elements from *j+1* to *i* are then shunted across one place, and finally the previous *i*th element is placed in position *j*. So, with the list:

```
960 DEFine PROCedure insertion_sort (number)
970 LOCal i,j,value
980 :
990 s(0) = -9999
1000 FOR i = 2 TO number
1010    value = s(i)
1020    j = i - 1
1030    REPeat place
1040       IF value >= s(j) THEN EXIT place
1050       s(j+1) = s(j)
1060       j = j - 1
1070    END REPeat place
1080    s(j+1) = value
1090 END FOR i
1100 :
1110 END DEFine insertion_sort
```

5 4 6 2 3 1

the steps would be:

4 5 6 2 3 1 (2nd element one place ←)
4 5 6 2 3 1 (don't move the 3rd element)
2 4 5 6 3 1 (4th element three places ←)
2 3 4 5 6 1 (5th element three places ←)
1 2 3 4 5 6 (move the 6th element five places ←)

Finally, note that the Insertion Sort requires that a first element in the list (ie, *S(0)*) contain a number smaller than any other number in the list.

Multiple Pile-Up

The next sort routine, known as a Heap Sort is shown below. This works on a simple strategy of successively removing the largest object from the unsorted elements. The

```
1150 DEFine PROCedure heap_sort (number)
1160 LOCal i
1170 :
1180 create_heap number
1190 FOR i = number TO 2 STEP -1
1200    swap i,1
1210    form_heap 1,i-1
1220 END FOR i
1230 :
1240 END DEFine heap_sort
1270 :
1280 DEFine PROCedure create_heap (number)
1290 LOCal i
1300 :
1310 FOR i = number DIV 2 TO 1 STEP -1
1320    form_heap i,number
1330 END FOR i
1340 :
1350 END DEFine create_heap
1380 :
1390 DEFine PROCedure form_heap (i,number)
1400 LOCal j,item
1410 :
1420 j = 2*i : item = s(i)
1430 REPeat combine
1440    IF j>number THEN EXIT combine
1450    IF j<number THEN IF s(j)<s(j+1) THEN j=j+1
1460    IF item>s(j) THEN EXIT combine
1470    s(j DIV 2) = s(j)
1480    j = 2 * j
1490 END REPeat combine
1500 s(j DIV 2) = item
1510 :
1520 END DEFine form_heap
```

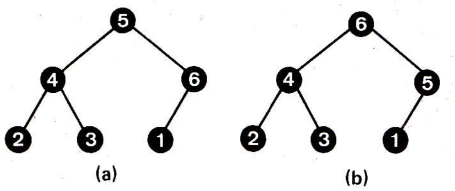
advantage of the Heap Sort is that elements can be found much quicker than in a normal list. Firstly, a 'heap' is a complete binary tree but with the values of the children at each node being at least as large as their parent. So, when the Heap Sort routine makes the initial call to 'create_a_heap' the following changes will be made to the unsorted list:

5 4 6 2 3 1

becomes:

6 4 5 2 3 1

Referring to the next diagram



makes things a little clearer. In *(5a)* the binary tree is not a heap, because the value '5' is not as large as both of the child node values, '4' and '6'. Therefore, the nodes are swapped to give (b). This tree maps onto a sequential list as an element *i* having children 2*i and 2*i+i.

Calling the procedure 'form_heap', either directly, or via 'create_heap' has the effect of moving the largest element to the top of the heap. (If it didn't, the structure would not be a heap!) Consequently, to sort the items, all that is necessary is to swap the largest item at the top, *s(1)*, with the item at the bottom, *s(i)*, then form a new heap of the items *i* to *i–1*, swap the top and bottom elements of this new heap, and so on. Of course, using this type of data structure it is not necessary to move quite so many items in order to move the largest value to the top. In fact, assuming N items, the movements will be a factor of $\log_2N$. This is repeated approximately N times, giving an overall factor of $N\times\log_2N$, as opposed to N^2 for the Bubble Sort algorithm – quite a considerable improvement. (These factors are true even if the algorithms are presented with the worst possible initial ordering of the data.)

Major Merger

Our penultimate routine, called Merge Sort (next), works using a

```
1560 DEFine PROCedure merge_sort (lower,upper)
1570 LOCal middle
1580 :
1590 IF lower < upper THEN
1600    middle = (lower + upper) DIV 2
1610    merge_sort lower,middle
1620    merge_sort middle+1,upper
1630    merge_array lower,middle,upper
1640 END IF
1650 :
1660 END DEFine merge_sort
1690 :
1700 DEFine PROCedure merge_array (lower,middle,up
per)
1710 LOCal t(upper),i,j,k,m
1720 :
1730 m = lower : i = lower
1740 j = middle + 1
1750 REPeat not_empty
1760    IF m>middle OR j>upper THEN EXIT not_empty
1770    IF s(m)>s(j) THEN
1780       t(i) = s(j)
1790       j = j + 1
1800    ELSE
1810       t(i) = s(m)
1820       m = m + 1
1830    END IF
1840    i = i + 1
1850 END REPeat not_empty
1860 IF m > middle THEN
1870    FOR k = j TO upper
1880       t(i) = s(k)
1890       i = i + 1
1900    END FOR k
1910 ELSE
1920    FOR k = m TO middle
1930       t(i) = s(k)
1940       i = i + 1
1950    END FOR k
1960 END IF
1970 FOR k = lower TO upper
1980    s(k) = t(k)
1990 END FOR k
2000 :
2010 END DEFine merge_array
```

strategy known as 'divide-and-conquer'. For instance, suppose you had to sort a list of say, 5,000 items. Using an exponential algorithm, the time will be a factor of N^2, giving 25,000,000. However, if we divide the set into two equal halves, and sort each separately, the time will be a factor of $2\times(N/2)^2$, giving a much reduced 12,500,000. The only problem then remaining is to combine, or 'merge' the two halves together. This can easily be done in a very short time by taking the next largest element from the beginning of each of the half-lists until both lists are exhausted.

Of course, there is no reason why each of the half-lists shouldn't be sorted by splitting them into two and sorting each quarter separately, giving a factor of 6,250,000. Continuing with this splitting process, we'll eventually have a factor of $N\times\log_2N$.

One of the remaining problems with the Merge Sort is that once the lists are reduced to only a few elements, most of the time is taken up by the recursion. A solution to this would be to put a test inside the 'merge_sort' procedure. This would carry out the operations as shown if the number of elements was still 'large', but use a non-recursive sort (eg, Insertion Sort) otherwise.

Particle Partition

The final sorting routine which we will consider is known as Quicksort (*Fig 7*). This works in a similar way to the Merge Sort, but instead of splitting up the elements, sorting them independantly, then later merging them, Quicksort ensures that all the elements in the first subset are smaller than those in the second. This process, known as 'partitioning' implies that the final merging is no longer necessary.

To split the list, the Partition Routine chooses a partition element, *i*, and rearranges all of the unsorted elements such that the first sublist contains all the elements with values less than or equal to the *i*th value, and the second sublist contains all values greater than or equal to the *i*th value. Obviously, it is unlikely that the *i*th value will split the data into two equal subsets. In the example, the last element is always chosen to be the partition element. This involves ensuring that an extra element exists at the end of the list which must be greater than or equal to all the elements in the list. There are many other means of choosing a partition element which are to be preferred in practice.

The Quicksort completes our analysis of some of the more common sorts currently being used. All that remains is to compare each one's performance on lists of varying sizes. To do this simply alter the value of *number* in line 110. The set of results given in *Fig 8* clearly highlight the differences in speed between the various methods.

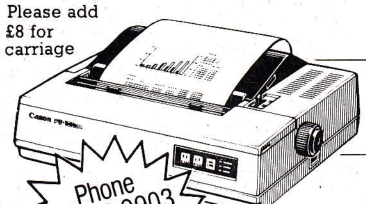
```
2050 DEFine PROCedure quick_sort (lower,upper)
2060 LOCal part_elem
2070 :
2080 IF lower < upper THEN
2090    part_elem = upper + 1
2100    partition lower,part_elem
2110    quick_sort lower,part_elem-1
2120    quick_sort part_elem+1,upper
2130 END IF
2140 :
2150 END DEFine quick_sort
2180 :
2190 DEFine PROCedure partition (i,j)
2200 LOCal k,value
2210 :
2220 value = s(i)
2230 k = i
2240 REPeat loop
2250    REPeat inc_k
2260       k = k + 1
2270       IF s(k) >= value THEN EXIT inc_k
2280    END REPeat inc_k
2290    REPeat dec_j
2300       j = j - 1
2310       IF s(j) <= value THEN EXIT dec_j
2320    END REPeat dec_j
2330    IF k < j THEN
2340       swap k,j
2350    ELSE
2360       EXIT loop
2370    END IF
2380 END REPeat loop
2390 s(i) = s(j)
2400 s(j) = value
2410 :
2420 END DEFine partition
```

	50 items	100 items	500 items
BUBBLE SORT	28	109	2815
INSERTION SORT	9	35	925
HEAP SORT	9	21	140
MERGE SORT	12	27	179
QUICKSORT	6	13	95

The best service, the best value in printers from Viglen

Dot Matrix Printers

Please add £8 for carriage



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

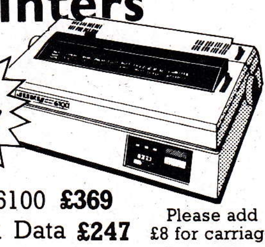
Epson RX80/FT **£275**

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.

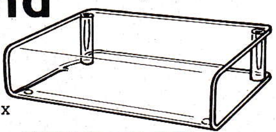
Viglen are also major suppliers to educational and government establishments and welcome further enquiries and orders.

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

All prices correct at time of going to press

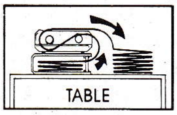
ALL PRICES INCLUDE VAT

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 ______________________ QU8

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 ______________________

VISA

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

QU8

TERMINAL EMULATION

In the final part of our terminal emulator Adam Denning covers job control and piping

Last issue we developed various routines for job control, this month we elaborate on these routines and explain how they work. We begin, understandably, with the **jobcontrol()** procedure itself. Called from **poptions()**, this produces a sub-menu with seven options and checks to see whether a valid key has been pressed. If it has, the key code is then passed as a parameter to **job.exec()**, which takes the appropriate action. If the key pressed was neither 'C' nor 'I', it asks for the internal job ID. This number is entered in ASCII hexadecimal, converted by the **get.hex()** function after which the machine code routine ISJOB is called to see if the specified job exists. If it doesn't, the question is asked again.

Once a valid job ID has been input, **job.exec()** switches on the key code and calls a designated subsidiary procedure. Job activation uses the **doactiv()** procedure, asking for a starting priority (which must be between 0 and 127) and a timeout (which must be 0 or −1). Killing a job invokes the KILLJOB machine code procedure, and releases a suspended job which uses our added routine RELEASE. The machine code extensions PRIORITY and SUSPEND are called to change the priority of a job or suspend it. These require numeric information from the user relating to the new priority (0 to 127) and to the timeout (0 to 32767, or −1 for infinite timeout), respectively.

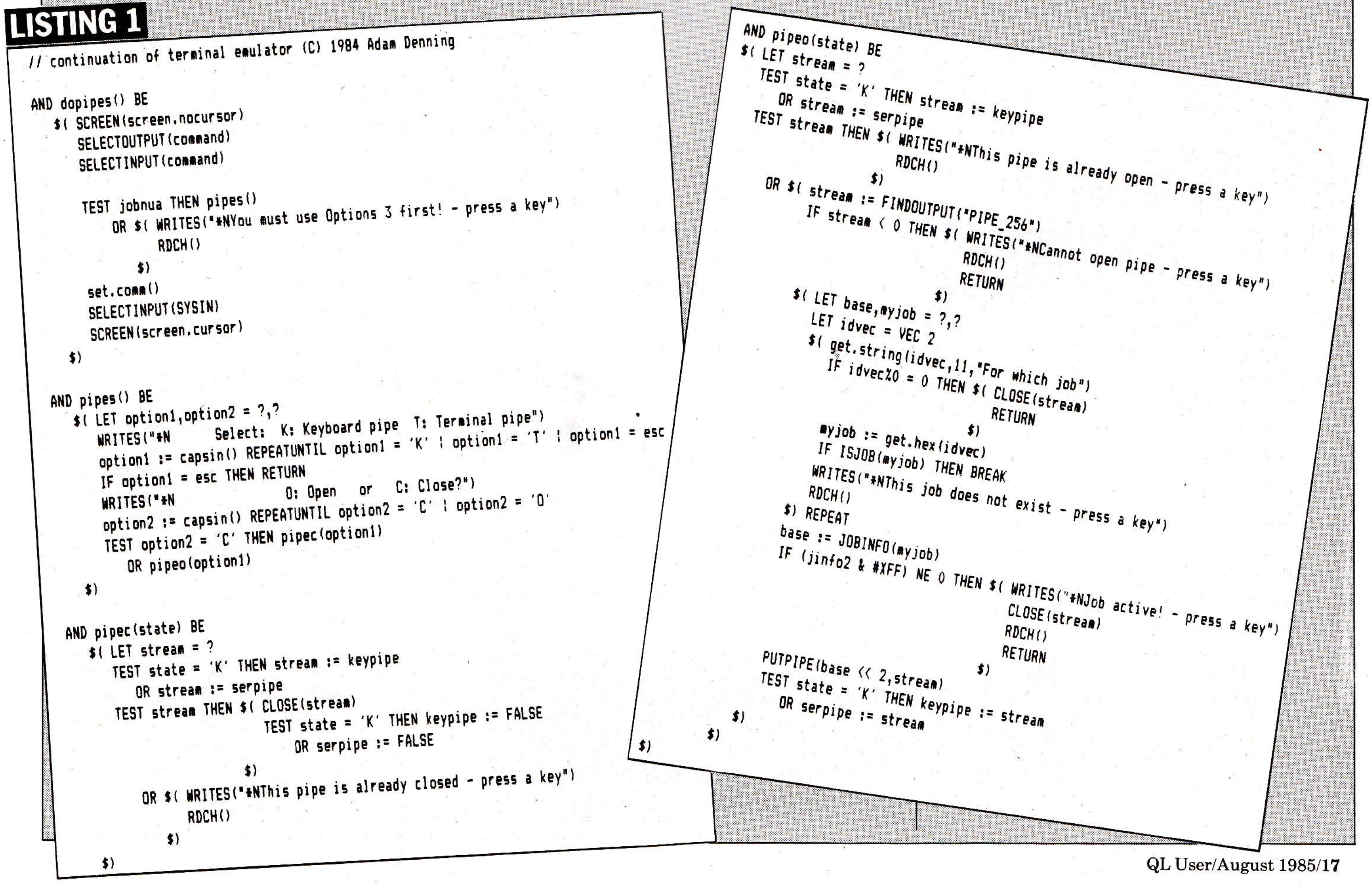
If the option selected in **jobcontrol** was 'C', for 'create job', then a filename must be collected and a file opened. After this READFILEHEADER is used to find the program's length and its data space size, both of which are passed as parameters to CREATEJOB. If QDOS condescends to create this job, a job ID and a base address are returned. The file is then loaded into memory at this base address before reporting the job ID to the user.

The final option, I for 'Information', has its own routine called **get.info()**. This procedure is a loop built around JOBINFO, which uses the QDOS routine MT.JINF to scan the job tree. Each job in the tree is reported in terms of job ID, owner job ID, followed by 'S' if it is suspended plus its priority. If the job in the tree is in 'standard format' (ie, if there is a word of $4AFB at offset 6), we assume that it has a name stored as a QDOS string starting at offset 8, and print this name out. When there are no more jobs in the tree, a message is printed before returning from the procedure.

The final part of the program reproduced in this issue (*listing 1*) comprises of the pipe control routines, which allow us to do things like print data at the same time as manipulating it in other ways, or to pass it to another program to generate monitor files. The **action** function shown a couple of months ago also has provision for an expression evaluator (accessed by pressing F5, which calls the **doexpr()** procedure). This expression evaluator is essentially a powerful 32-bit integer calculator, but as it is fairly long we have not incorporated it in the version of the program here. If anyone is interested, they can obtain the source from me via QL User if they send a disc or microdrive cartridge.

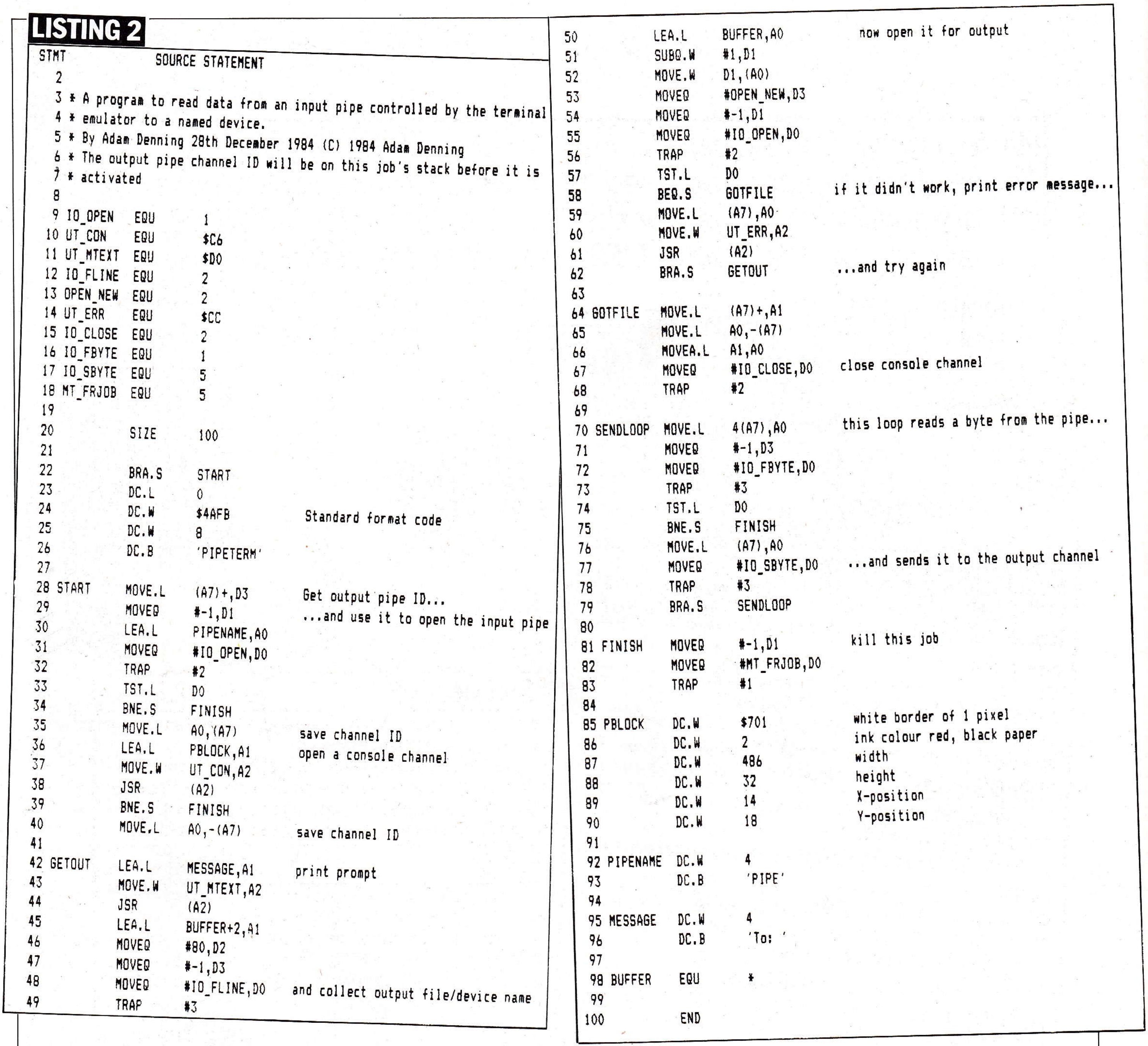
Moving on to the pipe routines, the first one called **dopipes()** simply checks if the job control routines have already been accessed by examining the global variable **jobnua**. If all's well, it calls **pipes()**, which asks you whether you want to open/close the keyboard pipe or the ter-

LISTING 1



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

AND dopipes() BE
   $( SCREEN(screen.nocursor)
      SELECTOUTPUT(command)
      SELECTINPUT(command)

      TEST jobnua THEN pipes()
          OR $( WRITES("*NYou must use Options 3 first! - press a key")
                RDCH()
             $)
      set.comm()
      SELECTINPUT(SYSIN)
      SCREEN(screen.cursor)
  $)

AND pipes() BE
   $( LET option1,option2 = ?,?
      WRITES("*N      Select:  K: Keyboard pipe  T: Terminal pipe")
      option1 := capsin() REPEATUNTIL option1 = 'K' | option1 = 'T' | option1 = esc
      IF option1 = esc THEN RETURN
      WRITES("*N                     O: Open   or   C: Close?")
      option2 := capsin() REPEATUNTIL option2 = 'C' | option2 = 'O'
      TEST option2 = 'C' THEN pipec(option1)
         OR pipeo(option1)
   $)

AND pipec(state) BE
   $( LET stream = ?
      TEST state = 'K' THEN stream := keypipe
         OR stream := serpipe
      TEST stream THEN $( CLOSE(stream)
                          TEST state = 'K' THEN keypipe := FALSE
                             OR serpipe := FALSE
                       $)
          OR $( WRITES("*NThis pipe is already closed - press a key")
                RDCH()
             $)
$)
```

```
AND pipeo(state) BE
$( LET stream = ?
   TEST state = 'K' THEN stream := keypipe
      OR stream := serpipe
   TEST stream THEN $( WRITES("*NThis pipe is already open - press a key")
                       RDCH()
                    $)
    OR $( stream := FINDOUTPUT("PIPE_256")
          IF stream < 0 THEN $( WRITES("*NCannot open pipe - press a key")
                                RDCH()
                                RETURN
                             $)
       $( LET base,myjob = ?,?
          LET idvec = VEC 2
          $( get.string(idvec,11,"For which job")
             IF idvec%0 = 0 THEN $( CLOSE(stream)
                                    RETURN
                                 $)
             myjob := get.hex(idvec)
             IF ISJOB(myjob) THEN BREAK
             WRITES("*NThis job does not exist - press a key")
             RDCH()
          $) REPEAT
          base := JOBINFO(myjob)
          IF (jinfo2 & #XFF) NE 0 THEN $( WRITES("*NJob active! - press a key")
                                          CLOSE(stream)
                                          RDCH()
                                          RETURN
                                       $)
          PUTPIPE(base << 2,stream)
          TEST state = 'K' THEN keypipe := stream
             OR serpipe := stream
       $)
    $)
$)
```



LISTING 2

```
STMT         SOURCE STATEMENT

* A program to read data from an input pipe controlled by the terminal
* emulator to a named device.
* By Adam Denning 28th December 1984 (C) 1984 Adam Denning
* The output pipe channel ID will be on this job's stack before it is
* activated

IO_OPEN   EQU      1
UT_CON    EQU      $C6
UT_MTEXT  EQU      $D0
IO_FLINE  EQU      2
OPEN_NEW  EQU      2
UT_ERR    EQU      $CC
IO_CLOSE  EQU      2
IO_FBYTE  EQU      1
IO_SBYTE  EQU      5
MT_FRJOB  EQU      5

          SIZE     100

          BRA.S    START
          DC.L     0
          DC.W     $4AFB          Standard format code
          DC.W     8
          DC.B     'PIPETERM'

START     MOVE.L   (A7)+,D3       Get output pipe ID...
          MOVEQ    #-1,D1         ...and use it to open the input pipe
          LEA.L    PIPENAME,A0
          MOVEQ    #IO_OPEN,D0
          TRAP     #2
          TST.L    D0
          BNE.S    FINISH
          MOVE.L   A0,(A7)        save channel ID
          LEA.L    PBLOCK,A1      open a console channel
          MOVE.W   UT_CON,A2
          JSR      (A2)
          BNE.S    FINISH
          MOVE.L   A0,-(A7)       save channel ID

GETOUT    LEA.L    MESSAGE,A1     print prompt
          MOVE.W   UT_MTEXT,A2
          JSR      (A2)
          LEA.L    BUFFER+2,A1
          MOVEQ    #80,D2
          MOVEQ    #-1,D3
          MOVEQ    #IO_FLINE,D0   and collect output file/device name
          TRAP     #3
          LEA.L    BUFFER,A0      now open it for output
          SUBQ.W   #1,D1
          MOVE.W   D1,(A0)
          MOVEQ    #OPEN_NEW,D3
          MOVEQ    #-1,D1
          MOVEQ    #IO_OPEN,D0
          TRAP     #2
          TST.L    D0
          BEQ.S    GOTFILE        if it didn't work, print error message...
          MOVE.L   (A7),A0
          MOVE.W   UT_ERR,A2
          JSR      (A2)
          BRA.S    GETOUT         ...and try again

GOTFILE   MOVE.L   (A7)+,A1
          MOVE.L   A0,-(A7)
          MOVEA.L  A1,A0
          MOVEQ    #IO_CLOSE,D0   close console channel
          TRAP     #2

SENDLOOP  MOVE.L   4(A7),A0       this loop reads a byte from the pipe...
          MOVEQ    #-1,D3
          MOVEQ    #IO_FBYTE,D0
          TRAP     #3
          TST.L    D0
          BNE.S    FINISH
          MOVE.L   (A7),A0
          MOVEQ    #IO_SBYTE,D0   ...and sends it to the output channel
          TRAP     #3
          BRA.S    SENDLOOP

FINISH    MOVEQ    #-1,D1         kill this job
          MOVEQ    #MT_FRJOB,D0
          TRAP     #1

PBLOCK    DC.W     $701           white border of 1 pixel
          DC.W     2              ink colour red, black paper
          DC.W     486            width
          DC.W     32             height
          DC.W     14             X-position
          DC.W     18             Y-position

PIPENAME  DC.W     4
          DC.B     'PIPE'

MESSAGE   DC.W     4
          DC.B     'To: '

BUFFER    EQU      *

          END
```

minal pipe. Both can be open at the same time. Along the keyboard pipe is sent all data typed at the keyboard which comes from the SYSIN channel and which is not an action character. Along the terminal pipe is sent all data originating from the serial line.

Pipe closure simply involves closing the channel. A job connected to the pipe should be written to detect end of file on the pipe (any QDOS I/O routine will return with ERR.EF in D0) and kill itself with MT.FRJOB when it finds it.

Opening a pipe is considerably more involved than closing one, as we need to establish which job the pipe is being opened for. Once a valid job ID is obtained, JOBINFO is called. This returns the job's base address and also sets the low byte of *jinfo2* to indicate if the job is active or not. A job cannot be passed a pipe ID if it is already active. The last machine code routine, PUTPIPE (shown last month) is then used to put the internal channel ID of the output pipe on the job's stack. This means that the job receiving the pipe ID must open it for input. The easiest way of doing this is as follows:

```
PIPENAME  DC.W      4
          DC.B      'PIPE'
JOB_START MOVE.L(A7),D3 put
                    pipe ID in D3
          MOVEQ #-1,D1
                    signify 'this
                    job'
          LEA.L     PIPENAME,
                    A0
          MOVEQ #IO.OPEN,
                    D0
TRAP      #2        perform the
                    open
                    the input pipe
                    ID is now in
                    A0
```

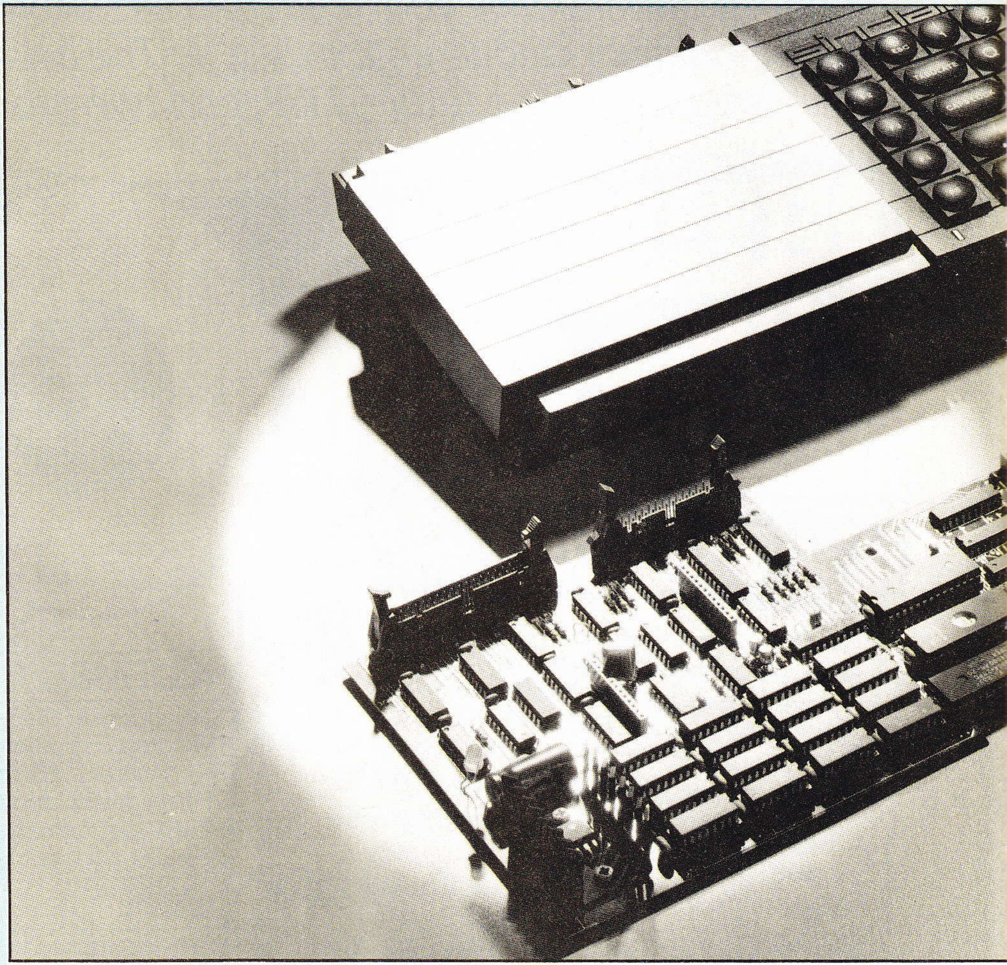
The short machine code program (*listing 2*) is a simple example of the technique. It opens the input pipe and then prompts for an output device. Once the output device is open, the program copies input data from the pipe to the output channel, until an error is discovered on the IO.FBYTE call. When this happens, the job kills itself.

The pipe routines conclude our terminal emulator. Those who have followed the series should now have a working program which will permit the QL to talk to a wide range of computers. Additionally, we hope that valuable insights have been gained concerning QDOS and BCPL.

Any readers encountering problems with their emulator may write to: Adam Denning, c/o QL User Magazine. (The emulator is, in fact, available commercially at a cost of £12.50 from Datamanagement, Tel No: 0904 760 351).

HARDWARE HOR

A compact disk system with built-in memory expansion and parallel printer port from Medic – can we believe our eyes? Paolo Baccanello thinks we can.



This Expansion system marks Medic's debut not only in the QL peripherals market but also in the computer field as a whole. It is all the more surprising then to find a product that would appear to be wholly in tune with the needs of QL owners.

Excepting printers and modems, most devices plug into the QL's main I/O port at the far left of the keyboard. Until recently this has meant that no more than one device may be plugged in at any one time. Users were then confronted with the problem of choosing between buying, for example, disk drives to make up for inadequacies in microdrive storage, or additional RAM to improve the performance of Psion's packages.

Two solutions are now available to perplexed owners. The first, which though simple in conception has proved particularly difficult to implement, has been to design an adaptor to increase the number of 64-way ports from one to four (see July *QL User*).

The second, adopted here by Medic, is to design a sort of multipurpose PCB or cartridge which may be upgraded to support a total of four different types of device – printer, additional RAM, disk drives and modem. Upgrading the cartridge involves installing more chips on the PCB. Installation, we understand, will be undertaken by Medic themselves at a small fee in excess of the upgrade cost.

Memorabilia

In its simplest configuration the Medic cartridge houses additional RAM, from 64K up to 256K. Memory chips used vary according to RAM capacity. At the lower end of the scale (64K & 128K) the unit will incorporate banks of 64 by 1 bit DRAMs. At the upper end (256K) these are replaced by more efficient 256 by 1 bit DRAMs. A 512K option is available but must be powered off Medic's disk drives. In addition to the DRAMs the unit also incorporates a sophisticated RAM controller and an 8K ROM which houses the firmware (drivers) to link the various devices to QDOS.

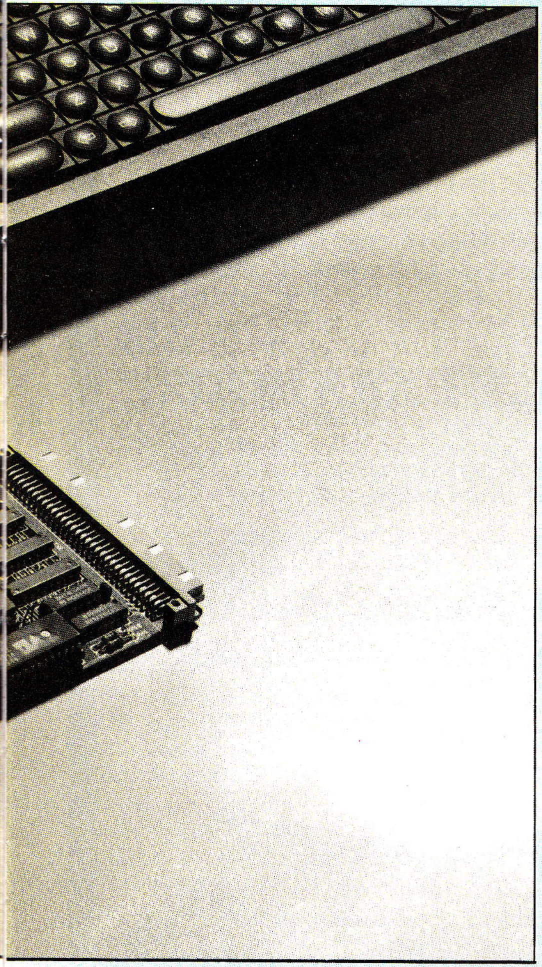
The RAM driver itself, by virtue of a simple FORMAT command, allows you to use extra memory as though it were a number of microdrives. This has the advantage that any I/O operation carried out in memory will be near enough instantaneous. The disadvantage is that, as RAM is volatile, files and programs must be loaded in from and saved to disk or microdrive at the beginning and end of a session. This can be time consuming. However, as a way round this problem Medic supply disk users with a boot program, known as M-SWITCH which automatically loads Psion's packages into memory and then permits the user to move from one to another in under a second. This 'psuedo' integration greatly enhances the programs' performance.

In addition to extra memory, Medic's expansion cartridge houses a disk interface. The controller appears alongside the 8K ROM on the PCB. As with most systems, the firmware has been written by Tony Tebby, author of QDOS, and conforms to the Sinclair standard with disks formatted 9 bytes per sector, 512 sectors per track and 40/80 tracks per disk. This means disks formatted on other (3.5″) systems may be interchanged with those formatted on Medic's drives.

Not surprisingly, as the cartridge's 8K ROM houses firmware relating to three other devices, Medic have been unable to fit in the usual assortment of extra file I/O and job control commands found on other systems. Nevertheless they have found space for microdrive emulation and extended the facility to cover additional memory as well. Using the simple commands FLP_USE or RAM_USE it's possible to assign any three letter device name to floppy or RAM disks respectively. More specifically, by assigning the name *mdv* to either device users will be able to run programs written for microdrive without the need to run an elaborate conversion program.

If there is little icing on Medic's ROM, then the company more than makes up for it with the free software they give away on disk. This we are told will be made available to those ordering disk interfaces (users without Medic disk drives will be

IZONS IV: MEDIC



required to specify the medium upon which they wish to receive the software).

Along with an enhanced version of the switching program mentioned earlier, Medic will be giving away an abridged version of Tony Tebby's *QL Toolkit,* a function key definer, an integrated accounts package, a mailshot program, file compression and backup routines and a spelling checker with a 3,000 word basic vocabulary which may be extended by the user. Unfortunately, only MSWITCH and MKEY were available at the time of review. Both programs performed satisfactorily, though the review versions, inadequately error trapped, were clearly not the final product. For this reason they will be reviewed separately as soon as they are made available.

Test Drive

Medic's disk interface has been designed to work with any standard Shugart compatible disk drive (3″, 3.5″, 5.25″). The interface may be powered off either the QL or from an external supply. Medic themselves provide single or dual 720K (formatted) 3.5″ Mitsubishi drives with built-in power supply and cables. The dual drives reviewed here were pre-production models, though no problems were encountered with their operation (refer to the`Bench marks). The drives operated silently with only the pilot lights at their front betraying any sign of activity. Additionally, the 3.5″ disks, enclosed in a rigid plastic cover and well protected from the elements, represent a distinct improvement upon their 5.25″ flexible cousins. However, at £6 a shot they certainly should do!

Included in the price of the disk interface is an 8-bit parallel printer port. As most low cost printers on the market are geared for parallel input the device should find use almost immediately. The port comes with its own QDOS driver, is assigned the name PAR and uses commands consistent with those governing the QL's serial port (except that here the user need not worry about such things as baud rate, parity and handshaking). The port has two distinct advantages over the QL's RS232 port. First, it can harness available RAM to act as a buffer (known as '*spooling*') which dramatically reduces the amount of time the computer is 'printer bound'. Second, transmission rates are limited only to the speed at which the printer is capable of digesting information.

Located at the centre of the PCB are two slots destined to house Medic's modem. At the time of review the unit was still at the design stage. However, we are informed that it will incorporate a 6580 (2MHz) UART and probably a 7911 modem chip. It is also worth noting that as the latter makes use of the QL's serial port (*ser2*) owners of the Medic system will not be compelled to wait and see how the unit turns out.

There can be little doubt that Medic have set out to manufacture the definitive expansion system and have, to a large extent, succeeded. They have taken all that is currently available for the QL and put it into a single box. This formula and a highly competitive pricing structure gives them a clear advantage over the competition. The promise of an attractive and well thought out design, matching accessories and free software all add to the appeal. However, with new products being announced weekly, Medic would do well to go into volume production as soon as possible – a consideration which has let them down in the past.

Prices (including VAT)
Extra Memory

64K	£99.95
128K	£129.95
256K	£169.95
Disk interface plus	£129.95
1×720K 3.5″ disk drive	£249.95
2×720K 3.5″ disk drives	£399.95
512K extra memory	£200.00
Modem	£120.00
Numeric Keypad	POA
Keyboard Cover	POA

Supplier: Medic Datasystems Ltd,
76 Grainger Close, Basingstoke,
Hants RG22 4EA

Bench marks

QL User bench marks show timings in seconds per single Archive operation. They are carried out on files containing 50 records with 20 fields apiece and an average of 10 randomly generated characters per field. A proportion of each operation must be attributed to software run time. This is especially true of *append* used to generate a record and its contents as well as write it out to disk or microdrive. The bench marks are designed to highlight improvements in performance whilst a program is running and not just in its loading time.

Bench mark	Micro-drive	Medic Disk
Create	37.00	4.00
Append	14.16	7.68
Display	4.24	3.88
Order	1.24	0.12
Locate	0.22	0.06
Find	0.66	0.36
Search	0.35	0.15

Note: The Medic System bench marked was the full 512K system with dual 720K drives. To prevent Archive from grabbing extra RAM for itself 1,000 sectors of memory were reserved using the command FORMAT RAM3_1000. Readers wishing to compare results with other QL disk systems should refer back to *QL User* June issue (p 10) (where it should be noted that Computamate's latest 1.09 version *does* include microdrive emulation).

New

SINCLAIR **QL** SOFTWARE BY

SHADOW GAMES

QUAZIMODO

Help our hero rescue Esmerelda. He must jump the battlements and avoid soldiers, rocks, arrows and the mad axeman. A popular concept skilfully converted for the QL. Exceptionally smooth and detailed graphics with 15 screens of brilliant arcade action.

£12.95

And

SPACE PARANOIDS

(Release date 1 August)

A fast moving alien zap game with 9 different types of alien and 40 different attack waves. Joystick or keyboard. You'll keep coming back for another go.

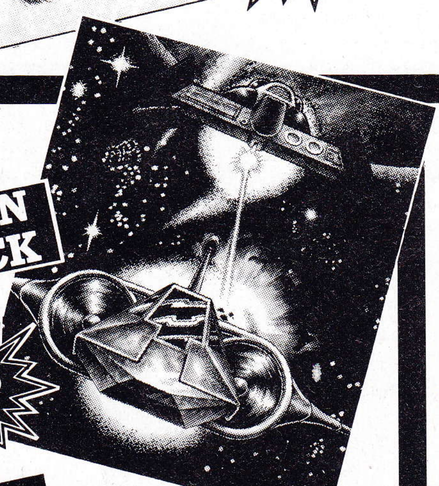
£12.95

GALACTIC INVADERS & STAR GUARD

TWIN PACK

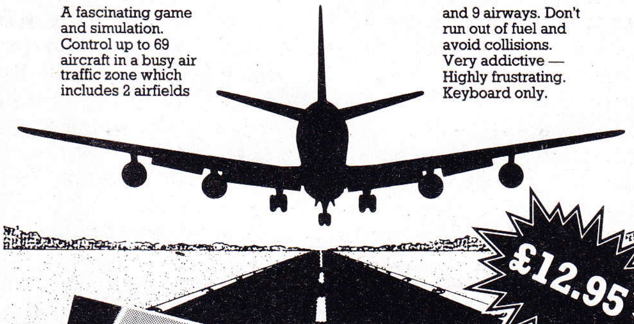
Unbeatable value, 2 good arcade games for the price of one. Keyboard or joystick.

£14.95



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.

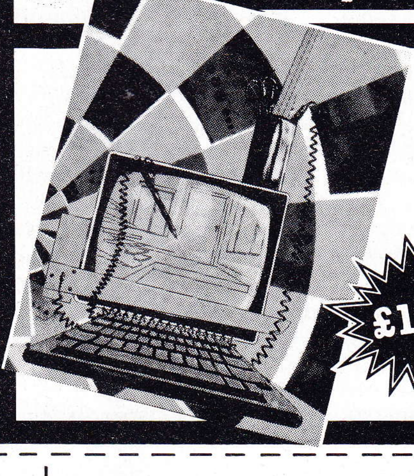


£12.95

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 joystick

£12.95



PAINT MASTER

A Graphics Package and Mini-Adventure. Keyboard or joystick.

£14.95

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

	cost
..... Quazimodo @ **£12.95**	....
..... Space Paranoids @ **£12.95**	....
..... Night Nurse @ **£12.95**	....
..... Galactic Invaders/Star Guard @ **£14.95**	....
..... Area Radar Controller @ **£12.95**	....
..... Paint Master @ **£14.95**	....
TOTAL	....

ALL PRICES INCLUDE VAT, POSTAGE & PACKING

Name ....

Address ....

....

....

Please make cheque or postal order payable to Shadow Games and send to:

Shadow Games
70 Gooseacre, Cheddington
Nr 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.

ARTICULATING WITH ARCHIVE

THE THEORY OF RELATIVITY

Genealogy remains the theme in the second episode of our series on how to use Archive. Dr Andy Carmicheal presents a set of procedures for displaying, editing and inputting information using a 'family' tree database as an example.

In recent years there's been an upsurge in interest in discovering one's "roots" and there are many books around to help with the process of research – once you've reached the limit of your grandfather's memory of course. But how are you going to store all that information when you've discovered it? The answer is to harness the power of the QL and Psion's versatile "Archive" package.

The main problem with drawing a family tree on paper is that once you've drawn it, there's never room to record any further information that you discover – its size is fixed. It's also more or less impossible to record more than one main branch of the family on a single sheet of paper, so usually the in-laws' side of the family gets left out in every generation, or maybe has to be recorded elsewhere. Although there's less room on a monitor screen than a large piece of paper (so we can't show as much information at one time) the great advantage of a computer-based method is its versatility. Using Archive we are able to record as many relations we wish and add them as and when they are discovered; even unrelated sections of family trees can be held in the same database. Having recorded the data, pressing one of the arrow keys (or one of several other keys) moves you round the family tree to any direct relation, or to a named individual. That person's immediate family (parents, spouses and children) can then be displayed on the screen. The process of recording and displaying family trees in this way is fascinating in itself, and most educational – what better way to teach the Kings and Queens of England for example!

The Archive package itself offers an ideal means of keeping track of all kinds of information. Addresses, club records, collections, garden plants and seeds, book or tape libraries, even your computer programs are all obvious examples of the sort data that can be stored and retrieved using the database program. Furthermore Archive allows us to write procedures to control our data in a language quite similar to BASIC. This can take out a lot of the tedium of repeatedly entering the same commands, and (as in this example) can also be used to ensure that data files are updated in a consistent manner. Simple information can usually be held in a "card index" type database file but other information is more complex and requires data to be held on a number of different kinds of entity, and relationships between entities also recorded. Unfortunately family tree data falls into the latter category. The program given here uses four data files in all: *people (P)* recording details of the individuals, *families (F)* recording which people are in any particular family, and two further files, *marriages (M)* and *parents (Pa)*, which contain no new information but which speed up the process of finding a spouse or parent of a given person. There are practically no limits by the way on the number of children in any family, nor on the number of spouses of any individual (Henry VIII – no problem!).

Last month's article gave the procedures (to be saved in the files *relation_prg* and *relat1_prg*) for setting up the data files and beginning to enter the data. This month's listing gives the remaining procedures (to be stored in *relat2_prg*), which display the family trees graphically and control the consistent entering and editing of data. This is most important since the database will get thoroughly confused if the information in the *families* file disagrees with that in either the *marriages* or *parents* file. (To enter the procedures first load Archive and then use its own built-in editor accessed through the command **edit**.)

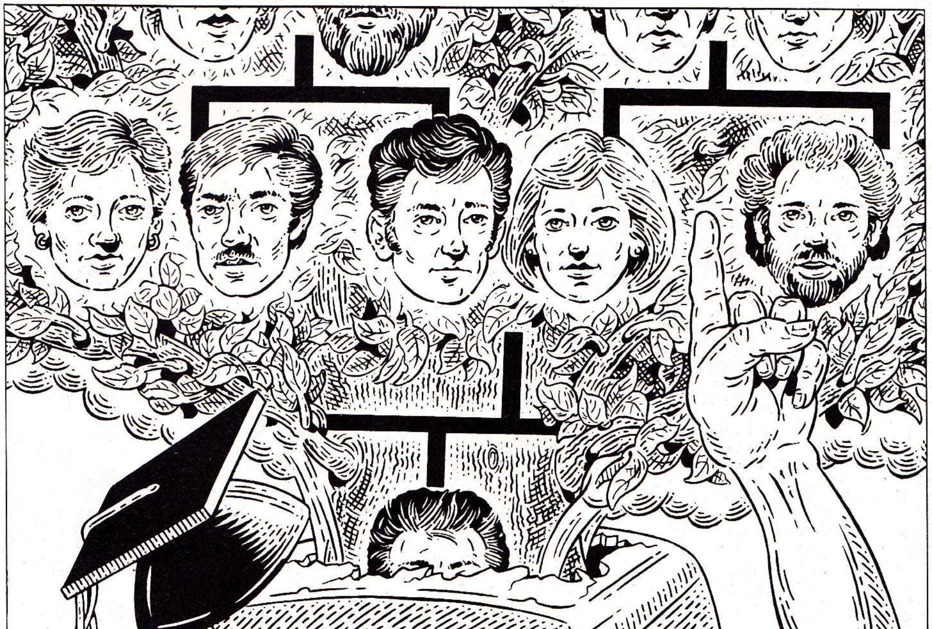


ILLUSTRATION BY STEPHEN WRIGHT

Some of the procedures in this month's listing are duplicates of the ones used in *relat1_prg* since they are also needed in this second phase. As you no doubt will have already typed them in once to re-enter them you need only load in last month's *relat1_prg*, enter the editor and delete [F3+D] those procedures NOT used in this month's *relat2_prg*.

START-UP

To set the ball rolling the user must enter the command **run "relation"**. The **run** command loads the specified program file, in this case *relation_prg*, looks for a procedure called **start** and if it exists, automatically calls it. This device is particularly useful as it allows one to execute or "chain" programs in a specific sequence without interruption. In our case the chain is set up so that *relation_prg* when complete will run *relat1_prg* which in turn will run *relat2_prg*.

Splitting procedures down into three separate files and chaining each in turn may seem puzzling. However, to do so saves memory for only those procedures which are required for a particular function are resident in memory at any one time leaving space for data.

Ignoring *relation* and *relat1* which were discussed last month, the procedure **Start** in *relat2* calls another procedure, **Go**, using the error trap (described later). Then after **Go** has completed (usually after <ESC> has been pressed) it checks for errors and if required closes the data files by calling **Bye**.

The procedure **Go** contains the main loop which is continually executed by the program. Within the loop, the current subject's name is printed (using procedure **Prin**) along with the amount of memory remaining and then **Menu** is called which responds to the next key-press (for example in the case of <UP> being pressed, moving the subject to the current subject's father). All the other procedures are called directly or indirectly from **Menu** to carry out the functions required by key-presses (see table 1). The procedure **Details**, for example, displays the family tree, calling a number of other procedures (**Prin, PrinFam, PrinKids** etc) in order to do so.

Archive provides a simple way of trapping errors. When a procedure call is preceded with the keyword **error** any mistakes in input or problems loading or saving files will not crash the called procedure (or any procedures called from within it) but simply cause it to terminate returning control to the calling procedure. To find out whether an error has in fact occurred during execution of a procedure the function **errnum()** should be called. This will return a non-zero value (ie an error code) if execution has been prematurely terminated. Examples in the listing of where error trapping has been used are the procedures **Start** when calling **Go** and **Bye**, and **Menu** when calling **EdFam** and **InFamily**.

A number of procedures appearing both in *relat1* and *relat2* may be regarded as general purpose. They are **bye, cenprint, yesno, getrec** and **chop**. As these will find use in most applications and not just setting up family trees, we look at them in greater detail.

Bye closes any files that happen to be open at the time by calling **close** within a perpetual loop. As a result this is bound to raise an error sooner or later, but by using the error handling facility described above we can avoid error messages and by this token ensure that all files will be closed without having to name them.

CenPrint is a procedure for centring output on the screen within a specified number of characters. It

```
proc Bye
  rem *** Close all files that may be open.
  rem *** An error will be caused by this
  rem *** proc and must be trapped.
  while 1: close : endwhile
  endproc
proc CenPrint;S$,Length
  rem *** Centres string within given length
  if len(S$)>=Length: print S$;: return : endif
  let T=int((Length-len(S$))/2)
  print rept(" ",T);S$;rept(" ",Length-T-len(S$));
  endproc
proc Chop;X$,n,Delim$
  rem *** Sets C$ to given string with first n
  rem *** 'words' removed
  local Words,Dlim
  let Words=n: let C$=X$
  let Dlim=instr(X$,Delim$)
  while Words
    if not Dlim or Dlim>=len(C$): let C$=""
      return : 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 assoc-
  rem *** iated with family "Fam"   (Does not
  rem *** 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$,1,","
    endwhile
  delete "F"
  print "*** Family DELETED ***"
  endproc
proc DelMar;Who,Fam
  rem *** Deletes marriage of "Who" if in family
  use "M": locate Who
  while M.RefNo=Who
    if M.Family=Fam
      delete "M"
      else : next : endif
    endwhile
  endproc
proc DelPar;Who,Fam
  rem *** Deletes parents of "Who" if in family
  use "Pa": locate Who
  while Pa.RefNo=Who
    if Pa.Family=Fam
      delete "Pa"
      else : next : endif
    endwhile
  endproc
proc Details;N
  rem *** Displays details of selected person
  MumAndDad;N,0
  cls : print at 5,0
  prin;dad,39,0
  print ink 4;"=";:prin;mum,40,0
  print tab 39; ink 4;"!"
  prin;N,80,1:Spouse;N
  if Spouses
    PrinFam;val(Fams$),1
    Chop;Fams$,1,",": let Fams$=C$
    while Fams$<>""
      print : paper 2:prin;N,80,0: paper 0
      PrinFam;val(Fams$),1
      Chop;Fams$,1,",": let Fams$=C$
      endwhile
    endif
  endproc
proc EdFam
  rem *** Edits subject and his/her relationships
  cls : print "Change personal details of ";
  prin;Subject,0,0:YesNo;"? Yes or No?"
  if yes
    use "P": locate Subject: alter : cls
    endif
  Spouse;Subject
  while val(Fams$)
    print "Do you wish to DELETE the subject's foll
owing relationships?"
    print :PrinFam;val(Fams$),1
    print :YesNo;"Yes or No? "
    if yes: paper 2:YesNo;"ARE YOU SURE?": paper 0
      if yes:DelFam;val(Fams$): endif : endif
    Chop;Fams$,1,",": let Fams$=C$
    endwhile
  while 1
    print "Input new family of ";:prin;Subject,0,0
    YesNo;"? Yes or No? "
    if not yes: return : endif
    if P.Male: let Dad=Subject: let Mum=0
      else : let Dad=0: let Mum=Subject: endif
    InFamily;P.Surname$,Mum,Dad
    endwhile
  endproc
proc ErrMess
  rem *** Prints error message
  cls : print paper 7; ink 0;" ERROR updating datab
ase "
  print "(Possibly caused by write protection on fi
les)"
  print "Press any key to continue..."
  let a$=getkey()
  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
  rem *** 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 Go
  rem *** Main procedure
  mode 0,8: let Ex=0
  Details;Subject
  let Stopping=0
  while not Stopping
    paper 6: ink 0
    print at 0,0;:prin;Subject,32,0: paper 0: ink 6
    print rept(" ",36);memory();" Bytes  "; at 0,0
    if memory()<999
      print "*** WARNING *** Memory low - Close & b
ackup files"
      endif
    Menu
    endwhile
  endproc
proc InFamily;Df$,M,D
  rem *** For inputting a family. Default surname
  rem *** is Df$, RefNo of Mother (M) or Father
  rem *** (D) may be given
  cls : print "Insert new family..."
  while 1
    use "F": last
    let F.Family=F.Family+1
    print "Father is ";
    if D:prin;D,0,0: print : else
      let Sx=1: let Def$=Df$:GetPerson: endif
    let F.Father=P.RefNo: let ChN$=P.Surname$
    print "Mother is ";
    if M:prin;M,0,0: print : else
      let Sx=0: let Def$="":GetPerson: endif
    let F.Mother=P.RefNo
    input "Date married? ";F.DateMarried$
    input "If divorced, give date: ";F.Dissolved$
    if F.Dissolved$<>""
      let F.Dissolved$="-"+F.Dissolved$: endif
    let F.Children$=""
    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
    if Subject=F.Father or Subject=F.Mother
      print :PrinFam;F.Family,1
      else : let Subject=F.Father:Details;Subject
      endif
    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
  rem *** (setting 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 Menu
  rem *** Responds to key presses
```

uses the function **len**(str$) which returns the length of the string.

YesNo waits for the reply Y(es) or N(o) to a given prompt. If any other key is pressed it merely asks the question again.

GetRec is a variation on a procedure in the Archive manual and will work with any database file, regardless of how many fields there are in the records or what they are called. It uses the command **find** to find a record containing the input string, then displays its contents and asks if this was the record you wanted. If not it looks for the next occurrence of the string.

The procedure **Chop** is useful for string manipulation. It chops characters of the string up to and including the first *n* occurrences of the substring *Delim$*, and puts the result in the variable *C$*. It's rather useful in this program as children in a family are stored as a string of numbers separated by commas (unfortunately there are no arrays defined in the Archive language). No doubt it could find many other uses though.

UP AND RUNNING

With the programming details behind us when can now move on to the actual running of our family tree database. Table 1 gives the valid keys which are used in the program and their effect. To use "E" (edit) or "I" (insert) you must have specified that you wanted to modify the database in reply to the prompt in phase 1, otherwise an error message is printed on the screen (by the procedure **ErrMess**). <SHIFT> = is used in the case of a person married more than once to move the subject to the next spouse, but first you must use "=" on its own.

When modifying the database with either "E" or "I", the program asks a series of questions about the family to be entered. Questions can usually be ignored, for example, if you don't know a date. However the people in a family, parents and children, must be entered together and if you make a mistake, either press <ESC> or reply "N" when asked if satisfactory and resubmit the details. It's done this way to keep the files consistent with each other and the program relatively simple. If you do delete a family note that only the relationships and not the records of the individuals are erased.

Also when the program asks for a person, eg:
Father was who?
you can EITHER reply with a name (or part of a name), in which case the database is searched for all occurrences of the string and a confirmation is asked for you to identify the correct person, OR if the person has not yet been entered in the database, type <ENTER> and a new person can be entered.

KEY PRESS	ACTION CARRIED OUT
<UP>	Moves subject to father (if known)
SHIFT <UP>	Moves subject to mother (if known)
<DOWN>	Moves subject to first child (if known)
<LEFT>	Moves subject to eldest brother or sister (if any)
<RIGHT>	Moves subject to next brother or sister (if any)
=	Moves subject to husband or wife (if any)
SHIFT =(ie. +)	Moves subject to NEXT spouse (if any). Only valid after "=".
?	Moves subject to named person
<SPACE> or D	Displays family tree of current subject
<ALT> +any-key or E	Edit details of current subject and his relationships
I	Insert a new family
<ESC>	Exit

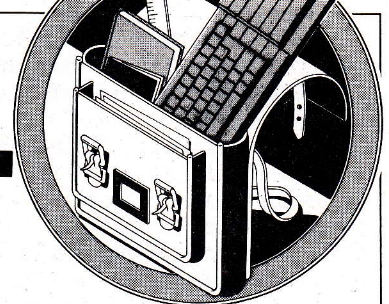
Finally a word of caution. Memory can get short with so many procedures taking up RAM and Archive is not totally reliable with less than 1K of remaining memory. If this seems the case, exit from Archive and back-up your data. If the program crashes while you are entering data ***all*** will be lost!

```
  let Z$=getkey(): let Z=code(Z$)
  if Z=27:rem (<ESC>)
    let Stopping=1: return : endif
  if Z=0 or Z$="e":rem (E or <ALT>+any_key)
    error EdFam: if errnum():ErrMess: endif
    Details;Subject: return : endif
  if Z$="i":rem (I for insert)
    error Infamily;P.Surname$,0,0
    if errnum():ErrMess: endif : return : endif
  if lower(Z$)=" " or Z=30:rem (SPACE or ENTER)
    Details;Subject: return : endif
  if Z=2 or Z=10:rem (Up or SHIFT Up)
    MumAndDad;Subject,Z: return : endif
  if Z=3 or Z=11:rem (Down or SHIFT Down)
    ToChild: return : endif
  if Z=5 or Z=4:rem (Left or Right)
    ToSibling: return : endif
  if Z$="=" or Z$="+":rem (= or SHIFT =)
    ToSpouse: return : endif
  if Z$="?":rem (SEARCH)
    ToSearch: return : endif
  endproc
proc MumAndDad;N,Option
  rem *** Sets mum & dad for subject N
  let dad=0
  let mum=0
  use "Pa"
  locate N
  if Pa.RefNo=N
    use "F"
    locate Pa.Family
    if F.Family=Pa.Family
      let dad=F.Father
      let mum=F.Mother
      if Option=2: let Subject=dad: endif
      if Option=10: let Subject=mum: endif
      endif
    endif
  endproc
proc Prin;N,Length,Extra
  rem *** Output name of person  N  centred in
  rem *** field of given length. If Extra=1
  rem *** additional details output (in red)
  local T$: use "P"
  locate N: if Extra: paper 2: ink 7: endif
  if (N<>P.RefNo or N=0)
    let T$="unknown to database"
    else
    let T$=P.Firstnames$+" "+upper(P.Surname$)
    endif
  CenPrint;T$,Length
  if Extra
    if P.Male: let T$="(M) "
      else : let T$="(F) ": endif
    CenPrint;T$+P.DateOfBirth$+" - "+P.DateOfDeath$
,Length
    CenPrint;P.Comments$,Length: paper 0
    endif
  endproc
proc PrinFam;N,plus
  rem *** Output details of family
  local Sp
  use "F": locate N: ink 4
  if N<>F.Family
    CenPrint;"Family unknown to database",80*plus
    else
    CenPrint;("= ("+F.DateMarried$+F.Dissolved$+")"
),80*plus: ink 7
    if F.Father<>Subject
      prin;F.Father,80,0
      else
      prin;F.Mother,80,0
```

```
    endif
  if plus:PrinKids: endif
  endif
endproc
proc PrinKids
  rem *** Output the kids of the current family
  let Ch$=F.Children$
  if not val(Ch$): return : endif
  let Ch=1
  let B1$="+"
  let B2$="|"
  Chop;Ch$,1,","
  while C$<>""
    let Ch=Ch+1
    if Ch<6
      let B1$=B1$+"-----------+"
      let B2$=B2$+"           |"
      endif
    Chop;C$,1,","
    endwhile
  let Sp=1
  if Ch<5
    let Sp=31-6*Ch
    endif
  print tab 39; ink 4;"|"
  print tab Sp+14; ink 4;B1$
  print tab Sp+14; ink 4;B2$;
  let C$=Ch$
  while val(C$)
    print : print tab Sp;
    prin;val(C$),30,0
    let Sp=Sp+12
    if Sp>55: let Sp=6: endif
    Chop;C$,1,","
    endwhile
  endproc
proc Spouse;N
  rem *** Sets Spouses & Fams$ for person N
  use "M": locate N
  let Spouses=0: let Fams$=""
  while N=M.RefNo and not eof()
    let Spouses=Spouses+1
    let Fams$=Fams$+str(M.Family,2,0)+","
    next "M": endwhile
  endproc
proc Start
  rem *** Start-up procedure
  ink 2: print "Press any key to continue..."
  let a$=getkey()
  ink 7: error Go: mode 1,8
  if errnum() and errnum()<>27
    print "Error ";errnum();" occurred": endif
  YesNo;"Close files? ": if yes: error bye: endif
  endproc
proc ToChild
  rem *** Sets subject to 1st child
  use "M"
  if Subject<>M.RefNo: locate Subject: endif
  if Subject=M.RefNo
    use "F": locate M.FAmily
    let N=val(F.Children$)
    if N: let Subject=N: endif
    endif
  endproc
proc ToSearch
  rem *** Sets subject to new person
  cls : print "Search for ";:GetRec;"P"
  if yes: let Subject=P.RefNo: endif
  cls :Details;Subject
  endproc
proc ToSibling
  rem *** Sets subject to brother or sister
  use "Pa": locate Subject
```

```
  if Subject<>Pa.RefNo: return : endif
  use "F": locate Pa.Family
  if F.Family<>Pa.Family: return : endif
  if Z=5 and val(F.Children$)
    let Subject=val(F.Children$): return
    endif
  if Z=4
    let X$=F.Children$
    while 1
      Chop;X$,1,","
      if not val(C$): return : endif
      if val(X$)=Subject
        let Subject=val(C$): return
        else : let X$=C$: endif
      endwhile
    endif
  endproc

proc ToSp;N
  rem *** Sets Subject to spouse of N
  rem *** if current M-record OK
  if N<>M.RefNo: return : endif
  use "F": locate M.Family
  if N=F.Father: let Subject=F.Mother: else
    if N=F.Mother: let Subject=F.Father: endif
    endif
  endproc
proc ToSpouse
  rem *** Sets subject to (next) spouse
  local Fnd: use "M"
  if Z$="=": let Ex=Subject: locate Subject
    ToSp;Subject
    else : locate Ex: let Fnd=0
    while Ex=M.RefNo and not Fnd
      use "F": locate M.Family
      let Fnd=(Subject=F.Father or Subject=F.Mother
)
      next "M": endwhile
    if Fnd:ToSp;Ex: endif
    endif
  endproc
proc UpSubs
  rem *** Update "Marriage" and "Parents" files
  use "F"
  let M.Family=F.Family
  let M.RefNo=F.Father
  append "M"
  let M.RefNo=F.Mother
  append "M"
  let Pa.Family=F.Family
  let C$=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 "yes"
  while 1
    print P$;: let Q$=lower(getkey())
    let yes=(Q$="y")
    if instr("ny",Q$): print " "+Q$
      return : endif
    print : endwhile
  endproc


NOTE:  The comments in the procedures are for
       information only but since they use  up a
       lot of very useful memory they should not be
       entered.  Saves typing time too!
```

QL AT SCHOOL



Following last month's article on one school's application of the QL, Mary Sargent investigates its wider use in education.

The fact is that the government sponsored MEP (Micro-Electronics Programme) scheme – designed to put a micro into every school in the country, and provide basic training for the teachers who would use them – was implemented in many areas with a minimum of understanding and goodwill, to the point where some areas still have not appointed Computer Liason Officers in schools and where information and advice on computer technology is very hard to come by. Even in regions where the scheme has apparently been implemented as the Government intended, there are individuals who take the attitude that one computer in each school is sufficient to teach computer awareness and that the majority of pupils will not need to have any closer encounter with a micro than can be gained in a half-hour class every two weeks. Teachers and pupils have not been the only victims of the patchy and inadequate MEP initiative. Sinclair Research, the company every one loves to hate, has fallen foul of this type of official attitude in a way which effectively means that the QL is unlikely ever to figure large in statistics on machines used in schools.

Distinctions, however, must be drawn. It's true that Sinclair Research is regarded with official reserve by many education authorities, and few will give active support to schools seeking to install Sinclair machines. This applied even to the Spectrum, of which many in-service teachers approved, on the grounds that it could be bought in sufficient numbers to make CAL (Computer Assisted Learning) a feasible option as opposed to a useless principle. It is almost universal when the QL is in question.

The official line in Cambridgeshire, ironically, is so anti-Sinclair that one headmaster who refused to confine his school micros to the prescribed BBC has met with downright hostility, only alleviated by the timely arrival of a Computing Adviser more concerned with education than politics. The Bradford Authority takes the line that any school buying a QL does so on its own, and there will be neither money nor encouragement available from the Authority. Sinclair may be forgiven for feeling that it has not been given a fair chance to prove the worth of its micros educationally.

However, the company, according to PR man, Julian Goldsmith, is undaunted. When Acorn's schools contract is up for renewal, Sinclair will tender for it again. Education was one of several markets identified as a possible habitat for the QL when it was launched, and there has been no fundamental change in attitude. With the launch of Pandora the portable Spectrum, scheduled for 1986, however, the emphasis on the QL as an educational machine may shift from secondary to tertiary level, although that doesn't mean schools at a lower level of education won't find uses for it. "Education is a tough market to crack," says Goldsmith, because the system is stacked against free choice, with the monopoly in the hands of Acorn and RML (Research Machines Limited), and the schools which take the line of least resistance receiving the most support from the Education Authorities.

Erik Elsom-Cook would agree with him. As the Headmaster of a primary school in St. Neots which has one BBC machine courtesy of the MEP and many more Spectrums as a result of school fund-raising, he claims that political considerations determine the decisions as to which micros get official approval. He would dearly love to have a QL in his school. The standard of computer literacy there is well above the national average, and he says that some of his eleven year-olds would benefit from the sophistication of the QL. He also makes the point (apparently trivial until you're called upon to cope with a room full of computers and children) that the QL presents a compact unit, without trailing wires which can be yanked out and misplaced by over-enthusiastic knowledge seekers. That alone would make it worth a place in the classroom!

Grange Secondary School in Buckinghamshire is a traditional secondary modern school with a sixth-form which offers A-level courses. It is also out of the ordinary because it has a QL. Peter Buchanan, Head of Maths and Computing at the school is very enthusiastic. He uses it for administration, keeping track of children's progress through the school, updating subject sets and generally dealing with a high volume of what would otherwise be paperwork. So far as the pupils are concerned, he uses Quill with a lower-ability group of fourth years who are producing a magazine using the word-processing and editing facilities, and he says that if the final result is unsatisfactory, it won't be the fault of the QL.

The New Learning Initiative fund, a government backed scheme to develop teaching aids for the least able children in schools, was used to acquire a QL for the Cherwell Upper School in Oxford. Because the fund has nothing to do with the MEP, and the machine is seen as a teaching aid rather than specifically as a computer, its use is restricted to a group of 14 to 16 year-olds whose educational needs are such that special courses are designed for them outside the main school curriculum. The six-month computing module of the course is devoted to such things as writing application letters with Quill, general word-processing and an overall view of a computer's functions and capabilities. The Head of Computer Studies at the school, Francis Glassborow, chose the QL because the bundled software which came with it was ideal for the NLI course, but although pleased with it, he recognises that the machine's teaching potential is restricted at Cherwell. He has no funds available to buy more QLs, and considers that the 68008 processor is sufficiently complex to deter many teachers. He himself uses the machine for administration.

Sinclair Research itself is unaware of any particular interest shown by educational institutions (apart from Strathclyde) and it's a case of seeking out places where individual teachers appreciate the QL's virtues. In Henley-upon-Thames, King James' Sixth Form College harbours three QLs, which were introduced by the Head of Computer Studies Stella Kendall, and are used both to demonstrate the benefits of computers on a general Computer Appreciation course and as advanced programming machines by A-level students. At one stage, they were also seriously considered for use by Martin West, Head of Computerised Administration at the College, but as he explains there are problems.

Microdrives are too slow for administration purposes. The database at the College is 360K and on the American Vector machine he's currently using, Mr West can access any file in 0.5 seconds. A microdrive system would require four microdrives to cope with the quantity of data and take about four minutes to get the same result, "And in any case, you can't write random files on the QL because of the microdrive system, and serial access takes forever."

The system is also inadequate to store the amount of data he is dealing with. There are 710 students at the college, 94 subject options and a possible range of five subjects in various combinations for each student. The QL could cope – the microdrives' storage capacity couldn't. If, however, Sinclair could produce a QL with disc access for £700, (that being the cost of the Nimbus, RML's 16-bit machine currently under consideration by various Education Authorities) Mr West would be more than happy to transfer the College's administration system to the QL.

He owns one himself which he programs for fun, claiming that not only has it the best graphics of any machine yet, but is also the easiest micro to program that he's tried. Despite this, and the fact that he is highly competent, having designed and written the administration programs currently in use at King James', he cites lack of software as another reason for not using the QL as an administrative machine. "We didn't have time to write the programs for it, because it's yet another language and yet another operating system".

Stella Kendall's attitude is slightly different, because she looks at computers from a different angle. Whereas Martin West needs a fast and efficient tool with a massive memory and equally massive storage capacity, Mrs Kendall wants a machine which displays well, demonstrates a large number of functions and demands structured thinking and logical programming. She finds the documentation for the QL poor and would have liked more information on QDOS, but the 68008 processor is not a problem. She aims to teach a structured approach to the principles of programming, and once the method of breaking information down into sections and a general knowledge of how to apply it is taught, then individual peculiarities of particular machines are quickly dealt with.

Reaction from her students to the QL's software has been positive. "My General Sixth (a low-ability group) find Quill easy to use. In fact they prefer it to the program recommended by Oxfordshire for primary schools. (This is RML's version of Wordstar.) It's lovely." Nevertheless, it will be the quantity and type of software available which will determine the QL's future at King James' College. Ironically, Computer Science A-level is not accepted by many Universities as a qualification for students intending to read Computer Science at degree level! Stella Kendall has already lost two of her more serious students, one of whom was told by Cambridge that she must take two Maths and one conventional Science A-level to gain entrance. The result is that the Computer Science A-level is taken as a second or third subject, and the students who might be most capable of using the QL to its full programming potential are religiously studying Physics, Maths and Chemistry instead.

So back to the packaged software problem and it's there that the QL can't compete with the officially approved educational machines. In addition to competition from RML and Acorn, Sinclair now has to contend with a new factor. "As far as we're concerned," says Mrs Kendall, "Apricot have suddenly dropped their prices to schools, and Macintosh are starting to look reasonable as well." Nevertheless, if software becomes available, she would be happy to promote the QL in her school.

Sinclair Research are aware of the problem. Chris Clifton, Educational Software Expert for the company has a number of projects on the drawing board. Among them are plans for LOGO, the Spectrum version which was well-thought of, and has now been accepted as a prime educational language, and Pascal and Fortran which would extend the QL's application well into university level education. A statistics tutorial package and a paint routine which uses a mouse take the QL back into Mrs Kendall's territory, whilst interactive programs and a Business simulation reinforce the work she does with students intending to go into industry or pursue business studies. There are as yet no launch dates attached to any of these development plans, but clearly, Sinclair are rather more in touch with the needs of their potential markets than the QL's shaky start indicated.

Hadrian James, Head of Computer Studies at Bulmershe Comprehensive School in Reading, would find a number of the proposed packages useful. Mr James is unusual in the context of schools and computers. Although he has taught Maths, he is in fact a graduate in Computer Science. As a result he is entirely at ease with and very knowledgeable about computers, and also has great enthusiasm. In three years, he has created a computing department where before there was one machine, a Piccolo, and plans to continue the expansion until he feels that the facilities are adequate to cope with the school's 1400 pupils. Starting with the present third years, every child who attends Bulmershe will receive instruction on, and about computers, either at a general level or in depth, according to the pupil's aptitudes and interests. The school has a high number of university entrants and Mr James is hopeful that the present anomaly regarding A-level Computer Science and degree courses will be resolved.

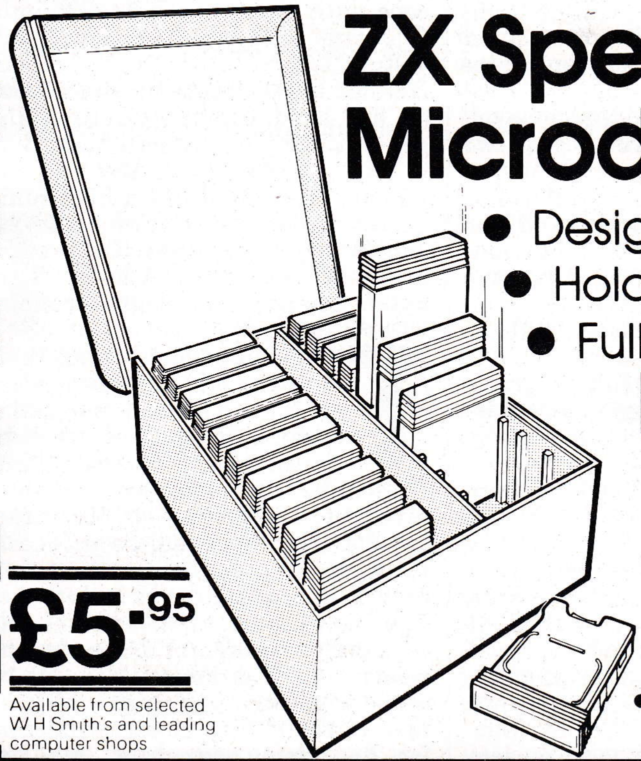
The Psion packages that come with the QL have been very useful so far, in particular Quill and Archive, and he considers the school's three QL's to have been an excellent investment. Because of its structured Basic, he considers it ideally suited to his aim of educating children to use computers as tools rather than as an end in themselves. He does not own a micro, although if he needs to work at home, it's the QL he takes with him, and with which he used to write simple maths routines for his young daughter. He has every intention of acquiring at least one more QL and is emphatic that schools should use them. So far as he is concerned, they represent the present state of computer hardware development at a realistic price. If computer education in schools is to have any meaning at all, the latest technology must be represented, even though the restrictions of exam syllabuses mean that the curriculum will always be out of date. He's equally realistic in accepting the machine as it is. He has had problems with microdrives but is shrewdly aware that a disc system would have taken the QL beyond his price-range.

Hopefully Hadrian James is representative of a new generation of computer-literate teachers who are capable of guiding children into the next century. Unlike many teachers he understands what he is doing and why he is doing it. It is highly probable that many teachers have been influenced by the bad press which Sinclair Research attracts, and there are too many ill-informed educationalists, even at County level who are promulgating the view that Sinclair machines are "toys", without even drawing distinctions between the ZX81, the Spectrum and the QL. In a sense, the very factors which contributed to Sinclair's popular success have worked against the image of the machines as educational tools. Hopefully, however, the QL's planned software explosion can change this emphasis.

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.

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

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

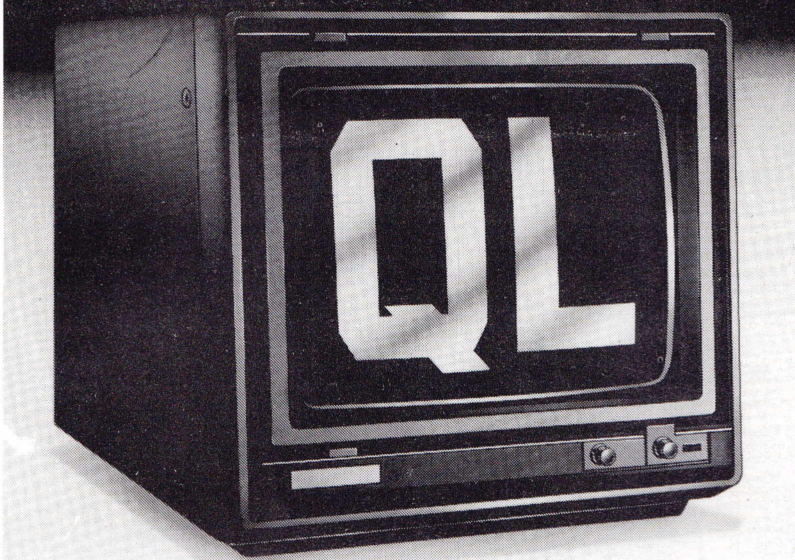
Manual dial V23 modem operating at 1200/75 baud and 1200/1200 baud half duplex.

Q-CALL

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

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

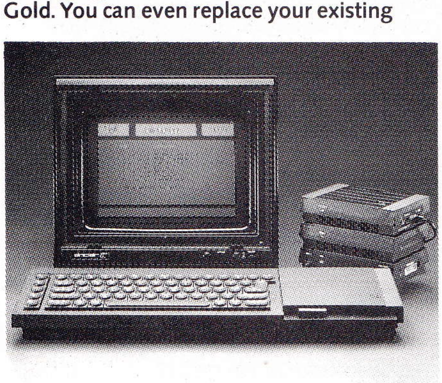
COMMUNICATIONS SINCLAIR QL

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.

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

Name ____________________

Address ____________________

____

____

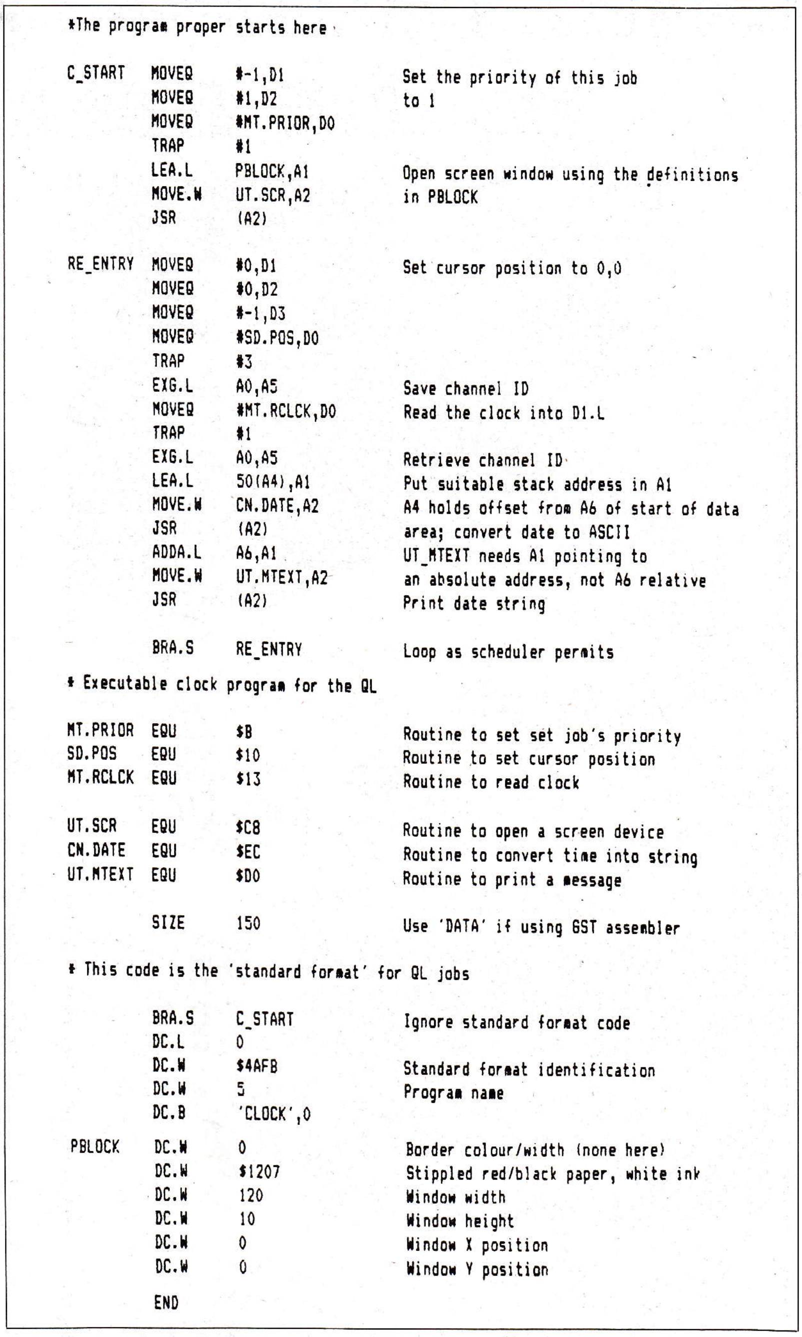
____

Tel No ____________________
QU8

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

Machine Code Tutorial

Adam Denning rounds off the first half of this series with a short multitasking clock program.



```
*The program proper starts here

C_START   MOVEQ     #-1,D1             Set the priority of this job
          MOVEQ     #1,D2              to 1
          MOVEQ     #MT.PRIOR,D0
          TRAP      #1
          LEA.L     PBLOCK,A1          Open screen window using the definitions
          MOVE.W    UT.SCR,A2          in PBLOCK
          JSR       (A2)

RE_ENTRY  MOVEQ     #0,D1              Set cursor position to 0,0
          MOVEQ     #0,D2
          MOVEQ     #-1,D3
          MOVEQ     #SD.POS,D0
          TRAP      #3
          EXG.L     A0,A5              Save channel ID
          MOVEQ     #MT.RCLCK,D0       Read the clock into D1.L
          TRAP      #1
          EXG.L     A0,A5              Retrieve channel ID
          LEA.L     50(A4),A1          Put suitable stack address in A1
          MOVE.W    CN.DATE,A2         A4 holds offset from A6 of start of data
          JSR       (A2)               area; convert date to ASCII
          ADDA.L    A6,A1              UT_MTEXT needs A1 pointing to
          MOVE.W    UT.MTEXT,A2        an absolute address, not A6 relative
          JSR       (A2)               Print date string

          BRA.S     RE_ENTRY           Loop as scheduler permits

* Executable clock program for the QL

MT.PRIOR  EQU       $B                 Routine to set set job's priority
SD.POS    EQU       $10                Routine to set cursor position
MT.RCLCK  EQU       $13                Routine to read clock

UT.SCR    EQU       $C8                Routine to open a screen device
CN.DATE   EQU       $EC                Routine to convert time into string
UT.MTEXT  EQU       $D0                Routine to print a message

          SIZE      150                Use 'DATA' if using GST assembler

* This code is the 'standard format' for QL jobs

          BRA.S     C_START            Ignore standard format code
          DC.L      0
          DC.W      $4AFB              Standard format identification
          DC.W      5                  Program name
          DC.B      'CLOCK',0

PBLOCK    DC.W      0                  Border colour/width (none here)
          DC.W      $1207              Stippled red/black paper, white ink
          DC.W      120                Window width
          DC.W      10                 Window height
          DC.W      0                  Window X position
          DC.W      0                  Window Y position

          END
```

To finish this series off, we're going to write a program that uses a large number of QDOS routines. The program *(listing 1)* is designed to run as an independant job, which means that it will only run with the EXEC or EXEC_W commands. You will need an assembler to be able to enter it, but if you've been following this course you ought to have one by now!

The first thing the program does is to set its priority to one. A job's priority determines how often it will be allowed to run by QDOS. By giving it a priority of one we ensure that it is updated every second or so, but does not get in the way of the SuperBasic interpreter (which runs at a higher priority of 32). We then use a utility routine called UT.SCR to open a screen device for us. This uses the size and colour definitions set out in a parameter block called 'PBLOCK'. By altering those values you can make the clock appear anywhere on the screen you like, with any colour of display. When UT.SCR returns, it holds the channel ID of the new window in A0.

The program loops continuously from now on, doing the same things each time around the loop. The first thing it does is to move the cursor in the window to the very first position, using a routine called SD.POS. The column number goes into D1 and the row number into D2. We put −1 into D3 to tell QDOS not to return from this routine until it has done what we asked it to. This is known as 'infinite timeout'.

Once we've done this, we need to save the channel ID (which is still in A0), and we do this by exchanging it with A5. We don't care what's in A5, so this is a simple way of putting the channel ID out of harm's way for a while. We then call another QDOS routine, MT.RCLCK, which reads the QL's internal clock and puts its value as a 32-bit number into D1. We exchange A0 and A5 back again.

Now, the next routine is CN.DATE, which converts the number in D1 into a string showing the time and date, like '1985 May 8 20:45:23'. This routine needs two things; the time in D1 and the address of a safe area in A1. This address should not be an absolute address, but it should be relative to A6. That is, the address in A1 plus the value in A6 should give us the real address of the safe area.

It just so happens that when we start a job, QDOS sets up register A4 to point to the bottom of the safe area (actually called the 'data space') for this job, and A4 is relative to A6. Just what we want? Not quite, as CN.DATE needs quite a few bytes of free space, and uses it from the high address down. So, we use LEA to put the value of A4 plus 50 into A1.

When this routine ends, it returns with A1 pointing to the bottom of the date string, relative to A6. To print it out, we use UT.MTEXT again, and this needs the address of the string in A1 too, but it expects it to be absolute rather than A6 relative. We simply add the value of A6 to it to make it absolute, with ADDA.L A6,A1.

There's not much more to do now except to branch back and do it all over again. As this executes as a separate job from Basic, we've written a multi-tasking program which displays a clock continuously, even while other programs are working.

If you buy the QL Technical Guide you'll find that 68000 assembly language gets no more daunting than this, but you can do an awful lot with it.

Machine Code Tutorial will restart in the autumn after a short break.

Viglen

offer the best alternative to the microdrive

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.

5¼ inch Drives	Drive on its own	Drive with Q Disk Interface
400K Single drive	£99	£199
800K Dual drive	£179	£249
800K Single drive	£139	£247
1.6M Dual drive	£269	£329
3½ inch Drives		
800K Single drive	£158	£258
1.6M Dual drive	£287	£387

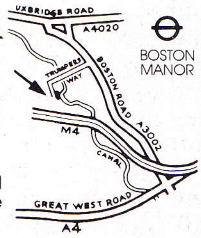
The above drives are compatible and ready to plug in and use with all currently available QL disc interfaces.

When bought with the disc interface they are ready to plug in and use.

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.

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.

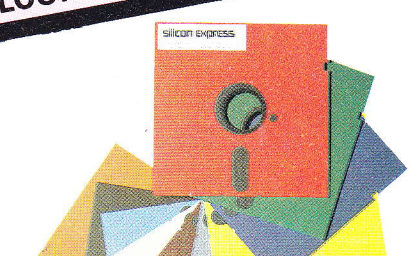
Or fill in and post the coupon – no stamp required – and we will despatch your goods within 14 days.

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

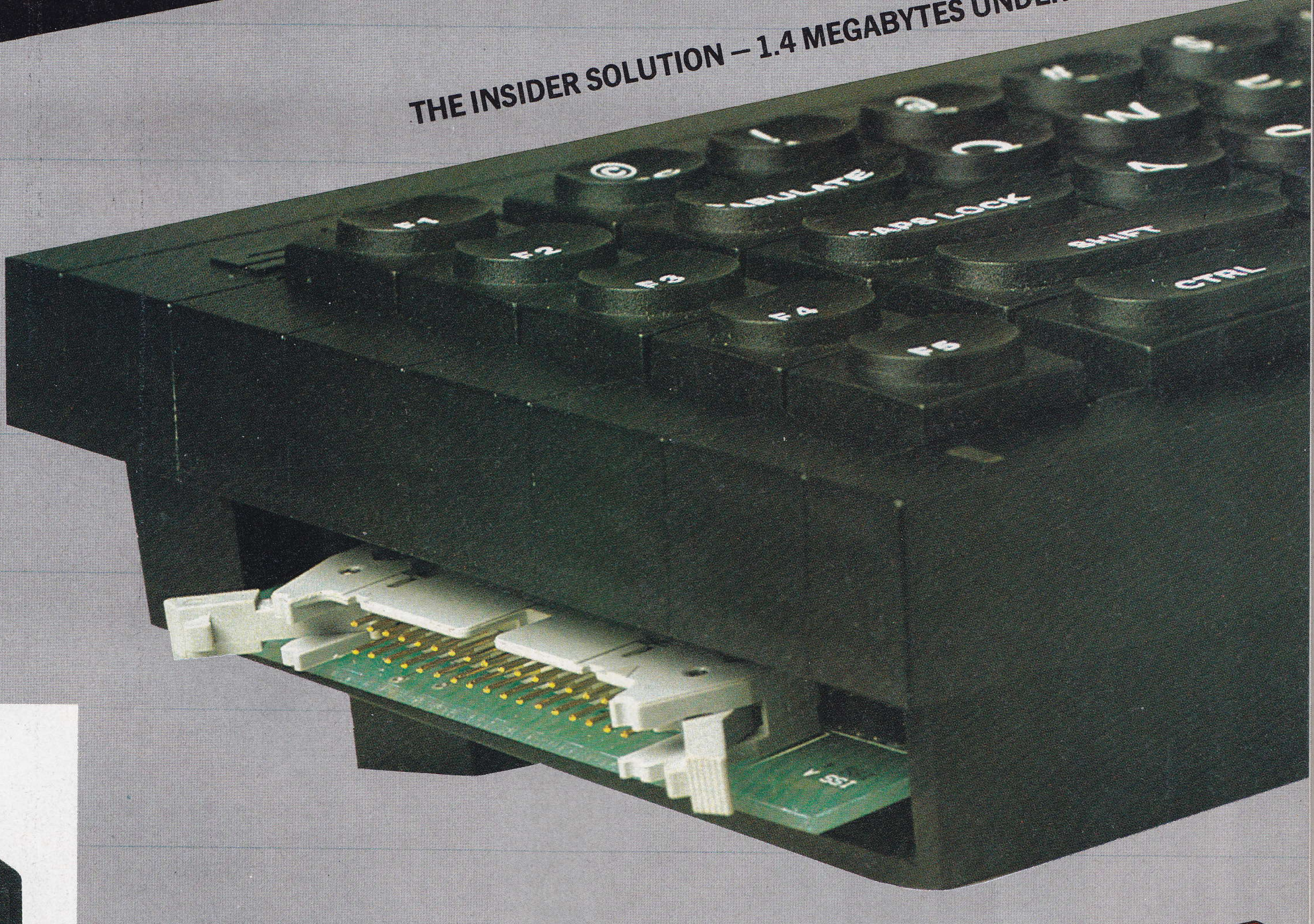
COLOURED DISKETTES



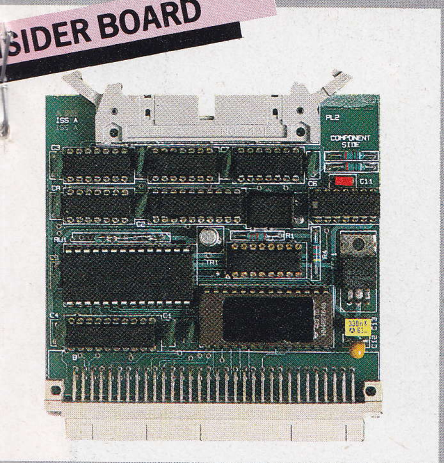
IN

ACE CARD....

THE INSIDER SOLUTION – 1.4 MEGABYTES UNDER COVER



SIDER 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 ______________________

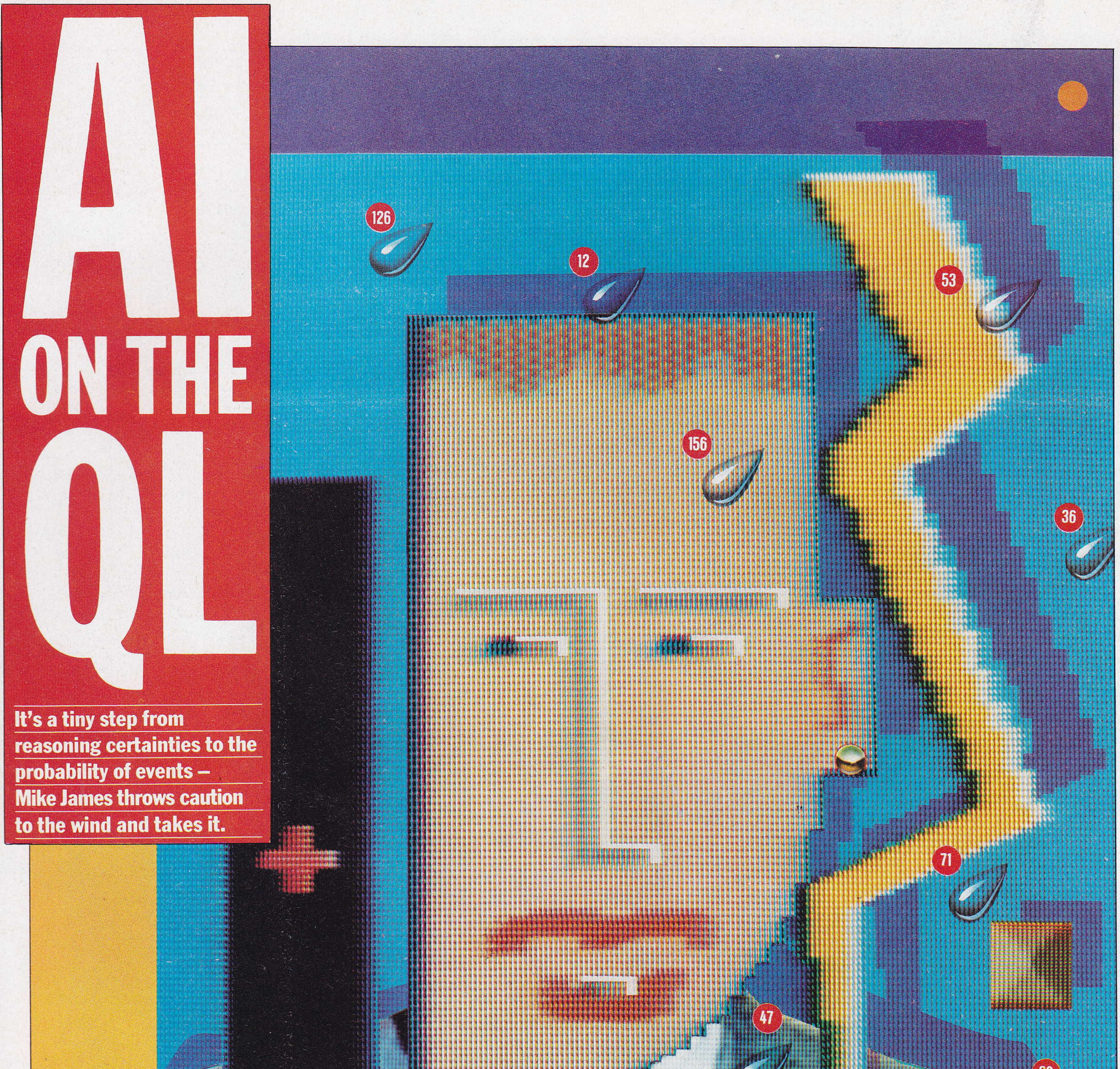
Address ______________________

QU8

Prices include VAT, post and packing.

AI ON THE QL

It's a tiny step from reasoning certainties to the probability of events – Mike James throws caution to the wind and takes it.



AI ON THE QL

Last month we introduced the concept of a knowledge-based 'expert' system. We did this by constructing a program capable of working out its own conclusions from a set of conditions and learning from its mistakes. The program, called *Aardvark* was crude as the conclusions it drew had to be based on certainty, yet it's not very often that a decision can be made with absolute certainty. It's more often the case that we say things like, *'I think the trouble might be'* or *'It could be . . .'* than saying, *'I know'* or *'It is'*. So far all the IF . . . THEN rules considered have assumed that the answer to any question was known with absolute certainty and the presence of any given condition was always an absolute sign that the conclusion should be drawn. However, this is not always the case. A component of human reasoning that we have ignored so far is *uncertainty*.

It is, even at this early stage, important to realise that there are two different sorts of uncertainty that occur in reasoning. The first is just not being sure of the condition. For example, you might have caught sight of an animal only for a moment and not been sure if it had a tail or not. This is being uncertain of the evidence and is generally easy to deal with. The second type of uncertainty is where the evidence is quite clear cut, that is you are sure of the facts, but there is no certain connection between what you know and the conclusion that you draw. For example, you may be sure that the weather is humid but this only increases the chances that there will be a thunder storm. This form of uncertainty is known as 'uncertainty in inference' and cannot be ignored as a component of reasoning.

Probably Certain

We are all familiar with the intuitive notion of probability as a way of summing up our certainty about an event but in fact it's a rather subtle concept that is very easy to misuse. The clearest interpretation of probability we have is in terms of the number of times something happens. For example, to say there is a probability of 0.5 of getting heads when a fair coin is tossed, is to say that you expect to see roughly 50% heads if the coin is tossed often enough. Thus a probability of 1 implies that something is certain to happen and a probability of 0 implies that something is certain not to happen; any value between 0 and 1 implies an uncertain outcome.

This works well when applied to events such as coin tossing but consider the situation where you are asked to say how likely something is to be true or false. In this case it is difficult to see how the idea of the number of times something is true or false in the long run can be used as an interpretation of probability.

For example, *'What do you reckon the probability is that life exists on other planets?'* is a question that you might be prepared to answer but it is difficult to see how the probability that you give as an answer can be interpreted as the number of times you are likely to be right in the long run. Even the wildest imagination will find it difficult to cope with more than one universe in which to repeat the event!

There are many ways round this difficulty but the easiest is to abandon the direct interpretation of probability as an indication of how often something would occur.

Even though there are philosophical problems with using probability in some areas of human knowledge and reasoning, it is still the system that we are most used to. For this reason it is worth examining how probability could be added to the sort of expert system represented by *Aardvark*. To be able to do this we need to look a little at the theory of probability.

The probability of an event x is normally written as $P(x)$, which should be read as a shorthand for 'the probability of x'. A slightly more complicated but very useful idea is that of 'conditional probability'. This is usually written as $P(A|B)$ and is read as 'the probability that A will happen given B has *already* happened'. In other words, a conditional probability is the probability of something *after* including any knowledge that we might already have.

You should be able to see that conditional probabilities are something like uncertain IF . . . THEN . . . rules. If you are certain about things then you write rules such as: 'IF dark clouds in sky THEN rain', whereas if you are admitting the existence of uncertainty then you would use the rule: $P(rain\ this\ afternoon|dark\ clouds)=.9$ which gives a reasonable indication of how certain you are that IF dark clouds THEN rain this afternoon.

Cloudy, With Rain . . .

Suppose you were sitting in a room without any window and absolutely no knowledge of what the weather might be doing. Your best estimate of the chance of it raining outside would be simply $P(rain)$. That is, simply the proportion of the time that it normally rains. In this sense $P(rain)$ summarises how much you believe it to be raining if you have no other knowledge. If you were given the information that there were black clouds in the sky outside, you would have to revise your belief in rain to take this information into account. Using:

$$P(rain|dark\ clouds)=$$
$$(P(dark\ clouds|rain)\ P(rain)$$
$$P(dark\ clouds)$$

you can change what you believed before the extra information, $P(rain)$, into your new belief, $P(rain|dark\ clouds)$. If some more information comes into your isolated room, perhaps a clap of thunder, the same method could be used to update what you believed once again:

$$P(rain|thunder|dark\ clouds)=P(thunder|rain)$$
$$P(rain|dark\ clouds)$$
$$P(thunder)$$

and so on. Each new piece of information would be used to change what you already believed into the new belief. To do this all you have to do is to multiply the probability that represents your old belief by a constant which indicates how much evidence the new information provides *for* or *against* the belief.

You should be able to see that this method could be added to *Aardvark* to produce an expert system that could cope with uncertain IF . . . THEN rules. At each decision point in the tree a new piece of information would be supplied to update the probability of each conclusion. Suppose the knowledge tree consisted of only three animals – lion, tiger and cheetah. At the start of the program the 'belief' that the computer had that each of the animals was the one you had thought of, would be simply $P(lion)$, $P(tiger)$, $P(cheetah)$. Each time you answer a question or supply some information the program would update its belief in each animal being the correct solution. For example, telling the program that the animal could run fast wouldn't make the program decide that your animal was a cheetah because lions and tigers can also run quite fast! Instead it would use this information to increase the probability that you had thought of a cheetah more than the probabilities of it being a lion or a tiger. In most cases, even after you had supplied it with everything that you knew about the animal, it would still not be able to tell you certainly what the answer was. Instead it would have to report the animal with the highest probability or even a number of very probable animals.

Obviously, this idea can be generalised to more serious and useful applications such as fault finding or medial diagnosis. A program that uses uncertain inference is not given here simply because the success of such a program depends very much on how well the knowledge base is constructed. In other words it takes a great deal of time to produce a good expert system but the principles by which it operates are not difficult to understand.

Circumstantial Evidence

The previous section discussed the most difficult aspect of uncertainty but hasn't really given a clue as to what to do when you are uncertain about the evidence. The answer is simple but unsatisfactory. If you include in an expert system a supplementary question such as, *'How sure are you (on a scale 0 to 1) of your last answer?'* then you will collect estimates of the certainty of each piece of information on which you are going to base your conclusion. If you treat these estimates as probabilities then the correct way of using them to estimate the uncertainty in your conclusion is very complicated and depends on knowing the relationships that exist between the different pieces of information. As these relationships are generally unknown the usual assumption is they are non-existent, and this implies that the correct estimate of the uncertainty of the conclusion should be obtained by multiplying all the probabilities together. This sounds reasonable enough until you notice that .5 times .5 is rather small (ie, 0.25) and even if you start out with large certainties, such as .9, the final certainty of the conclusion will be very small if you have very much evidence. Multiplying probabilities gives a very conservative estimate of the certainty of the conclusion because we are ignoring the relationships between the evidence.

There is no real acceptable solution to this problem and most expert systems use *ad hoc* methods to give a rough answer. For example, taking the minimum of the uncertainties gives a value greater than would be obtained by multiplying them together but still takes account of any very uncertain observation.

Expert systems are by no means the answer to all our problems but they do at least start to give some idea of the way knowledge can be represented and manipulated inside a computer. There are still many advances needed in our techniques. In particular, current expert systems use very crude methods of dealing with uncertainty and most do not examine and evaluate the rules that they use. What *is* surprising is that a large collection of very simple rules does seem to give the impression of human-like reasoning!

When expert systems emerge in conjunction with other recent AI products, such as programs that can understand standard English, the age of the useful intelligent computer will be upon us.

QL QUIDNUNC

Rifling through the rumours and sifting the speculation at the start of a new gossip column, penned by Silhoutte.

Here's the first, monthly, 'ear to the ground' report devoted to the proliferation of astounding revelations and other material of a suitably contentious nature. And all in the cause of truth and justice (!)

Where to begin? In the place where it all started – deepest Cambridge, university city and home of the Science Park, computing's equivalent to Alton Towers. Here, the amusement is freely available upon request and you'd be surprised by the number of people willing to spill the beans on their rivals.

Acornetti, well-known ice cream manufacturer, is a source of some ridicule this summer. Apparently the company is due to release a state-of-the-art machine real soon now. Does this mean we can expect another 6502-based computer? Ah, the wonders of modern technology.

And then there's the boys from the software side. 68000 is now in vogue but who said anything about the QL? And who gives a TOS about furtive attempts to conceal priority parcels from the States?

Then there's the scarcity of microdrive cartridges (how long have *you* been waiting?) Scouts have information that may be of interest to the Monopolies Commission. The sole producer of these marvels of modern tape technology makes up to 60,000 a week, but they are all snapped up before WHS's can say "45% please". Somewhat surprising, since it's been postured that there are only 60,000 QL users in the entire universe. This implies that each exponent of the 'bad or changed medium' principle buys on average one cartridge a week!

The solution lies, it seems, in the hands of the younger generation. The machine which is responsible for the microdrive fancier's weekly purchase is none other than the ZX Spectrum. No offence to personal dignity (OPD for short) intended. Also, have you heard the one about the one megabyte microdrive. It's an 8 track car stereo in disguise.

Next to busy Basingstoke, where the waiting is over provided, of course, that you like the Swedish style of presentation. Are bare PCB's and exposed connectors to be the stuff of that country's electronics industry? With the organic equivalent dominating their literature (and film industry) anything goes and just about everything does!

And now for something completely different – another QL magazine. Not on the bookstand, nor totally unconnected with its namesake. But the real surprise lies in its origins . . . From the people who brought you the highly entertaining 'Prism Goes Bust' comes 'Son of Prism', or as it will soon appear 'QL World' – yes, the phoenix has risen once more.

But is that all? Judge for yourselves as this so-called "independent" freebie is purportedly edited by a master of impartiality, the editor of QLUB news. Not one to bitch about a rival, you understand, but is it conceivable that such a product can claim even one iota of objectivity? Methinks not.

Finally, what's new but old, multiuser as well as multitasking and looking for home on QL. Gem, *no way!* Unix, *not yet!* 68K-OS *not quite!* OS9, *spot on!*

We end with a little speculation. Will the QL spearhead the comms revolution? Is Prestel under siege? Can Reuters compete with networked QL's? Has IT finally made it to Fleet Street? Will the Daily Mirror follow Strathclyde's lead and issue every reader with a QL? What does WSI mean and does it matter? These questions and others besides may or may not be answered next month.

QL User would like to emphasise that Silhouette's views and opinions as presented here do not necessarily reflect those of this magazine or its publishers.

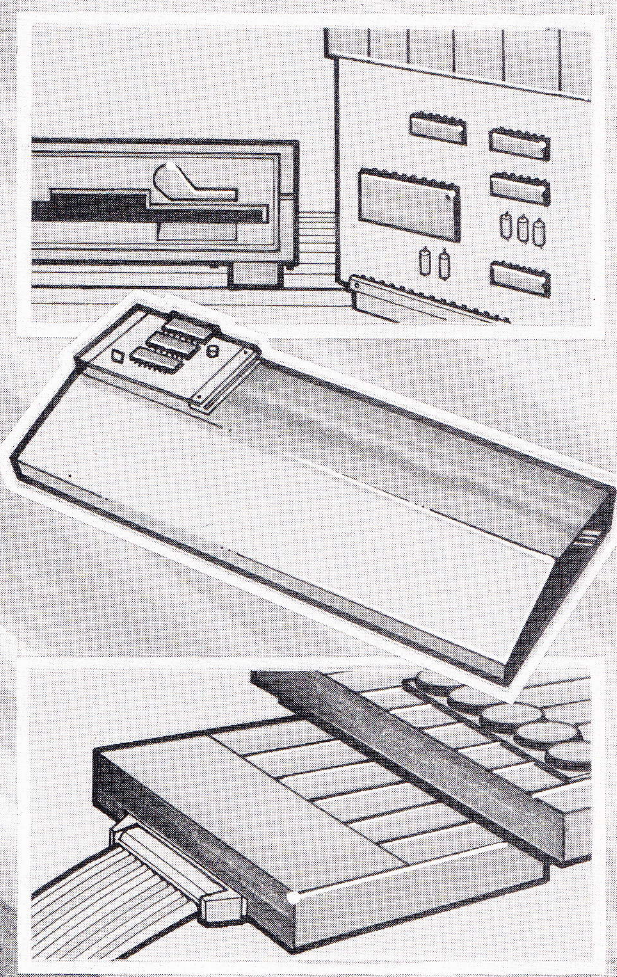
QL Future Growth

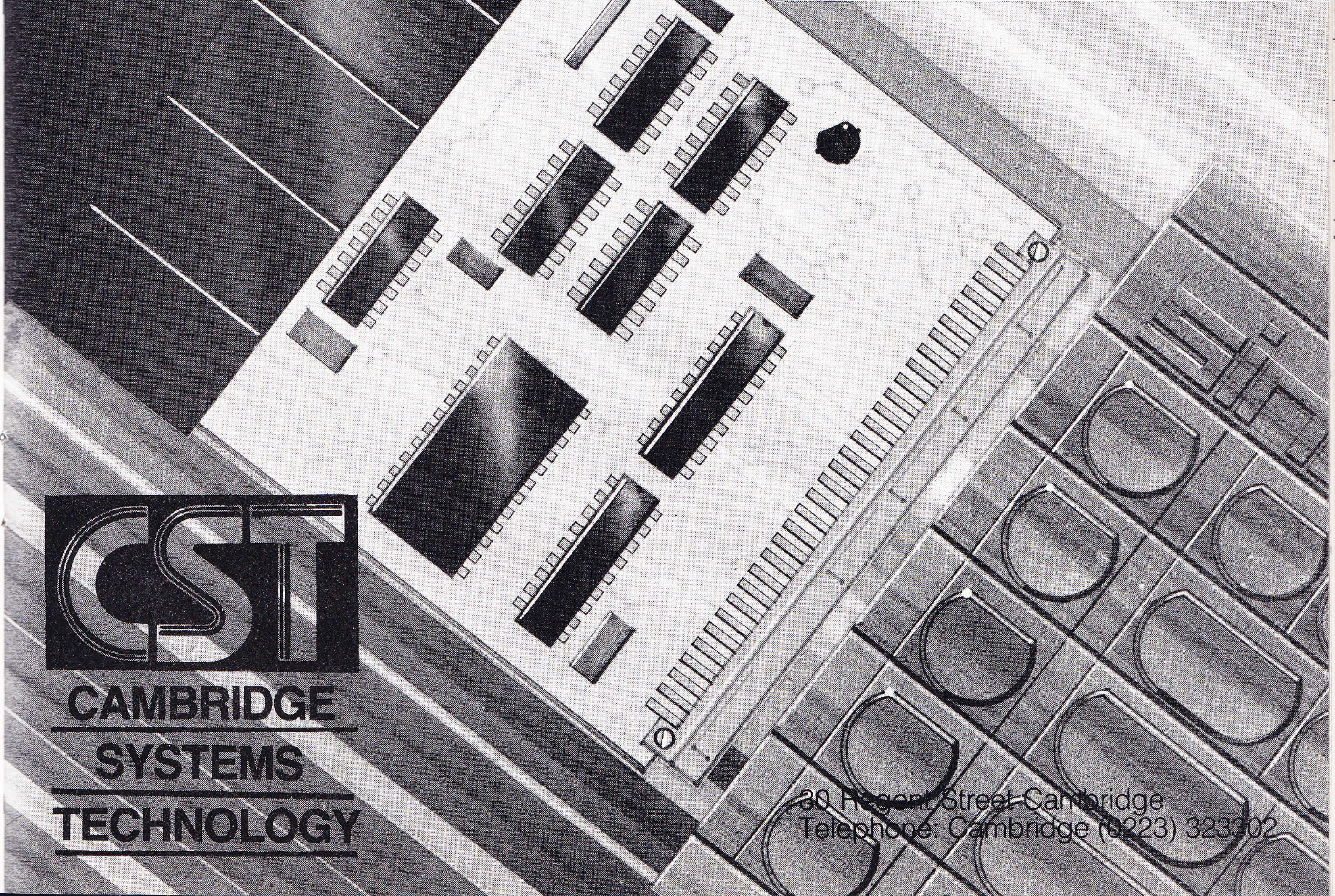
New horizons for your microcomputer from CAMBRIDGE SYSTEMS TECHNOLOGY, the dedicated specialists in expansion peripherals for the Sinclair QL.

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

THE PERFECT ALTERNATIVE

An operating system and development environment based entirely around Pascal. Evolution or entropy on the QL? Adam Denning investigates.

A long time ago, the University of California at San Diego (UCSD) decided to revitalise its computing course. It adopted the Pascal language as the basis for the new course, and set about writing a uniform system on which to implement the compiler. This became the *P-System*, and was soon adopted by a major American software house called Softech Microsystems as a commercially viable operating system and development environment.

Softech began to sell the *P*-system to educational establishments, computer manufacturers and software developers, improving on the basic system all the time. A Fortran 77 compiler, a Basic compiler, various assemblers and numerous utilities were written and sold.

As the *p*-system was written for an idealised Pascal machine (the *P-machine*), it was relatively easy to transport the system and all its tools from one machine to another. All you needed to do was re-write the interpreter and the low-level I/O system, and you had a new version of the system for your new machine. The interpreter is responsible for implementing the *p*-machine on the host processor, interpreting *p*-code instructions, which (almost) every *p*-system program consists of, so that the programs may be run on almost any microprocessor. The low-level I/O system is the interface between the *p*-system I/O and the I/O system of the host operating system or underlying machine.

The *p*-machine was envisioned as a 16-bit machine with a large instruction set. Its interpreter, therefore, turns out far from simple to write. Nevertheless, *p*-code interpreter and software may be found running not only off 16-bit processors (ie, 8088 & 8086) but also 8-bit processors such as the Z80, 6809 and 6502.

To recap briefly then, the fundamental aspects of the *p*-system are:

1) It is based entirely around the Pascal programming language and permits high-level languages to be compiled into *p*-code.
2) *P*-code is a code designed to run on an idealised Pascal machine known as the *p*–machine.
3) At the core of the *p*-system lies an interpreter which bridges the gap between idealised machine and existing computer (ie QL).

The major exponent of the *p*-system in the UK is Bristol based TDI. The company has recently released two versions of the system for the QL which complement those already out for other 6800 based computers (Mac, Sage and Pinnacle). Both versions cost £99.95 and consist of the same interpreter, filer, editor and so on. Where they differ is in the compiler and supporting libraries supplied. One version compiles Pascal into *p*-code and the other Fortran 77.

Also available is an Advanced Development Toolkit, which costs £49.95 and has numerous tools and utilities which people who take the *p*-system seriously will find useful. Surprisingly, TDI has no plans to release a Basic compiler.

AN INHERENTLY PASCAL, 16-BIT INTERPRETER

As the *p*-system is inherently a Pascal system, it makes sense to concentrate on the version which comes equipped with a Pascal compiler. The package consists of four microdrive cartridges and three manuals. There is also a short implementation-specific guide which contains details of those aspects of the system which are not portable.

Before he can get stuck in, the novice *p*-system user must read the tome known as the 'Operating System Reference Manual'. This gives details of the system's menu-driven structure, integral screen editor and filing system (the 'filer'). Some of the common utilities, such as SETUP, are described here as well.

The novice user should then move onto the UCSD Pascal handbook. This is the reference guide to the syntax and semantics of the language accepted by the compiler. In common with all *p*-system compilers, the end product of the Pascal compiler is a *p*-code module which may be a program in its own right, or may need to be linked with other 'units', or even possibly with some native code produced by the assembler.

The *p*-system always expects to find certain things in certain files; the interpreter is held in a file called SYSTEM.INTERP, the compiler in SYSTEM.COMPILER, the linker in SYSTEM.LINKER, and the runtime library in SYSTEM.LIBRARY. From this it follows that all that needs to be done to change from one compiler to another is to simply exchange versions of the SYSTEM.COMPILER and SYSTEM.LIBRARY files. There is no need to go around patching any files to alter the name of the compiler or library, as they always have the same names.

The *p*-system uses the concept of a *work file*. This is a file which will be edited, compiled or executed whenever the relevant options are taken from the menus. If the work file has not been explicitly named, then it will be given the default name 'SYSTEM.WRK.TEXT' (for the source file) and 'SYSTEM.WRK.CODE' (for the compiled result).

The screen editor, held in the file SYSTEM.EDITOR seems better suited to an antiquated teletype device rather than a VDU screen. Insertions and deletions, for example, cannot be carried out using the cursor alone but require the operator to explicitly specify the requisite editor mode. However, despite its wrinkles the editor is very powerful.

The filer (held in the file SYSTEM.FILER) allows you to access all the various elements of the filing system. The operations which you can perform cover almost every foreseeable need. For example, you can delete, copy and rename files, check what volumes are on-line, change the date and catalogue discs, amongst other things. It seems a little bit silly that while MS-DOS allows you to catalogue a disc simply by typing DIR, the *p*-system only allows you to do so by entering the filer, typing 'F', pressing 'L' (or 'E' for extended directory listings), entering the volume name, and finally pressing 'Q' to leave the filer. There must be some reason why the filer should be distinct from the rest of the operating system, but it evades me.

RECOMMENDED FOR DISK SYSTEMS OR RAM EXTENSIONS

This highlights the principal failing of the p-system: it is far too I/O bound! This means that it accesses the discs almost every time you press a key. If your system is equipped only with microdrives you are in for a shock. This is probably why TDI recommends the system be used on a QL with disc drives or added RAM. You can take advantage of the extra RAM by converting it to RAM-discs, this increases performance dramatically. It is also possible to make the filer RAM-resident, with or without extra RAM, but this is at the expense of user memory.

A program compiled into *p*-code is fairly compact – more so than an equivalent 68000 native code program – but there are occasions when it is desirable to write a section of code in assembler, generally to speed up execution (not that p-code is slow). For this reason, a file called SYSTEM.ASSMBLER is provided.

Unfortunately, the short manual that explains the workings of the assembler does not limit itself to 68000 but extends into the realms of the various 8-bit and 16-bit chips. This means that no chip is covered in sufficient depth.

The 68000 *p*-system assembler has two major problems. It is implemented on a 16-bit operating system, which means that you cannot use LONG constants or expressions (ie, a value greater than 16 bits). Hence, instructions like:

MOVE.L #$12345678,D3

and

MOVEA.L $9ABCDEF0,A6

cannot be assembled. Another problem is that aside register names and instruction set mnemonics nothing else follows the Motorola standard syntax. Additionally, the assembler is fairly unintelligent as each directive must be preceded by a period.

As the underlying *p*-system philosophy is to write programs that are not machine and processor specific inadequacies in the assembler may be forgiven. That is, provided that the programmer is content to accept the limitations imposed by being restricted to the facilities afforded by a Pascal Compiler.

It is possible to write programs in the *p*-system environment and then sell them to people who do not have the operating system, but this would undoubtedly require some sort of run-time licence from TDI. It also has the problem that each program you sell which is developed in this way will have the *p*-code interpreter and *p*-system bootstrap loader tacked onto the end, greatly increasing load time and the program size.

Perhaps TDI have launched this product not for program development but for education. The Open University has a computer science course based around the *p*-system and UCSD Pascal, so a student equipped with a QL and the *p*-system will be able to try out all the lessons and get the same results as the person with an IBM PC and the *p*-system.

If TDI is aiming principally for this market, it is almost certain to succeed; if it really expects the product to sell as a program development system, the company should think again. With numerous programming languages already available (ISO Pascal, APL, BCPL, C, assembler and so on), a developer is likely to need a good deal of persuasion to take this on. Having said that, the very first Occam development kits released by Inmos for the transputer were written on the *p*-system. A major defence electronics firm which was supplied such a system for its numerous Vax computers also had to be supplied with a *p*-system environment to be able to use it.

As far as the QL implementation goes, its performance far exceeds the Acornsoft/TDI implementation for the BBC Micro. The system has device drivers for the console, the serial ports (and therefore a printer), RAMdiscs and can cope with microdrives and floppy discs.

Also when Benchmarked against other machines running the p-system the QL obtained a very respectable 5th place, beating the IBM PC.

One thing the QL implementation is not capable of, is multi-tasking. It's not important that two versions of the system cannot run simultaneously (who needs it, anyway?), but it is an ommission that once booted up, only the *p*-system is operative. There's no room for a printer spooler, a clock, an assembler or anything else which you are likely to want to run behind other things.

The device drivers provided cater for the majority of devices, and if you buy the advanced toolkit you can edit the source files, re-assemble them and reconfigure the system to take advantage of aspects which may not be covered. TDI is thinking about incorporating a device driver (or an independent program) which allow you to read normal QDOS files from disc or microdrive into the *p*-system filing system. As the screen editors available for QDOS are rather nicer than that supplied with the *p*-system, this could be a step well worth taking.

All the *p*-system files are held in one large QDOS file per medium, called 'psystem'. Floppy discs and microdrives can be set up for the *p*-system with the provided PREPARE SuperBasic program. This program asks how much of the medium is to be devoted to the *p*-system, and then creates the requisite QDOS file. It also stores vital pieces of information inside the psystem file, such as the name of the volume and a basic directly structure. It is almost invariably the case that the best way of backing up a *p*-system medium is by copying the 'psystem' file from SuperBasic:

COPY flp1_psystem,flp2_psystem

rather than by entering the *p*-system filer and using option (T)ransfer to copy each individual file.

The filer also allows you to create subsidiary volumes, but this is only practical with floppy discs as there simply isn't enough space on a microdrive cartridge. Subsidiary volumes are roughly equivalent to sub-directories found under MS-DOS.

To sum up then, TDI's implementation of the p-system on the QL is a professional well turned out product. It compares favourably with similar systems appearing on larger and considerably more expensive machines. However, because of its heavy dependence on external I/O it is not suited to unexpanded QLs where even the simplest 30 line Pascal program takes minutes rather than seconds to compile.

ALTERNATIVE OS AND PROGRAMMING ENVIRONMENT

As an alternative operating system and program development environment competing directly with the likes of ISO and Computer One Pascal and indirectly, with QDOS, CP/M 68K, 68K-OS, the *p*-system would seem to have a limited appeal. Its likely home would be amongst avid Pascal Purists, educationalists and students where academia rather than commerce comes first.

However, beyond the Pascal front, the fact that the *p*-system currently provides the only way of compiling Fortran programs on the QL may well extend its appeal into an hitherto untapped market.

GRAPHICS ON DISPLAY

Recursion and in-betweening are two graphic methods that can be used to great effect, with little extra effort. Ian Stewart demonstrates how.

When it comes to graphics, computers have a big advantage over people. When they have learned to draw a picture once, they can draw it many times in different positions and orientations with very little extra effort. And they can combine elements from different pictures to produce new effects. In this article I'll look at two such methods. The first, *recursion*, uses procedures that call themselves to produce pictures with a high level of detail and pattern. The second, *in-betweening*, transforms one picture step by step into another. By defining your own input data you can easily produce your own variations.

Wheels Within Wheels

The fundamental unit of a SuperBasic program is the *PROCedure*, a named self-contained segment of program which can be called when required. You can pass parameters to it, and with a little more effort, pass them back. Any procedure can call any other; and in particular it can call itself. This technique is called recursive programming.

For our first example, We will draw a large circular disc. On its rim, at 12, 3, 6 and 9 o'clock, will appear four smaller discs and on each of these will be four even smaller discs, and so on until we reach the limit of screen resolution.

listing 1

```
100 MODE 8
110 blob 80,50,25,0
120 STOP
1000 DEFine PROCedure blob (x,y,r,n)
1010   LOCal i,x0,y0,r0
1020   IF n=5 THEN END DEFine blob
1030   INK n+2
1040   FILL 1
1050   CIRCLE x,y,r
1060   FILL 0
1070   r0=r/2.3
1080   FOR i=0 TO 3
1090     x0=x+r*COS(i*PI/2)
1100     y0=y+r*SIN(i*PI/2)
1110     blob x0,y0,r0,n+1
1120   END FOR i
1130 END DEFine blob
```

The main unit in *listing 1* is the procedure **blob**. Lines 1030-1060 on their own would just draw a filled circular disc. But in addition **blob** calls itself four times at line 1110 (inside the loop from 1080 to 1120). The counter *n* determines the 'level' of recursion. Every time **blob** is called from one level it descends one stage deeper thus *n* keeps track of what's going on. In lines 1090 and 1100 the program computes the coordinates of the points on the given circle at 12, 3, 6 and 9 o'clock. The radius of the circle is divided by 2.3 (a value found by trial and error) in line 1070, so that successive discs get smaller.

Line 1020 terminates the recursion whenever it descends to level 5 ensuring that the program does not run forever, trying to draw ever-decreasing discs smaller than the screen can display. Notice that in line 1010 the variables *x0,y0* and *r0* used by the recursive call to **blob** are defined as *local* variables. If you don't do this, then each call of **blob** changes the variables in such a way that the procedure 'forgets' previous values that it will need again, and everything goes haywire. (The loop counter *i* is also made LOCal, for safety's sake. It's a good rule to make all variables LOCal unless you have good reason not to.)

Angling In

When drawing graphics it's very common to want to select a number of equally spaced points around a circle. The trigonometric functions SIN and COS can be used to do this. If the circle has radius *r* and centre *(x,y)*, and you want to space out *i* points, then their coordinates are

x + r*COS(2*PI/i) or
x + r*COS(RAD(360/i))
y + r*SIN(2*PI/i) or
y + r*SIN(RAD(360/i)).

In the program above we spaced out four discs, so *i* = 4; and since _2*PI/4 = PI/2_ lines 1060 and 1070 result. The next program uses *i*=5 to obtain pentagonal spacing, and the result is an attractive fractal flower.

listing 2

```
100 array
110 MODE 8
120 tree 80,50,30,0
130 STOP
1000 DEFine PROCedure tree(x,y,r,n)
1010   LOCal i,x0,y0,r0
1020   INK n+2
1030   IF n=5 THEN END DEFine tree
1040   star x,y,r
1050   FOR i=0 TO 4
1060     r0=r/2.8
1070     x0=x+r*c(i):y0=y+r*s(i)
1080     tree x0,y0,r0,n+1
1090   END FOR i
1100 END DEFine tree
2000 DEFine PROCedure star (x,y,r)
2010   LOCal i,x0,y0
2020   FOR i=0 TO 4
2030     x0=x+r*c(i):y0=y+r*s(i)
2040     LINE x,y TO x0,y0
2050     FILL 1
2060     CIRCLE .5*(x+x0),.5*(y+y0),r/8
2070     FILL 0
2080   END FOR i
2090 END DEFine star
3000 DEFine PROCedure array
3010   DIM s(4),c(4)
3020   ,FOR t=0 TO 4
3030 s(t)=SIN(2*t*PI/5):c(t)=COS(2*t*PI/5)
3040   END FOR t
3050 END DEFine array
```

The first point to notice in *listing 2* is that the values of the trigonometric functions – which are used repeatedly by the *PROC*edures **tree** and **star** are calculated once and for all at the start, by the *PROC*edure **array**, and placed in *s(t)* and *c(t)*. If you don't do this, the program proceeds at a crawl. Lines 3030 and 3040 perform the pentagonal spacing. The main *PROC*edure **tree** draws five radiating lines (using **star**), puts a blob on each to fancy it up a little in line 2060, and then calls itself, to place a similar set of radiating lines at each of the five extremities.

In the previous article in this series I mentioned fractals, curves and surfaces with structure on all scales. Our flower is one example. Fractals can often be generated by using recursive programs, and the final example of recursion will do this for the Sierpinski Carpet (*listing 3*). This is obtained by taking a square, dividing it into a 3×3 array of smaller squares, and removing the central one; then repeating this process on the eight squares that remain, and so on. The area is multiplied by 8/9 at each stage, so the final area is zero!

listing 3

```
100 MODE 4
110 PAPER 4:CLS
120 carpet 80,50,45,2
130 STOP
1000 DEFine PROCedure carpet(x,y,s,n)
1010   LOCal x0,y0,s0,a,b,p
1020   INK 0
1030   FILL 1
1040   LINE x-s,y-s TO x-s,y+s TO x+s,y+s TO x+s,y
-s TO x-s,y-s
1050   FILL 0
1060   INK 7
1070   FILL 1
1080   p=s/3
1090   LINE x-p,y-p TO x+p,y-p TO x+p,y+p TO x-p,y
+p TO x-p,y-p
1100   FILL 0
1110   IF n=5 THEN END DEFine carpet
1120   s0=s/3
1130   FOR a=-1 TO 1
1140     FOR b=-1 TO 1
1150       IF a<>0 OR b<>0 THEN
1160         x0=x+2*a*s0:y0=y+2*b*s0
1170         carpet x0,y0,s0,n+1
1180       END IF
1190     END FOR b
1200   END FOR a
1210 END DEFine carpet
```

In-Betweening

Imagine you're a cartoonist working for Walt Disney. You've got to draw Donald Duck slipping up on a banana-skin. You've drawn the picture where his webbed foot has just trodden on the slippery fruit; and another of him flat on his face. Wouldn't it be nice to persuade a computer to fill in the bit in between?

It's possible though at that level of sophistication it needs a big computer and lots of fancy software. But we can get some idea of how it's done by seeing how to deform one object gradually into another one. This technique is sometimes called *In-Betweening*.

The basic problem to solve is this. Suppose a moving point starts at *(x,y)* and travels in a straight line to *(u,v)*, taking exactly one second to do so. What will be its position after *p* seconds, for *p* between 0 and 1? Using coordinate geometry, the answer turns out to be:

$$((1-p)x+pu,(1-p)y+pv)).$$

You can see this makes sense, because when $p = 0$ it gives *(x,y)*, the starting position, and when $p = 1$ it gives *(u,v)*; further when $p = \frac{1}{2}$ it gives $(\frac{1}{2}x+\frac{1}{2}u,\frac{1}{2}y+\frac{1}{2}v)$, which is midway between the two.

Now suppose you want to deform one shape, A say, into another, B. Represent them in the computer as dots joined by straight lines. Pair off corresponding dots between the two shapes. Now imagine each dot in shape A travelling towards it counterpart in shape B along a straight line, taking exactly one second to get there. Then the whole shape deforms from A to B in that second. To see what it looks like at the halfway mark, we work out where all the points are after half a second. More generally, if we take a series of numbers *p* running from 0 to 1 in small steps, we can use the formula above to see how the shape moves after *p* seconds.

Apart from that, all you have to do is input the data for the two shapes A and B, and devise a program to draw everything on the screen. As an example (*listing 4*), I'll give a new twist to an age-old philosophical problem.

listing 4

```
100 DATA 54,150,44,136,26,125,45,124
110 DATA 60,116,84,65,117,52,120,40
120 DATA 110,27,100,23,130,23,122,27
130 DATA 130,40,125,52,150,52,180,70
140 DATA 200,94,205,115,160,110,100,130,80,150
150 DATA 120,200,100,182,80,160,67,140
160 DATA 58,120,53,100,54,80,57,57
170 DATA 65,40,80,27,100,23,125,20
180 DATA 150,26,172,41,186,70,188,90
190 DATA 180,120,170,150,160,170,140,190,130,200
200 np=21
210 CLS#0
220 RESTORE
230 SCALE 280,0,0
240 PAPER 1:INK 6:CLS
250 CSIZE 3,1
260 AT 1,5
270 PRINT "WHICH CAME FIRST?"
280 INK 7
290 DIM x(np),y(np),u(np),v(np)
300 FOR n=1 TO np
310   READ x(n),y(n)
320 END FOR n
330 x(0)=x(np):y(0)=y(np)
340 FOR n=1 TO np
350   READ u(n),v(n)
360   u(n)=u(n)+250
370 END FOR n
380 u(0)=u(np):v(0)=v(np)
390 FOR t=1 TO 2
400   stp=1
410   IF t=2 THEN stp = 5E-2
420   FOR p=0 TO 1 STEP stp
430     q=1-p
440     FOR n=0 TO np-1
450       LINE q*x(n)+p*u(n),q*y(n)+p*v(n) TO q*x(
n+1)+p*u(n+1),q*y(n+1)+p*v(n+1)
460     END FOR n
470     IF p<.9999 THEN RECOL 0,1,2,3,4,5,6,0
480   END FOR p
490 END FOR t
```

Both chicken and egg have 21 dots. I drew them out on a sheet of graph paper, and used the coordinates to set up two data lists, from lines 100-140 for the chicken and 150-190 for the egg. Lines 200-280 set up the screen for display. Lines 300-330 read the chicken data into arrays *x* and *y*; lines 340-380 read the egg into arrays *u* and *v*.

The main in-betweening routine is between lines 420 and 480. Here *p* runs from 0 to 1 in steps of 0.05, and the in-betweening formula is applied in line 450 to all the points in the arrays *x,y,u,v*. Some additional recolouring effects are obtained from lines 390-410 and 470-490. The original coordinate data place the chicken on the left of screen and the egg on the right, so the intermediate shapes move slowly across from one to the other.

By using different data you can change the shapes: the main program stays the same (but you may need to change the value np, the number of points). Make sure both shapes have the same number of points in them.

For a slightly different effect you can modify the listing a bit:

omit line 500
add:
90 k=200
421 FOR z=1 to 2
422 INK 0:FILL 0
423 IF z=1 THEN k=k+1:INK k:
FILL 1
465 FILL 0
466 END FOR z

EXTRAVEHICULAR ACTIVITY

EVA

WIN

a copy of Westway's great new space adventure

20 EVA games to be won!

This must be one of the easiest competitions we've run yet, but it does require a bit of imagination . . . and patience!

TO ENTER Look at the letters contained in the phrase EXTRAVEHICULAR ACTIVITY (which is the current vernacular for floating around your spacecraft) and using any or all of them construct an anagram which means something similar, but in a lighter vein (eg, EXIT CIRCLE might be one way to say *leaving orbit*.

RULES Letters may only be used up to the number of times they appear in the phrase, but you do not need to use all of them. Anagrams must comprise words found in the dictionary used during the judging (New Collins Concise Dictionary). Entries will only be accepted if written on a postcard. Only one anagram per card (one entry per person).

THE PRIZES Each of the 20 winners will receive one copy of Westway's space adventure – EVA.

In common with other QL games, EVA takes advantage of the machine's advanced features. The EV Activity in question is a combination of repair work, self-preservation and alien zapping. Using cursor keys or joystick your task is to put together various pieces of interplanetary junk that litter your space station. Once an assembly is complete you are free to teleport off to another location and begin work over again.

Fireballs, spectres, giant jellyfish, malevolent robots, rotor droids and other nasties make your job very much more difficult.

To begin with you need only take evasive action such as hiding behind a platform or hightailing it to another part of the screen. However, as the game progresses and the aliens become increasingly more intelligent, moving faster and homing in on your every movement, you have no choice but to use your rivet gun to blast them into oblivion.

With 26 different screens to work through, each with its unique assortment of aliens, EVA should provide many hours entertainment. Certainly, it took us the better part of two hours solid play to work through just three screens. With a meagre four lives, this meant countless replays before we finally got the hang of it.

As QL programs go EVA is relatively short (45K). However, written entirely in machine code it packs quite a punch. It uses three frame animation throughout and contains a full 40K dedicated to sprite definitions. This means the action is fluid and the sprites are varied, colourful and often complex.

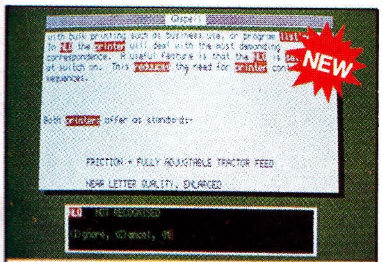
JUDGING Entries will be judged firstly to check that they comply with the above rules. Then the 20 considered most humorously and aptly to redefine 'Extravehicular Activity' will win the prizes.

CLOSING DATE Entries must be received on or before **30th August 1985** at the following address:

QL User,
EVA COMP,
Priory Court,
30-32 Farringdon Lane,
EC1R 3AU

No employee of EMAP and its associate companies, nor any of their families may enter this competition. Judging will be performed by the Editor and one other member of QL User's editorial staff, whose joint decision is final.

No correspondence will be entered into about the results, though the winners' names will be published in a future issue of QL User.



with bulk printing such as business use, or program list... In ... the printer will deal with the most demanding correspondence. A useful feature is that the ... is ... at switch on. This reduces the need for printer control sequences.

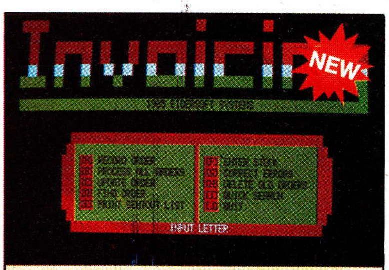
Both printers offer as standard:-

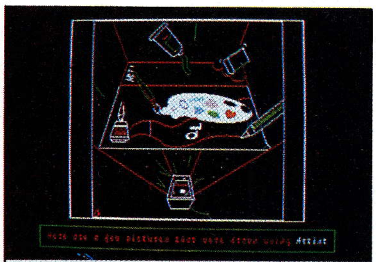
FRICTION + FULLY ADJUSTABLE TRACTOR FEED

NEAR LETTER QUALITY, ENLARGED

NOT RECOGNISED

(I)gnore, (C)ancel, (A)





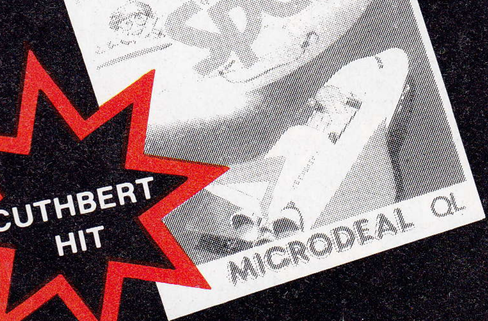
MICRODEALS

QUANTUM LEAP

HELP HOPPY TO HIS HOME
MICRODRIVE £14.95
POST & PACKING 75p

1ST ARCADE GAME FOR QL

INTO ARCADE GAMES



OUR FAVOURITE ARCADE GAME
HELP CUTHBERT LOOT THE MORONIANS PLANET
EX NO 1 HIT FOR THE DRAGON
MICRODRIVE £14.95
POST & PACKING 75p

CUTHBERT HIT

BACKUP FACILITY INCLUDED

UNBELIEVABLE 2000
SCREEN ARCADE ADVENTURE
OVER 1 YEAR IN THE MAKING
COMES COMPLETE WITH
9 FULL COLOUR MAPS
NOW AVAILABLE ON THE QL
£19.95
POST & PACKING 75p

2000 SCREENS

TO ORDER
PHONE 0726 68020 (SALES ONLY)
WITH CREDIT CARD
OR POST ORDER TO:
MICROPOST 41 TRURO RD., ST. AUSTELL, CORNWALL PL25 5JE.

LANDS OF HAVOC

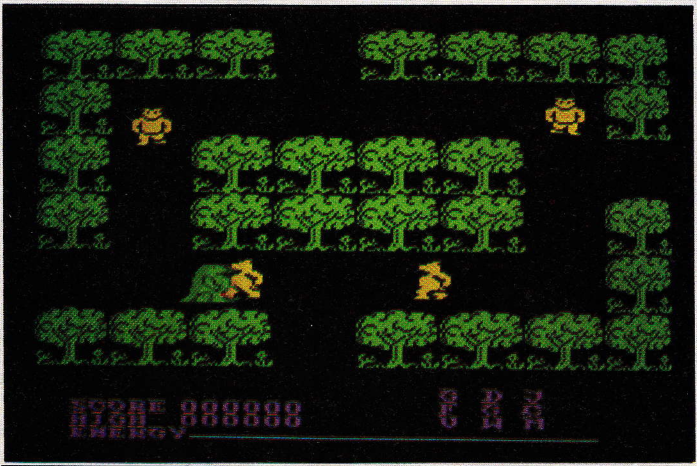
Arcade action adventure for the QL. Is it a welcome change from text or should we wait for the icon-based variety?

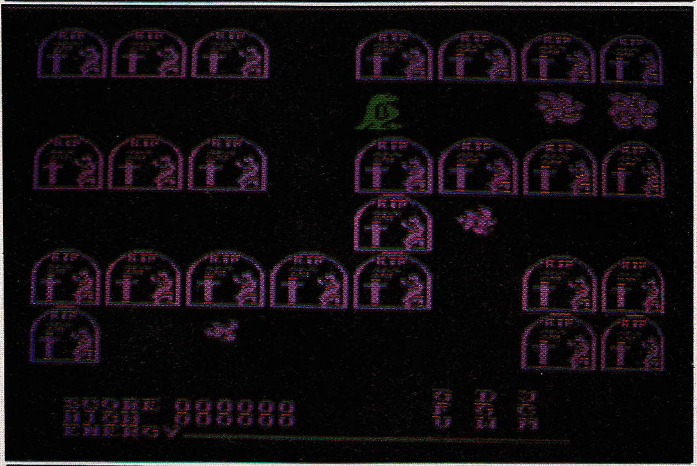
One of the big mysteries of the QL has been the time it has taken for good games to start appearing. *Lands of Havoc*, from Microdeal, is evidence that the situation is improving for the better but it still doesn't offer the kind of 'quantum leap' in performance that one might reasonably expect from such a high-powered machine.

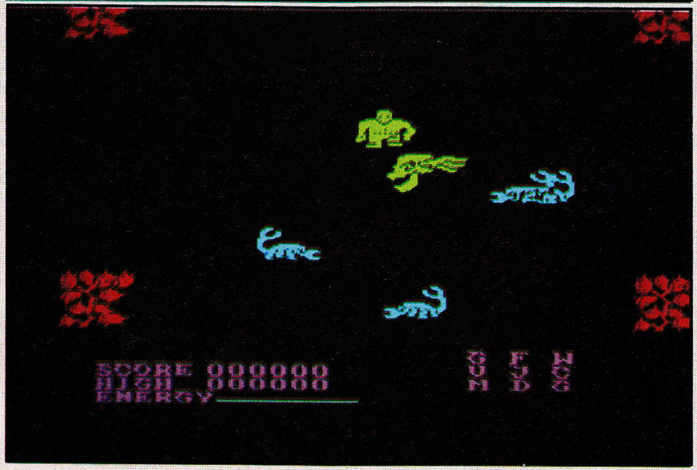
However, it would be churlish to moan too much when Microdeal, probably best known for the immense range of Dragon software they pioneered in an earlier stage of microcomputer evolution, have provided a genuinely playable game which should keep players happy for a considerable time. Perhaps some of that MC6809 experience was transferable to the 68008?

"Should please those who prefer the breathless *what next* of most arcade games."

If you've ever owned a Spectrum, you will instantly recognise the general classification into which *Lands of Havoc* falls: the so-called arcade adventure. You control a little man – well, actually he's a humanoid lizard rather like those in 'V' on the telly – who rushes around the maze-like screen, which shifts from one location to another as he rushes off the edge of the display. There are numerous nasties in close attendance and no immediately obvious objective springs to mind. Only a cryptic excerpt from a long-forgotten manuscript guides you in your mysterious quest . . . welcome to the Ultimate school of games design!







Actually, the differences between *Lands of Havoc* and Ultimate's variations on a theme (*Knight Lore, Sabre Wulf, Alien 8 etc . . .*) are significant, and the skills all you ex-Spectrum hackers developed in those days are unlikely to serve you well in this new game, but you could be forgiven for thinking that Microdeal took careful note of Ultimate's sales figures before launching this game.

A novel inclusion in the package is what seems at first to be a set of nine pretty post-cards, each with a different maze-like scene printed upon it. Booting the game from microdrive soon reveals that these should be fitted together to form a map of the first stage of the game. An immediate difference from other arcade adventures is that these initial screens come in a different order every time you play. Consequently you can't learn your way around the game as you can with most examples of the *genre*. This will frustrate seasoned adventure lovers, who like to be able to build up experience against future forays into the valley of death, but it should please those who prefer the breathless 'what next' of most arcade games.

It's not giving too much away to say that after some practice, you arrive at a village in which an alchemist's storeroom conceals clues to the real aim of the game, and from that point things start to hot up.

One of the main points of interest in the game is the colourful and interesting graphics display which makes up each location. The animation of the moving characters is competently carried out, though personally I still think Ultimate does it better. The sound is pretty uninspiring, and after a while rather irritating (fortunately you can turn it off).

"A total of 2000 screens, but there are secret ways of zapping from place to place."

Theoretically you could spend a long, long time playing this game. It has a total of 2000 screens, according to Microdeal, but in common with many text adventures there are secret ways of zapping instantly from place to place, making the game merely difficult rather than impossible.

What happens when my little lizard-man Sador finally meets the Dark Lords for the final confrontation, I have yet to learn. Judging by my progress so far it will take at least another week. Still, the QL's graphics are used well in the areas visited so far, so it should be worth the wait. Now, if I can only remember how I got to the instant space-warp location . . .

Meanwhile, where's the *real* blockbuster game for the QL? This one is OK, but as it's also available on the Commodore 64, it can't be stretching the machine that much. Come back Psion Chess, all is forgiven!

£19.95 – Mail Order from Micropost, 41 Truro Road, St Austell, Cornwall PL25 5JE (0726 68020).

FOR THOSE 'GREAT' MICRODEAL GAMES

QL

JOYSTICKS & INTERFACES

INTERFACES Only £4·75

Allows any ATARI or COMMODORE Joystick to plug into the QL

QUICKSHOT II Joystick £9·95 For above interface

SPECIAL OFFER both the above £12·95

Order Form by POST or PHONE 0726 68020 with your Credit Card No.

Access VISA AMERICAN EXPRESS

Please Sendme

QL INTERFACE £4·75 □

QUICKSHOT II £9·95 □

SPECIAL OFFER £12·95 □

Add **75**p Post & Packing

Name ____________

Address ____________

Postcode ____________

MICROPOST 41 Truro Rd.
St. Austell. Cornwall
PL25 5JE

QL8

QL TRANSACT – Book-keeping System

TRANSACT is a Book-keeping System and Nominal Ledger package
Provides a flexible alternative to manually operated systems
Observes normal accounting conventions
Suitable for Accountants and small business applications

Transaction Files emulate the manual Books of Prime Entry Sales, Sales, Purchases, Bank and Petty Cash transactions types
Standardised and fast Input format
Optional down-calculation of VAT & Net transaction values
Automatic assignment of Debit and Credit Values
Validation checks to ensure that entries balance
Double Entry principals maintained at all times
VAT Input and Output accumulations recorded separately
Journal facilities provided for Accounts adjustments
Standard Reports & Audit Trail to screen and printer

Accounts File emulates the manual Nominal Ledger
100 Accounts available under 10 Section Headings
Bank, Cash, Debtor, Creditor & VAT Controls are pre-set
Two Accounts reserved for Accrual and Prepayment adjustments
Remaining 90 Account Headings may be defined and/or amended
Account Balances are updated by each Transaction File created
View or Print Account Heading, Balances & Section Totals
Full Trial Balance printout excludes null Accounts
VAT payments and rebates controlled through Clearance Account
Produces Profit & Loss figure and 'quick' Balance Sheet

Price £34.95

QL SALES LEDGER and invoice processor

Open Item Accounting System
Fast creation of invoices from product tables
Multiple rates of VAT allowable on invoices
Suspense accounts for 'unrecognised' payments
Trade and settlement discounts allowable
Extensive reporting to screen and printer
Clients referenced by name or part name

Create Invoices and Credit Notes
Receive and Allocate payments
Print individual Statements of Account
Account Enquiries
Amend Terms
Amend Client Details
Suspense Accounts featuring automatic transfers of unallocated payments

Outstanding Invoices
Full Transaction Reports
Full Statement Runs
Monthly Sales Reports
Aged Debtors Lists
Mail Labels for clients Names and Addresses

Auto Invoice Numbering
Monthly deletions of paid invoices
Deletion Reports
Change date option

Requires no additional hardware
Runs on single or twin drive systems

Price £34.95

DIALOG SOFTWARE
20 NEW ROW
LONDON WC2N 4LA
Tel: 01-502 4328
01-502 4423

BARCLAYCARD VISA Access

Barclaycard and Access accepted

DIALOG SOFTWARE IS DISTRIBUTED BY
MICRO DEALER (UK) LTD. TEL: 07073-28181
CENTRESOFT LTD. 021-359-3020
TERRY BLOOD 0782-620321

COMPLETE SYSTEM for SINCLAIR QL from ZEAL MARKETING – The Professionals

★JUST PLUG IN AND GO

128K Micro plus choice of

PRINTER ● MONITOR ● DISK DRIVE

★QL COMPUTER – £385 –

tested before despatch

PRINTER & MONITOR PACKAGES

Factory tested before shipped on to you

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		£599		£445
Daisy Junior (7K Buffer)		£525		£385
Daisystep 2000		£599		£445
Juki 6100		£675		£525
Brother HR15		£715		£569
Epson DX100		£745		£590
Epson JX80 COLOUR (Superb with QL)		£895		

NOW ADD DISK DRIVES

Computamatic	
3·5″ or 5·25″ 720K	£335
Dual 2× 720K	£499
Microperipherals	
3·5″ 720K	£285
Dual 2× 720K	£425

ALL DRIVES INCLUDE INTERFACE AND PSU

All printers supplied with RS232C interfacing. Monitors display full 85 columns – connecting leads all inclusive.

MONITORS

Factory set for full 85 column display

COLOUR	
Microvitec 1451 DQ3	£275
1451 DQT3	£295
MONO	
Philips-BM 7502 green	£92
BM 7522 amber	£95

MEMORY EXPANSION

Two-way expander	£69.90
Four-way expander	£126.50
64K RAM	£99.90
128K RAM	£149.95
256K RAM	£175.00
512K RAM	£320

ACCESSORIES

Box of 10 SS DD 3·5″ disks	£52.50
Box of 10 SS DD 5·25″ disks	£20.00
RS232 C Printer Cable	£9.20
Miracle Systems Interface	£29.95
Epson RS232 C Interface	£30.00
Epson 2K Buffer Interface	£65.00
MONO MONITOR CABLES	£4.95

NEW – Protect your system from **SPIKES** and tidy up at the same time with **"Z" GUARD**
Stylish, metal, mains socket block with built-in filter and illuminated master switch to control entire system – 1-way £19.95
4-way £31.95

Full range of filters available for industrial conditions – contact factory

PAY US A VISIT AND SEE A SYSTEM IN ACTION – OPEN MONDAY TO FRIDAY

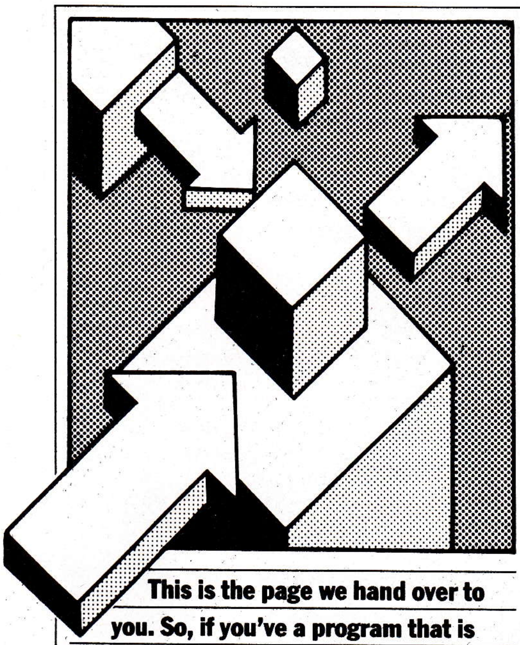
0900 – 1800

All prices include VAT

Delivery express – £10 Accessories – £2 p&p

ZEAL MARKETING LIMITED

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



This is the page we hand over to you. So, if you've a program that is worthy of consideration, send it to 'The Progs', QL User, Priory Court, 30-32 Farringdon Lane, London EC1R 3AU. We pay for everything published at the usual page rates.

Miners

Matthew Capp

This interesting simulation puts you in the role of the NCB – can you do better than MacGregor? You have to make decisions about whether to buy or sell coal (and mines), hire or fire miners as well as increase or decrease wages. All this is set against a background of ever-changing market forces.

The object is to remain profitable, maintain reasonable levels of satisfaction amongst your employees and, if possible, even expand the industry. Inexperienced players, however, will find it difficult enough to remain solvent, let alone satisfy any of the other conditions. The game calls for quick thinking, anticipation and, of course, the ability to make the right decision at the right time. A quality that seems singularly lacking in certain quarters!

```
1 REMark  **** QL User 1985 Miners ****
2 CLEAR:MODE 1:PAPER 4:CLS
3 OPEN #6,con_448x200a32x16:PAPER #6,0:CLS #6
4 OPEN #11,scr_512x256a0x0:PAPER #11,4:CLS #11
5 OPEN #10,scr_220x80a254x144
6 OPEN #9,scr_220x100a254x20
7 OPEN #8,scr_170x204a37x20
8 OPEN #5,scr_220x80a250x140
9 OPEN #4,con_220X100A250X16
10 OPEN #3,SCR_170X204A33X16
11 REMark *****CLS CHANNELS FOR MAIN WINDOW*****
12 DEFine PROCedure windows
13 FOR chan=8 TO 10:PAPER #chan,0:CLS #chan
14 FOR chan2=3 TO 5
15 INK #chan2,0:CLS # chan:BORDER #chan2,1,2
16 PAPER #chan2,7:NEXT chan2:END DEFine windows
17 REMark **********SETTING UP VARIABLES*********
18 M=0:y=1985:DIM g$(12,3)
19 FOR n=1 TO 12:READ g$(n)
20 DATA "JAN","FEB","MAR","APR","MAY","JUN"
21 DATA "JUL","AUG","SEP","OCT","NOV","DEC"
22 mines= RND(5 TO 10):MINERS_S=0
23 miners=RND(1000 TO 2000):miners_w=miners
24 LET money=800*miners+RND(30000)
25 tacks=money/7:money=money-tacks:z=0
26 mpay=800:coal=0:sat%=50:wage=mpay*miners
27 windows:procedures
28 REMark *****PROCEDURE TO CHANGE VARIABLES*****
29 DEFine PROCedure change
30 M=M+1:IF M=13 THEN LET y=y+1:LET M=1
31 IF coal<0 THEN coal=0
32 IF mines<0 THEN mines=0
33 IF mpay>900 THEN sat%=sat%+INT((mpay-800)/20)
34 IF mpay<750 THEN sat%=sat%-INT((750-mpay)/5)
35 sat%=sat%-RND(0 TO 2):IF sat%>50 THEN
36 miners_w=miners_w+MINERS_S:MINERS_S=0:END IF
37 coal=coal+(miners_w*RND(2 TO 5)+sat%)* mines
38 mprice=1000*(RND(8000 TO 12000))
39 mvalue=mprice-100*(RND(1000 TO 20000))
40 cprice=RND(32 TO 48)+z
41 cvalue=cprice-RND(0 TO 8)+z:z=0
42 END DEFine change
43 REMark ************REPORT PRCEDURE************
44 DEFine PROCedure REPORT
45 IF sat%<0 THEN LET sat%=0
46 IF sat%>100 THEN LET sat%=100
47 IF sat%<35 THEN strike
48 CLS #3:IF money <0 THEN bankrupt
49 PRINT #3,\"         REPORT"
50 PRINT #3,\\"  YEAR:";y;"  MONTH:";g$(M)
51 PRINT #3,\\"MINERS EMPLOYED =";miners
52 PRINT #3,"MINERS ON STRIKE = ";MINERS_S
53 PRINT #3,"MINERS WORKING = ";miners_w
54 PRINT #3,"COAL PRODUCED =";coal
55 PRINT #3,"MINES = ";mines
56 PRINT #3,"AVERAGE PAY/ MINER ='";mpay
57 PRINT #3,"BILLS(1000's) ='";INT(tacks/1000)
58 PRINT #3,"YOUR MONEY(1000's)='";INT(money/1000)
59 PRINT #3,\"SATISFACTION = ";sat%;"%":END DEFine

60 REMark **********COMMAND PROCEDURES***********
61 DEFine PROCedure procedures
62 REPeat main_loop
63 tax:change:news:choice:pay_men:effects
64 END REPeat main_loop:END DEFine procedures
65 DEFine PROCedure choice
66 REPORT:CLS #4:PRINT #4,\" (F1)   Buy  Mines"
67 PRINT #4," (F2)   Sell Mines"
68 PRINT #4," (F3)   Sell Coal"
69 PRINT #4," (F4)   Hire Miners"
70 PRINT #4," (F5)   Fire Miners"
71 PRINT #4," (TAB)  Change Miners Pay"
72 PRINT #4," (ESC)  To Continue"
73 FOR n=1 TO 75
74 IF KEYROW(0)=2 THEN Buy_Mines
75 IF KEYROW(0)=8 THEN Sell_Mines
76 IF KEYROW(0)=16 THEN Sell_Coal
77 IF KEYROW(0)=1 THEN Hire_Miners
78 IF KEYROW(0)=32 THEN Fire_Miners
79 IF KEYROW(5)=8 THEN Pay_Workers
80 IF KEYROW(1)=8 THEN pay_men
81 NEXT n:pay_men:END DEFine choice
82 DEFine PROCedure Sell_Mines
83 CLS #4
84 INPUT #4,"How many mines to sell ? ";msell$
85 IF mscil$="" THEN GO TO 83
86 mscil=msell$:BEEP 6000,255
87 IF msell <0 OR msell >mines THEN GO TO 83
88 IF msell<mines-2 THEN GO TO 96
89 IF msell>=mines-2 THEN
90 PRINT #4,"Are you sure about this ?(Y or N)"
91 END IF
92 REPeat loop
93 IF KEYROW(5)=64 THEN EXIT loop
94 IF KEYROW(7)=64 THEN GO TO 83
95 END REPeat loop
96 money=money+(msell*mprice):mines=mines-msell
97 choice:END DEFine Sell_Mines
98 DEFine PROCedure Buy_Mines
99 CLS #4:INPUT #4,"How many mines to buy ? ";mbuy
$
100 IF mbuy$="" THEN GO TO 99
101 mbuy=mbuy$:BEEP 6000,255
102 IF mbuy*mprice>money THEN GO TO 99
103 LET money=money-(mbuy*mprice)
104 LET mines=mines+mbuy:choice:END DEFine
105 DEFine PROCedure Sell_Coal
106 CLS #4:INPUT #4,"How much coal to sell?";csell
$
107 IF csell$="" THEN GO TO 106
108 csell=csell$:BEEP 6000,255
109 IF csell <0 OR csell >coal THEN GO TO 106
110 coal=coal-csell:money=money+(csell*cvalue)
111 choice:END DEFine Sell_Coal
112 DEFine PROCedure Pay_Workers
113 CLS #4:INPUT #4,"How much pay per miner?";mpay
$
114 IF mpay$="" THEN GO TO 113
115 mpay=mpay$:BEEP 6000,255
116 IF mpay*miners >money THEN GO TO 113
117 IF mpay<0 THEN GO TO 113
118 choice:END DEFine Pay_Workers
119 DEFine PROCedure Fire_Miners
120 CLS #4
121 INPUT #4,"How many miners to fire ? ";mfire$
122 IF mfire$="" THEN GO TO 120
123 mfire=mfire$:BEEP 6000,255
124 IF mfire<0 OR mfire>miners-(30*mines) THEN GO
TO 121
125 miners=miners-mfire:miners_w=miners_w-mfire
126 choice:END DEFine Fire_Miners
127 DEFine PROCedure Hire_Miners
128 CLS #4
129 INPUT #4,"How many miners to employ ?";mihire$
130 IF mihire$="" THEN GO TO 128
131 mihire=mihire$:BEEP 6000,255
132 IF mihire=0 THEN GO TO 139
133 IF mihire<0 THEN GO TO 129
134 IF money/(miners_w+mihire)<5000 THEN
135 PRINT #4,"You haven't enough money"
136 PRINT #4,"to pay wages for them !":GO TO 129
137 END IF
138 miners=miners+mihire:miners_w=miners_w+mihire
139 choice:END DEFine Hire_Miners
140 DEFine PROCedure pay_men
141 IF money<(mpay*miners_w) THEN
142 mpay=mpay-10:GO TO 141:END IF
143 effects:procedures:END DEFine pay_men
144 REMark ************NEWS PROCEDURE************
145 DEFine PROCedure news
146 CLS #5
147 PRINT #5,"-------------- NEWS --------------
"
148 PRINT #5,,"       BUYING"
149 PRINT #5,"  Mine prices (1000's)= '";(mprice/1
000)
150 PRINT #5,\,"       SELLING"
151 PRINT #5,"  Mine prices (1000's)= '";(mvalue/1
000)
152 PRINT #5,"  Coal price/ton = '";cvalue
153 END DEFine news
154 REMark ******RANDOM EFECTS DURING YEAR*******
155 DEFine PROCedure effects
156 CLS #4:INK #4,0:rd=RND(1 TO 35):PRINT #4
157 SELect ON rd
158 =1:PRINT #4,"Explosion at one of your pits cau
sed"
159 PRINT #4,"by an excess of coal gas."\"Irrepair
able damage."
160 PRINT#4,"Unavoidably the pit is closed down."
161 a=RND(40 TO 250):mines=mines-1
162 miners=miners-a:miners_w=miners_w-a
163 =2 TO 4:z= RND(4 TO 16)
```

```
164 PRINT #4,"Rival company goes bankrupt.More"
165 PRINT#4,"people turn towards you as coal"\"ord
er ";
166 PRINT#4,"increases"\"Coal prices rise by ` ";z
;"."
167 cprice=cprice+z:cvalue=cvalue+z
168 =5 TO 6:c=RND(4 TO 16):x=RND(500 TO 1500)
169 PRINT #4," Price glut due to recession."
170 PRINT #4,"Mine value drops by ` ";(x*1000)
171 PRINT #4,"Coal value drops by ` ";c
172 mprice=mprice-(x*1000):cvalue=cvalue-c
173 =7:x=RND(32 TO 58)
174 PRINT #4," Unfortunate coach crash means that"
175 PRINT#4,x;" miners died on their way to work."
176 miners=miners-x:miners_w=miners_w-x
177 =8 TO 9:c=RND(5000 TO 20000)
178 PRINT #4," The computer firm who were"
179 PRINT #4,"installing your companies new"
180 PRINT#4,"computer syatem have gone bankrupt."
181 PRINT #4,"   `";(10*c);" lost"
182 money=money-(10*c)
183 =10 TO 12:z=RND(1 TO 8)
184 PRINT#4,"You and rival companies are having a"
185 PRINT #4,"price war.You must therefore cut"
186 PRINT #4,"the price of coal down `";z;" per to
n."
187 cvalue=cvalue-z:cprice=cprice-z
188 =13 TO 15:x=RND(5000 TO 15000)
189 PRINT #4,"Because of profits in the coal"
190 PRINT #4,"industry,you receive a sudden rush"\
"of perspective shareholders.The"\"company gains c
ash to the value of"\" ` ";x
191 money=money+x
192 =16 TO 17:c=RND(10000 TO 30000)
193 PRINT #4," You receive a government grant "
194 PRINT #4,"for new computerized equipment.This"
195 PRINT #4,"enables your miners to retrieve "\;c
;" extra tons of coal."
196 coal=coal+c
197 =18 TO 20:x=RND(7000 TO 25000)
198 PRINT#4,"At one pit the main power generator"
199 PRINT#4,"explodes causing several weeks of"
200 PRINT#4,"no mining so you lose approx. ";x
201 PRINT #4,"tons of coal."
202 IF coal>x THEN coal=coal-x
203 =21:c=RND(4000 TO 14000)
204 PRINT #4,"Bomb scare at one of the pits means"
\"that an order to the value of"\"` ";c;" was lost
."
205 money=money-c
206 =22 TO 24:z=RND(1 TO 10)
207 PRINT #4,"Opec raise the price of oil $";z*3
208 PRINT#4," on the barrel.Because of this your"
209 PRINT#4,"company get more foreign orders and"
210 PRINT #4,"the price of coal rises ` ";z;\"per
ton."
211 cprice=cprice+z:cvalue=cvalue+z
212 =25:z=RND(1 TO 6)
213 PRINT #4,"The government decide to build"\"sev
eral new power stations.To meet"\"the demand you a
nd the other"
214 PRINT #4,"companies rise the price of ore by"
215 PRINT #4,"  ' ";z
216 cprice=cprice+z:cvalue=cvalue+z
217 =26 TO 27:c=RND(4 TO 10)
218 PRINT #4," At the advice of some of your "
219 PRINT#4,"committee you decide to give the"
220 PRINT#4,"local school an old computer.This"
221 PRINT#4,"raises the satisfaction by ";c;" %"
222 sat%=sat%+c
223 =REMAINDER :PRINT #4,"   A quiet month."
224 END SELect
225 PRINT #4,\\"Press any key .....":END SELect
226 FOR n=1 TO 500:NEXT n:PAUSE:CLS #4
227 END DEFine effects
228 REMark *************TAX PROCEDURE************
229 DEFine PROCedure tax
230 tacks=money/(7-RND(0 TO 2))+5010
231 money=money-tacks:END DEFine tax
232 REMark ***********STRIKE PROCEDURE***********
233 DEFine PROCedure strike
234 MINERS_S=INT(miners*(100-sat%)/100)
235 miners_w=miners-MINERS_S:END DEFine strike
236 REMark ****PROCEDURE TO CHECK IF BANKRUPT****
237 DEFine PROCedure bankrupt
238 tacks=5010:REPeat loop
239 IF money >0 THEN EXIT loop
240 IF mines=0 AND coal=0 THEN BANKRUPCY
241 money=money+cvalue*coal:CLS #4
242 PRINT #4, \"Forced to sell off coal":coal=0
243 IF money >0 THEN EXIT loop
244 IF mines=0 THEN BANKRUPCY
245 money=money+mvalue:CLS #4
246 PRINT #4, \"Forced to sell off a mine"
247 mines=mines-1:END REPeat loop
248 PRINT #4,\\\"Press any key .....":PAUSE
249 procedures:END DEFine bankrupt
250 REMark *********BANKRUPTCY FINISH************
251 DEFine PROCedure BANKRUPCY
252 y1=y-1985:CLS #7
253 PRINT #7,,\\\"YOU WENT BANKRUPT AFTER ";y1;
254 PRINT #7," YEARS AND ";M;" MONTHS"
255 PRINT #7,"OF OFFICE.YOU SUCEEDED IN LOSING ALL
 YOUR "
256 PRINT #7,"MINES, ALL YOUR COAL AND ALL OF YOUR
 MONEY."\\\
257 IF y1>8 THEN
258 PRINT #7,"BUT YOU DIDN'T DO TOO BADLY !!!!"
259 GO TO 261:END IF
260 PRINT #7,"YOU WERE COMPLETE AND UTTER RUBBISH
!!!!"
261 PRINT #7,\\\\,"PRESS (alt) FOR ANOTHER GAME."
262 REPeat loop
263 REPeat loop:IF KEYROW(7)=4 THEN RUN
264 END REPeat loop:END DEFine BANKRUPCY
```

Type Right

S Ackers

The importance of keyboard skills cannot be understated when it comes to computers. This is especially true of the QL where a word processing package is thrown in for free. The following program helps you develop such skills.

It consists of a four-lesson course where you have to type in letters, words and even phrases. The program displays the number of typing errors made, time taken and where appropriate, words per minute. A score is also displayed. Here accuracy counts more than speed as programmers can ill-afford to make simple typing mistakes. Entries are accompanied by a beep when correct, and a buzz when not. Instructions are provided throughout.

As a parting shot, we should add that this program is a whittled-down version of the one we originally received which contained 13 lessons, a 700-word vocabulary, interactive keyboard display and fingering chart (for those learning to touch type). With more than 30K of code, this was unfortunately too long to publish but is available on our microdrive exchange service.

```
10 REMark **** QL User 1985 ****
96 REMark ****TYPE  RIGHT****
97 REMark *****COPYRIGHT*****
98 REMark ***JOHN.S.ACKERS***
99 REMark ***THIMBLE  SOFT***
100 CLEAR
110 DIM w$(122,11)
120 RESTORE 2360
130 FOR word=0 TO 122:READ w$(word):END FOR word
140 DIM article$(3,4),adjective$(3,9),colour$(3,9)
,animal$(3,12),verb$(3,9),adverb$(3,9)
150 RESTORE 2550
160 FOR j=0 TO 3:READ article$(j),adjective$(j),co
lour$(j),animal$(j),verb$(j),adverb$(j):END FOR j
170 midtest$="ASDFG;LKJHASDFG;LKJH"
180 alltest$="QAZWSXEDCRFVTG'/P;.OL,IKMUJNYHB"
190 OPEN#1,scr_512x256a0x0
200 sellev
210 DEFine PROCedure sellev
220 MODE 4
230 score=0
240 PAPER 0:CLS
250 OPEN#3,scr_512x203a0x0:BORDER#3,2,0,7
260 CURSOR 136,4:CSIZE 1,0:PAPER 2:INK 4:PRINT' PR
ESS KEY  FOR DESIRED LEVEL '
270 PAPER 0
280  CSIZE 2,0:INK 6:AT 2,1:PRINT'1':CSIZE 0,0:INK
 4:AT 2,6:PRINT'FOR LEVEL ONE':INK 6:AT 2,31:PRINT
'ASD.....    ROW':INK 2:AT 2,56:PRINT'IN KEYBOARD
ORDER'
290  CSIZE 2,0:INK 6:AT 8,1:PRINT'6':CSIZE 0,0:INK
 4:AT 8,6:PRINT'FOR LEVEL SIX':INK 6:AT 8,31:PRINT
'ASD/QWE..   ROWS':INK 2:AT 8,56:PRINT'RANDOM WORD
S'
300  CSIZE 2,0:INK 6:AT 15,1:PRINT'-':CSIZE 0,0:IN
K 4:AT 15,6:PRINT'FOR LEVEL ELEVEN':INK 6:AT 15,31
:PRINT'ASD/QWE/ZXC.ROWS':INK 2:AT 15,56:PRINT'RAND
OM LETTERS'
310  CSIZE 2,0:INK 6:AT 17,1:PRINT'`':CSIZE 0,0:IN
K 4:AT 17,6:PRINT'FOR LEVEL THIRTEEN':INK 6:AT 17,
31:PRINT'ASD/QWE/ZXC.ROWS':INK 2:AT 17,56:PRINT'RA
NDOM LONG  WORD PHRASES'
320 CLOSE#3
330 OPEN#4,SCR_512X40A0X205:PAPER#4,0:CLS#4:BORDER
#4,2,0,7
340  CSIZE 2,0:AT 21,1:INK 4,2:PRINT'TEST SYMBOLS
& LETTERS WILL APPEAR       ':AT 21,36:INK 7:PRINT'
WHITE'
350  AT 22,1:INK 2,7:PRINT'WHEN KEYED IN, THEY WIL
L CHANGE TO         ':AT 22,36:INK 4:PRINT'GREEN'
360  AT 23,1:INK 4,7:PRINT'WRONGLY KEYED ITEMS WIL
L APPEAR IN    ':AT 23,37:INK 2:PRINT'RED'
370 CLOSE#4
380 c=CODE(INKEY$(-1))
390 SELect ON c
400  =49:level_one
410  =54:level_six
420  =45:level_eleven
430  =96:level_thirteen
440  =REMAINDER :GO TO 380
450 END SELect
460 END DEFine sellev
470 DEFine PROCedure command
480 CSIZE 0,0:INK 2:AT 0,21:PRINT"TYPE THE WORDS,L
ETTERS OR SYMBOLS DISPLAYED"
490 PAUSE 150:AT 0,21:PRINT FILL$(" ",43)
500 END DEFine command
510 DEFine PROCedure level_to_level
520 OPEN#5,SCR_138X50A232X195:BORDER#5,2,7,0
530 OPEN#6,SCR_230X50A0X195:BORDER#6,2,7,0
540 OPEN#7,scr_140x50a372x195:BORDER#7,2,7,0
550 t$=DATE$
560 IF t$(20)<1:t$(20)=1:END IF
570 mins=t$(17)+(((t$(19)*10)+t$(20))/60)
580 wpm=wds/mins
590 CSIZE 0,0:INK 4:AT 20,56:PRINT'M:SE'
```

```
600 INK 2:AT 21,40:PRINT'THIS TEST TOOK '!t$(17 TO
 20)
610 AT 22,40:PRINT'TYPING ERRORS ='!mtake
620 AT 23,40:PRINT'WORDS PER MIN ='!INT(wpm)
630 IF mtake=0:mtake=.1:END IF
640 highscore=INT(((1/mtake)*(1/mins))*timefactor)
650 CSIZE 1,0:INK 4:CURSOR 382,200:PRINT'YOUR SCOR
E'
660 PAPER 2:CURSOR 464,200:PRINT'
670 lh=INT(LEN(highscore))
680 SELect ON lh
690  =1:hpos=476
700  =2:hpos=471
710  =3:hpos=464
720 END SELect
730 CSIZE 2,0:INK 0:CURSOR hpos,200:PRINT highscor
e
740 IF highscore>=score:LET score=highscore
750 CSIZE 0,0:PAPER 0:INK 7:CURSOR 382,210:PRINT'T
HIS IS THE BEST YET'
760 CURSOR 382,220:PRINT'FOR THIS SESSION, AT'
770 END IF
780 CSIZE 0,0:PAPER 0:INK 4:AT 20,1:PRINT"PRESS
  FOR ANOTHER TRY"
790 AT 21,1:PRINT'PRESS     TO MOVE UP A LEVEL'
800 AT 22,1:PRINT'PRESS     TO MOVE DOWN A LEVEL'
810 AT 23,1:PRINT'PRESS     TO MOVE TO ANY OTHER L
EVEL'
820 CSIZE 2,0:INK 6:CURSOR 45,200:PRINT'<'
830 CURSOR 45,210:PRINT'>'
840 CURSOR 45,220:PRINT'?'
850 CURSOR 45,230:PRINT'='
860 CLOSE#5:CLOSE#6:CLOSE#7
870 END DEFine level_to_level
880 DEFine PROCedure level_one
890 CLS
900 INK 2:CSIZE 0,0:AT 23,69:PRINT'LEVEL ONE'
910 lvl_1
920 CSIZE 0,0:AT 22,1:PRINT FILL$(" ",31)
930 c=CODE(INKEY$(-1))
940 SELect ON c
950  =192:level_one
960  =200:sellev
970  =208:score=0:level_six
980  =REMAINDER :GO TO 930
990 END SELect
1000 END DEFine level_one
1010 DEFine PROCedure level_six
1020 CLS
1030 INK 2:CSIZE 0,0:AT 23,69:PRINT'LEVEL SIX'
1040 lvl_6
1050 c=CODE(INKEY$(-1))
1060 SELect ON c
1070  =192:level_six
1080  =200:sellev
1090  =208:score=0:level_eleven
1100  =216:score=0:level_one
1110  =REMAINDER :GO TO 1050
1120 END SELect
1130 END DEFine level_six
1140 DEFine PROCedure level_eleven
1150 CLS
1160 INK 2:CSIZE 0,0:AT 23,67:PRINT'LEVEL  ELEVEN'
1170 lvl_11
1180 c=CODE(INKEY$(-1))
1190 SELect ON c
1200  =192:level_eleven
1210  =200:sellev
1220  =216:score=0:level_six
1230  =208:score=0:level_thirteen
1240  =REMAINDER :GO TO 1180
1250 END SELect
1260 END DEFine level_eleven
1270 DEFine PROCedure level_thirteen
1280 mtake=0:wds=45:timefactor=40
1290 CLS
1300 INK 2:CSIZE 0,0:AT 23,66:PRINT'LEVEL  THIRTEE
N'
1310 command
1320 SDATE 1984,12,1,0,0,0
1330 FOR phrase=1 TO 5
1340 CSIZE 1,1:AT 5,0:PRINT FILL$(" ",65)
1350 CSIZE 1,1:AT 6,0:PRINT FILL$(" ",65)
1360 rar=RND(3):rd=RND(3):rc=RND(3):ran=RND(3):rv=
RND(3):rdv=RND(3):rar2=RND(3):rd2=RND(3):ran2=RND(
3)
1370 phrase$=article$(rar) &' '& adjective$(rd) &'
 '& colour$(rc) &' '& animal$(ran) &' '& verb$(rv)
 &' '& adverb$(rdv) &' '& article$(rar2) &' '& adj
ective$(rd2) &' '& animal$(ran2)
1380 tab=(63-LEN(phrase$)) DIV 2
1390 IF rar=rar2 OR rd=rd2 OR ran=ran2 OR LEN(phra
se$)>64:GO TO 1360:END IF
1400 CSIZE 1,1:INK 6:AT 5,tab+1:PRINT phrase$
1410 FOR X=1 TO LEN(phrase$)
1420 an$=INKEY$(-1)
1430 CASE=CODE(an$)
1440 IF CASE>64 AND CASE<91:CASE=CASE+32:END IF
1450 IF an$ INSTR phrase$(X)
1460 BEEP 1000,10:CSIZE 1,1:INK 4:AT 5,tab+X:PRINT
 CHR$(CASE)
1470 ELSE
1480 BEEP 1000,255:CSIZE 1,1:INK 2:AT 6,tab+X:PRIN
T CHR$(CASE)
1490 mtake=mtake+1
1500 END IF
1510 END FOR X
1520 END FOR phrase
1530 level_to_level
1540 CSIZE 0,0:AT 21,1:PRINT FILL$(" ",28)
1550 c=CODE(INKEY$(-1))
1560 SELect ON c
1570  =192:level_thirteen
1580  =200:sellev
1590  =216:score=0:level_eleven
1600  =REMAINDER :GO TO 1550
1610 END SELect
1620 END DEFine
1630 DEFine PROCedure lvl_1
1640 mtake=0:wds=0:timefactor=1
1650 command
1660 CSIZE 3,1:INK 6:AT 5,7:PRINT midtest$
1670 SDATE 1985,3,14,0,0,0
1680 FOR X=1 TO 20
1690 an$=INKEY$(-1)
1700 CASE=CODE(an$)
1710 IF CASE>96 AND CASE<123:CASE=CASE-32:END IF
1720 IF an$ INSTR midtest$(X)
1730 BEEP 300,10:CSIZE 3,1:INK 4:AT 5,6+X:PRINT CH
R$(CASE)
1740 ELSE
1750 BEEP 300,255:CSIZE 3,1:INK 2:AT 6,6+X:PRINT C
HR$(CASE)
1760 mtake=mtake+1
1770 END IF
1780 END FOR X
1790 level_to_level
1800 CSIZE 0,0:AT 23,40:PRINT FILL$(" ",20)
1810 END DEFine
1820 DEFine PROCedure lvl_11
1830 mtake=0:wds=0:timefactor=4
1840 command
1850 SDATE 1985,3,14,0,0,0
1860 a=2
1870 FOR letter=2 TO 39
1880 r=RND(31)
1890 IF r=0:GO TO 1880
1900 CSIZE 2,1:INK 6:AT 5,a:PRINT alltest$(r)
1910 a=a+1
1920 REPeat wrong
1930 an$=INKEY$(-1)
1940 CASE=CODE(an$)
1950 IF CASE>96 AND CASE<123:CASE=CASE-32:END IF
1960 IF an$ INSTR alltest$(r)
1970 BEEP 1000,10
1980 CSIZE 2,1:INK 4:AT 5,letter:PRINT CHR$(CASE)
1990 EXIT wrong
2000 ELSE
2010 BEEP 1000,255
2020 CSIZE 2,1:INK 2:AT 6,letter:PRINT CHR$(CASE)
2030 mtake=mtake+1
2040 END IF
2050 END REPeat wrong
2060 END FOR letter
2070 level_to_level
2080 CSIZE 0,0:AT 23,40:PRINT FILL$(" ",20)
2090 END DEFine
2100 DEFine PROCedure lvl_6
2110 mtake=0:wds=20:timefactor=8
2120 command
2130 SDATE 1985,3,14,0,0,0
2140 FOR word=1 TO 20
2150 r=RND(122)
2160 tab=INT(30-LEN(w$(r)))/2
2170 CSIZE 3,1:AT 5,9:PRINT FILL$(" ",14)
2180 CSIZE 3,1:AT 6,9:PRINT FILL$(" ",14)
2190 CSIZE 3,1:INK 6:AT 5,tab+1:PRINT w$(r)
2200 FOR X=1 TO LEN(w$(r))
2210 an$=INKEY$(-1)
2220 CASE=CODE(an$)
2230 IF CASE>64 AND CASE<91:CASE=CASE+32:END IF
2240 IF CHR$(CASE) INSTR w$(r)(X)
2250 BEEP 1000,10
2260 CSIZE 3,1:INK 4:AT 5,tab+X:PRINT CHR$(CASE)
2270 ELSE
2280 BEEP 1000,255
2290 CSIZE 3,1:INK 2:AT 6,tab+X:PRINT CHR$(CASE)
2300 mtake=mtake+1
2310 END IF
2320 END FOR X
2330 END FOR word
2340 level_to_level
2350 END DEFine
2360 DATA 'aerial','ahead','airedale','altogether'
,'apotheosis','argosy','aleatery'
2370 DATA 'default','didgeridoo','dyke','dilapidat
ed','doughty','dowager','drogue'
2380 DATA 'early','edify','egregious','eldorado','
epiglottis','erupt','euphoria'
2390 DATA 'fastidious','fedora','fidelity','flatfo
oted','fragile','freight','fridge'
2400 DATA 'gargoyle','geisha','glower','gooseflesh
','grapefruit','guerilla','guitar'
2410 DATA 'halter','heredity','hieroglyph','holida
y','hugely','hysteresis'
2420 DATA 'ideology','igloo','illustrious','iota',
'iraqi','isolate'
2430 DATA 'jaguar','jeopardy','jitter','joist','ju
stify'
2440 DATA 'kaiser','kestrel','khaki','kikuyu','koa
la','kurdish'
2450 DATA 'laureate','lethargy','liquefy','lisle',
'literature','loosestrife','luffa'
2460 DATA 'oasis','odyssey','okapi','opaque','orat
or','otter','outlaw','owl','oyster'
2470 DATA 'palaeolith','photography','pipistrelle'
,'poltergeist','psaltery'
2480 DATA 'quadruple','querulous','quest','ql','qu
eue','quiet','quota'
2490 DATA 'raffia','register','rhapsody','ridge','
rooster','ruffle'
2500 DATA 'salutary','separate','shapely','silhoue
tte','spaghetti','stalwart'
2510 DATA 'tapestry','teasel','thwarted','tigersha
rk','tolerated','twelfth','typify'
2520 DATA 'ufology','ukulele','ulterior','uphold',
'urged','usage','utility','usurp'
2530 DATA 'wallflower','werewolf','wherewithal','w
```

```
ithhold','world','wreak'
2540 DATA 'yak','yesterday','yielded','yourself'
2550 DATA 'the','quick','brown','fox','jumped','ov
er','this','slow','black','dog','ran','under','tha
t','fast','yellow','aardvark','crawled','round','a
','lazy','white','hen','leaped','onto'
```

Maze 1-2-3

Simon Wallis

No adventure game is complete without its obligatory maze. This set of three programs takes you all the way from drawing a maze to constructing ever more intricate routines around it.

Maze Drawer

The first program, *Maze Drawer* is a maze generating program. It draws a true maze (ie, where there is only one solution). The maze measures 28 × 14 and is drawn in under 20 seconds. Size and therefore the speed of drawing can be changed by altering the variables *xl* and *yl* at the beginning of the program. This program should be saved under the name *mdv1_Maze_drawer* and is used in the two succeeding programs.

3D-Maze

If the mazes produced by the previous program *(Maze Drawer)* look easy, then this program proves otherwise. It draws a three-dimensional view of the maze and times your progress in looking for a solution (best recorded time 45 seconds).

Controls are as follows:
J=Jump (random, only 3 per game)
V=View the maze (3 per game)
Q=Quit (coward!)
Cursor left=Rotate 90° left
Cursor right=Rotate 90° right
1-9=Move this number of spaces forward.

Mouse Maze

When you are fed up of solving mazes, run this third program. It draws a maze, using program 1, and then solves it with a mouse that lurks in the QL character set. The mouse is reasonably intelligent, and will show you up by solving the maze in an average of five seconds. This is a reasonable example of artificial intelligence on the QL, even though the program is only about 50 lines of SuperBasic.

```
100 REMark **** Maze drawer by S.Wallis ****
101 REMark **** QL User  1985 ****
110 PAPER 7:CLS:CLS#0:CSIZE 2,0:CSIZE#0,2,0
120 xl=17:yl=29:INK 4
130 DIM a$(xl,yl)
140 b$=FILL$(CHR$(0),yl)
150 a$(1)=b$:a$(xl)=b$:a$(xl-1)=b$
160 c$=FILL$(CHR$(255)&CHR$(0),yl):c$(1 TO 2)=CHR$
(0)&CHR$(0):c$(yl-1 TO yl)=CHR$(0)&CHR$(0)
170 FOR n=2 TO xl-2 STEP 2
180 a$(n)=b$
190 a$(n+1)=c$
200 NEXT n
210 F$=""
220 REPeat nn:x1=RND(3 TO xl):y1=RND(3 TO yl):IF a
$(x1,y1)<>CHR$(0)THEN EXIT nn
230 a$(x1,y1)=" "
240 PRINT a$
250 RESTORE :FOR n=1 TO 4
260 READ x,y:IF a$(x,y)=CHR$(255)THEN a$(x,y)="F":
F$=F$&CHR$(x)&CHR$(y)
270 NEXT n
280 DATA x1-2,y1,x1+2,y1,x1,y1-2,x1,y1+2
290 :
300 REPeat loop
310 IF F$=""THEN EXIT loop
320 P=RND(1 TO LEN(F$)/2)*2-1
330 IF P<1 THEN P=1
340 x=CODE(F$(P)):y=CODE(F$(P+1))
350 P$=F$:F$=""
360 IF P>1 THEN F$=P$(1 TO P-1)
370 IF P<LEN(P$)-2 THEN F$=F$&P$(P+2 TO LEN(P$))
380 b$=""
390 IF a$(x-2,y)=" "THEN b$="1"
400 IF a$(x,y+2)=" "THEN b$=b$&"2"
410 IF a$(x+2,y)=" "THEN b$=b$&"3"
420 IF a$(x,y-2)=" "THEN b$=b$&"4"
430 B=b$(RND(1 TO LEN(b$)))
440 x1=-1:y1=0
450 IF B=2 THEN x1=0:y1=1
460 IF B=3 THEN x1=1:y1=0
470 IF B=4 THEN x1=0:y1=-1
480 a$(x,y)=" ":a$(x+x1,y+y1)=" "
490 a$(x+x1+x1,y+y1+y1)=" "
500 IF a$(x+2,y)=CHR$(255)THEN a$(x+2,y)="F":F$=F$
&CHR$(x+2)&CHR$(y)
510 IF a$(x-2,y)=CHR$(255)THEN a$(x-2,y)="F":F$=F$
&CHR$(x-2)&CHR$(y)
520 IF a$(x,y+2)=CHR$(255)THEN a$(x,y+2)="F":F$=F$
&CHR$(x)&CHR$(y+2)
530 IF a$(x,y-2)=CHR$(255)THEN a$(x,y-2)="F":F$=F$
&CHR$(x)&CHR$(y-2)
540 AT x,y-1:PRINT" ":AT x+x1,y+y1-1:PRINT" "
550 END REPeat loop
560 :
```

```
100 MERGE mdv1_Maze_drawer
1000 REMark MOUSE_MAZE
1010 f=0
1020 xl=xl-2:yl=yl-2
1030 c$="":b$=CHR$(183)
1040 x=3:y=3
1050 X1=0:Y1=1
1060 CLS#0
1070 :
1080 REPeat LOOP
1090 a$(x,y)="."
1100 AT x,y-1:PRINT b$;
1110 IF x=xl AND y=yl THEN EXIT LOOP
1120 IF f=0 THEN GO TO 1240
1130 IF a$(x,y+1)=" "THEN X1=0:Y1=1:GO TO 1350
1140 IF a$(x+1,y)=" "THEN X1=1:Y1=0:GO TO 1350
1150 IF a$(x-1,y)=" "THEN X1=-1:Y1=0:GO TO 1350
1160 IF a$(x,y-1)=" "THEN X1=0:Y1=-1:GO TO 1350
1170 a$(x,y)="#"
1180 IF a$(x+X1,y+Y1)="."THEN GO TO 1350
1190 IF a$(x+1,y)="."AND X1<>1 THEN X1=1:Y1=0:GO T
O 1350
1200 IF a$(x,y+1)="."AND Y1<>1 THEN Y1=1:X1=0:GO T
O 1350
1210 IF a$(x-1,y)="."AND X1<>-1 THEN X1=-1:Y1=0:GO
 TO 1350
1220 IF a$(x,y-1)="."AND Y1<>-1 THEN Y1=-1:X1=0:GO
 TO 1350
1230 X1=0:Y1=0:GO TO 1190
1240 IF a$(x+1,y)=" "OR a$(x-1,y)=" "OR a$(x,y+1)=
" "OR a$(x,y-1)=" "THEN GO TO 1280
1250 f=1
1260 X1=-X1:Y1=-Y1
1270 GO TO 1350
1280 IF a$(x+X1,Y1+y)=" "THEN GO TO 1330
1290 IF a$(x,y+1)=" "THEN X1=0:Y1=1:GO TO 1350
1300 IF a$(x+1,y)=" "THEN X1=1:Y1=0:GO TO 1350
1310 IF a$(x-1,y)=" "THEN X1=-1:Y1=0:GO TO 1350
1320 X1=0:Y1=-1:GO TO 1350
1330 IF a$(x+Y1,y+X1)<>" "THEN GO TO 1350
1340 IF ABS(xl/(x+Y1))-yl/(y+X1)>ABS(xl/(x+X1))-yl
/(y+Y1) THEN Z=X1:X1=Y1:Y1=Z
1350 AT x,y-1:PRINT"."
1360 x=x+X1:y=y+Y1
1370 END REPeat LOOP
1380 PRINT #0\\"Gimme the cheese!":BEEP 22000,0,50
,200,1,0
1390 FOR n=3 TO xl
1400 FOR m=3 TO yl
1410 IF a$(n,m)="."THEN AT n,m-1:PRINT" ":a$(n,m)=
" "
1420 IF a$(n,m)="#"THEN AT n,m-1:PRINT" "
1430 NEXT m:NEXT n
1440 PAUSE
1450 GO TO 1040
```

```
100 MERGE mdv1_maze_drawer
1000 REMark 3d Maze  by S.Wallis
1010 x=xl-2:y=yl-2:a$(3,2)=" "
1020 d=1:out=0
1030 jumps=0:views=0
1040 DIM d$(4,5):FOR n=1 TO 4
1050 READ d$(n):NEXT n
1060 DATA "North","East","South","West"
1070 SCALE 300,0,0
1080 INK 0:INK#0,0
1090 PAPER 7:PAPER#0,7
1100 CLS:CLS#0
1110 CSIZE 3,1:AT 4,6:FLASH 1:PRINT"GO GO GO !!!":
FLASH 0
1120 CSIZE 2,0
1130 SDATE 1985,0,0,0,0,0
1140 :
1150 REPeat loop
1160 IF out THEN EXIT loop
1170 move_
1180 print_
1190 END REPeat loop
1200 :
1210 time:PRINT#0\'Again ?'
1220 REPeat z:q$=INKEY$(-1):IF q$ INSTR 'yn':EXIT
z
1230 IF q$=='y':RUN
1240 STOP
1250 :
1260 DEFine PROCedure time
1270 time$=DATE$
1280 time$=time$(17 TO 20)
1290 AT#0,0,28:PRINT#0;time$
1300 END DEFine
1310 :
1320 DEFine PROCedure move_
1330 time:in$=INKEY$
1340 IF in$=""THEN GO TO 1330
1350 in=CODE(in$)
1360 SELect ON in
1370 =74,106:jump
1380 =86,118:view
1390 =81,113:quit
1400 =192:left
1410 =200:right
1420 =48 TO 57:forward
1430 END SELect
1440 AT#0,0,0
1450 PRINT#0;" Facing ";d$(d)!\
1460 PRINT#0;x-2;" South,  ";y-1;" East.     "
1470 time
1480 END DEFine
1490 :
1500 DEFine PROCedure jump
1510 IF jumps>3 THEN PRINT#0;"You've already had 3
 jumps!":PAUSE 100:RETurn
1520 x=RND(7 TO xl-2):y=RND(7 TO yl-2)
```

```
1530 IF a$(x,y)<>" "THEN GO TO 1520
1540 BEEP 22000,0,50,200,1
1550 d=RND(1 TO 4):jumps=jumps+1
1560 END DEFine
1570 :
1580 DEFine PROCedure view
1590 IF views>3 THEN PRINT#0;"You've already had 3
 views!":PAUSE 100:RETurn
1600 CLS:INK 4:PAPER 7
1610 PRINT a$
1620 AT 3,1:PRINT"*"
1630 AT x,y-1:
1640 SELect ON d
1650 =1:PRINT CHR$(190)
1660 =2:PRINT CHR$(189)
1670 =3:PRINT CHR$(191)
1680 =4:PRINT CHR$(188)
1690 END SELect
1700 views=views+1
1710 FOR n=1 TO 200
1720 time:IF INKEY$=""THEN NEXT n
1730 END DEFine
1740 :
1750 DEFine PROCedure quit
1760 CLS
1770 views=0:view
1780 PRINT#0\\\"You quit."
1790 STOP
1800 END DEFine
1810 :
1820 DEFine PROCedure left
1830 d=d-1:IF d=0 THEN d=4
1840 END DEFine
1850 :
1860 DEFine PROCedure right
1870 d=d+1:IF d=5 THEN d=1
1880 END DEFine
1890 :
1900 DEFine PROCedure forward
1910 IF in$=0 THEN RETurn
1920 x1=0:y1=0
1930 SELect ON d
1940 =1:x1=-1
1950 =2:y1=1
1960 =3:x1=1
1970 =4:y1=-1
1980 END SELect
1990 FOR n=1 TO in$
2000 IF a$(x+x1,y+y1)<>" "THEN
2010 PRINT#0;"Ouch! That's a wall !":BEEP 10000,0,
20,1,1:PAUSE 100:RETurn
2020 ELSE
2030 x=x+x1:y=y+y1
2040 END IF
2050 NEXT n
2060 END DEFine
2070 :
2080 DEFine PROCedure print_
2090 CLS:x1=0:y1=0
2100 SELect ON d
2110 =1:x1=-1
2120 =2:y1=1
2130 =3:x1=1
2140 =4:y1=-1
2150 END SELect
2160 FOR n=0 TO 5
2170 floor n:time
2180 IF a$(x+n*x1-y1,y+n*y1+x1)=" "THEN left_ n
2190 IF a$(x+n*x1+y1,y+n*yi-x1)=" "THEN right_ n
2200 IF y+y1*n<3 THEN exit_ n+1:RETurn
2210 IF a$(x+n*x1+x1,y+n*y1+y1)<>" "THEN wall_ n:R
ETurn
2220 NEXT n
2230 END DEFine
2240 :
2250 DEFine PROCedure floor(di)
2260 FILL 0:INK 0
2270 SELect ON di
2280 =0:
2290 LINE 450,0 TO 450,300,50,0 TO 50,300:FILL 1
2300 LINE 50,0 TO 90,30,450,0 TO 410,30,450,300 TO
 410,270,50,300 TO 90,270
2310 =1:
2320 LINE 410,30 TO 410,270,90,30 TO 90,270:FILL 1
2330 LINE 90,30 TO 140,67,410,30 TO 360,67,410,270
 TO 360,233,90,270 TO 140,233
2340 =2:
2350 LINE 360,67 TO 360,233,140,67 TO 140,233:FILL
 1
2360 LINE 140,67 TO 175,95,360,67 TO 325,95,360,23
3 TO 325,205,140,233 TO 175,205
2370 =3:
2380 LINE 325,95 TO 325,205,175,95 TO 175,205:FILL
 1
2390 LINE 175,95 TO 205,115,325,95 TO 295,115,325,
205 TO 295,185,175,205 TO 205,185
2400 =4:
2410 LINE 295,115 TO 295,185,205,115 TO 205,185:FI
LL 1
2420 LINE 205,115 TO 230,135,295,115 TO 270,135,29
5,185 TO 270,165,205,185 TO 230,165
2430 =5:
2440 LINE 230,135 TO 230,165,270,135 TO 270,165:FI
LL 1
2450 LINE 230,135 TO 270,165,270,135 TO 230,165
2460 END SELect
2470 END DEFine
2480 :
2490 DEFine PROCedure left_(di)
2500 SELect ON di
2510 =0:LINE 50,0 TO 50,300
2520 =1:LINE 90,30 TO 90,270
2530 =2:LINE 140,60 TO 140,233
2540 =3:LINE 175,95 TO 175,205
2550 =4:LINE 205,115 TO 205,185
2560 END SELect
2570 END DEFine
2580 :
2590 DEFine PROCedure right_(di)
2600 SELect ON di
2610 =0:LINE 450,0 TO 450,30,450,270 TO 450,300
2620 =1:LINE 410,30 TO 410,66,410,233 TO 410,270
2630 =2:LINE 360,66 TO 360,95,360,205 TO 360,233
2640 =3:LINE 325,95 TO 325,115,325,185 TO 325,205
2650 =4:LINE 295,115 TO 295,135,295,165 TO 295,185
2660 END SELect
2670 END DEFine
2680 :
2690 DEFine PROCedure wall_(di)
2700 FILL 1:INK 4
2710 SELect ON di
2720 =0:LINE 90,30 TO 90,270,410,30 TO 410,270
2730 =1:LINE 140,66 TO 140,233,360,66 TO 360,233
2740 =2:LINE 175,95 TO 175,205,325,95 TO 325,205
2750 =3:LINE 205,115 TO 205,185,295,115 TO 295,185
2760 =4:LINE 230,135 TO 230,165,270,135 TO 270,165
2770 END SELect
2780 END DEFine
2790 :
2800 DEFine PROCedure exit_(di)
2810 wall_ di-1
2820 FILL 1:INK 6
2830 SELect ON di
2840 =1:LINE 175,30 TO 175,205,325,30 TO 325,205:o
ut=1
2850 =2:LINE 205,66 TO 205,185,295,66 TO 295,185
2860 =3:LINE 230,95 TO 230,165,270,95 TO 270,165
2870 =4:LINE 240,115 TO 240,155,260,115 TO 260,155
2880 END SELect
2890 INK 0
2900 END DEFine
```

Qthello

Richard Green

This QL version of the classic Othello board game can be played by one or two players. Peppered with procedures and meaningful variable names, the program should be easy to follow.

The object of the game is to capture as many of your opponent's counters without endangering your own. A capture is made by sandwiching one or more enemy pieces between your own. This can occur along the vertical, horizontal and/or diagonal.

The display is a 3D representation of the board, drawn in MODE 4 for the benefit of those with TVs (this may easily be changed). 'THETA' determines the angle of elevation (a negative angle shows the board from beneath), 'DIST' and 'FAR' alter the perspective and magnification, respectively. The 'SCALE' also gives control of location and magnification.

Counters are placed on the board using the cursor keys (or joystick). Pressing the space bar ends the move. The computer checks for illegal moves and the game ends either when the board is covered or neither player can move. A legal move must involve a capture.

Finally, the computer may be selected as an opponent. Without giving too much away, it can be beaten but should provide those new to the game with a number of unpleasant surprises. Average response time is about 15 seconds.

```
1000 REMark ***************************
1010 REMark *      Q T H E L L O      *
1020 REMark *    by  Richard Green    *
1030 REMark ***************************
1040 RANDOMISE
1050 m$="3334353663646566643534656"
1060 SCALE 110,-91,-55
1070 theta=RAD(20): dist=400: far=10: gamma=ATAN(1
60/dist)
1080 both_cannot_move=0
1090 cannot_move=0
1100 cursor_x=1: cursor_y=1
1110 DIM score(2)
1120 score(1)=2: score(2)=2
1130 DIM board(9,9)
1140 board(4,4)=1: `board(5,5)=1
1150 board(4,5)=2: board(5,4)=2
1160 DIM ddata(8,2)
1170 DATA -1,0,-1,1,0,1,1,1,1,0,1,-1,0,-1,-1,-1
1180 RESTORE 1170
1190,FOR f=1 TO 8: READ ddata(f,1),ddata(f,2)
1200 MODE 8
1210 INPUT#0;("1 or 2 players :");num$
1220 num_players=1: IF num$="2" THEN num_players=2
1230 MODE 4
1240 set_up_screen
1250 IF RND>.5 AND num_players=1
```

```
1260  player=2: move_computer: jump_for_scores=end
ed
1270 END IF
1280 REPeat game
1290  FOR player=1 TO num_players
1300   move_player
1310   IF ended THEN EXIT game
1320  END FOR player
1330   IF num_players=2 THEN NEXT game
1340   player=2
1350   move_computer
1360   IF ended THEN EXIT game
1370 END REPeat game
1380 game_ended
1390 STOP
1400 REMark ---------------------
1410 DEFine PROCedure move_player
1420 REMark ---------------------
1430 REPeat human_player
1440  REPeat cursors
1450   move_cursor
1460   x=cursor_x: y=cursor_y
1470   IF board(x,y)=0 THEN EXIT cursors
1480  END REPeat cursors
1490  CLS#0
1500  cannot_move=0
1510  main_mode=1
1520  main_sub
1530 IF count>0 THEN EXIT human_player
1540  CLS#0: INPUT#0;("Can you go (y/n) :");enter$
:CLS#0
1550  IF NOT(enter$=="n") THEN NEXT human_player
1560  IF cannot_move=1 THEN both_cannot_move=1
1570  cannot_move=1
1580  RETurn
1590 END REPeat human_player
1600 END DEFine
1610 REMark ----------------------
1620 DEFine PROCedure move_computer
1630 REMark ----------------------
1640 main_mode=0
1650 max$="11": max=-100
1660 FOR s=1 TO LEN(m$) STEP 2
1670  x=m$(s): y=m$(s+1)
1680  IF board(x,y)<>0 THEN NEXT s: EXIT s
1690  main_sub
1700  IF count=0 THEN NEXT s: EXIT s
1710  IF x=1 OR x=8 OR y=1 OR y=8 THEN count=count
+5
1720  IF x=2 OR x=7 OR y=2 OR y=7 THEN count=count
-5
1730  IF (x=1 AND x=8) OR (y=1 AND y=8) THEN count
=count+10
1740  IF count<max OR (count=max AND RND>.5) THEN
NEXT s: EXIT s
1750  max=count
1760  max$=x&y
1770 END FOR s
1780 IF max<-80
1790  PRINT#0;"I cannot move !!"
1800  IF cannot_move=1 THEN both_cannot_move=1
1810  cannot_move=1
1820  RETurn
1830 END IF
1840 main_mode=1: cannot_move=0
1850 x=max$(1): y=max$(2)
1860 main_sub
1870 board(x,y)=2
1880 cursor_x=x: cursor_y=y
1890 END DEFine
1900 REMark -----------------
1910 DEFine PROCedure main_sub
1920 REMark -----------------
1930 count=0
1940 FOR a=1 TO 8
```

```
1950  xx=x: yy=y
1960  FOR b=1 TO 7
1970   xx=xx+ddata(a,1): yy=yy+ddata(a,2)
1980   IF xx<1 OR xx>8 OR yy<1 OR yy>8 THEN NEXT a
: EXIT a
1990   c=board(xx,yy)
2000   IF c=0 THEN NEXT a: EXIT a
2010   IF c=player AND b>1 THEN EXIT b
2020   IF c=player THEN NEXT a: EXIT a
2030  END FOR b
2040  IF c<>player THEN NEXT a: EXIT a
2050  count=count+b-1
2060  IF main_mode=0 THEN NEXT a: EXIT a
2070  xx=x: yy=y
2080  FOR d=1 TO b-1
2090   xx=xx+ddata(a,1): yy=yy+ddata(a,2)
2100   board(xx,yy)=player
2110   INK 7*(player=1): put_piece
2120   BEEP 1400,7
2130  END FOR d
2140 END FOR a
2150 IF main_mode=0 OR count=0 THEN RETurn
2160 xx=x: yy=y
2170 board(xx,yy)=player
2180 INK 7*(player=1): put_piece
2190 BEEP 1400,7
2200 IF num_players=1 THEN alter_m_string
2210 END DEFine
2220 REMark ------------------
2230 DEFine PROCedure put_piece
2240 REMark ------------------
2250 row=xx: piece=yy-4.5
2260 wide=TAN(gamma)*(COS(theta)*20*row+far+dis
2270 ratio=160/wide: FILL 1
2280 CIRCLE piece*20*ratio,(row-4.5)*20*SIN(the
,8*ratio,SIN(theta),PI/2
2290 FILL 0
2300 END DEFine
2310 REMark ----------------------
2320 DEFine PROCedure set_up_screen
2330 REMark ----------------------
2340 PAPER 0: CLS: INK 4
2350 CSIZE 3,1: CURSOR 168,0: PRINT "QTHELLO"
2360 CSIZE 0,0: PRINT TO 29;"by Richard Green"
2370 CSIZE 3,0:c$="PLAYER 2":IF num_players=1 TH
 c$="COMPUTER"
2380 AT 16,0:INK 2:PRINT "PLAYER 1";TO 20;c$
2390 DATA 0,0,0,9,9,9,9,0,0,0
2400 RESTORE 2390
2410 FILL 1: INK 4: DIM v(4)
2420 FOR g=1 TO 5
2430 READ a,b
2440 wide=TAN(gamma)*(COS(theta)*20*a+far+dist)
2450 ratio=160/wide
2460 IF g=1
2470 LINE (b-4.5)*20*ratio,(a-4.5)*20*SIN(theta)
2480 ELSE
2490 LINE TO (b-4.5)*20*ratio,(a-4.5)*20*SIN(thet
)
2500 END IF
2510 w1=(b-4.5)*20*ratio: w2=(a-4.5)*20*SIN(theta
2520 IF g<3 THEN v(g*2-1)=w1: v(g*2)=w2
2530 END FOR g
2540 INK 0: FILL 0
2550 FOR c=.5 TO 8.5
2560  FOR d=1 TO 2
2570   FOR e=.5 TO 8.5 STEP 8
2580    a=c:b=e:IF d=2 THEN b=c:a=e
2590    wide=TAN(gamma)*(COS(theta)*20*a+far+dist)
2600    ratio=160/wide
2610   IF e=.5
2620    LINE (b-4.5)*20*ratio,(a-4.5)*20*SIN(theta
)
2630   ELSE
2640    LINE TO (b-4.5)*20*ratio,(a-4.5)*20*SIN(th
```

```
eta)
2650   END IF
2660   END FOR e
2670  END FOR d
2680 END FOR c
2690 FOR xx=4 TO 5
2700 FOR yy=4 TO 5
2710 INK 7*(board(xx,yy)=1)
2720 put_piece: END FOR yy: END FOR xx
2730 END DEFine
2740 REMark --------------
2750 DEFine FuNction ended
2760 REMark --------------
2770 IF NOT cannot_move THEN
2780  score(player)=score(player)+count+1
2790  score(1+(player=1))=score(1+(player=1))-coun
t
2800 END IF
2810 AT 18,3:INK 2:PRINT score(1);" ";TO 23;score(
2);" "
2820 IF both_cannot_move THEN RETurn 1
2830 IF score(1)+score(2)=64 THEN RETurn 1
2840 IF score(1)*score(2)=0 THEN RETurn 1
2850 RETurn 0
2860 END DEFine
2870 REMark ---------------------
2880 DEFine PROCedure move_cursor
2890 REMark ---------------------
2900 REPeat cursor_move
2910  xx=cursor_x: yy=cursor_y
2920  INK 2
2930  put_piece
2940   key$=INKEY$(50)
2950   BEEP 140,20
2960   key=CODE(key$)
2970  cursor_x=cursor_x+(key=208 AND cursor_x<8)-(
key=216 AND cursor_x>1)
2980  cursor_y=cursor_y+(key=200 AND cursor_y<8)-(
key=192 AND cursor_y>1)
2990  INK "470"(board(xx,yy)+1): put_piece
3000  IF key=248 OR key=32 THEN EXIT cursor_move
3010 END REPeat cursor_move
3020 END DEFine
3030 REMark ------------------------
3040 DEFine PROCedure alter_m_string
3050 REMark ------------------------
3060 FOR f=1 TO LEN(m$)-2 STEP 2
3070  IF m$(f TO f+1)=xx&yy
3080   m$=m$(1 TO f-1)&m$(f+2 TO LEN(m$)): EXIT f
3090  END IF
3100 END FOR f
3110 FOR loop=1 TO 8
3120  x1=xx+ddata(loop,1): y1=yy+ddata(loop,2)
3130  it$=x1&y1
3140  IF x1<1 OR x1>8 OR y1<1 OR y1>8 THEN NEXT lo
op: EXIT loop
3150  IF board(x1,y1)<>0 THEN NEXT loop: EXIT loop
3160  p=0: REPeat short
3170   pp=it$ INSTR m$(p+1 TO): p=p+pp
3180   IF pp=0 THEN m$=m$&it$: NEXT loop: EXIT loo
p
3190   IF p MOD 2 THEN EXIT short
3200  END REPeat short
3210 END FOR loop
3220 END DEFine
3230 REMark -------------------
3240 DEFine PROCedure game_ended
3250 REMark -------------------
3260 FOR f=1 TO 3: BEEP 3000,24: PAUSE 10
3270 BEEP 14000,33
3280 sc=score(1): sc2=score(2)
3290 INK 7:AT 17,12*(sc=sc2)+20*(sc2>sc)
3300 IF sc<>sc2 THEN PRINT "WINNER !": ELSE PRINT
"DRAW"
3310 END DEFine
```

SOFTWARE FILE

Simon Craven pits wit and digits against a diverse range of QL software.

Shhh... don't tell anybody, but there's actually quite a bit of QL software on the market now. Still only about a tenth of what is available for Spectrum, Commodore 64 or even (shock!) the BBC Micro, and regrettably, much of it is about a tenth of the quality. But it's a start, and things have certainly improved from the days when "quality software" meant something that would *load!*

HOPPER

To get the fingers flexing and the adrenalin flowing, what could be better than a traditional arcade game? Well, maybe a *new* arcade game, but meanwhile there are still plenty of joystick junkies who like a faithful reproduction of the arcade classics on their home computers and *QL Hopper* from Microdeal is just that.

I was using a pre-production version which caused a few hours of intense irritation one afternoon. This wasn't because of any defect in the game, or difficulty in loading – it's just that this is a pretty tough version of the ubiquitous frogs-versus-motorways game. Naturally, the speed and complexity intensify as you hop through the different screens, so it's with some embarrassment that I must report I only just made it onto the second screen after a lot of practice. A possible excuse for my lack of success is that Microdeal haven't finally decided how difficult to make the first screen – production copies may be fine-tuned a little. Programmer Steve Bak says "....as an experienced arcade player I often tend to make the initial versions of my own games rather hard... personally I think Hopper is dead easy!".

The graphics and sound are better than you get on most versions of this game, with some interesting uses of the QL's colour range and a wide variety of vehicles to get flattened by.

One improvement from the original is that the spacing between trucks and cars (Sinclair C5s?!) is subject to slight variations. This isn't a game cracked by sussing out the natural rhythm of the opposing forces – you have to play it by the seat of your pants all the way.

Steve says production units will get a pretty title screen for players to look at while their Microdrives whirr, but even the temporary display is quite exciting. Every time it loads, it looks as if the machine has found a new and spectacular way to bomb out. So *that's* how programmers get revenge on their critics!

Microdeal

BLACKJACK

If you find the idea of juggling with real money on a QL rather frightening, help is at hand. Quest's *Blackjack* lets you fool around with all the imaginary money you like. At first, the idea of playing Blackjack (like Pontoon, in case you aren't a Maverick fan) against a computer seems pretty stupid. Let's face it, most of the card game programs we old-timers have seen through the years on various machines have been about as much fun as watching paint dry. However, *Blackjack* shows that it's always better to play the game first and make your preconceptions afterwards.

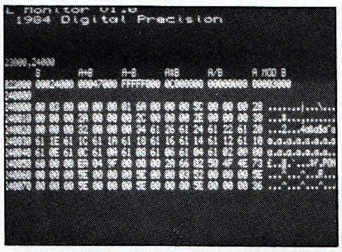
Firstly, the whole thing is carried out with real style. The graphics are superb, the commentary nicely ironic and the whole atmosphere just right. After extensive play-testing (losing an imaginary £2,000 in the cause of duty) I reckon it plays a straight game, relying on the house percentage to stay ahead.

The execution speed is very impressive, as is the loading time. Machine code has a lot going for it. The game is well error-trapped, and you can't use any of the unrealistic tricks that many gambling games allow, such as betting a negative amount and winning when you lose. I *still* think it's a dumb idea to play cards against a computer, but that won't stop me sneaking back for another try every so often. Make sure you're alone first, though – it's amazing how everyone becomes an instant expert when they see you playing this game.

Quest

MONITOR

At last, an updated version of the Computer One *Monitor* has arrived, which now does a lot more than its modest title would have you believe. It does everything the old one did (see *QL User* April 85 for info on Computer One's Assembler), but with the addition of full symbolic debugging, as well as a built-in facility to assemble code without leaving the monitor. Although in theory you could assemble a whole program using this package, it would take a real masochist. Where it comes in very useful, though, is when you just want to modify a few lines or maybe just a couple of bytes to delete a bug, or improve the spelling in the screen prompts of your latest educational software package. It beats patching in hex, and makes debugging a lot quicker than if you reassembled the whole program every time you made a little change.



```
L Monitor V1.0
 1984 Digital Precision
```

Since I'm more of a high-level language man (in other words, I program in BASIC), I waved this package in front of arch bit-twiddler Denning, and even he thought it was a good one.

Computer One

FANTASIA ADVENTURE

As a heavy user of adventure games it always causes a few twinges of regret when I have to forgo the hi-tech delights of the QL and resort to the BBC Micro or even (shudder) the IBM PC to play a decent adventure. No disrespect to Talent, whose duo *West* and *Lost Kingdom of Zkul* are certainly worth exploring, but I've already played those two.

It was with some glee, then, that I spotted *Fantasia Adventure* from SB Software in Norfolk. The plot concerns a spy in a hostile country who has to collect various treasures and bump off various nefarious characters if possible.

Instead of the traditional format of text adventure games, Fantasia displays its information in windows. One window tells you where you are, another gives the possible exits, another accepts the player's commands and so on. There is a certain logic to this with windowing functions built into SuperBasic, but in practice it detracts from the smooth flow of narrative that is the hallmark of a good text adventure. Words are what it's all about, and instead of a clear, interesting description which displays writing as well as programming skills, all you get is a rather disjointed clump of information.

Something else which will annoy seasoned adventurers is the awkward and slow way the program accepts instructions. You have to make a conscious effort to type slowly, and put up with a lot of superfluous beeping noises.

These things apart, *Fantasia Adventure* has the basis of a good game. It tends to kill you off rather arbitrarily, but if you keep your wits about you, you soon start to recognise the traps. The vocabulary is good, and the parsing algorithms are flexible letting you use language in a number of different ways.

SB Software

SUPPLIERS

Microdeal Ltd,
41 Truro Road,
St Austell,
Cornwall PL25 5JE

Quest Automation,
PO Box 49, Torquay,
Devon TQ1 4UR

Computer One Ltd,
Cambridge Technopark,
Newmarket Road,
Cambridge CB5 8BP

SB Software,
20 St Nicholas Street,
Diss, Norfolk

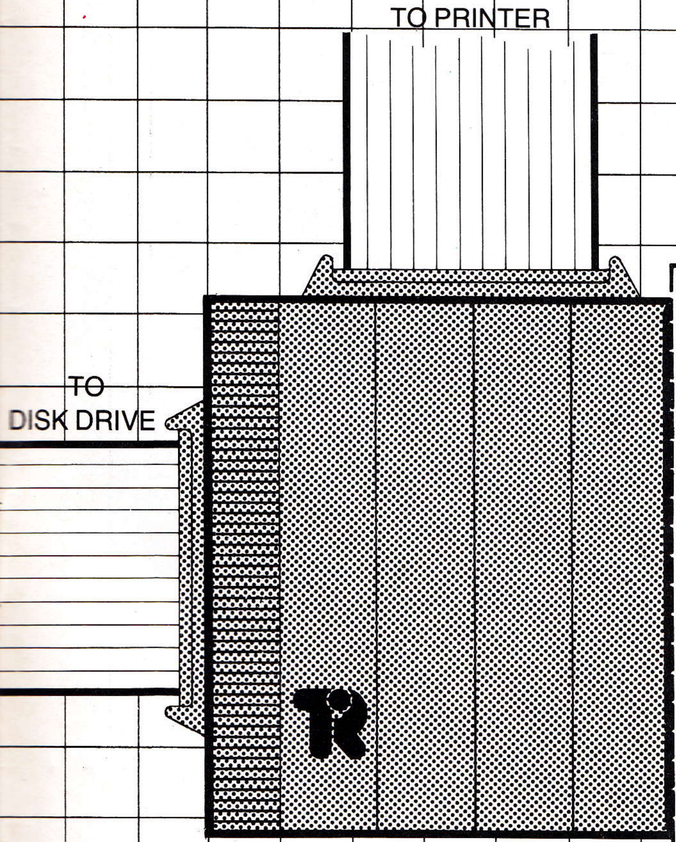
3 IN 1 DELTA INTERFACE

A UNIQUE TRANSFORMER FOR QL

The Delta Disk Interface enhances the Quantum Leap and transforms the QL to a truly professional standard. The Delta Disk has a unique 3 in 1 feature. It is a Disk Interface, a Centronic Printer Interface and a Memory Expansion (Optional extra).

■ Professional Quality

The disk drive system has been regarded as a standard feature on the professional systems. It is fast, reliable, economical and flexible. The Delta Disk is designed to meet the most stringent professional standard and quality giving the maximum performance, compatibility and high speed storage.



■ Compatibility

The operating system of Delta Disk is in the EPROM. The operating software is unique and is designed with the full compatibility of the QL software in mind. The Delta Disk can work with any disk drives which BBC Micro uses (Shugart Standard). It works perfectly well with the Microdrive. It means you can run both Microdrive and disk in complete harmony.

■ Unique 3 in 1 feature

The design of Delta Disk allows total flexibility and upgradibility. It is a Disk Interface, a Centronic Printer Interface and it is also a Memory Expansion up to a total RAM of 128Kbytes. Three models are available:

a. Delta Basic
b. Delta 64 with 64 Kbyte RAM
c. Delta 128 with 128 Kbytes RAM

The Delta Disk requires **no** external power supply. The Delta Disk slots into the left side of your QL to give instant operating compatibility with the disk drive and the printer.

■ The Delta disk from Technology Research, a truly inseparable partner for your QL.

CUT OUT THIS COUPON AND SEND WITH CHEQUE OR POSTAL ORDER TO:
TECHNOLOGY RESEARCH LIMITED
UNIT 18, CENTRAL TRADING ESTATE, STAINES, MIDDLESEX TW18 4UX

PLEASE SEND ME:		QTY		TOTAL
DELTA BASIC	@ £129.50		£4.00 p&p	£
DELTA 64	@ £199.00		£4.00 p&p	£
DELTA 128	@ £249.50		£4.00 p&p	£
				£

NAME ______________________________

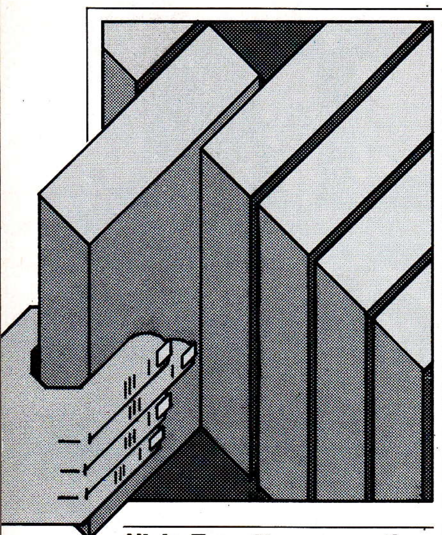
ADDRESS ______________________________

____

______________________ POSTCODE ______________

*Allow 28 days for delivery. Prices include VAT.

Technology Research Limited
Unit 18, Central Trading Estate, Staines, Middlesex TW18 4UX Telephone Staines (0784) 63547 Telex 896691 TLXIR G



BOOKMARKS

Nicky Trevett assesses the 'ins and outs' of the latest on QL User's bookshelves.

For those who want to know what goes on in a physical sense inside their QL, *Inside the Sinclair QL* by Jeff Naylor and Diane Rogers aims to explain it all.

Priced at £6.95 and published by Sunshine, a third of the book is devoted to general electronic and computer principles. The idea is a sound one. If a reader with a non-technical background is to make any sense of the QL's inner workings, he or she must know something about basic electronics – circuits, semiconductors and logic gates, as well as have some understanding of binary and hexadecimal, the way a microprocessor is organized, and what all the various bits of a computer do. The first part then, only relates to the QL in passing.

Most of the meat comes in the chapter on the 68008 instruction set, which is arranged alphabetically as a reference guide and not intended to be read through, as such. This seems a rather odd way to handle material in a book written principally for those interested in finding out about a topic new to them.

A chapter on machine code includes a SuperBasic program for installing a machine code routine in the QL's memory, then passing control to the code, along with a program designed to allow you to experiment with machine code. The rest of the book looks at the memory map, video display, input and output, and a quick glance at SuperBasic and QDOS.

I found it all a bit of a puzzle. There is a great deal of information provided here, and a lot of work has gone into putting it all together, but it's neither a general interest book, nor a straightforward hardware reference book. It is simply not lively enough to grip and hold the imagination either of the 'lay' QL user, or the dedicated hobbyist.

Risqué Business

Talking of liveliness, the business user or programmer who would welcome a ray of sunshine into a drab working day will enjoy *Sinclair QL in Business* by Arnold Handley, published by Newnes Microcomputer Books at £9.95.

It's a remarkable book, with something for everyone who uses a QL, business user or home enthusiast. It is principally about the software applications that come with the QL and how they can best be exploited for use in business, but the author has seen fit to fling in anything else about the QL, business, or computing that has occurred to him in the course of writing the book.

This is best illustrated by the contents page. 'Is Quill the best, or are other word processors better?' asks one chapter heading, provocatively. 'Quill, the unofficial ways – tricks not in the manual' promises another. 'Abacus as a word processor? Writing a report to the Boss' and 'Abacus with no numbers at all – the fastest Duty List' illustrates the author's refreshingly open-minded approach to Abacus. In other chapters, Easel is used to solve the mystery of a terrible epidemic in Russia, Archive is used to select a suspect in a murder hunt, put to work as a calculator, and made to draw a graph.

It's a witty and highly readable approach to putting QL applications effectively to work, and between the jokes there's a great deal of very sound advice. It's an approach that probably won't suit everybody, and there are plenty of 'straight' guides to Quill, Abacus, Easel and Archive around. But they won't beat this one for entertainment!

LISP Lifeline

If your interest in artificial intelligence centres on the programming techniques involved, an understanding of LISP is indispensable. LISP, however, with its huge extended vocabulary, has a reputation for being tough to learn.

Enter *LISP: A Gentle Introduction to Symbolic Computation* by David Touretzky, published by Harper & Row at £11.95. The 'gentle' might sound comforting, but 'symbolic computation' is likely to frighten off all but mathematicians and scientists 'in the know'.

Non-programmers need have no fear. The book is written just for you. It's aimed at those without a mathematical, scientific or even a computing background, written perhaps with the student programmer in mind, but equally of value to anyone interested in learning LISP.

It sets out to avoid technical jargon, and largely succeeds. There are plenty of exercises to allow you to practice your new-found skills – most of them, believe it or not, are fun – and the whole thing is presented in a large-format typeface that doesn't repel you as soon as you open the book. It's a lively text-book introduction to LISP which I can strongly recommend.

And on to yet another book about LISP, the artificial intelligence language currently arousing such interest. *LISP, The Language of Artificial Intelligence* by A A Berk, published by Collins at £9.95, aims to offer practical instruction on programming in LISP, orientated towards artificial intelligence applications.

It starts with an introduction to AI, aimed at showing exactly why a list processing language such as LISP is good for AI programming, then moves on to describing LISP itself, using BASIC as a point of comparison throughout. Knowledge of BASIC is assumed, but none of LISP.

There are plenty of diagrams and practical examples to work through, but if you're seriously interested in becoming a LISP programmer, it's worth checking out some of the other books available first before deciding which approach suits you best.

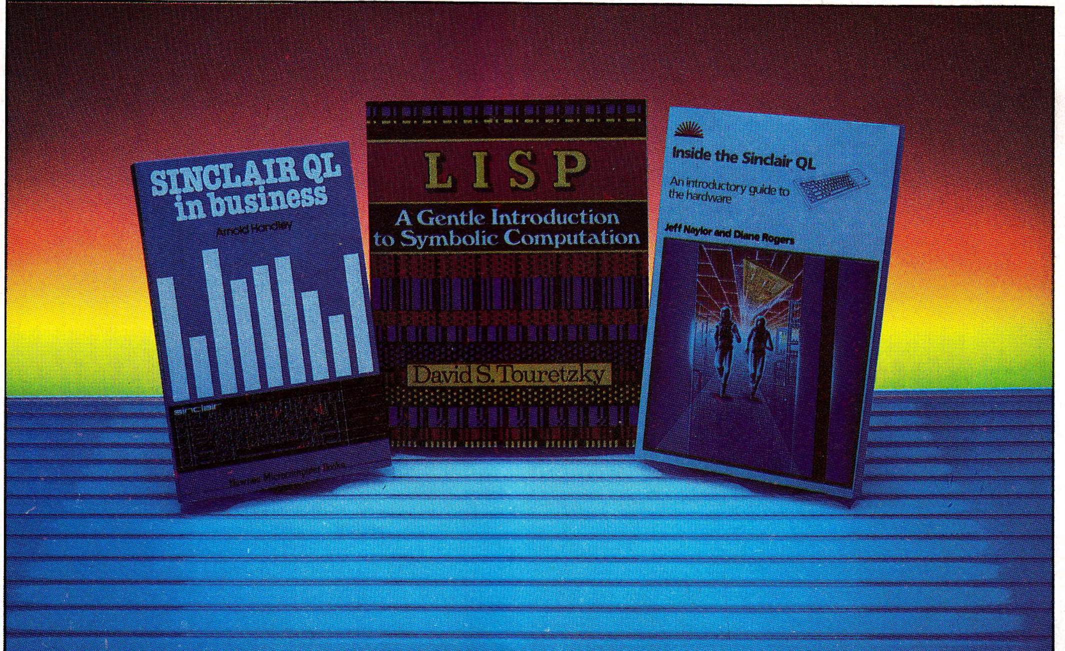
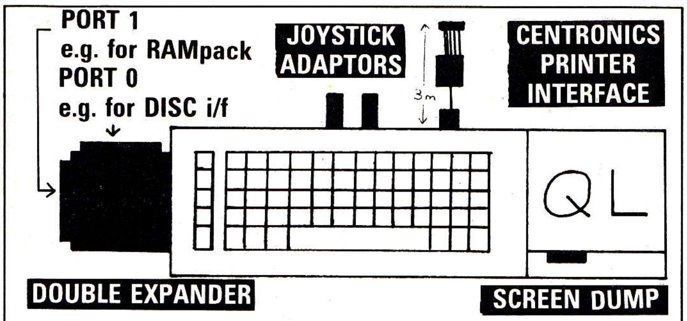

SINCLAIR QL in business
Arnold Handley
Newnes Microcomputer Books
LISP
A Gentle Introduction to Symbolic Computation
David S. Touretzky
Inside the Sinclair QL
An introductory guide to the hardware
Jeff Naylor and Diane Rogers

PHOTO BY TERRY BEDDIS



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.

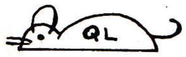
★ Copies QL screen image to printer.
★ Supplied on microdrive cartridge.
★ Contains other useful software.

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.



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 orders welcome

DataManagement QL software — leading the way

SBUTIL — (£9.00) — Possibly the last word in Basic programming utilities. Constant readout of: Line No & Statement/Data Line & Statement/Dynamic Basic Size. Full SuperBasic speed control via the Basic Editor. 4 formats for all Screens and tastes. No more wondering where things go wrong. This tells you exactly where you are at all times.

MBACKUP — (£8.00) — An independent, fast & reliable backup to/from Mdv or Disk. File additions, over-writes and file sizes fully displayed.

TERMINAL — (£12.50) — A superb full featured terminal package. Full protocol control and ability to recall previous setups. Job control, clock and alarm modules, together with Piping for mdv and printer. Full mdv and disk control without exiting: copy, delete or save to files. Monitor version only.

CHARGEN — (£10.00) — A very comprehensive character generating package. Includes extra functions for SuperBasic to control your creations. Founts can be saved to disk or mdv for recall and use. No copyright restriction on the subsequent publication using these features.

SBEXTRAS — (£6.00) — Many extra features and procedures to add to SuperBasic. Includes full job control, dynamic file access with byte or variable GET/PUT. LENGTH, REMAINDER, QUOTIENT, HEX, DEC and many more.

FM — (£12.50) — A full feature file management package. If you need to do it to a file, this will do it for you. Supports disks/mdvs, wildcard and formatted printing to help your library control.

All the above are 100% m/language, multi-tasking and will run independently under QDOS. Prices shown include VAT & P/P, but add £4.95 per order for a mdv cartridge.

Cheques or P/Os to DataManagement: 12 Larch Way, Haxby, YORK, YO3 8RU or telephone (0904) 760351 (anytime) for our latest list, which is growing daily.

Have **you** written any good **QL** software. We are looking for the **best** software and are prepared to pay the **best** royalties.

Coming Soon: FED — a full feature file editor: **DisAssem** (if you don't know what it is — you won't need it!); **NeatList** — to keep those printouts beautiful; and many, many more. Write or Phone for your list — **Now.**

QL Books

ADVANCED ARCHIVE ON THE SINCLAIR QL

Malcolm Davison shows you how you can do far more if you know how to design Archive programs correctly. Firstly – in Archive Update, you'll discover the best use of Archive commands and go on, in the second part of the book to learn by applying your expertise and to tailor your own application programs. In all, 201 pages of discovery, increased enjoyment and efficiency. £7.95

So Close to Reality?

That's the question Patrick Hall asks in THE REAL THING? Microcomputer simulations on the Sinclair QL. After reading the 231 pages of this book you'll be well equipped to program your own simulations and to engineer your own dreams. You'll learn the art and science of simulation by following the carefully structured program descriptions, and seeing the 19 fully listed programs at work. £7.95

How to find your way around the MC 68000 processor?

Lionel Fleetwood has the answer – the 181 pages of THE 68000 USER GUIDE takes you from starting with assembler, through instructions and modes; stacks, registers and modes; moving data around; sub-routines; register and arithmetic verbs; bit level operations; and more. A technical book full of practical help but not without humanity – there's also a chapter called "when things go wrong". £8.95

PUBLISHED BY SIGMA PRESS AND MARKETED BY JOHN WILEY & SONS LTD

ORDER FORM

To: Lesley Valentine, John Wiley & Sons Ltd, FREEPOST,
Baffins Lane, CHICHESTER, West Sussex PO19 1YP

□ Please send me .... copy(ies) of
Advanced Archive on the Sinclair QL
by Malcolm Davison at £7.95 per copy

□ Please send me .... copy(ies) of
The Real Thing by Patrick Hall at £7.95 per copy

□ Please send me .... copy(ies) of
The 68000 User Guide by Lionel Fleetwood
at £8.95 per copy

Postage and packing free - please allow 14 days for delivery

□ I enclose a cheque/PO for £ .... payable to
John Wiley & Sons Ltd

□ Please debit my credit card account £ ....

Card No.................. Expiry date ....

Access □ American Express □ Barclaycard/Visa □ Diners Club □

Name/Address ....

....

....

Signature ....

JOHN WILEY & SONS LTD REG. NO. 641132 ENGLAND QL USER

▼ 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					
Andy Carmicheal	(B)	Family Tree	£3.00	Aug	50 □
Archive program and database for setting up and displaying large family trees					
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					
Mathew Capp	(B)	Miners	£2.00	Aug	30 □
A nail biting management simulation that puts you in charge of the NCB					
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					
R Green	(B)	Qthello	£1.00	Aug	25 □
A 3D version of the well known board game Othello for one or two players					
S J Ackers	(B)	*Touch Type	£4.00	Aug	80 □
Touch typing course – 14 lessons, on-screen keyboard, 800+ word vocabulary and WPM readout					

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
Redhill 65080
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 6711863
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

Slave Software
050 846 8866
Q-Disc

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 475244

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 475244

PCML Ltd
0372 67282/68631

QL+RAM cards
Quest
0421 566488
Simplex Data Ltd
01 575 7531

Extras

A>Line Computer Systems
0533 778724
4-way mains filter/adapter
Action Computer Supplies
01 903 3921
Mains spike eliminator
Broomspring Specialities
0742 737000
Table-top stands
Classified Product & Services
0930 52204
Leads etc.
Computer Supplies
146 Church Road, 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
Spectrotek
0669 20565
QL repair service
Transform Ltd
089 283 4783
QL dust cover, microdrive storage box, RS232 lead
Viglen Computer Supplies
01 843 9903
Voltmace Ltd
Park Drive, Baldock, Herts.
Printer stand
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 Road, Bedford, Beds
Screen Editor
Computer One
0223 862616
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
0582 696421
MonQL
J&D Software
3 Alfred Road, Lowton, Warrington
Metacomco
0272 428781
Assembler, BCPL, Lisp
MicroAPL
01 622 0395
Micrologic Consultants Ltd
57 Station Road
Micro Processor Engineering Ltd
21 Hanley Road, Southwater, Horsham, E Sussex
QL Terminator Emulator
PCS
8 Oak Grove Way, Bridgwater, 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 Road, 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
Slave Software
050 846 8866
Arable Farmer Software
Strong Computer Systems
0267 231246
Super Plant Software
097 231246
Plant & gardening software
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 Road, 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 Road, 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 Road, Exeter, Devon
Executive Adventure
New Horizon Software
Fourwinds, Cwn Lane, Rogerstone, Gwent
Pacman, QBERT, Gold
Peak Electronics
32 Clifton Avenue, Hartlepool, Cleveland
QL Colour Quest
Printerland
0484 513105/687875
Psion Chess
Psion
01 723 9408
Psion Chess
Rodent Software
3 Brookend Crescent, Henley-in-Arden, W Midlands
Adventure Writer, QL Artist, 2×7 Games cartridges
S&B Software
20 St Nicholas Street, Diss, Norfolk
Fantasia Adventure
Shadowsoft
0296 669740
Area Radar Controller, Strategy
Snowsoft
6 Bousefield Crescent, Newton Aycliffe, Durham
Hungry Harry in the Haunted House, Doom Room: Excod. VIII
Sinclair Research
0276 685311
Psion Chess
Summit Software
36 Wood Crescent, Rogerstone, Newport, Gwent
Frogger, Dungeon
Swansoft
164 Vicarage Road, Morriston, Swansea
Space Trek
Talent Computer Systems
041 552 2128
ZKUL, WEST, GraphicQL
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 Street, W8
McGraw Hill
0628 23431
MicroPress
0892 39606
Melbourne House
01 940 6064
Prentice Hall
0442 58531
Sunshine
01 437 4343

Book Titles

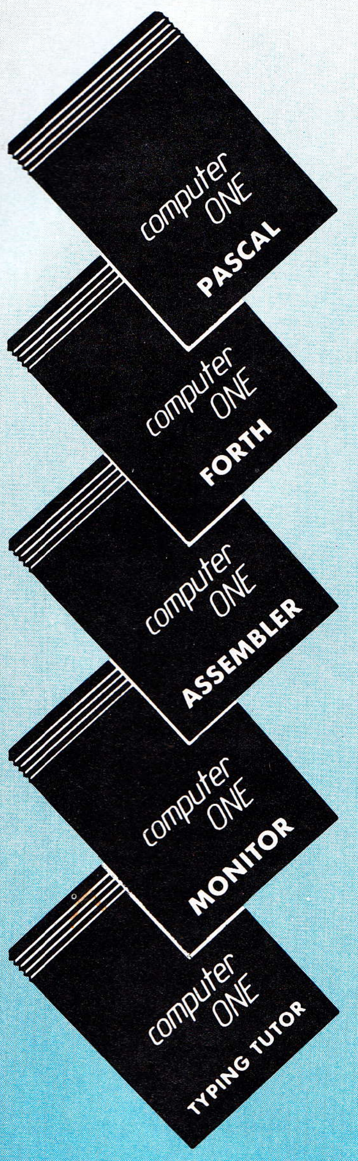
Quill, Easel, Archive and Abacus on the Sinclair QL
McCallum Varey
(Sunshine £6.95)
Mathematics on the Sinclair QL
Kosniowski
(Sunshine £6.95)
Programming in C
Traister
(Prentice/Hall £19.50)
Getting To Know Your Sinclair QL
Morris
(Duckworth £7.95)
The C Programming Language
Kernigham & Ritchie
(Prentice/Hall £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)

CLASSIFIED

AD INDEX

QL POWER

AT YOUR FINGERTIPS FROM COMPUTER ONE



PASCAL £39.95

Computer One PASCAL is a powerful implementation of this classic programming language. Produced specifically for the QL, this package provides a professional and highly educational programming system. Complete with comprehensive 100 page manual, this is the ultimate language package that no QL enthusiast can do without.
FEATURES: ★ Full screen editor ★ QL graphics and sound extensions
★ Example programs ★ Complete 100 page User Guide ★ Produce stand-alone programs
★ Full implementation close to ISO-standard ★ Typically 15× faster than BASIC

FORTH £39.95

A full implementation of the latest FORTH-83 standard with full integration to QDOS. This compiled language runs very fast and allows you to produce fast machine code application programs for your QL.
FEATURES: ★ FORTH-83 system ★ Forth screen editor ★ 100-page manual
★ Produce stand-alone programs ★ QL-graphics and sound extensions
★ Example FORTH programs ★ FORTH macro-assembler
★ Multitasking extensions with many powerful features

ASSEMBLER £29.95 NEW: Now includes FREE linker

COMPUTER ONE Assembler is a comprehensive assembler program toolkit comprising a full 68008 assembler and full screen editor. This is a professional multitasking assembler and is written in machine code for speed and compactness.
FEATURES: ★ 68008 Assembler ★ Full syntax checking ★ Full screen editor
★ Assembler linker ★ Comprehensive User Manual ★ Integrates to Pascal and Superbasic
★ INCLUDE file facility ★ Full assembler listing

MONITOR £24.95 (£19.95 when purchased with the Assembler)

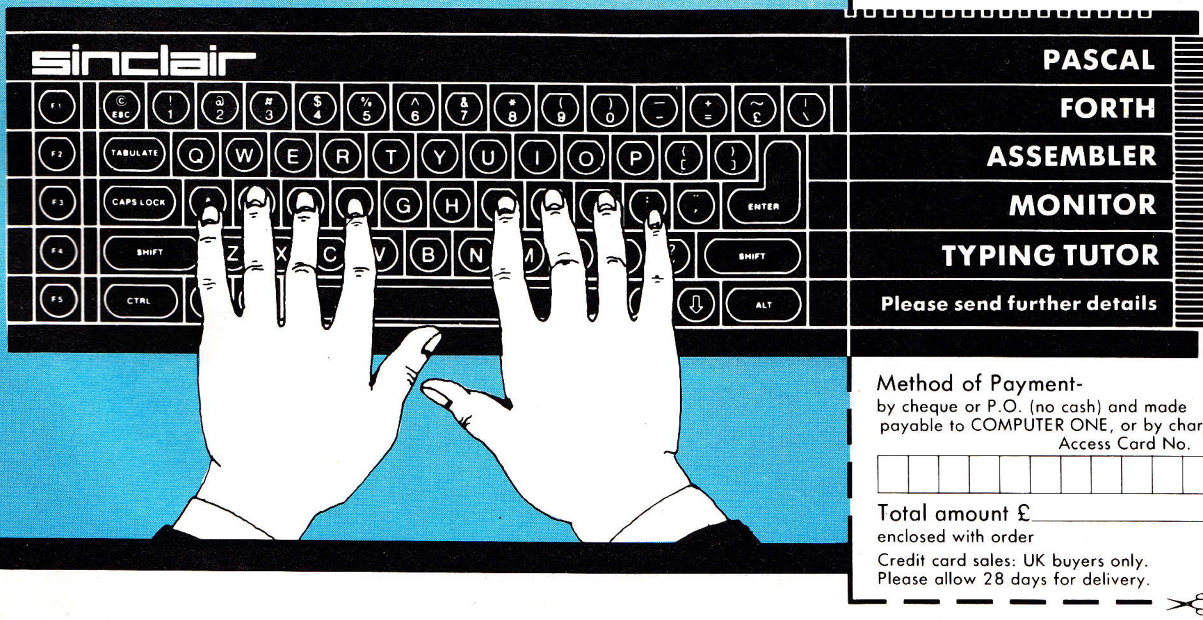
The Computer One MONITOR is a powerful multitasking, multi-window debugging monitor. The MONITOR is an invaluable software tool for anyone developing software for the QL and is ideal for use in conjunction with the Computer One Assembler.
FEATURES: ★ Memory disassembly and display ★ Trace code in RAM or ROM
★ Breakpoints and watchpoints ★ Full channel management ★ User defined screen layout
★ Set and display registers ★ Alter memory ★ Move and search memory ★ Full job control
★ Ability to monitor several jobs simultaneously

TYPING TUTOR £19.95

Let your Sinclair QL teach you to type! Computer One's Typing Tutor is a fast, easy and enjoyable way of learning true touch-typing on your QL.
FEATURES: ★ Self-paced tutorials
★ Speed and accuracy diagnosis
★ Interactive colour graphics ★ User Guide

computer ONE
SCIENCE PARK, MILTON ROAD, CAMBRIDGE
(0223) 862616

Name____________________
Address____________________
____
Signed____________________



	Tick Order
PASCAL	□
FORTH	□
ASSEMBLER	□
MONITOR	□
TYPING TUTOR	□
Please send further details	□

Access

Method of Payment-
by cheque or P.O. (no cash) and made payable to COMPUTER ONE, or by charge card.
Access Card No.
□□□□□□□□□□□□□□□□

Total amount £____________________
enclosed with order
Credit card sales: UK buyers only.
Please allow 28 days for delivery.

QL8/85
Inc. VAT & p&p.

The first QL adventures from TALENT!

The Lost Kingdom of ZKUL

Explore the ancient dwelling-place of the Dwarves – where the Wizard guards the last precious secret.

A classic and complex text adventure with hundreds of locations and a huge vocabulary. Set in real-time with traps, tasks, puzzles and mazes – and a special note-pad feature to aid you.

"the quality of the game is superb"
Micro Adventurer

PRICE
£19.95
+ 50p postage and packing

WEST

You are on the track of a notorious gang of bank robbers who have gone to ground near an abandoned mining town deep in Indian territory. Your task – to outwit and outgun the robbers, collect as much loot as possible and piece together clues on how to escape.

A demanding and exciting text adventure for the experienced games-player, using over 200 words and as many phrases. There are over 130 locations to search – not all of them easy to find. Events happen in real-time outside your control – Indians charge, rattle-snakes slither past and robbers appear and shoot at you. How long can you survive?

PRICE
£19.95
+ 50p postage and packing
(Also on CBM 64, BBC & Electron)

Available from selected branches of:
John Menzies
WHSMITH

TALENT
COMPUTER SYSTEMS
CURRAN BUILDING, 101 ST JAMES ROAD, GLASGOW G4 0NS
24 HOUR CREDIT CARD HOT-LINE 041-552 2128